Pesquisa Operacional (2022) 42: e260406 p.1-24
doi: 10.1590/0101-7438.2022.042.00260406

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

IMPROVED BIASED RANDOM KEY GENETIC ALGORITHM FOR THE TWO-DIMENSIONAL NON-GUILLOTINE CUTTING PROBLEM

Eliane Vendramini de Oliveira[1*] and Rubén Romero[2]

Received January 26, 2022 / Accepted June 29, 2022

ABSTRACT. The two-dimensional cutting problem has a direct relationship with industry problems. There are several proposals to solve these problems. In particular, solution proposals using metaheuristics are the focus of this research. Thus, in this paper, we present a specialized biased random key genetic algorithm. Several tests were performed using known instances in the specific literature, and the results found by the metaheuristics proposed were, in many cases, equal or superior to the results already published in the literature. Another comparison of results presented in this paper is related to the results obtained by specialized metaheuristics and the results found by a mathematical model using commercial software. Once again, in this case, the genetic algorithm presented results equal to or very close to the optimum found by the mathematical model. In addition, the optimization proposal was extended to two-dimensional non-guillotine cutting without parts orientation.

Keywords: Genetic algorithm, metaheuristics, two-dimensional cutting problem.

1 INTRODUCTION

In the stage of globalization which the markets currently are, products with competitive prices have been increasingly sought by industries. For that reason, we have invested in optimizing industrial processes to achieve the necessary competitiveness. Several types of problems arise when it comes to optimizing production processes, among them, there is the two-dimensional non-guillotine cutting problem.

The problem was introduced in literature by Gilmore and Gomory (1961). According to Alvarez-Valdes et al. (2007), it consists of cutting rectangular pieces from a larger rectangular plate, made of steel, cloth, paper, glass, or wood, or even placing advertisements on the pages of newspapers and magazines to maximize the total value of the pieces cut, or to minimize material waste if

*Corresponding author

[1] Department of Analysis and Systems Development, Sao Paulo State Technological College, Presidente Prudente, SP, Brazil.
Department of Electrical Engineering, Sao Paulo State University, Ilha Solteira, SP, Brazil – E-mail: eliane.oliveira5@fatec.sp.gov.br – http://orcid.org/0000-0001-9652-1385

[2] Department of Electrical Engineering, Sao Paulo State University, Ilha Solteira, SP, Brazil – E-mail: ruben.romero@unesp.br – http://orcid.org/0000-0002-7744-254X

the value of a part is taken to be proportional to its area (Hadjiconstantinou; Christofides, 1995), where only two dimensions are considered by the problem under study in which the laser cutting technique is used.

Often, what is observed in the industry are pieces with high added value that must be cut from a plate, but these isolated pieces are not the product ready to be delivered. For the product to be finished by the industry, this high-value piece should interact with other parts whose value is smaller, but no less important for the delivery of the final product. Thus, the raw material cutting industry is forced to cut the pieces of lesser value, and from such an impasse, the lower and upper bounds applied to the quantity of each piece to be cut arise. Although it is interesting for the industry to cut the most valuable parts, the constraints of the two bounds must be respected.

Among the cutting problems, there are those that are guillotined and those that are non-guillotined. In several industries, the cutting equipment operates in such a way that any cut made on a rectangle should be performed in a straight line from one edge of the plate to another. This type of cut is referred to as a guillotine cut (Lai, Chan, 1997). However, in some applications, the guillotine cut is not the most suitable one, making it possible to use another cutting technique called laser-cut, allowing non-guillotine cutting.

This paper aims to identify the application of a metaheuristic in the resolution of the two-dimensional non-guillotine cutting problem. Particularly, as a comparative basis, the papers analyzed were the ones by Alvarez-Valdes et al. (2007), in which the tabu search metaheuristic was used, by Lai and Chan (1997), in which the simulated annealing metaheuristic was chosen, and by Gonçalves and Resende (2013), in which the genetic algorithm metaheuristic was applied to solve the problem. As a specific objective, this work proposed to apply metaheuristics to solve the problem, which is the biased random key genetic algorithm (BRKGA), proposed by Gonçalves and Resende (2013), adding significant improvements to its implementation, and therefore to formulate a proposal that is better performing. A secondary objective is to show the performance of the optimization solver to solve the mathematical model of the problem.

There are several additional proposals in the specialized literature that address the two-dimensional non-guillotined cutting problem. Arenales and Morabito (1995) address the unconstrained problem using a branch and bound strategy by using AND/OR logic and using additional heuristic rules to reduce the search space. In Leung et al. (2001), the unconstrained problem is approached by using the genetic algorithm and simulated annealing metaheuristics for the problem with fixed orientation and by performing tests using eight instances. In Beasley (2004), the constrained problem without rotation is approached by using a population heuristic which is a modified genetic algorithm. In Baldaci and Boschetti (2007), a two-level proposal is presented in which a subset of parts to be allocated is selected at the first level, and the possibility of finding feasible solutions is verified at the second level. The proposal can improve the upper bounds of several complex instances. In Ayadi et al., a new implementation version of the particle swarm optimization (PSO) metaheuristics is proposed which incorporates a heuristic strategy existing in the specialized literature. In Wei et al. (2018), a proposal based on the best-fit branch and bound method is presented. This proposal is supported by a greedy heuristic to generate complete solution proposals from a partial solution within the branch and bound structure. Due to the diversity of variants used in tests, only a comparative analysis with some of these proposals is presented in the testing phase.

This paper is organized into sections: section 2 presents the mathematical model of the problem; section 3 describes the specialized BRKGA to optimize the two-dimensional non-guillotine cutting problem; in section 4, the results found in this work are presented and analyzed; section 5 describes the conclusions and final considerations of the paper.

2 MATHEMATICAL MODEL

Hadjiconstantinou and Christofides (1995) describe the mathematical model for the two-dimensional, orthogonal, non-guillotine, part-oriented, and constrained related to the upper bound problems in the following way:

Let $A_0 = (\alpha_0, \beta_0)$ be a rectangular plate with length α_0 and height β_0, and let R be a set of m smaller rectangular pieces $R_1, R_2, \ldots, R_m$ with dimensions (α_1, β_1), (α_2, β_2),..., (α_m, β_m) and with corresponding values $v_1, v_2, \ldots, v_m$ for each piece of R. The objective is to build a non-guillotine cutting pattern for A_0 with the highest total value using no more than Q_i copies of each piece R_i for all $i = 1, \ldots, m$. There is also a minimum requirement of P_i copies of each piece R_i to be used in the cutting pattern.

Hadjiconstantinou and Christofides (1995) assumed the following hypotheses: (i) All dimensions (α_i, β_i), for $i = 0, ..., m$, are assumed to be integers, therefore, cuts on the plates must be made in whole steps along the axes x or y. This limitation is not considered serious by the authors, since the real dimensions can be extended in practice; (ii) The orientation of the pieces is considered fixed, that is to say, a piece of length e and height f is different from a piece of length f and height e; (iii) Each piece should be cut with its edges parallel to the edges of the plate (orthogonal cuts).

To formulate the problem as a binary programming problem, the following was defined

$$L_i = \{x | 0 \leq x \leq \alpha_0 - \alpha_i, x integer\}$$

as the set of all possible points x along the length of A_0, so that any piece R_i of the set R can be cut from A_0 with its height parallel to the axis y and the cut piece has its lower-left corner in any $x \in L_i$. In the same way, the authors defined the following:

$$W_i = \{y | 0 \leq y \leq \beta_0 - \beta_i, y integer\}.$$

The sets L_i and W_i are defined for $i = 1, \ldots, m$. In addition, it is defined that:

$$x_{ijp} = \begin{cases} 1, if the\ j^{th} \text{ copy of the piece } i \text{ is cut with its lower-left corner at x-position p} \\ \text{where j =1,..., } \mathrm{Q_i} \text{ and p} \in \mathrm{L_i}, \\ 0, otherwise. \end{cases}$$

$$y_{ijq} = \begin{cases} 1, \textit{if the } j^{th} \text{ copy of the piece } i \text{ is cut with its lower-left corner at y-position q} \\ \text{where j =1,..., } \mathrm{Q_i} \text{ and q } \in \text{ W}_i.\square \\ 0, \textit{otherwise}. \end{cases}$$

$z_{rs} = 1 \textit{if the point} (r,s) \in A_0, \textit{where} r = 0,\ldots,\alpha_0 - 1 \wedge s = 0,\ldots,\beta_0 - 1$, was not cut by any piece of the set R.

The mathematical formulation of Hadjiconstantinou and Christofides (1995) takes the following form:

$$\textit{maximize } Z = \sum_{i=1}^{m} v_i \sum_{j=1}^{Q} \sum_{p \in L} x_{ijp} \tag{1}$$

subject to

$$\sum_{s=q}^{q+\beta_1-1} \sum_{r=p}^{p+\alpha_1-1} z_{rs} \leq (2 - x_{ijp} - y_{ijq})\, \alpha_i \beta_i, i = 1,\ldots,m, j = 1,\ldots,Q_i p \in L_i, q \in W_i, \tag{2}$$

$$\sum_{p \in L_i} x_{ijp} \leq 1, i = 1,\ldots,m, j = 1,\ldots,Q_i, \tag{3}$$

$$\sum_{p \in L_i} x_{ijp} = \sum_{q \in W_i} y_{ijq}, i = 1,\ldots,m, j = 1,\ldots,Q_i, \tag{4}$$

$$\sum_{i=1}^{m} \beta_i \sum_{j=1}^{Q_i} \sum_{p=r-\alpha_1+1, p \in L_i}^{r} x_{ijp} + \sum_{s=0}^{\beta_0-1} z_{rs} = \beta_0, r = 0,\ldots,\alpha_0 - 1, \tag{5}$$

$$\sum_{i=1}^{m} \alpha_i \sum_{j=1}^{Q_i} \sum_{q=s-\beta_1+1, q \in W_i}^{s} y_{ijq} + \sum_{r=0}^{\alpha_0-1} z_{rs} = \alpha_0, s = 0,\ldots,\beta_0 - 1, \tag{6}$$

$$x_{ijp}, y_{ijq} \in \{0,1\}, i = 1,\ldots,m, j = 1,\ldots,Q_i, p \in L_i, q \in W_i, \tag{7a}$$

$$z_{rs} \in \{0,1\}, r = 0,\ldots,\alpha_0 - 1, s = 0,\ldots,\beta_0 - 1. \tag{7b}$$

Constraint (2) ensures that any point in A_0 be cut in one piece at most. Constraints (3) and (4) express the fact that any piece is cut at most once in A_0. Constraint (5) ensures that not all pieces of the set R, whose sum of heights exceeds β_0, can be cut in A_0 with the same length, having the left lower corners of each piece as the origin. Likewise, constraint (6) ensures that if the sum of the lengths of the pieces exceeds α_0, then not all of these pieces can be cut into A_0 with the same height, having its lower-left corners as origin.

This model is a complete formulation of the two-dimensional cutting, orthogonal, non-guillotined problem involving approximately constraints and integer variables in the following way:

$$N_c = 2\sum_{i=1}^{m} Q_i + \sum_{i=1}^{m} Q_i \left|L_i\right| \left|W_i\right| + \alpha_0 + \beta_0 \tag{8}$$

and

$$N_v = \sum_{i=1}^{m} Q_i \left(|L_i| + |W_i| \right) + \alpha_0 \beta_0. \tag{9}$$

According to Hadjiconstantinou and Christofides (1995), the size of this formulation depends on the total number of pieces in R and the maximum number Q_i of copies of each piece that can be used for cutting. A sequence of cuts in R is called the cutting pattern.

According to Alvarez-Valdes et al. (2007), depending on the values of the lower bound, P_i, and the upper bound, Q_i, of the pieces R_i to be cut on the plate R, three types of problems can be distinguished: unconstrained, constrained, and doubly constrained. The first problem is called unconstrained, in which there are no lower and upper constraints that must be met, the only constraint is related to respecting the boundaries of the plate area R. In the constrained problem, there is only one constraint related to the upper bound. In this problem, all pieces R_i to be cut in R, do not have a minimum number of copies to be cut, but there is a constraint on the maximum number of copies for any R_i pieces. The upper bound indicates that up to Q_i copies of pieces R_i can be cut into a plate R. If the sum of the areas of the Q_i copies of the piece R_i is smaller than the area of the plate R, and all copies of the piece R_i have already been used in cutting the plate, even if there is a possibility of cutting more copies of this piece R_i, it cannot be done, because the upper bound has been reached and this constraint must be respected. Similarly, if the sum of the areas of Q_i pieces R_i is greater than the area of the plate R, not all copies of this part will be used, so the constraint continues to be respected but without using all the copies available through the upper bound.

The latter type of problem is doubly constrained when there are lower and upper bounds that must be respected by any R_i pieces to be cut on the plate R. Therefore, it is possible to cut a number of copies of the part ranging from the value of its lower bound to the value of its upper bound, in a way that the two constraints imposed by the latter problem are respected.

3 SPECIALIZED BIASED RANDOM KEY GENETIC ALGORITHM

The BRKGA was proposed by Gonçalves and Resende (2013) and brings some differences in relation to the traditional genetic algorithm. However, to understand its operation, it is necessary first to define the random key genetic algorithm (RKGA).

3.1 Random Key Genetic Algorithm

Introduced by Bean (1994), the RKGAs have their chromosomes represented by vectors with n elements, which assume values generated randomly in the real interval $[0,1]$. For that reason, a proposed solution is depicted in encoded form and must be decoded using a decoder.

The decoder is a deterministic algorithm that uses a chromosome as input, associating it with a solution of the combinatorial optimization problem, for which a target value or fitness can be calculated (GONÇALVES; RESENDE, 2013).

According to Gonçalves and Resende (2013), an RKGA generates a population of random key vectors during a number of generations. The initial population consists of p vectors of n random keys. Each component of the solution vector, or random key, is independently and randomly generated in the real interval $[0,1]$. After the fitness of each individual is calculated by the decoder in the generation g, the population is partitioned into two groups of individuals: a small group of p_e elite individuals, in other words, those with the best fitness values, and a group formed by the rest of the population, set of $p - p_e$ non-elite individuals. To evolve from generation g to generation $g+1$, a new generation of individuals is produced. The RKGA algorithm uses elitism. All the elite individuals of the population of generation g are copied unmodified to the population of generation $g+1$. Two other strategies are used by the RKGAs to complete the formation of generation $g+1$, that is, the mutation and recombination operators.

The RKGAs implement mutation by introducing mutants into the current population. A mutant is created as a vector of random keys, generated in the same way as an element of the initial population. With each generation, a small number of mutants p_m is introduced into the population. Mutants replace the mutation strategy used in the traditional genetic algorithm (GONÇALVES; RESENDE, 2013).

The population of the next generations will be composed of elite individuals (p_e), mutant individuals (p_m), and individuals produced by the recombination process. With p_e+p_m individuals representing the population of the $g+1$, $p-p_e-p_m$, additional individuals need to be generated to complete the p individuals that make up the population of the $g+1$. This is accomplished by producing $p-p_e-p_m$ solutions through the recombination process.

In the RKGA, as in the BRKGA, there is no conventional selection operator. This is exactly where the two random key algorithms differ.

3.2 Biased Random Key Genetic Algorithm

In the BRKGA proposed by Gonçalves and Resende (2013), each descendant is generated by combining an individual randomly selected from the elite partition of the current population. For the selection of the second individual, there are two strategies: in the first strategy, the second individual is obtained by selecting it from the non-elite partition of the current population; and in the second strategy, the second individual is obtained by selecting it from the current population, which includes both the elite and the non-elite partitions. It should be noted that the selection operator is implicitly present.

The word biased in the BRKGA, incorporated by Gonçalves and Resende (2013), is used to show that recombination is not purely random. That is, one of the generating solutions must necessarily be an elite (biased), as opposed to the original proposal, in which the two generating solutions can be of the population as a whole. The other biased component appears when a single descendant is generated.

It should be noted that each recombination is performed independently and, therefore, one individual can produce more than one descendant in the same generation. As in RKGAs, BRKGAs implement parameterized uniform recombination (SPEARS; DEJONG, 1991).

In this type of recombination, there is the probability q_e which will serve as a flag parameter to know if the component n, of the descending vector C, will be inherited from parent A or parent B. It is important to remember that n denotes the number of components in a solution vector of an individual. For $i = 1,...,n$, the i$^{\text{th}}$ component of $c(i)$ of the downstream vector C assumes the value of the i^{th} component $a(i)$ of elite parent A with probability q_e or value of i^{th} component $b(i)$ of parent B from the population with probability 1 $-q_e$.

The value used for q_e is high and is typically chosen q_e= 0,7. Thus, in each case, there is a probability of 0,7 for the value stored in A to be copied into the descending C being generated. This strategy allows most of the elements copied in C to be elements stored in the elite solution, making the reasons why only one descendant is generated clear.

When the next population is generated, the values of fitness of these individuals are calculated using the decoder, and the population is partitioned into elite and non-elite individuals to start a new generation until the stopping criterion is satisfied.

3.3 Specialized Genetic Algorithm

The specialized algorithm was based on the BRKGA proposed by Gonçalves and Resende (2013), but it considers some processes that were necessary to improve the performance when solving the problem under consideration.

3.3.1 Functioning of the Specialized BRKGA

Next, in Algorithm 1, we have a description of the operation of the specialized BRKGA.

Algorithm 1 - Specialized Biased Random Key Genetic Algorithm

```
set the initial population size $p$.
while (stopping criterion)
    for $(i = 1; i \leq p; i++)$ do
        decode $(individual_i)$
    end for
    for $(i = 1; i \leq p; i++)$ do
        cutting_pattern(i) $\leftarrow$ CHA $(individual_i)$
    end for
    for $(i = 1; i \leq p; i++)$ do
        fitness_calculation $(cutting\_pattern_i)$
    end for
    elite $\leftarrow$ best cutting_pattern
    non-elite $\leftarrow$ *cutting_pattern* - *elite*
    create mutants
    crossover ( $elite, non_{elite}$)
    current_population $\leftarrow$ next generation
end while
return incumbent
```

In Algorithm 1, it is noticed that the specialized genetic algorithm maintains some characteristics of the traditional genetic algorithm, such as the initial population generation, fitness calculation, recombination, mutation, and stopping criterion. Only a few phases were added, such as the decoding of each individual (solution proposal), the transformation of these individuals into cutting patterns, and the use of the elite group in the recombination.

In Algorithm 2, there is the description of the CHA sub-algorithm.

Algorithm 2 - Constructive Heuristic Algorithm (CHA)

```
1: set $free_{rectangles}(1,\ldots,j)$, $pieces_{available}(1,\ldots,m)$,
2: for $(i = 1; i \leq m; i++)$ do
3:     choose piece $i$
4:     create a block with the samples of the piece $i(quantity\ of\ pieces\ i\ of\ the\ block \leq Q_i)$
5:     while $(block_{area} \leq free_{area}(j))$
6:         cut block from step 4 closest to the plate origin (lower-left corner)
7:         update the list of free_rectangles and the list of pieces_ available
8:     end while
9: end for
10: return cutting_pattern
```

It can be seen in Algorithm 2 that the CHA determines how and where each piece is cut on the plate, picking the plate and the free rectangle to cut from it.

3.3.2 Formation of blocks

In the algorithm proposed in this paper, the block with equal parts is constructed. This procedure of grouping the plates into blocks was not used in the BRKGA proposed by Gonçalves and Resende (2013), so it is an improvement applied to the specialized RKGA proposed in this paper.

Pieces p_i are placed side by side until no more space is available within the free rectangle chosen to allocate more pieces, or until the availability of the pieces p_i is exhausted. Only after pieces p_i are placed next to each other, and if more pieces of the same type are available to be used in the block construction, pieces p_i are to be allocated above the pieces p_i already allocated in the free rectangle, so it will occur successively until the quantity of pieces p_i used in the construction of the block is $\leq Q_i$. Consequently, a block with a rectangular or square geometric shape is guaranteed.

3.3.3 Scanning Free Rectangles

During the research, it was found that the identification of free rectangles on the plate is important for the success of the algorithm because identifying free rectangles makes it possible to choose which rectangle to use in the next cut inside the plate.

The purpose of this paper is to trace the plate by mapping the spaces that are available for further cuts. The procedure works as a scanner by mapping the board from top to bottom and from left to right. This is another improvement, proposed in this paper, to be applied to the specialized BRKGA, and it is also an unprecedented contribution of this work since the type of scanning done on the plate is different from those found in the specialized literature.

When a space is identified as free, it is added to it all the free and sequential spaces that are just below it. Then, following the same procedure, the free spaces to the right are added, obeying the height delimitation performed by the first column of the scan, thus originating a free rectangle. Using this logic, scanning is performed from top to bottom.

In the same way, scanning from left to right is used. When a space is identified as free, all the free and sequential spaces that are to its right are added to it.

After obtaining the free rectangles, the process of difference and elimination proposed by Lai and Chan (1997) is applied.

4 RESULTS AND DISCUSSION

4.1 Tests

The test instances were obtained from the OR-Library website (2017). The OR-Library (2017) set of tests is composed of 21 cases extracted from the specialized literature, 12 instances of Beasley (1985), 2 instances of Hadjiconstantinou and Christofides (1995), 1 instance of Wang (1983), 1 instance of Christofides and Whitlock (1977), and 5 instances of Fekete and Scheppers (1997).

The algorithm was run using a laptop with an Intel Core i7 processor, 8GB of RAM, and the FORTRAN programming language was used. The time required for processing did not increase substantially in the different test cases, reaching an average execution time of 5 seconds.

As a comparison, the test cases were also submitted to the mathematical model proposed by Hadjiconstantinou and Christofides (1995) to solve the two-dimensional non-guillotine cutting problem with fixed orientation pieces. The model was programmed and tested in AMPL, using the solver CPLEX version 12.6.3 as the solver. In the execution of AMPL, it was used a Dell PowerEdge T430, with Linux Operating System (Debian distribution), with two 12 Cores/24 Threads Intel® Xeon® Processors E5-2650 v4 2.2GHz and 64 GB of RAM. The first 14 cases had an execution time of 1 to 10 seconds, but the last 6 cases had execution time greater than 24 hours.

To perform a fair comparison with the results obtained by the mathematical model tested in AMPL, the specialized algorithm proposed in this paper also considered the pieces with fixed orientation. In addition, in the specialized literature, both metaheuristic algorithms and exact methods also consider only this type of orientation.

Therefore, the 21 cases were tested with two algorithms, which are:

System I - The mathematical model, programmed in the AMPL proposed by Hadjiconstantinou and Christofides (1995) to solve the two-dimensional non-guillotine cutting problem with fixed orientation of the pieces.

System II - Specialized random key genetic algorithm with fixed orientation of the pieces.

System III - Specialized random key genetic algorithm with free orientation of the pieces which are tests that cannot be compared with other optimization proposals.

4.2 Analysis and Discussion of Results

The specialized algorithms proposed in this work were able to solve the two-dimensional non-guillotine cutting problem. The results are shown in Table 1, where it is possible to analyze the results obtained by this paper, comparing them with the results of works that are already published and taken as references in the specialized literature.

When analyzing the data contained in Table 1, the following are shown: in the second column, the data obtained by Beasley (1985) are presented using an exact method as a solution technique;

Table 1 – Comparison of Computational Results Considering Fixed and Free Orientation.

	Fixed Orientation					Free Orientation
Test Case	Beasley [1]	GRASP [2]	Tabu Search[3]	System I	System II	System III
1	247	247	247	247	234	259
2	230	230	230	230	230	250
3	164	164	164	164	156	193
4	358	358	358	358	358	370
5	268	268	268	268	268	268
6	289	289	289	289	289	298
7	834	834	834	834	828	856
8	430	430	430	430	430	430
9	924	924	924	924	924	924
10	1688	1688	1688	1688	1688	1786
11	1178	1178	1178	1178	1178	1272
12	1801	1865	1865	1865	1865	1932
13	1452	1452	1452	1452	1452	1452
14	1270	1270	1270	1270	1270	1431
15	2721	2726	2726	2726	2667	2793
16	1720	1860	1860	1860	1820	1840
17	27486	27589	27718	27718	27539	28090
18	21976	21976	22502	22502	22502	22852
19	23743	23743	24019	24019	24019	24811
20	31269	32893	32893	32893	32893	32893
21	26332	27923	27923	27923	27923	27983

Source: [1]Found by Beasley (1985). [2]Found by Alvarez-Valdes et al. (2005). [3]Found by Alvarez-Valdes et al. (2007).

the third column shows the results obtained by Alvarez-Valdes et al. (2005) in which the GRASP metaheuristics to solve the problem under study were used; the fourth column shows the results obtained by Alvarez-Valdes et al. (2007) in which the Tabu Search metaheuristics to solve the same problem were used; the fifth column presents the results obtained by System I by using the mathematical model proposed by Hadjiconstantinou and Christofides (1995), and the problem was solved using CPLEX; the following columns show the results obtained by System II with a specialized algorithm using fixed orientation of parts, and by System III with the specialized algorithm using free orientation of parts.

Comparing the results, System I presented better solutions in relation to Beasley (1985), which compares two exact methods applied to solve the same problem with the same tested instances. The results obtained by System I are considered optimal solutions and were used as comparison parameters to evaluate specialized metaheuristics.

The results presented by the GRASP metaheuristic of Alvarez-Valdes et al. (2005) are very close to the optimal results of System I; only in three test cases the results were inferior (cases 17, 18, and 19).

The results presented by the tabu search metaheuristics of Alvarez-Valdes et al. (2007) are identical to the results found by System I.

In terms of the results obtained by System II proposed in this work, using specialized metaheuristics, it was also found that the results are close to the optimal solutions of System I, so they are good-quality results; still, there were six test cases with results lower than optimal (cases 1, 3, 7, 15, 16, 17).

Comparing the results of the test cases of Alvarez-Valdes et al. (2005) with the results obtained by System II proposed in this work, the algorithm of Alvarez-Valdes et al. (2005) was better in smaller instances such as 1, 3, 7, 15, and 16. In these cases, the plates have dimensions ranging from 10 x 10 to 40 x 70, and the number of types of parts varies from 5 to 20. These instances are considered small and medium-sized instances.

In cases 18 and 19, which are considered to be large, with plates of 100 x 100 dimensions and number of types of pieces reaching 30 different types, Systems II proposed by the author was better than the algorithm of Alvarez-Valdes et al. (2005), presenting, therefore, its contribution and improvement for the resolution of the two-dimensional non-guillotine cutting problem.

As previously mentioned, a specialized BRKGA was proposed for the two-dimensional non-guillotined cutting problem with free orientation of parts (System III). This way, a problem with search space larger than the problem with fixed orientation is analyzed. For this reason, our results with fixed orientation of parts cannot be compared directly with the results considering free orientation of parts. But the comparison is valid if it is observed that the results of System III should be greater than or equal to the results of System II.

5 CONCLUSIONS

This paper used a specialized metaheuristic, that is, the biased random key genetic algorithm, to solve the two-dimensional non-guillotine cutting problem considering the alternatives with and without orientation of parts. In the alternative with a smaller search space, the proposal was compared with results obtained by other optimization proposals. In the case of optimization considering free orientation, it was not possible to compare it with other proposals since these proposals are not found in the specialized literature. The specialized algorithm uses specific strategies to address the problem: the first strategy is the formation of blocks using the same type of parts, and the second strategy was the scanning of free rectangles on the plate.

The algorithm proposed in this work is considered contributes to the specialized literature because it presents combinations of strategies not yet tested to solve the problem under study.

A mathematical model was programmed in AMPL and solved using the CPLEX solver to aid in the comparisons between the results from the exact method and the results obtained by the specialized algorithm proposed by the current work, in addition to comparing the latter and the results obtained by works already published and used as a reference in the literature, making it simpler to check the quality of each one of the results. When comparing the results obtained by the specialized genetic algorithm proposed in this work with the results available in the literature and found in Table 1 of this paper, it is concluded that the proposed algorithm surpassed in two test cases the results already published, giving another novelty to the current work. Finally, a specialized algorithm was tested which was configured to turn the parts, if need be. The results of the twenty-one tests, in this case, were equal to or better than those already published in the literature but considering the fixed orientation parts.

As suggestion for future work, there is the possibility of extending the problem resolution to three dimensions, using the specialized algorithm developed in this work and implementing other metaheuristics, such as the variable neighborhood search algorithm, for the resolution of the two-dimensional non-guillotine cutting problem.

5.1 Supplementary Materials

The part and plate size data used to support the findings of this study are included in the supplementary information file.

References

ALVAREZ-VALDES R, PARREÑO F & TAMARIT JM. 2005. A GRASP algorithm for constrained two- dimensional non- guillotine cutting problems. *Journal of the Operational Research Society*, **56**: 414-425.

ALVAREZ-VALDES R, PARREÑO F & TAMARIT JM. 2007. A Tabu Search algorithm for two-dimensional non-guillotine cutting problems. *European Journal of Operational Research*, **183**: 1167-1182.

ARENALES M & MORABITO R. 1995. An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems. *European Journal of Operational Research*, **84**: 599-617.

AYADI O, MASMOUDI M, AMEUR MB & MASMOUDI F. 2017. A new PSO-based algorithm for two-dimensional non-guillotine non-oriented cutting stock problem. *Applied Artificial Intelligence*, **31**(4): 376-393.

BALDACI R & BOSCHETTI M. 2007. A cutting-plane approach for the two-dimensional orthogonal non-guillotine cutting problems. *European Journal of Operational Research*, **183**: 1136-1149.

BEAN JC. 1994. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, **6**: 154-160.

BEASLEY JE. 1985. An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, **33**: 49-64.

BEASLEY JE. 2004. A population heuristic for constrained two-dimensional non-guillotine cutting. *European Journal of Operational Research*, **156**: 601-627.

CHRISTOFIDES N & WHITLOCK C. 1977. An algorithm for two-dimensional cutting problems. *Operations Research*, **25**(1): 30-44.

FEKETE SP & SCHEPERS J. 1997. On more-dimensional packing iii: Exact algorithms. *Technical Report 97-290*. Technical University of Berlin.

GILMORE P & GOMORY R. 1961. A linear programming approach to the cutting stock problem. *Operational Research*, **9**: 849-859.

GONÇALVES JF & RESENDE MGC. 2013. A Biased Random-Key Genetic Algorithm for a 2D and 3D Bin Packing Problem. *International Journal of Production Economics*, **145**: 500-510.

HADJICONSTANTINOU E & CHRISTOFIDES N. 1995. An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *European Journal of Operational Research*, **83**: 39-56.

LAI KK & CHAN JWM. 1997. Developing a Simulated Annealing Algorithm for the Cutting Stock Problem. *Computers & Industrial Engineering*, **32**: 115-127.

LEUNG TW, YUNG CH & TROUTT MD. 2001. Application of genetic search and Simulated Annealing to the two-dimensional non-guillotine cutting stock problem. *Computers & Industrial Engineering*, **40**: 201-214.

OR-LIBRARY. 2017. *Operations Research (OR) problems*. Online. http://people.brunel.ac.uk/mastjjb/jeb/info.html. Access on: 20 Apr. 2017.

SPEARS WM & DEJONG KA. 1991. On the virtues of parameterized uniform crossover. *International Conference on Genetic Algorithms*, 4., San Diego. Proceedings. San Diego: University of California, p. 230-236.

WANG PY. 1983. Two algorithms for constrained two-dimensional cutting stock problems. *Operations Research*, **31**(3): 573–586.

WEI L, HU Q, LIM A & LIU Q. 2018. A best-fit branch and bound heuristic for the unconstrained two-dimensional non-guillotine cutting problems. *European Journal of Operational Research*, **270**: 448–474.

How to cite

OLIVEIRA EV & ROMERO R. 2022. Improved biased random key genetic algorithm for the two-dimensional non-guillotine cutting problem. *Pesquisa Operacional*, **42**: e260406. doi: 10.1590/0101-7438.2022.042.00260406.

APPENDIX A

Research data obtained from the OR-LIBRARY (2017).

For a better understanding of the data from Table A1 through Table A21, the column headers are described below.

$R = (lengthxheight)$: are the length and height dimensions of the board for each test case.

Pieces: Represents the types of parts available in each test case;

C_i: length of the piece i;

A_i: height of the piece i;

P_i: lower part limitation i;

Q_i: upper limit of the pieces i, where Q_i also represents the quantity of pieces i; available for cutting;

V_i: value of the pieces i;

Table A1 – Pieces data and test case plate 1.

$R = (10x10)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	3	2	0	2	7
2	7	2	0	3	20
3	4	5	0	2	60
4	4	1	0	2	9
5	1	10	0	1	14
6	8	4	0	1	79
7	4	2	0	2	11
8	3	7	0	3	52
9	6	2	0	3	13
10	9	1	0	2	21

Table A2 – Pieces data and test case plate 2.

$R = (10x10)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	1	10	0	3	28
2	5	3	0	2	40
3	9	3	0	3	63
4	6	1	0	3	13
5	3	8	0	3	31
6	4	1	0	1	10
7	6	3	0	2	44

Table A3 – Pieces data and test case plate 3.

$R = (10x10)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	3	7	0	2	35
2	8	2	0	2	40
3	10	2	0	1	27
4	5	4	0	3	23
5	2	9	0	2	43

Table A4 – Pieces data and test case plate 4.

$R = (15x10)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	10	3	0	2	34
2	9	3	0	1	48
3	12	2	0	3	72
4	11	3	0	1	91
5	12	3	0	3	37
6	11	1	0	3	15
7	2	10	0	1	36

Table A5 – Pieces data and test case plate 5.

$R = (15x10)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	15	2	0	2	34
2	7	3	0	1	27
3	9	1	0	1	14
4	8	3	0	1	71
5	12	2	0	2	61

Table A6 – Pieces data and test case plate 6.

$R = (15x10)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	2	9	0	1	50
2	10	2	0	3	29
3	2	4	0	1	11
4	2	8	0	2	30
5	3	8	0	1	46
6	4	1	0	1	7
7	6	4	0	3	32
8	10	3	0	1	74
9	11	2	0	1	48
10	4	5	0	1	32

Table A7 – Pieces data and test case plate 7.

$R = (20x20)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	15	3	0	1	110
2	6	6	0	3	93
3	13	1	0	2	38
4	8	7	0	2	144
5	18	5	0	3	185
6	12	4	0	1	78
7	8	3	0	1	25

Table A8 – Pieces data and test case plate 8.

$R = (20x20)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	1	9	0	3	26
2	16	3	0	1	113
3	18	3	0	1	127
4	20	2	0	1	104
5	3	1	0	2	4

Table A9 – Pieces data and test case plate 9.

$R = (20x20)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	14	2	0	1	45
2	1	5	0	1	10
3	20	4	0	2	139
4	12	3	0	3	92
5	11	8	0	2	145
6	11	6	0	2	123
7	7	9	0	1	166
8	17	5	0	1	252
9	7	14	0	2	161
10	1	7	0	3	12

Table A10 – Pieces data and test case plate 10.

$R = (30x30)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	3	23	0	3	118
2	29	5	0	1	244
3	21	2	0	3	81
4	17	11	0	3	371
5	14	7	0	2	254
6	8	5	0	2	78
7	21	8	0	1	293

Table A11 – Pieces data and test case plate 11.

$R = (30x30)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	12	21	0	1	450
2	17	21	0	1	570
3	6	6	0	1	49
4	9	15	0	1	166
5	5	12	0	1	69
6	16	6	0	1	92
7	5	4	0	1	17

Table A12 – Pieces data and test case plate 12.

$R = (30x30)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	16	6	0	1	132
2	24	1	0	3	29
3	6	28	0	3	351
4	8	23	0	3	275
5	5	1	0	3	10
6	6	26	0	2	259
7	2	30	0	2	110
8	9	11	0	3	240
9	4	30	0	1	265
10	16	13	0	1	507

Table A13 – Pieces data and test case plate 13.

$R = (30x30)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	1	30	0	3	75
2	25	7	0	1	279
3	27	9	0	3	409
4	30	2	0	3	65
5	26	7	0	3	235

Table A14 – Pieces data and test case plate 14.

$R = (30x30)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	22	21	0	1	828
2	12	21	0	1	450
3	11	18	0	1	335
4	17	21	0	1	570
5	8	21	0	1	268
6	9	17	0	1	236
7	5	18	0	1	128
8	6	6	0	1	49
9	9	15	0	1	166
10	1	19	0	1	22
11	5	12	0	1	69
12	22	21	0	1	530
13	5	6	0	1	33
14	16	6	0	1	92
15	5	4	0	1	17

Table A15 – Pieces data and test case plate 15.

$R = (70x40)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	17	9	0	1	153
2	11	19	0	4	209
3	12	21	0	3	252
4	14	23	0	4	322
5	24	15	0	3	360
6	25	16	0	4	400
7	27	17	0	2	459
8	18	29	0	3	522
9	21	31	0	3	651
10	32	22	0	2	704
11	23	33	0	3	759
12	34	24	0	2	816
13	35	25	0	2	875
14	36	26	0	1	936
15	37	27	0	1	999
16	38	28	0	1	1064
17	39	29	0	1	1131
18	41	30	0	1	1230
19	43	31	0	1	1333

Table A16 – Pieces data and test case plate 16.

$R = (40x70)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	31	43	0	4	500
2	30	41	0	2	480
3	29	39	0	4	460
4	28	38	0	4	440
5	27	37	0	3	420
6	26	36	0	4	410
7	25	35	0	3	400
8	24	34	0	4	380
9	33	23	0	4	360
10	22	32	0	3	340
11	31	21	0	3	320
12	29	18	0	3	300
13	17	27	0	2	280
14	15	24	0	2	240
15	16	25	0	4	260
16	15	24	0	1	240
17	23	14	0	4	220
18	21	12	0	3	180
19	19	11	0	4	160
20	9	17	0	1	140

Table A17 – Pieces data and test case plate 17.

$R = (100x100)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	4	90	0	5	838
2	22	21	0	2	521
3	22	80	0	3	4735
4	1	88	0	5	181
5	6	40	0	5	706
6	100	9	0	5	2538
7	46	14	0	3	1349
8	10	96	0	1	1685
9	70	27	0	3	5336
10	57	18	0	1	1775
11	10	84	0	1	1131
12	100	1	0	5	129
13	2	41	0	5	179
14	36	63	0	2	6668
15	51	24	0	4	3551

Table A18 – Pieces data and test case plate 18.

$R = (100x100)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	8	81	0	1	953
2	5	76	0	1	389
3	42	19	0	1	1668
4	6	80	0	1	676
5	41	48	0	1	3580
6	6	86	0	1	1416
7	58	20	0	1	3166
8	99	3	0	1	537
9	9	52	0	1	1176
10	100	14	0	1	3434
11	7	53	0	1	676
12	24	54	0	1	1408
13	23	77	0	1	2362
14	42	32	0	1	4031
15	17	30	0	1	1152
16	11	90	0	1	2255
17	26	65	0	1	3570
18	11	84	0	1	1913
19	100	11	0	1	1552
20	29	81	0	1	4559
21	10	64	0	1	713
22	25	48	0	1	1279
23	17	93	0	1	3989
24	77	31	0	1	4850
25	3	71	0	1	299
26	89	9	0	1	1577
27	1	6	0	1	12
28	12	99	0	1	2116
29	33	72	0	1	2932
30	21	26	0	1	1214

Table A19 – Pieces data and test case plate 19.

$R = (100x100)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	3	98	0	1	756
2	34	36	0	1	2712
3	100	6	0	1	1633
4	49	26	0	1	2332
5	14	56	0	1	2187
6	100	3	0	1	470
7	10	90	0	1	1569
8	23	95	0	1	4947
9	10	97	0	1	2757
10	50	47	0	1	4274
11	41	45	0	1	4347
12	13	12	0	1	396
13	19	68	0	1	3866
14	50	46	0	1	5447
15	23	70	0	1	2906
16	28	82	0	1	6032
17	12	65	0	1	1799
18	9	86	0	1	929
19	21	96	0	1	5186
20	19	64	0	1	2120
21	21	75	0	1	1629
22	45	26	0	1	2059
23	19	77	0	1	2583
24	5	84	0	1	953
25	16	21	0	1	1000
26	23	69	0	1	2900
27	5	89	0	1	1102
28	22	63	0	1	2234
29	41	6	0	1	458
30	76	30	0	1	5458

Table A20 – Pieces data and test case plate 20.

$R = (100x100)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	48	48	0	1	5145
2	6	85	0	2	874
3	100	14	0	1	2924
4	17	85	0	1	3182
5	69	20	0	1	2862
6	12	72	0	1	1224
7	5	48	0	3	531
8	1	97	0	3	249
9	66	36	0	2	6601
10	15	53	0	1	1005
11	29	80	0	3	6228
12	19	77	0	1	3362
13	97	7	0	1	907
14	7	57	0	2	473
15	63	37	0	2	6137
16	71	14	0	1	1556
17	3	76	0	3	313
18	34	54	0	1	4123
19	5	91	0	2	581
20	14	87	0	1	1999
21	62	28	0	3	5004
22	6	7	0	3	2040
23	20	71	0	1	3143
24	92	7	0	1	795
25	10	77	0	2	1460
26	99	4	0	3	841
27	14	44	0	2	1107
28	100	2	0	3	280
29	56	40	0	2	5898
30	86	14	0	1	2096
31	22	93	0	1	4411
32	13	99	0	3	3456
33	7	76	0	3	1406

Table A21 – Pieces data and test case plate 21.

$R = (100x100)$					
Pieces	C_i	A_i	P_i	Q_i	V_i
1	8	81	0	3	953
2	5	76	0	4	389
3	42	19	0	4	1668
4	6	80	0	4	676
5	41	48	0	1	3580
6	6	86	0	5	1416
7	58	20	0	5	3166
8	99	3	0	5	537
9	9	52	0	5	1176
10	100	14	0	4	3434
11	7	53	0	5	676
12	24	54	0	1	1408
13	23	77	0	1	2362
14	42	32	0	5	4031
15	17	30	0	5	1152
16	11	90	0	4	2255
17	26	65	0	2	3570
18	11	84	0	3	1913
19	100	11	0	1	1552
20	29	81	0	1	4559
21	10	64	0	2	713
22	25	48	0	1	1279
23	17	93	0	4	3989
24	77	31	0	1	4850
25	3	71	0	5	299
26	89	9	0	4	1577
27	1	6	0	5	12
28	12	99	0	2	2116
29	21	26	0	5	1214