Orthogonal packing of identical rectangles within isotropic convex regions*

Ernesto G. Birgin† Rafael D. Lobato†

June 18, 2010

Abstract

A mixed integer continuous nonlinear model and a solution method for the problem of orthogonally packing identical rectangles within an arbitrary convex region are introduced in the present work. The convex region is assumed to be made of an isotropic material in such a way that arbitrary rotations of the items, preserving the orthogonality constraint, are allowed. The solution method is based on a combination of branch and bound and active-set strategies for bound-constrained minimization of smooth functions. Numerical results show the reliability of the presented approach.

Key words: Packing and cutting of rectangles, orthogonal packing, isotropic convex regions, feasibility problems, nonlinear programming, models

1 Introduction

Many packing and cutting problems can be adequately modeled by nonlinear programming (NLP). On the one hand, nonlinearities can easily handle the essence of the original problem, while discrete models may be such that their solutions are only approximations to the solutions of the original problem. On the other hand, in most cases, the global solution of the NLP models may be required, and finding the global solution of general NLP problems is a difficult task for which much research is expected in the following years (see, for example, [4, 17] and the references therein). Whether or not dealing with nonlinearities is profitable depends on the packing problem at hand.

Packing problems for which nonlinear approaches proved to be effective and efficient include the packing of circles (or cylinders) and spheres [5, 13, 15], and the packing of rectangles within arbitrary convex regions with a variety of positioning constraints [9, 10, 27]. In [25, 26], Packmol was introduced as successful tool for building initial configurations for molecular dynamics simulations, based on the packing of spheres by nonlinear optimization. Hybrid methods that combine nonlinear models with heuristics have also been considered in [21, 22, 24, 28, 31, 32],

*This work was partly supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.510/2006 - APQ1), FAPESP (Grants 2006/53768-0, 2006/57984-6 and 2006/58313-0) and CNPq.

†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 ({egbirgin | lobato}@ime.usp.br).

among others, for packing identical or different circular pieces within several types of objects. In [36], mixed integer linear and nonlinear formulations for staged cutting problems and 2-stage two-dimensional guillotine cutting patterns are reviewed. In [29, 34, 35], mixed integer nonlinear models for cutting problems based on p-group cutting patterns are introduced. The introduced models are linearized by classical techniques – paying the price of adding new binary variables to the models – and then solved with the modelling language GAMS and the solver CPLEX.

In the present work, we deal with the problem of packing (or cutting) identical rectangular items within an arbitrary convex two-dimensional object. The object is made of an isotropic material and therefore it does not impose any constraints on the orientation of the items being packed. The cutting process, however, requires the items to satisfy an orthogonality constraint, i.e. only rotations of ninety degrees with respect to a unique angle of rotation for all the items are allowed. The unique angle of rotations (other than the ninety degrees rotations) for the whole set of items can also be regarded as a rotation of the object. Given the object and the dimensions of the identical items, the goal is to pack as many items as possible within the object and without overlapping. The problem can be classified as "two-dimensional, rectangular identical item packing problem (IIPP)" according to the typology introduced in [33].

The present approach is related to two previously published works [9, 10] that also deal with the packing of identical rectangles within arbitrary convex regions. The difference between the present approach and previous ones pertains to the positioning constraints applied to the rectangular items. In [10], the orientation of the object and the items is fixed and only ninety-degree rotations are allowed. In this sense, the decision space of the problem tackled in the present research is larger and, as a result, solutions with more packed items are expected. On the other hand, the problem presented in [9] considers an individual angle of rotation for each rectangle. The overlapping of such rectangles is a hard-to-model computational geometry problem that was accomplished using continuous and differentiable constraints, with the help of the Sentinels concept [9, 27]. While this is the model whose global optimal solution implies the largest number of packed items, the difficulty in finding such global optimal solution may result in poor quality solutions for the associated packing problem. Using the model introduced in the present work may be useful, even when the orthogonality constraint between the items is not imposed by the real packing problem. Figure 1 illustrates the kind of solutions found by the three different approaches.

A nonlinear model and a solution method are introduced in the present paper. For a given number of items, the packing model consists of minimizing the overlapping between the items subject to being accommodated within the object. The objective function is continuous and differentiable with respect to the continuous variables and there are integrality constraints in a subset of the decision variables. The solution method is based on a combination of branch and bound and a modern active-set strategy for bound-constrained minimization of smooth functions [8]. In order to be able to apply such techniques, a desirable property of the introduced model is pursued: the relaxed version of the model (ignoring the integrality constraints) to be continuous and differentiable with respect to *all* its decision variables. For packing as many items as possible, problems with an increasing number of items are considered.

The paper is organized as follows. In Section 2, the mixed integer continuous nonlinear model is derived. Section 3 describes the solution method. Numerical results are presented and analyzed in Section 4. Section 5 summarizes the conclusions and includes some lines for future

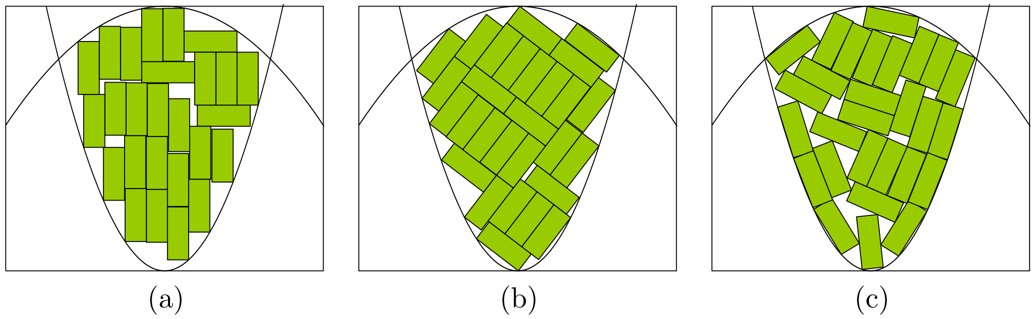


Figure 1: The three problems consist in packing identical rectangular items within a convex region, but different positioning constraints are imposed on the items. Graphics (a) and (b) represent the case in which an orthogonality constraint is imposed to the items. (a) deals with an anisotropic object while the isotropic-object case is represented in (b). In (c), the case of an isotropic object without any positioning constraint imposed on the items is illustrated. The packing in (a) contains 26 rectangles, while the packings in (b) and (c) contain 28 rectangles.

research.

2 Mixed integer continuous nonlinear model

Let $\Omega = \{x \in \mathbb{R}^2 \mid g_j(x) \leq 0,\ j = 1, \ldots, m\}$ be a convex subset of $\mathbb{R}^2$. For all $k = 1, \ldots, N$, consider a rectangle $R(a^k, b^k)$ centered at the origin of the two-dimensional Cartesian coordinate system with $a^k, b^k > 0$ being the fixed values of its horizontal and vertical sides, respectively. Assume that we want to place those N rectangles in such a way that the interior of the intersection of any pair of different rectangles is empty and they are contained in Ω. (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.) Moreover, assume that an orthogonality constraint is imposed on any pair of rectangles, i.e. sides of any two different rectangles must be parallel or perpendicular to each other.

Let $\theta \in \mathbb{R}$ be a variable anticlockwise rotation angle common to all the N rectangles. Let $C^k \in \mathbb{R}^2$ be the variable center of $R(a^k, b^k)$, and let $p^k \in \{0, 1\}$ be a binary variable, which indicates whether an extra ninety-degree rotation is being applied to $R(a^k, b^k)$ ($p^k = 1$) or not ($p^k = 0$), independently of the common rotation angle θ. Then, the problem, called PackN from now on, consists of finding values for the $3N + 1$ variables $\theta \in \mathbb{R}$, $C^k \in \mathbb{R}^2$ and $p^k \in \{0, 1\}$, for $k = 1, \ldots, N$, such that the N rectangles $R(a^k, b^k)$ – with displacements C^k, ninety-degree rotations represented by p^k and the common rotation given by θ – are contained in Ω without overlapping.

Given a rectangle $R(a, b)$ with horizontal side a and vertical side b and $p \in [0, 1] \subset \mathbb{R}$, define the length and the height as

$$\ell(a, b, p) = (1 - p)\, a + p\, b \quad \text{and} \quad h(a, b, p) = (1 - p)\, b + p\, a, \quad k = 1, \ldots, N. \tag{1}$$

Note that if $p = 0$ then the rectangle with length $\ell(a, b, p)$ and height $h(a, b, p)$ coincides with $R(a,b)$. If $p = 1$, however, the rectangle with length $\ell(a,b,p)$ and height $h(a,b,p)$ coincides with a ninety-degree rotation of $R(a,b)$, i.e. coincides with $R(b,a)$.

Let $Q(\theta)$ be the anticlockwise rotation matrix

$$Q(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{2}$$

Using (1–2) and considering an angle of rotation θ, a displacement C with respect to the origin and an orthogonal rotation represented by $p \in \{0,1\}$, it becomes clear that the four vertices of a rectangle $R(a,b)$ are given by:

$$\begin{aligned} V_{\rm sw}(R(a,b),C,p,\theta) &= C + Q(\theta) \begin{pmatrix} -0.5\ \ell(a,b,p) \\ -0.5\ h(a,b,p) \end{pmatrix}, \\ V_{\rm se}(R(a,b),C,p,\theta) &= C + Q(\theta) \begin{pmatrix} 0.5\ \ell(a,b,p) \\ -0.5\ h(a,b,p) \end{pmatrix}, \\ V_{\rm ne}(R(a,b),C,p,\theta) &= C + Q(\theta) \begin{pmatrix} 0.5\ \ell(a,b,p) \\ 0.5\ h(a,b,p) \end{pmatrix}, \\ V_{\rm nw}(R(a,b),C,p,\theta) &= C + Q(\theta) \begin{pmatrix} -0.5\ \ell(a,b,p) \\ 0.5\ h(a,b,p) \end{pmatrix}. \end{aligned} \tag{3}$$

Referring to the four vertices of rectangle $R(a^k,b^k)$ as $V_i^k \equiv V_i(R(a^k,b^k),C^k,p^k,\theta)$ for all $i \in D = \{\rm sw, se, ne, nw\}$ and $k = 1,\dots,N$, the constraints that state the rectangles must belong to Ω can be modeled as

$$V_i^k \in \Omega \text{ for all } i \in D \text{ and } k = 1,\dots,N$$

or

$$g_j(V_i^k) \le 0 \text{ for all } i \in D, j = 1,\dots,m, \text{ and } k = 1,\dots,N \tag{4}$$

plus $p^k \in \{0,1\}$, for $k = 1,\dots,N$. Provided $g_j(\cdot)$, $j = 1,\dots,m$, are continuous and differentiable functions and by the differentiability of (1–3), we have that constraints (4) are continuous and differentiable with respect to the decision variables C^k, p^k and θ.

Consider a pair of rectangles $R(a^{k_1},b^{k_1})$ and $R(a^{k_2},b^{k_2})$ with displacements C^{k_1} and C^{k_2}, orthogonal rotations represented by $p^{k_1},p^{k_2} \in \{0,1\}$, and a common angle of rotation θ. The horizontal and vertical coordinate-wise distances with respect to the θ anticlockwise rotation of the Cartesian system of coordinates, between the displacements C^{k_1} and C^{k_2} are given by $|d(C^{k_1},C^{k_2},\theta)_x|$ and $|d(C^{k_1},C^{k_2},\theta)_y|$, respectively, where

$$d(C^{k_1},C^{k_2},\theta) \equiv \begin{pmatrix} d(C^{k_1},C^{k_2},\theta)_x \\ d(C^{k_1},C^{k_2},\theta)_y \end{pmatrix} = Q(\theta)^T (C^{k_1} - C^{k_2}).$$

Then, the non-overlapping constraint between $R(a^{k_1},b^{k_1})$ and $R(a^{k_2},b^{k_2})$ can be modeled as

$$\begin{gathered} |\ d(C^{k_1},C^{k_2},\theta)_x\ | \ge (\ell(a^{k_1},b^{k_1},p^{k_1}) + \ell(a^{k_2},b^{k_2},p^{k_2}))/2 \\ \text{or} \\ |\ d(C^{k_1},C^{k_2},\theta)_y\ | \ge (h(a^{k_1},b^{k_1},p^{k_1}) + h(a^{k_2},b^{k_2},p^{k_2}))/2. \end{gathered} \tag{5}$$

Squaring both sides of inequalities in (5), substituting $t \le 0$ by $\max\{0,t\}^2 = 0$ and replacing ($t_1 = 0$ or $t_2 = 0$) by $t_1 \times t_2 = 0$, the non-overlapping constraints between every pair of rectangles can be written as the continuous and differentiable constraints

$$\begin{array}{rcl} \max\left\{0, (\ell(a^{k_1}, b^{k_1}, p^{k_1}) + \ell(a^{k_2}, b^{k_2}, p^{k_2}))^2/4 - [d(C^{k_1}, C^{k_2}, \theta)_x]^2\right\}^2 & \times & \\ \max\left\{0, (h(a^{k_1}, b^{k_1}, p^{k_1}) + h(a^{k_2}, b^{k_2}, p^{k_2}))^2/4 - [d(C^{k_1}, C^{k_2}, \theta)_y]^2\right\}^2 & = & 0, \\ \text{for all } k_1 = 1, \dots, N-1 \text{ and } k_2 = k_1 + 1, \dots, N, & & \end{array} \tag{6}$$

and $p^k \in \{0,1\}$, for $k = 1, \dots, N$. Note that squaring the maximum with zero is necessary to obtain differentiability.

So, problem PackN can be modeled as a continuous and differentiable feasibility problem given by (4) and (6) plus the constraints $p^k \in \{0,1\}$, for $k = 1, \dots, N$. Solving the feasibility problem is equivalent to finding a global minimizer with zero-valued objective function of the optimization problem

$$\text{minimize } f(\theta, C^1, \dots, C^N, p^1, \dots, p^N) \text{ subject to } p^k \in \{0,1\} \text{ for } k = 1, \dots, N, \tag{7}$$

where

$$\begin{array}{l} f(\theta, C^1, \dots, C^N, p^1, \dots, p^N) \;=\; \sum_{k=1}^{N} \sum_{j=1}^{m} \sum_{i \in D} \max\{0, g_j(V_i^k)\}^2 + \\ \sum_{k_1=1}^{N-1} \sum_{k_2=k_1+1}^{N} \quad \max\left\{0, (\ell(a^{k_1}, b^{k_1}, p^{k_1}) + \ell(a^{k_2}, b^{k_2}, p^{k_2}))^2/4 - [d(C^{k_1}, C^{k_2}, \theta)_x]^2\right\}^2 \times \\ \qquad\qquad\qquad\qquad \max\left\{0, (h(a^{k_1}, b^{k_1}, p^{k_1}) + h(a^{k_2}, b^{k_2}, p^{k_2}))^2/4 - [d(C^{k_1}, C^{k_2}, \theta)_y]^2\right\}^2 . \end{array} \tag{8}$$

Note that non-identical rectangles are being considered. Hence, the presented model can be used to pack a given fixed set of non-identical rectangles within a convex region. Considering that all rectangles are identical, the packing problem of packing as many identical rectangles as possible can be modeled as finding the largest integer value of N such that the minimum of (7–8) is equal to zero (or such that the feasibility problem given by (4) and (6) plus $p^k \in \{0,1\}$, for $k = 1, \dots, N$, is solvable).

3 Solution method

In this section, we describe a branch and bound method to solve problems of the form:

$$\text{minimize } f(x) \text{ subject to } \ell \le x \le u,\; x_i \in \mathbb{Z} \text{ for all } i \in I,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth nonlinear and generally nonconvex function, $I \subseteq \{1, \dots, n\}$ is the set of indices of the variables with integrality constraints, and $-\infty < \ell_i \le u_i < \infty, \forall\, i \in I$. In the branch and bound algorithm, each node of the tree corresponds to a subproblem, which is defined by a mixed integer bound-constrained minimization problem. The *relaxed* subproblem associated with a subproblem is defined as the subproblem itself without its integrality constraints. In other words, relaxed subproblems are bound-constrained minimization problems.

In the search tree, a node is *fathomed* in three different situations: **S1.** The associated subproblem is infeasible (it can be trivially checked); **S2.** The optimal solution of the relaxed associated subproblem satisfies the integrality constraints (and, therefore, there is no further need for branching); and **S3.** The optimal value of the relaxed associated subproblem, that is a lower bound on the optimal value of the (non-relaxed) associated subproblem, is greater than or equal to the value of the current incumbent solution.

The selection of a node to solve follows the depth-first rule. When a node is *branched*, two nodes are generated by splitting a bound constraint. The selection of the variable whose bound will be splitted follows a rule based on pseudocosts [3]. Pseudocost is a measure associated with each variable of a problem. It aims to quantify the importance of a variable within a problem and to predict the deterioration in the optimal value of the problem when the range of variation of the variable is reduced.

Consider that a node $\mathcal{N}$ is selected to be solved. The first step is to solve its relaxed subproblem. Let $\hat{x}^{\mathcal{N}}$ be the solution of the relaxed subproblem. If by S1–S3 the node can be fathomed, we are done. Otherwise, we have that $\hat{I} = \{i \in I \mid \hat{x}_i^{\mathcal{N}} \notin \mathbb{Z}\} \neq \emptyset$. Let us assume that, by a rule that will be detailed below, x_i with $i \in \hat{I}$ is selected to have its bound splitted in the branching process. In the two new nodes $\mathcal{N}_D$ and $\mathcal{N}_U$, the subproblem bound constraint $\ell_i \leq x_i \leq u_i$ is replaced by $\ell_i \leq x_i \leq \lfloor \hat{x}_i^{\mathcal{N}} \rfloor$ and $\lceil \hat{x}_i^{\mathcal{N}} \rceil \leq x_i \leq u_i$, respectively. Let now $\hat{x}^{\mathcal{N}_D}$ and $\hat{x}^{\mathcal{N}_U}$ be the solutions of the relaxed subproblems associated with nodes $\mathcal{N}_D$ and $\mathcal{N}_U$, respectively. In the two subproblems $\mathcal{N}_D$ and $\mathcal{N}_U$, variable x_i had its range reduced with respect to its range in subproblem $\mathcal{N}$. There are many ways of computing pseudocosts for x_i. In the present work, following the suggestion given in [16], *local* "down" and "up" pseudocosts for x_i are computed as

$$\delta_i^D = \frac{f(\hat{x}^{\mathcal{N}_D}) - f(\hat{x}^{\mathcal{N}})}{\hat{x}_i - \lfloor \hat{x}_i \rfloor} \quad \text{and} \quad \delta_i^U = \frac{f(\hat{x}^{\mathcal{N}_U}) - f(\hat{x}^{\mathcal{N}})}{\lceil \hat{x}_i \rceil - \hat{x}_i}.$$

For those variables x_i with $i \in I$ for which at least a local down (up) pseudocost was computed, their *global* down (up) pseudocosts Δ_i^D (Δ_i^U) are given by the average of its local down (up) pseudocosts. Global down (up) pseudocosts of variables whose local down (up) pseudocosts have not been computed are defined as the average of the global down (up) pseudocosts of the other variables.

The rule for selecting a variable, whose explanation was delayed in the previous paragraph, is based on the global pseudocosts (see [3]) and merely says to select a variable x_{i^*} such that

$$i^* = \arg\max_{i \in I} \{\min\{\Delta_i^D, \Delta_i^U\}\}.$$

In case of tie, we select the variable with the smallest index. The expectation is that new generated nodes with a large deterioration in their optimal values will be rapidly fathomed.

The incumbent solution is updated considering the solutions of the relaxed subproblems that satisfy the integrality constraints (i.e., that are also solutions of the corresponding subproblem). When the solution of a relaxed subproblem does not satisfy the integrality constraints, its rounded counterpart is considered. Rounding each component to its nearest feasible integer value is trivial and provides a feasible solution for the subproblem that may improve the incumbent solution.

In S3 it is stated that a node is fathomed if the optimal value of the relaxed subproblem is greater than or equal to the value of the incumbent solution. This fathoming rule is correct only if by optimal value of the relaxed subproblem we mean *global* optimal value, which can only be easily computed for convex objective functions. This is not the case of the packing problem being considered. To overcome this inconvenience, we use two combined strategies:

Multistart: By running a local bound-constraints minimization solver from $N_{\text{multi}} \geq 1$ different initial points and considering the solution to be the best local minimizer so far obtained, we aim to enhance the probability of finding global solution of relaxed subproblem.

Untightened fathoming: Let f_{best} be the value of the incumbent solution and let $\mathcal{N}$ and $\hat{x}^{\mathcal{N}}$ be a node and the best local solution obtained for its relaxed subproblem, respectively. By S3, $\mathcal{N}$ should be fathomed if $f(\hat{x}^{\mathcal{N}}) \geq f_{\text{best}}$. We consider an untightened version of that condition. The untightened condition uses a known lower bound $\hat{f}_{\text{lb}}$ on the optimal value of the relaxed subproblem and a parameter $\alpha \in [0,1]$ that expresses the degree of confidence on finding the global solution of the relaxed subproblem. The untightened condition states that node $\mathcal{N}$ should be fathomed if

$$\alpha f(\hat{x}^{\mathcal{N}}) + (1-\alpha)\hat{f}_{\text{lb}} \geq f_{\text{best}}. \tag{9}$$

Using $\alpha = 1$, nodes are fathomed as if the global solutions of the relaxed subproblems were being computed. Using $\alpha = 0$, inequality (9) is reduced to $\hat{f}_{\text{lb}} \geq f_{\text{best}}$. In this case, a node is fathomed only if a known lower bound for the node relaxed subproblem guarantees that the node is useless.

We use Gencan [8], an active-set method for bound constrained local minimization for solving the branch and bound relaxed subproblems. Gencan adopts the leaving-face criterion presented in [7] that employs the spectral projected gradients defined in [11, 12]. For the internal-to-the-face minimization, Gencan uses a general algorithm with a line search that combines backtracking and extrapolation. As we computed first and second derivatives of problem (7–8), each step of Gencan computes the direction inside the face using the Newton direction and subroutine MA57 from HSL for solving the linear systems. For a description of basic techniques of continuous optimization and active-set methods see, for example, [18] and [23] (pp. 326–330). For a publicly available version of Gencan, see the Tango Project web site `http://www.ime.usp.br/~egbirgin/tango/`.

4 Implementation details and numerical experiments

We implemented the branch and bound method described in Section 3 that uses the continuous bound-constraints minimization solver Gencan for solving the node relaxed subproblem. Codes are in Fortran 77 and Fortran 90. They were compiled with gfortran (GNU Fortran version 4.2.1) and the compiler option "-O3" was adopted. All the experiments were run on a 2.4GHz Intel Core2 Quad Q6600 processor, 4Gb of RAM memory and Linux operating system.

We consider the same set of rectangular items and convex regions considered in [10] and [9]. For the sake of completeness[1], Table 1 shows the description of each problem (inequalities that describe the convex region Ω, dimensions of the rectangular items, and areas of the convex regions and the rectangles).

[1]There are a few typographical errors in the definition of $g_1(\cdot)$ for problems 12–16 in [10] and [9].

There is a trade-off between computational cost and losing the optimal solution by wrongly fathoming a node in the selection of the untightened S3 threshold parameter $\alpha \in [0,1]$. Moreover, the probability of finding the global solution of a relaxed subproblem is enhanced by solving each relaxed subproblem starting from $N_{\text{multi}} \geq 1$ different initial points. Preliminary numerical experiments varying $\alpha \in \{1, 10^{-1}, 10^{-2}, \ldots\}$ and $N_{\text{multi}} \in \{5, 100\}$ showed that combination $\alpha = 10^{-4}$ and $N_{\text{multi}} = 5$ allows the method to find solutions at least as good as the ones reported in [10]. The untightened version of fathoming rule S3 requires a lower bound $\hat{f}_{\text{lb}}$ for the optimal value of the node relaxed subproblems. We use $\hat{f}_{\text{lb}} = 0$. It is easy to see that the objective function (8) of the problem at the root node, which coincides with the objective function of the subproblems and the relaxed subproblems, is greater than or equal to zero as it is a sum of squares.

The branch and bound scheme also considers a lower bound f_{lb} for the optimal value of the problem at the root node. The whole search process is stopped if the value of the incumbent solution achieves the given lower bound. As explained in the previous paragraph, zero is the natural candidate for f_{lb}. This lower bound is tight only if a feasible solution for the packing problem exists. In the numerical experiments, we set $f_{\text{lb}} = 10^{-8}$. It means that if a feasible point x^* of the problem at the root node such that the objective function (8) evaluated at x^* is smaller than or equal to 10^{-8} is found, the method will be stopped and x^* will be returned as a solution of the packing problem given by (7–8). In other words, it means that a packing with N rectangles was found. Otherwise, if the method stops with an incumbent solution whose value is larger than 10^{-8}, it will be said that a packing with N rectangles was not found.

The whole process starts by trying to solve problem (7–8) with $N = 1$. If a solution of the packing problem is found, we set $N \leftarrow N + 1$ and we try again. The process stops when a packing with N rectangles cannot be found, and the solution found for the packing problem with $N^* = N - 1$ is considered as the solution of packing as many rectangles as possible. In [10], an explanation empirically confirmed with numerical experiments justifies the use of this kind of sequential process of increasing N one by one instead of other strategies such as bisection. Roughly speaking, packing a few less rectangles than the maximum capacity of the object is a very easy task, while packing the optimal quantity or trying to pack more than the maximum object load are very time-consuming problems. In the "easy cases", the branch and bound is rapidly terminated by achieving the known zero-valued lower bound on the optimal value, while in the "hard cases" the branch and bound search tree is fully explored. As in [10], the whole process of increasing N can be stopped if a known upper bound for its value is achieved, but this is never the case for *arbitrary* convex regions, where upper bounds based on a quotients of areas are never tight. The process can also be started from a lower bound for N different from 1 as suggested at the beginning of this paragraph. As explained above, however, the first problems (with small values for N) are simple and solving them or not makes no difference.

Table 2 shows, for each problem, the number of rectangles that were packed in [9] allowing arbitrary rotations, the number of rectangles that were packed in [10] allowing only ninety-degree rotations, and the number of rectangles that were packed in this study (allowing ninety-degree rotations and an extra common angle of rotation for the whole set of rectangles). On the one hand, the results obtained solving the model introduced in the present approach are expected to have at least as many packed items as the ones obtained in [10], where no rotations are allowed. The experiments confirm that expectation: the same number of items was packed in problems

1, 3, 7, 10, 11, 12, 13, 14, 15 and 16, while one or two more items were packed for the remaining problems. On the other hand, the comparison is not as clear with the free-rotations model introduced in [9]. While the present model has a smaller feasible set, it seems to be easier to find global solutions for larger values of N. We found solutions with the same number of items in problems 1, 2, 4, 6, 9 and 14 (even imposing the orthogonality constraint between the items), solutions with one or two less items in problems 5, 7, 12, 13, 15 and 16, and solutions with up to three more items in problems 3, 8 and 10. Figure 2 illustrates the solutions. Figure 3 compares the solutions found for problem 8 in [10], in [9] and in the present approach.

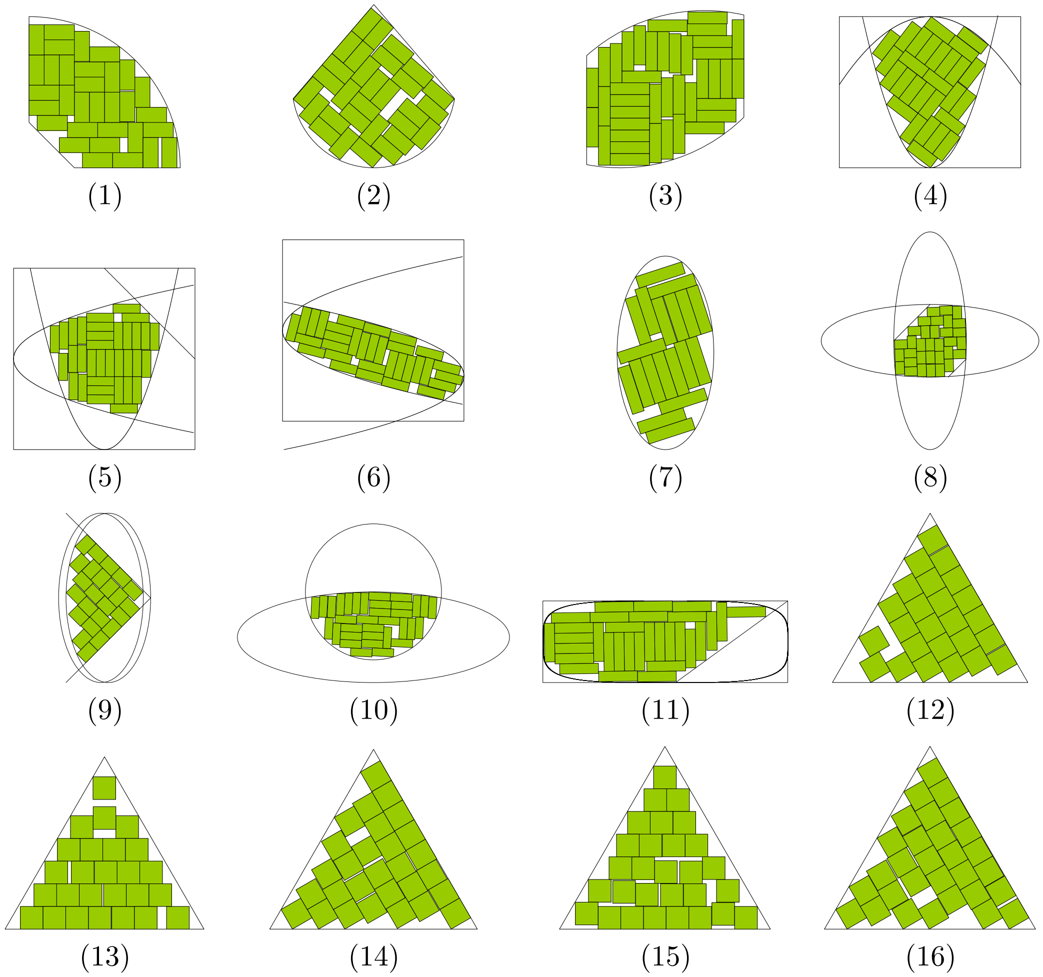
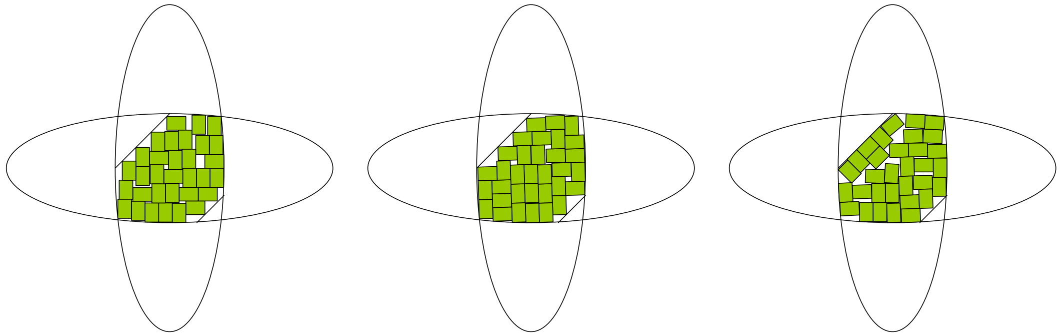


Figure 2: Graphical representation of the solutions.

A few words about the accuracy of the obtained solutions are in order. We considered solutions points such that the value of the objective function (8) is not greater than 10^{-8}. The objective function consists of the sum of two terms: one that penalizes the violation of the object constraints and other that penalizes the overlapping between the items. The independent values of those terms at the reported solutions are showed in Table 3. Irrespective of that, another



(a) 32 items (b) 33 items (c) 32 items

Figure 3: Graphical representation of the solutions found for a problem with the convex regions and the item dimensions of problem 8. In (a), the orthogonality constraint between the items is imposed and only ninety-degree rotations are allowed. It corresponds to the problem introduced in [10]. In (b), the orthogonality constraint is maintained and a common angle of rotation for all the items is added. It corresponds to the problem been tackled in the present work. Finally, (c) corresponds to the free-rotations model introduced in [9] based on the Sentinels concept. With the present approach, we were able to find a solution with one more item when compared to the other two approaches. Although it is hard to see from the picture, there is a small common angle of rotation of $\theta \approx 1.60$ degree for the items in (b).

measure of overlapping between the items is being computed. Clearly, the intersection between a pair of orthogonal rectangles is null or gives a rectangle. We will call this rectangle an "intersection rectangle". We computed the area of the intersection rectangles between every pair of rectangular items. The last column of Table 3 displays the area of the intersection rectangle with the largest area.

Finally, Table 4 shows some measures of the computational effort made by the branch and bound strategy to find the reported solutions. The last column shows the accumulated CPU time used by the method to solve problem (7–8) for increasing values of N starting from $N = 1$ and up to $N = N^*$. The other two columns show the number of explored nodes in the branch and bound tree and the corresponding CPU time, both related to the problem with $N = N^*$. Subtracting the time spent in the last problem from the accumulated CPU time, it is easy to see that, for the hardest instances 1, 3, 4, 5 and 11, solving problems with less than N^* items is very cheap. In general, to find a "near-to-the-best" packing is an easy task. This observation justifies, as already observed in [9, 10], the sequential strategy adopted to determine N^* instead of any other strategy based on bisection.

Last but not least, the present approach, as are the ones introduced in [9, 10], is suitable for packing rectangles within general convex regions. When the convex region takes the particular form of a rectangle, we are faced with the well known pallet loading problem [6, 14, 20, 30], for which dedicated solution methods exist. Numerical experiments presented in [9] show that nonlinear-based methods, such as the one presented here, are not competitive with clever methods developed for this particular case.

5 Conclusions and future work

The problem of orthogonally packing identical rectangles within isotropic convex regions was modeled as mixed integer continuous feasibility and optimization problems. A straightforward extension of a well established continuous bound-constrained minimization solver was developed to solve mixed integer nonlinear bound-constrained optimization problems. Its application to the packing problem models showed that the method is reliable. As a side result, the introduced models, together with the ones presented in [10] and [9], constitute a nice set of test problems for global mixed integer continuous solvers. As future work, two different topics deserve further investigation. First, strategies for eliminating the undesirable symmetry property [19] of the introduced packing models may be studied and incorporated into them. Second, the extension of Gencan for mixed integer bound-constrained problems can be incorporated in an augmented Lagrangian framework, like the one implemented in Algencan [1, 2], to solve mixed integer (general-constrained) nonlinear programming problems, obtaining a MINLP solver based on augmented Lagrangians.

Acknowledgements. The authors are also indebted to the anonymous referees whose comments helped to improve this paper.

Problem	Convex Region		Rectangular item	
	Description	Area	$a \times b$	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	$g_1(x_1,x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(4 + 8/\sqrt{3})$ $g_2(x_1,x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1,x_2) = -x_2$	32.2	1×1	1
13	$g_1(x_1,x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(3 + 10/\sqrt{3})$ $g_2(x_1,x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1,x_2) = -x_2$	33.3	1×1	1
14	$g_1(x_1,x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(8 + 2/\sqrt{3})$ $g_2(x_1,x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1,x_2) = -x_2$	36.3	1×1	1
15	$g_1(x_1,x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(9.302)$ $g_2(x_1,x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1,x_2) = -x_2$	37.5	1×1	1
16	$g_1(x_1,x_2) = \sqrt{3}x_1 + x_2 - \sqrt{3}(7 + 4/\sqrt{3})$ $g_2(x_1,x_2) = -\sqrt{3}x_1 + x_2$ $g_3(x_1,x_2) = -x_2$	37.5	1×1	1

Table 1: Definition of the problems.

Problem	Number of packed items		
	Orthogonal items		Free rotations [9]
	Only ninety-degree rotations [10]	ninety-degree rotations plus a common rotation angle	
1	32	32	32
2	28	30	30
3	40	40	37
4	26	28	28
5	33	34	35
6	30	32	32
7	19	19	20
8	32	33	32
9	22	24	24
10	34	34	32
11	31	31	–
12	25	25	27
13	26	26	28
14	29	29	29
15	29	29	30
16	30	30	31

Table 2: Number of packed items. When compared to the number of packed items in [10], where only orthogonal rotations are allowed, it can be seen that the extra common angle of rotation allows more items to be packed within the same object. Comparing these results with the number of packed items in [9], where no orthogonality constraint is imposed and each item has its own angle of rotation, it can be seen that, even with additional positioning constraints, the simplicity of the present model allows one to find better quality solutions in some cases.

Problem	Objective function value at the solution		Maximum overlapping area
	Constraints violation term	Overlapping violation term	
1	5.2E−13	2.4E−16	1.44432E−07
2	2.1E−11	1.1E−15	1.88327E−06
3	1.3E−13	1.1E−14	4.60587E−08
4	8.3E−14	2.0E−15	1.98382E−07
5	3.6E−10	6.2E−11	1.43665E−05
6	5.3E−17	1.2E−18	7.20967E−09
7	0.0E+00	1.4E−23	0.00000E+00
8	6.3E−14	6.4E−18	1.80352E−07
9	1.5E−17	0.0E+00	1.89104E−09
10	5.4E−17	5.5E−18	3.88860E−09
11	1.1E−12	2.1E−12	1.89224E−07
12	0.0E+00	8.0E−19	0.00000E+00
13	8.9E−19	1.1E−18	2.50940E−10
14	2.2E−17	0.0E+00	1.21657E−09
15	0.0E+00	5.0E−23	0.00000E+00
16	1.5E−23	0.0E+00	1.21323E−12

Table 3: Accuracy of the solutions.

Problem	Computational cost		
	Last problem (with N^* items)		Increasing values of N from 1 to N^*
	# nodes	CPU time (secs)	CPU time (secs)
1	39816	10350.08	10574.47
2	7	11.57	94.78
3	1482841	620836.38	622073.00
4	3090896	304453.28	304674.76
5	11128	4717.57	4953.22
6	2121	906.31	1056.20
7	46	3.35	6.21
8	11306	1462.63	1754.43
9	341	23.34	47.45
10	641	114.30	276.47
11	251245	64088.21	64298.39
12	6	0.35	1.60
13	2	0.18	1.27
14	1	0.33	1.78
15	5	3.09	4.34
16	3	1.58	2.89

Table 4: Computational cost of the branch and bound strategy. Note that solving the packing problem with N^* items is very expensive compared to solving all the other problems with less than N^* items.

References

[1] 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*, 18:1286–1309, 2007.

[2] 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*, 111:5–32, 2008.

[3] M. Benichou, J. M. Gauthier, P. Girodet, G. Hentges, G. Ribiere, and O. Vincent. Experiments in mixed-integer linear programming. *Mathematical Programming*, 1:76–94, 1971.

[4] E. G. Birgin, C. A. Floudas, and J. M. Martínez. Global minimization using an Augmented Lagrangian method with variable lower-level constraints. *Mathematical Programming*, 2009. to appear (DOI: 10.1007/s10107-009-0264-y).

[5] E. G. Birgin and J. M. Gentil. New and improved results for packing identical unitary radius circles within triangles, rectangles and strips. *Computers & Operations Research*, 37:1318–1327, 2010.

[6] E. G. Birgin, R. D. Lobato, and R. Morabito. An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. *Journal of the Operational Research Society*, 61:306–320, 2010.

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

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

[9] E. G. Birgin, J. M. Martínez, W. F. Mascarenhas, and D. P. Ronconi. Method of sentinels for packing items within arbitrary convex regions. *Journal of the Operational Research Society*, 57:735–746, 2006.

[10] E. G. Birgin, J. M. Martínez, F. H. Nishihara, and D. P. Ronconi. Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization. *Computers & Operations Research*, 33:3535–3548, 2006.

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

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

[13] 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*, 160:19–33, 2005.

[14] 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*, 56:1448–1451, 2005.

[15] E. G. Birgin and F. N. C. Sobral. Minimizing the object dimensions in circle and sphere packing problems. *Computers & Operations Research*, 35:2357–2375, 2008.

[16] J. Eckstein. Parallel Branch-and-Bound Algorithms for General Mixed Integer Programming on the CM-5. *SIAM Journal on Optimization*, 4:794–814, 1994.

[17] C. A. Floudas. *Deterministic global optimization: theory, methods and application.* Kluwer Academic Publishers, DorDrecht, Boston, London, 1999.

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

[19] L. Liberti. Reformulations in mathematical programming: Definitions and systematics. *RAIRO – Operations Research*, 43:55–85, 2009.

[20] 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*, 54:777–789, 2003.

[21] Jingfa Liu, Shengjun Xue, Zhaoxia Liu, and Danhua Xu. An improved energy landscape paving algorithm for the problem of packing circles into a larger containing circle. *Computers & Industrial Engineering*, 57:1144–1149, 2009.

[22] M. Locatelli and U. Raber. Packing equal circles in a square: a deterministic global optimization approach. *Discrete Applied Mathematics*, 122:139–166, 2002.

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

[24] C. D. Maranas, C. A. Floudas, and P. M. Pardalos. New results in the packing of equal circles in a square. *Discrete Mathematics*, 142:287–293, 1995.

[25] 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*, 24:819–825, 2003.

[26] L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of Computational Chemistry*, 30:2157–2164, 2009.

[27] W. F. Mascarenhas and E. G. Birgin. Using sentinels to detect intersections of convex and nonconvex polygons. *Computational & Applied Mathematics*, 2010. to appear.

[28] N. Mladenović, F. Plastria, and D. Urosevic. Reformulation descent applied to circle packing problems. *Computers & Operations Research*, 32:2419–2434, 2005.

[29] R. Morabito and M. N. Arenales. Optimizing the cutting of stock plates in a furniture company. *International Journal of Production Research*, 38:2725–2742, 2000.

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

[31] K. J. Nurmela and P. R. Östergård. Packing up to 50 equal circles in a square. *Discrete & Computational Geometry*, 18:111–120, 1997.

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

[33] G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183:1109–1130, 2007.

[34] H. H. Yanasse and R. Morabito. Linear models for one-group two-dimensional guillotine cutting problems. *International Journal of Production Research*, 44:3471–3491, 2006.

[35] H. H. Yanasse and R. Morabito. A note on linear models for two-group and three-group two-dimensional guillotine cutting problems. *International Journal of Production Research*, 46:6189–6206, 2008.

[36] H. H. Yanasse, R. Morabito, and M. N. Arenales. Linear and nonlinear models for staged two-dimensional guillotine cutting problems. In A. Bortfeldt, J. Homberger, H. Kopfer, G. Pankratz, and R. Strangmeier, editors, *Intelligent Decision Support. Current Challenges and Approaches. Part 1: Decision Support for Cutting and Packing Problems*, pages 69–88. Gabler-Verlag Springer, Wiesbaden, Germany, 2008.