Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization

E. G. Birgin *   J. M. Martínez †   F. H. Nishihara ‡   D. P. Ronconi §

February 3, 2005

Abstract

The orthogonal packing of rectangular items in an arbitrary convex region is considered in this work. The packing problem is modeled as the problem of deciding for the feasibility or infeasibility of a set of nonlinear equality and inequality constraints. A procedure based on nonlinear programming is introduced and numerical experiments which show that the new procedure is reliable are exhibited.

Scope and purpose: We address the problem of packing orthogonal rectangles within an arbitrary convex region. We aim to show that smooth nonlinear programming models are a reliable alternative for packing problems and that well-known general-purpose methods based on continuous optimization can be used to solve the models. Numerical experiments illustrate the capabilities and limitations of the approach.

Key words: Packing of rectangles, orthogonal packing, feasibility 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

*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. This author was supported by FAPESP (Grant 03/00460-0). e-mail: fhn@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

pallet loading, packing of ship containers and establishing of layout in clothing industry. Many papers have been published dealing with packing problems. For a complete review we refer to the annotated bibliography [13] and to the survey [16].

One of the most popular and useful problems in this area consists in finding the maximum number of identical rectangles that can be orthogonally packed into a larger rectangle. Polynomial algorithms for the guillotine version of the problem exist [21] and the NP-completeness of the non-guillotine problem has been conjectured [12, 19]. In [14] a very efficient heuristic to solve this problem was introduced (see also [10] for an available implementation and extensive numerical experiments). The authors conjectured that their method always finds the optimal solution and solved hard instances that could not be solved by other heuristics.

A nonlinear formulation for the constrained two-dimensional non-guillotine cutting problem (which deals with limits in the number of each different type of rectangle that can be packed) was presented in [2], where the model was used for the definition of a populational heuristic. The Method of Sentinels for packing polygonal items within arbitrary objects was introduced in [6]. The method is based on convex analysis and nonlinear formulations for overlapping and belonging. Nonlinear models have also been successfully used for other related packing problems, such as the packing of molecules with a specified minimum-atom distance [17]; and the packing of identical or different circular pieces within several types of objects [9, 18, 20, 22]; among others.

In the present work we are concerned with the problem of packing rectangular items in an arbitrary region, generally convex. The axes of the rectangles must be parallel to the coordinate axes and the loading is not restricted to guillotine cutting patterns. The strategy for solving the problem is based on the fact that the question "Is it possible to pack m rectangles?" can be answered by means of the resolution of a finite (although possible large) number of nonlinear optimization problems. Therefore, the strategy for packing the largest possible number of rectangles consists of answering the above question for increasing values of m.

This paper is organized as follows. Section 2 describes the nonlinear model and a procedure to pack as many rectangles as possible. In Section 3 numerical results are presented. The last section contains final remarks.

2 Nonlinear approach

We first consider the problem of packing a given set of m rectangles without rotations. Then, the possibility of orthogonal rotations is incorporated and a procedure for identical rectangles is devised. Finally, we deal with the problem of packing as many identical rectangles as possible allowing orthogonal rotations.

2.1 Fixed-orientation-pack problem

Let Ω be a convex subset of $I\!R^2$. We want to pack m rectangles in such a way that they are contained in Ω and the interior of the intersection of any pair of different rectangles is empty. Since Ω is convex, the fact that the vertices of a rectangle are in Ω is enough to guarantee that the rectangle is contained in Ω. In addition, the axes of all the rectangles must be parallel, so, we may consider that they are parallel to the natural x and y axes of $I\!R^2$. This assumption represents no loss of generality if the original packing problem has such kind of constraint or if we are dealing with the cutting problem of an anisotropic material.

The *Fixed-orientation-pack* problem consists of packing a set of m given rectangles in a given region Ω. Rotations (of any type) are not allowed. This problem is the nonlinear programming basis of our general procedure. For all $i = 1, \dots, m$, let $C^i = (c_1^i, c_2^i)$ be the (variable) center of the rectangle $R^i \equiv R^i(a^i, b^i)$ and let $a^i, b^i > 0$ be the (fixed) values of the horizontal and vertical sides, respectively. Let

$$V_{\mathrm{sw}}^i \equiv V_{\mathrm{sw}}^i(R^i, C^i) = (c_1^i - a^i/2, c_2^i - b^i/2), \quad V_{\mathrm{se}}^i \equiv V_{\mathrm{se}}^i(R^i, C^i) = (c_1^i + a^i/2, c_2^i - b^i/2),$$

$$V_{\mathrm{ne}}^i \equiv V_{\mathrm{ne}}^i(R^i, C^i) = (c_1^i + a^i/2, c_2^i + b^i/2), \quad \text{and} \quad V_{\mathrm{nw}}^i \equiv V_{\mathrm{nw}}^i(R^i, C^i) = (c_1^i - a^i/2, c_2^i + b^i/2),$$

be the four vertices of R^i centered at C^i. So we refer to the four vertices as V_k^i for all $k \in D \equiv \{\mathrm{sw}, \mathrm{se}, \mathrm{ne}, \mathrm{nw}\}$.

Fixed-orientation-pack (FOP) problem:

Given a convex set $\Omega \subset I\!R^2$ and a set of m rectangles $R^i \equiv R^i(a^i, b^i), i = 1, \dots, m$, find $C^1, \dots, C^m \in I\!R^2$ such that

1. *For all $i = 1, \dots, m$ and $k \in D$,*
$$V_k^i \in \Omega. \tag{1}$$

2. *For all $i, j = 1, \dots, m, i \neq j$,*
$$|c_1^i - c_1^j| \geq \frac{a^i + a^j}{2} \tag{2}$$
or
$$|c_2^i - c_2^j| \geq \frac{b^i + b^j}{2}. \tag{3}$$

Condition (1) says that the vertices of R^i are in Ω. The fact that at least one of the conditions (2) or (3) is satisfied means that rectangles R^i and R^j do not overlap (the interior of their intersection is empty).

We will show now how to obtain a solution of FOP by means of the resolution of a continuous feasibility problem. Let us define

$$\bar{f}(C^1, \dots, C^m) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left[\max\left\{0, \frac{(a^i + a^j)^2}{4} - (c_1^i - c_1^j)^2\right\}^2 \times \max\left\{0, \frac{(b^i + b^j)^2}{4} - (c_2^i - c_2^j)^2\right\}^2 \right]$$

Clearly, the FOP problem consists in finding $C^1, \dots, C^m \in I\!R^2$ such that $\bar{f}(C^1, \dots, C^m) = 0$ with the constraints (1). The equation $\bar{f}(C^1, \dots, C^m) = 0$ and the inequalities (1) define a feasibility problem that represents well our goal.

The FOP problem can be reduced to find a global minimizer of

$$\text{Minimize } \bar{f}(C^1, \dots, C^m) \text{ subject to (1)} \tag{4}$$

The feasibility FOP problem has a solution if, and only if, the objective function value is null at a global minimizer of (4). Our strategy for solving (4) is to consider this problem as an ordinary nonlinear programming problem and try to solve it using a "local" solver, starting from different initial points in order to enhance the probability of convergence to a global minimizer. The objective function of (4) has continuous first (but not second) derivatives.

If $\Omega \subset I\!R^2$ is an easy set (i.e., a set onto which is computationally inexpensive to project an arbitrary point), like a box, a ball, or a polyhedron, then problem (4) can be solved using an optimization method which deals explicitly with the constraints (1), like, for example, GENCAN [5], SPG [7, 8] or IVM [1], respectively. If an optimization method that deals explicitly with constraints (1) is not available then (4) can be solved using an Augmented Lagrangian approach for inequality constraints as the one implemented in ALGENCAN [4]. Finally, consider the case in which Ω is given by $\Omega = \Omega_1 \cap \Omega_2$, where Ω_1 is an easy set and $\Omega_2 = \{x \in I\!R^2 \mid g(x) \leq 0\}$ is not. Clearly, constraints (1) can be splitted into

$$V_k^i \in \Omega_1, \tag{5}$$

and

$$V_k^i \in \Omega_2. \tag{6}$$

As before, it is easy to see that to find $C^1, \ldots, C^m$ satisfying (6) is equivalent to finding $C^1, \ldots, C^m$ that annihilates $\sum_{i=1}^m \sum_{k \in D} \max\{0, g(V_k^i)\}^2$. Therefore, problem (4) can be reformulated as

$$\text{Minimize } f(C^1, \ldots, C^m) \equiv \bar{f}(C^1, \ldots, C^m) + \sum_{i=1}^m \sum_{k \in D} \max\{0, g(V_k^i)\}^2 \text{ subject to (5)}. \tag{7}$$

We chose the approach (7) in this study. We consider $\Omega \equiv \Omega_1 \cap \Omega_2$, where Ω_1 consists of the box constraints in the definition of Ω and Ω_2 consists of all the other constraints. So, problem (7) becomes a box-constrained problem and GENCAN [5] is a good option for solving it. Note that, as in problem (4), by the smoothness of g, the objective function of (7) has continuous first (but not second) derivatives.

2.2 Allowing 90-degrees rotations

In many practical applications, the rectangle R^i with horizontal side a^i and vertical side b^i can be considered equivalent to the rectangle with horizontal side b^i and vertical side a^i. This means that rotations of 90 degrees are admissible for packing purposes. Assume that, for all $i = 1, \ldots, m$, a choice is made according to which a^i (or b^i) is the horizontal side of R^i and, of course, b^i (respectively a^i) is the vertical side. If, given this choice, we solve the corresponding FOP problem and we obtain that f is null at the solution, the packing problem with orthogonal rotations is solved.

Clearly, one has 2^m possible choices but, in specific situations, this number can be considerably reduced. For example, if $a^i = a$, $b^i = b$ for all $i = 1, \ldots, m$ then the number of different possibilities is reduced to $m + 1$. In this case, for all $p = 0, 1, \ldots, m$, we may think that p rectangles have horizontal side equal to a and vertical side equal to b and $m - p$ rectangles have horizontal side equal to b and vertical side equal to a. So, we have $m+1$ FOP problems of the form (7). If we solve one of them finding $f=0$ the new packing problem is solved.

The *Orthogonal-rotations-pack* problem defined below consists of packing m identical rectangles in a given region Ω, allowing 90-degrees rotations.

Orthogonal-rotations-pack (ORP) problem:

Given a convex set $\Omega \subset I\!R^2$ and a set of m identical rectangles $R^i \equiv R^i(a,b)$, find $C^1,\dots,C^m \in I\!R^2$ and $p \in \{0,1,\dots,m\}$ such that $C^1,\dots,C^m$ is a solution of the FOP problem defined by the rectangles $\bar{R}^i \equiv \bar{R}^i(a,b), i=1,\dots,p$, $\bar{R}^i \equiv \bar{R}^i(b,a), i=p+1,\dots,m$, and the set Ω.

2.3 Packing as many rectangles as possible allowing 90-degrees rotations

Given a convex set Ω and an unlimited number of identical rectangles, the problem of packing as many rectangles as possible (allowing 90-degrees rotations) is equivalent to solve the ORP problem for the biggest possible value of m. For a fixed value of m, solving the ORP problem is equivalent to solve one of the $m+1$ associated FOP problems. Finally, solving one of these FOP problems is equivalent to find a global solution of its associated nonlinear programming problem (7) such that the objective function value is null. Moreover, most nonlinear optimization methods find first-order stationary points (very likely, local minimizers) of problem (7). 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 $m \equiv k-1$ items were packed. On the other hand, while our effort is not exhausted, we carry out the following steps:

Step 1: Set $k \leftarrow m_{\rm lb}$, 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 an option).

Step 2: For solving the ORP problem of packing k rectangles, we select a random integer $p \in \{0,1,\dots,k\}$ and try to solve its p-th associated FOP problem. For solving the p-th FOP problem we consider a random initial guess from which we start a local minimization with the expectation of finding a global solution of (7) that annihilates the objective function.

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

Step 4: If a solution with null optimal cost was found but $k<m_{\rm ub}$ then we increase k by one and go back to Step 2. In this case, an ORP problem with k 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 go back to Step 2. In this case, the current ORP problem we are trying to solve remains the same and we will just try to solve it choosing another random number p of horizontal (and consequently vertical) rectangles and another random initial guess.

Figure 1 gives a graphical representation of these steps, including the stopping criterion based on effort exhaustion. If the algorithm stops with $m \equiv m_{\mathrm{lb}} - 1$ it means that even a packing with m_{lb} items was not found. Moreover, when the method stops saying "A packing with m items was found" it 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.

The whole packing procedure consists of solving a potentially large number of box-constrained optimization problems. Each time we try to solve an instance of the smooth nonlinear bound-constrained optimization problem (7), a random configuration is used as initial guess to run a local solver. No information of the previous trials is considered. Information of preceding problems with a small number of items is not useful because the optimal configuration may vary a lot when the number of items increases. Information of preceding problems with the same number of items is not useful either. Note that, if we are still trying to solve a problem repeating the number of items, this is because we failed in the previous trials. So, the final points of those trials are undesired stationary point of the nonlinear model, and to start near to them would leave us trapped 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.

The influence of the lower and upper bounds on the behaviour of the algorithm is a relevant practical issue. First, note that there is no other stopping criterion, before the effort exhaustion, than reaching the upper bound. So, to have a sharp upper bound or, in other words, to know the optimal number of items that can be packed, is the unique alternative to confirm that an optimal solution was attained and, consequently, to stop the process. On the other hand, a lower bound could be useful to skip the resolution of some problems. However, as the numerical experiments will show, while finding a packing of a "near-to-the-optimal" number of items is a relatively hard task, to pack e few less items than this number is a trivial problem. Therefore, the availability of a lower bound is, in practice, useless. This is the reason why we start the process packing one item and then increasing one item per step instead of using bisection approaches.

We chose GENCAN [5] 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, [11] and [15] (pp. 326–330). GENCAN adopts the leaving-face criterion of [3], that employs the spectral projected gradients defined in [7, 8]. For the internal-to-the-face minimization it uses a general algorithm with a line search that combines backtracking and extrapolation. In the present form, GENCAN uses, 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 [5] where extensive numerical experiments assess its reliability.

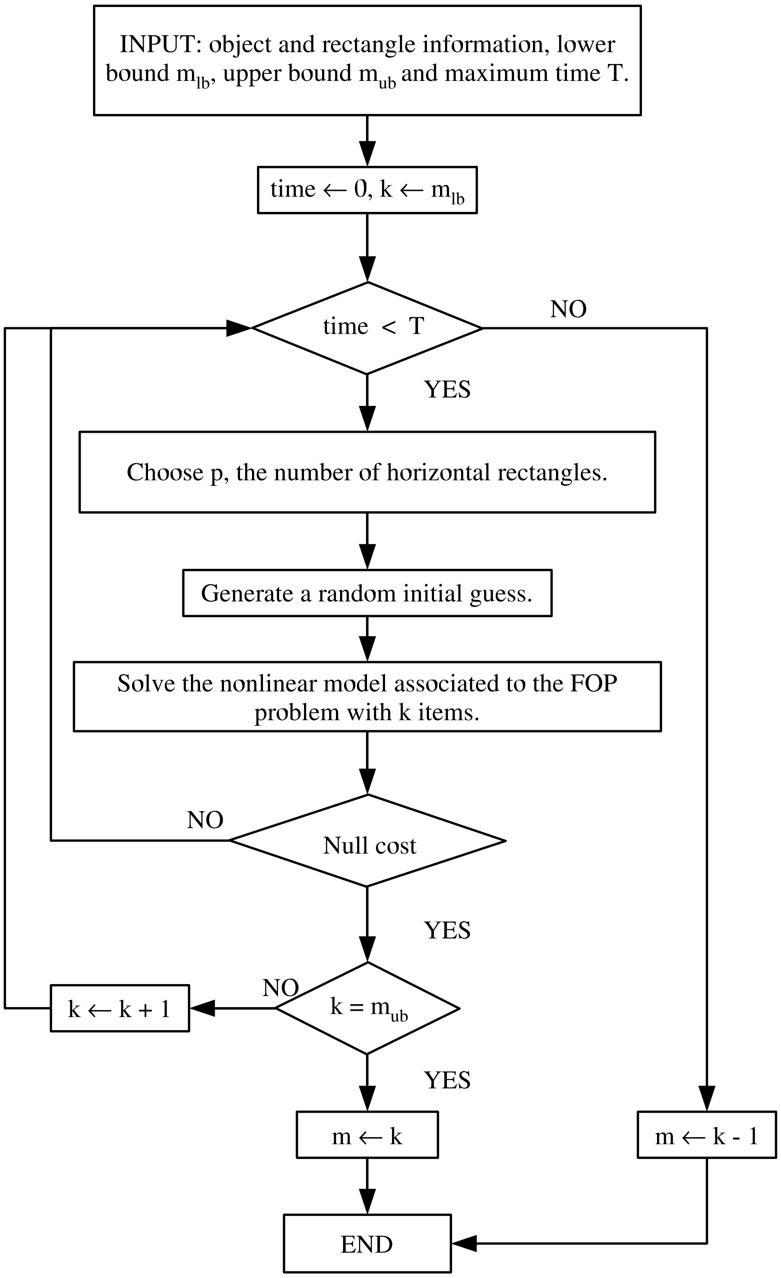

Figure 1: Fluxogram of the algorithm. When the method stops, m represents the number of packed items. It is assumed that the value of "time" corresponds with the elapsed time since it was zeroed at the beginning of the process.

3 Numerical experiments

We consider the set of rectangle packing problems in arbitrary convex regions introduced in [6], where the possibility of arbitrary θ-rotations was considered. In addition, we considered an interesting problem in which the convex region is given by the Lamé curve. 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).

Problem	Convex Region		Rectangular item	
	Description	Area	Dimensions	Area
1	$g_1(x_1, x_2) = -x_1$ $g_2(x_1, x_2) = -x_2$ $g_3(x_1, x_2) = -x_1 - x_2 + 3$ $g_4(x_1, x_2) = x_1^2 + x_2^2 - 100$	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	$g_1(x_1, x_2) = -x_1$ $g_2(x_1, x_2) = x_1 - 8$ $g_3(x_1, x_2) = (x_1 - 6)^2 + x_2^2 - 81$ $g_4(x_1, x_2) = (x_1 - 1.7)^2 + (x_2 - 10)^2 - 81$	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	$g_1(x_1, x_2) = (x_1 - 3)^2/4 + (x_2 - 4)^2/16 - 1$ $g_2(x_1, x_2) = (x_1 - 2.65)^2/4 + (x_2 - 4)^2/16 - 1$ $g_3(x_1, x_2) = -x_1 + 1$ $g_4(x_1, x_2) = x_1 - x_2 - 1$ $g_5(x_1, x_2) = x_1 + x_2 - 9$	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	$g_1(x_1, x_2) = (x_1/6)^4 + (x_2/2)^4 - 1$ $g_2(x_1, x_2) = 8x_1 - 11x_2 - 26$	34.7	1.9×0.5	0.95
12	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(3/2 + \sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	32.2	1×1	1
13	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(2 + 4/\sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	33.3	1×1	1
14	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(3 + 4/\sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	36.3	1×1	1
15	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(2 + 2\sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	37.5	1×1	1
16	$g_1(x_1, x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(4 + 4/\sqrt{3})$ $g_2(x_1, x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1, x_2) = -x_2$	37.5	1×1	1

Table 1: Problems definition.

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 "-O4" 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 rectangles that were packed in [6] allowing arbitrary θ-rotations and the number of rectangles that were packed in this study considering only orthogonal rotations. Table 2 also shows 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.) Figure 2 illustrates the solutions.

Problem	Upper bound	Number of packed rectangles		Effort measurements	
		Using arbitrary rotations	Using only orthogonal rotations	Number of trials	CPU time in seconds
1	37	32	32	94921	6453.80
2	35	30	28	417	31.94
3	45	38	**40**	148870	19508.71
4	33	28	26	14734	485.29
5	40	35	33	4778	329.65
6	37	28	**30**	14573	780.77
7	25	20	19	6396	157.71
8	37	30	**32**	19422	880.40
9	28	23	22	6041	108.11
10	40	32	**34**	149935	5554.20
11	36	Not available	31	70613	4199.44
12	32	27	25	68	1.13
13	33	28	26	67	1.21
14	36	29	29	49	0.35
15	37	30	29	42	0.34
16	37	31	30	68	1.19

Table 2: Performance of the nonlinear orthogonal-packing procedure.

Table 2 deserves some explanations. Since the feasible set of the orthogonal packing problem is included in the feasible set of the free-rotations version of the problem, the number of orthogonal-packed rectangles should not be greater than the number of free-rotated-packed rectangles. This is confirmed in most of the considered problems with the exception of problems 3, 6, 8 and 10 in which the orthogonal packing approach found solutions with more rectangles.

In addition to the previous set of problems, and based on the small effort needed for solving problems 12–16, we also considered these problems but trying to pack a larger number of smaller rectangles. In particular, we consider the problem of packing rectangles two times smaller (dividing one of their sides by two). We call these problems 12'–16'. Table 3 shows the obtained results and Figure 3 illustrates the solutions. It is easy to see that the number of rectangles packed in problems 12–16 multiplied by two is a lower bound for the expected number of packed rectangles in problems 12'–16'. Note in Table 3 that these lower bounds were improved by a

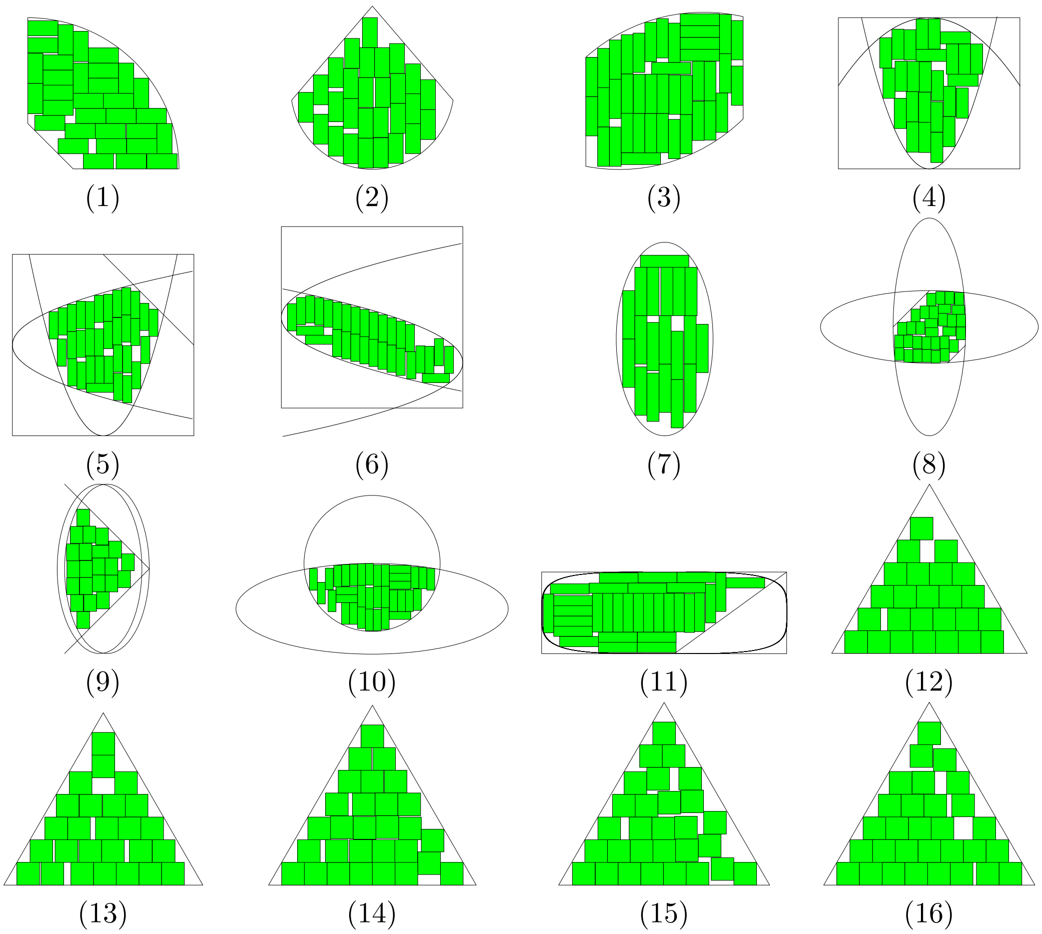

Figure 2: Graphical representation of the solutions. The pictures are automatically generated by the software using MetaPost.

number of rectangles between 4 and 6.

Finally, to give an idea of the amount of time that could be saved if a reasonably tight lower bound is available, Table 4 show the (relatively small) effort needed by the nonlinear orthogonal-packing procedure to find a "near-to-the-best" packing. On average, to find a packing with 2.62 items less than the best packing found requires less than 0.44% of the time.

4 Final remarks

This work presented a methodology, based on a feasibility problem and its nonlinear programming reformulation, to solve the problem of orthogonally packing rectangles in an arbitrary region. The numerical results show that this is a promising approach. Moreover, we provided a family of challenging global optimization problems that may be useful for future research on this subject.

Problem	Lower bound	Upper bound	Number of packed rectangles	Number of trials	CPU time in seconds
12’	50	64	55	114945	12812.07
13’	52	66	57	113685	13575.78
14’	58	72	62	4031	640.35
15’	58	74	64	89271	14811.55
16’	60	74	64	6138	1126.21

Table 3: Performance of the nonlinear orthogonal-packing procedure for problems 12’–16’ – packing a larger number of smaller rectangles.

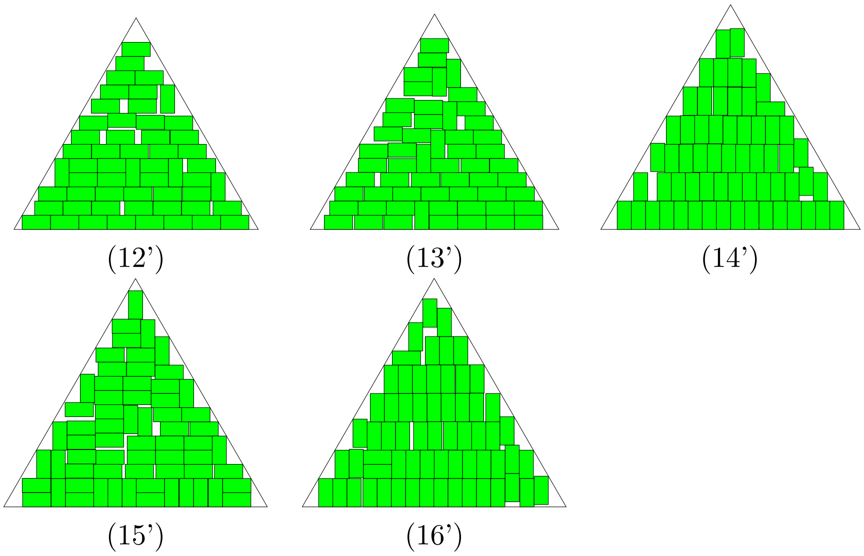

Figure 3: Graphical representation of the solutions for problems 12’–16’.

For the case of a packing problem which does not impose the condition that the sides of the rectangles must be parallel to some fixed axes or the case of a cutting problem of an isotropic material, the nonlinear formulations presented in this work are applicable but may give an incomplete answer. The whole package (formulations plus algorithms) can be extended to give a complete answer to the mentioned problems just adding a unique angle of rotation for all the rectangular items.

The complete Fortran 77 sources codes of the algorithms presented in this paper are available in www.ime.usp.br/∼egbirgin.

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

Problem	Difference to the best solution found	Number of trials	CPU time in seconds
1	-1	53	2.54
2	-1	57	1.61
3	-3	47	0.74
4	-1	298	6.58
5	-3	116	2.90
6	-4	133	3.66
7	-1	36	0.27
8	-3	177	2.75
9	-1	112	1.35
10	-3	376	6.94
11	-4	40	0.89
12	-1	30	0.13
13	-1	42	0.40
14	-1	47	0.27
15	-1	41	0.32
16	-1	51	0.34
12'	-2	316	22.40
13'	-3	615	71.68
14'	-2	452	39.65
15'	-2	665	95.09
16'	-3	659	96.93

Table 4: Effort measurements of the nonlinear orthogonal-packing procedure showing how relatively easy is to find a "near-to-the-best" packing.

References

[1] R. Andreani, E. G. Birgin, J. M. Martínez and J. Yuan, Spectral projected gradient and variable metric methods for optimization with linear inequalities, *IMA Journal of Numerical Analysis*, to appear.

[2] J. E Beasley, A population heuristic for constrained two-dimensional non-guillotine cutting, *European Journal of Operational Research* 156, pp. 601–627, 2004.

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

[4] E. G. Birgin, R. Castillo and J. M. Martínez, Numerical comparison of Augmented Lagrangian algorithms for nonconvex problems, *Computational Optimization and Applications*, to appear.

[5] 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.

[6] E. G. Birgin, J. M. Martínez, W. F. Mascarenhas and D. P. Ronconi, Method of Sentinels for Packing Objects whitin Arbitrary Regions, submitted.

[7] E. G. Birgin, J. M. Martínez and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization* 10, pp. 1196–1211, 2000.

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

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

[10] E. G. Birgin, R. Morabito and F. H. Nishihara, A note on an L-approach for solving the manufacturer's pallet loading problem, *Journal of the Operational Research Society*, to appear.

[11] J. E. Dennis Jr., R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewoods Cliffs, 1983.

[12] K. A. Dowsland, An exact algorithm for the pallet loading problem, *European Journal of Operational Research* 84, pp. 78–84, 1987.

[13] H. Dyckhoff, G. Scheithauer and J. Terno, Cutting and packing: An annotated bibliography. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pp. 393–412, Wiley, 1997.

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

[15] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, California, London, Amsterdam, Ontario, Sydney, 1984.

[16] A. Lodi, S. Martello and M. Monaci, Two-dimensional packing problems: a survey, *European Journal of Operational Research* 141, pp. 241–252, 2003.

[17] J. M. Martínez and L. Martínez, Packing optimization for automated generation of complex system's initial configurations for molecular dynamics and docking, *Journal of Computational Chemistry* 24, pp. 819–825, 2003.

[18] N. Mladenović, F. Plastria and D. Urošević, Reformulation descent applied to circle packing problems, *Computers & Operations Research*, to appear.

[19] R. Morabito and S. Morales, A simple and effective recursive procedure for the manufacturer's pallet loading problem, *Journal of the Operational Research Society* 49, pp. 819–828, 1998.

[20] Y. G. Stoyan and G. Yaskov, A mathematical model and a solution method for the problem of placing various-sized circles into a strip, *European Journal of Operational Research* 156, pp 590–600, 2004.

[21] A. Tarnowski, J. Terno and G. Scheithauer, A polynomial-time algorithm for the guillotine pallet loading problem, *INFOR* 32, pp. 275–287, 1994.

[22] H. Wang, W. Huang, Q. Zhang and D. Xu, An improved algorithm for the packing of unequal circles within a larger containing circle, *European Journal of Operational Research* 141, pp 440–453, 2002.

Ernesto G. Birgin is an Associate Professor in the Department of Computer Science at the University of São Paulo. He received his Ph.D. from the State University of Campinas. His research interests are numerical methods for nonlinear programming and their application to practical problems. He has previously published in ACM Transactions on Mathematical Software, Computational Optimization and Applications, Computing, European Journal of Operational Research, IMA Journal on Numerical Analysis, Journal of Computational Physics, Numerical Algorithms, SIAM Journal of Optimization and SIAM Journal on Scientific Computing, among others.

José Mario Martínez is Professor at the Department of Applied Mathematics of the State University of Campinas. His research interests lie in numerical optimization, theory and applications.

Fabio H. Nishihara is a graduate student in Computer Science at the University of São Paulo.

Débora P. Ronconi is an Assistant Professor in the Department of Production Engineering at the University of São Paulo. She received her M.Sc. and Ph.D. in Operational Research from the State University of Campinas. Her research interests include combinatorial optimization and scheduling of manufacturing systems. She has previously published in Annals of Operations Research, Computers and Operations Research, European Journal of Operational Research, International Journal of Production Economics, International Transactions in Operational Research and Journal of the Operational Research Society, among others.