

# A biased random-key genetic algorithm using dotted board model for solving two-dimensional irregular strip packing problems

Bonfim Amaro Junior  
Federal University of Ceará  
Russas, Brazil  
bonfimamaro@ufc.br

Rodrigo Lima da Costa  
Federal University of Ceará  
Russas, Brazil  
rodrigoferlima.12@gmail.com

Plácido Rogério Pinheiro  
University of Fortaleza  
Fortaleza, Brazil  
placido@unifor.br

Luiz Jonatã Pires de Araújo  
Innopolis University  
Innopolis, Russia  
l.araujo@innopolis.university

Alexandr Grichshenko  
Innopolis University  
Innopolis, Russia  
a.grishchenko@innopolis.university

**Abstract**—The irregular strip packing problem (ISPP) is a combinatorial optimisation problem that has applicability in several industrial processes since it aims for the efficient use of material. Most of the techniques reported in the literature for solving the ISPP employ metaheuristics as they can cope with complex requirements that prevent the use of exact model formulations. This paper presents a biased random-key genetic algorithm (BRKGA) that uses the dotted board model to compute the fitnesses of candidate solutions aiming for the minimisation of the height of the large object. The algorithm allows the pieces to rotate in order to achieve better layouts. Computational experiments using instances from the literature were conducted to demonstrate the efficiency of the proposed method, with promising results.

**Index Terms**—cutting and packing, biased random-key genetic algorithm, linear programming.

## I. INTRODUCTION

Cutting and packing (C&P) are combinatorial optimisation problems that concern the arrangement of pieces (i.e. items) within larger objects without overlap aiming the minimisation of the necessary space and resultant waste [1]. C&P problems have outstanding importance in several industrial processes including furniture production [2], clothing manufacturing [3], shipbuilding industry [4] and container loading [5], [6], to mention few applications.

C&P belong to the class of problems called NP-Hard, which implies that there are no efficient algorithms for solving large problems as those found in real-world scenarios. Hence most of the solutions employ metaheuristics capable of finding satisfactory results in a reasonable amount of computation time. Examples of approaches for C&P problems include placement policies [7], [8], linear programming [9], [10], and combination of placement algorithms with search metaheuristics [11].

This study addressed the two-dimensional irregular strip packing problem (ISPP) - a C&P variant in which the items

are arranged in such a way that the length of the container is minimised [1]. According to Araújo et al.'s extended taxonomy for C&P [12], this problem is represented by the four-tuple  $2|Si|Oo|A$ , which denotes two-dimensional problems, a single input or container ( $Si$ ) with an open dimension ( $Oo$ ) with different assortment of items ( $A$ ). This paper analyses the performance of biased random-key genetic algorithm [13] combined with the dotted board model employed for assessing candidate solutions. Such an approach [14] has demonstrated reasonable results in satisfactory running time for larger instances than previous methods in the literature.

The remaining part of this paper is organized as follows. Section II reviews the literature on the two-dimensional packing algorithms. Next, section III details the implementation of a biased random-key genetic algorithm and the data structures to encode candidate solutions. Section IV presents the used datasets and compare the results with other algorithmic solutions from the literature. Section VI concludes by summarising the results and limitations of the presented approach and comments on future research directions.

## II. LITERATURE REVIEW

This section presents the main algorithms solutions for two-dimensional C&P problems in the literature, which can be categorised into placement policies, mathematical formulations, and combinations of the previous approaches with metaheuristics [12], [15]. It also presents two packing methods that are combined in the implemented approach: the biased random-key genetic algorithm and the dotted board model formulation.

### A. Two-dimensional packing approaches

One of the first approaches for addressing C&P problem is the use of placement policies such as the bottom-left (BL) heuristic [16]. In this method, items are packed one at a time,

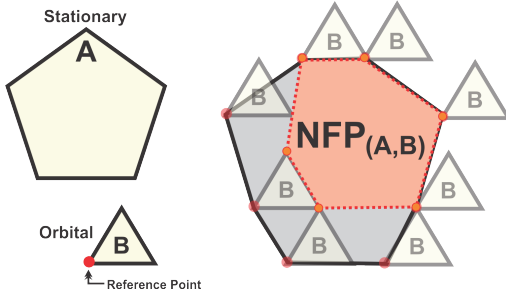


Fig. 1. Example of No-fit Polygon.

being slid towards the origin and the most bottom-left available position. Most of BL methods in the literature vary according to criteria for determining the sequence in which items are processed [17]–[19]. For example, some algorithms prioritize items using properties like height or area [20].

Another method for solving C&P is the use of mathematical formulations to ensure the feasibility of placement configurations. For example, the No-fit polygon (NFP) is a method for determining legal placements between a pair of items [18], [21]. In this method, an item (called stationary) is fixed while another item is translated around the latter to form a region in which the placement leads to overlap, as illustrated in Figure 1.

Other approaches are based on Minkowski's sum, given by  $A \oplus B = \{a + b : a \in A, b \in B\}$ . This method consists of the combined addition of each point in two sets A and B, each representing a piece. Through algebraic calculations, the Minkowski's difference, given by  $A \oplus -B$ , is equivalent to NFP between two figures A and B, where A is the stationary figure and B is the orbital figure.

An alternative method is the collision free region (CFR) illustrated in Figure 2 [22], and it has been successfully used in the literature [11], [23]. The CFR combines IFP and NFP techniques for generating viable layouts for placing algorithms. Mathematically, we can define CFR between two items as follows:

$$CFR_{P,S} = IFP_{P,S} - \bigcup_{i=1}^N NFP_{S_i,P} \quad (1)$$

In equation 1,  $S$  represents the *stage*, and  $P$  corresponds the next polygon to be packed. The quantity of pieces nested is represented by  $N$ .  $S_i$  is a piece of sequence  $i$  inside  $S$  and  $CFR_{P,S}$  is computed by difference of feasible region for  $P$  ( $IFP_{P,S}$ ) and the union between all the pieces arranged on the surface, i.e.  $NFP_{S_i,P}$  for all located polygon inside  $S$ . Figure 2 illustrates the CFR of two items.

A major drawback of approaches like NFP, IFP and CFR is the need for a preprocessing phase before the packing to calculate the valid placement positions for each pair of items. Moreover, the presence of complex shapes and higher freedom of rotation contribute to infeasible running time and computational effort.

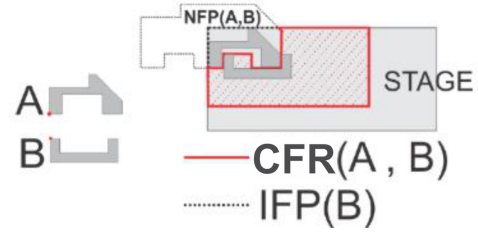


Fig. 2. Example of collision free region polygon.

Mixed integer linear programming models have also been proposed to tackle C&P problems [24], [25]. In this type of approach, the surface of the large object is typically discretised, although there are examples of methods allowing higher degrees of freedom in the placement of item [9].

There have been several approaches combining the strengths of placement heuristics and mathematical formulations. For example, [26] used a local search algorithm to minimise the overlap between the polygons, coupled with a nonlinear model that seeks solutions within a time limit.

Another type of solutions uses metaheuristics for finding the most suitable sequence in which items should be processed by a placement heuristic. Examples of metaheuristics include tabu search and the hill-climbing method [18], beam search [20], simulated annealing [22], guided local and cuckoo search [27]. The spectrum of sorting techniques includes random sorting [23], [28].

The combination of genetic algorithms (GA) and BL heuristic have been the prevalent approach for C&P problems [23], [28], [29]. GA are search algorithms inspired by the natural selection processes, in which a population of individuals evolve through selection and reproduction of the fittest individuals, which transmit their genetic characteristics to the next generation of individuals [30]. The literature contains several examples of packing techniques based on such an approach and employed into real-world problems like circuit design and 3D Printing.

### B. Biased random-key genetic algorithm

Random-key genetic algorithm (RKGA) [31] are a GA variant in which each individual is represented by an array of real numbers in the  $[0, 1]$  interval. A function (called a decoder) takes an individual's chromosome of random keys and determined one candidate solution. In short, the main difference to original GA occurs in selection, crossover and mutation processes. In RKGA, the population is partitioned into two groups of individuals: a small group of  $P_e$  elite individuals, i.e., those solutions with the best fitness values, and the residual set of  $(P - P_e)$  non-elite individuals. To expand the population, a new generation of individuals must be produced. An RKGA applies an elitist strategy copying all elite set at population  $k$  to  $k + 1$ . The mutation is implemented by introducing  $P_m$  mutants into the population. Discounting  $P_e$  elite individuals and  $P_m$  mutants, new individuals need to be created to compose the  $P$  individuals that make up the

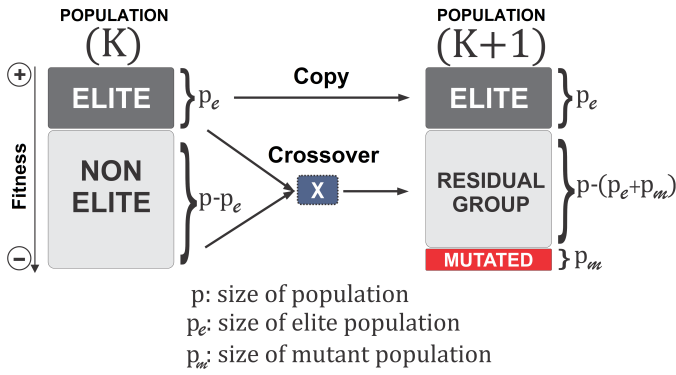


Fig. 3. Generation of a new population in BRKGA.

population of generation  $k + 1$ . This is performed by producing  $P - P_e - P_m$  offspring through the crossover process.

An extension of the RKG algorithm is the biased random-key genetic algorithm (BRKGA), in which individuals are distinguished into two sets: high-performing (or elite), and the remaining (non-elite) individuals of the generation. Elite individuals are copied directly to the next generation, while random-key vectors or mutations are introduced to the remaining population of the next generation.

In the BRKGA, individuals are obtained through parameterized uniform crossover [32], i.e. combinations of pairs in which one parent is an elite and the other a non-elite individual, as illustrated in Figure 3. An additional parameter, the “probability of choice”, is applied to each allele and is used to determine the probability of the elite parent transmitting its genetic material.

A noteworthy work on a variant of BRKGA for packing was proposed by Bonfim et al. [23], which employed the CFR technique for validating placement feasibility. In this method, each individual encompasses the position, the rotation of a part and the CFR was used to place pieces. Moreover, the BRKGA process was adapted to apply multiple parallel populations through the fork-join structure. Due to the parallel aspect populations, an information exchange mechanism was applied. This work was later extended by Mundim et al. [28], which differs in how viable layouts are generated using a discretisation of the packaging surface.

### C. The dotted board model

The dotted board model is a mixed linear programming model that considers an item as a projection on a grid of rows and columns, i.e. a set of feasible points in a mesh data structure [14]. The stage is represented by a discrete grid of points. Take into consideration, the reference point of each item is arranged at a given point of this, aforementioned, grid and the cost of having precision lean on board discretization. In short, this model represents mathematically a needed length to generate a feasible layout however, it does not consider polygon rotations.

In the model above,  $c$  is the column of grid where a piece has its reference point chosen, while  $g_x$  indicates the *dotted*

*board* resolution for  $x$  axis, i.e. the distance between each column.  $X_t^M$  is the horizontal range of reference point for a piece of a type  $t$ , considering its rightmost point.  $\delta_t^d$  is 1 if each piece of type  $t$  has its reference point nested over  $d$  of *grid*, and 0 otherwise.  $q_t$  is the quantity of type pieces  $t$  that should be packed.

The objective function (equation 2) aims the minimal height to accommodate all the items. The constraint 3 ensures that the length is sufficient to pack all pieces of initial demand. The placement of all types of pieces  $t \in T$  and their quantities  $q_t$  are guaranteed by constraint 4. The constraint 5 prevents the overlap of any pair of items  $u$  and  $t$ , placed at points  $e$  and  $d$  respectively. The constraint 6 sets a binary  $\delta$  to be associated to each place possibility to pieces of type  $t \in T$ , for each  $d$  feasible. Lastly, the constraint 7 prevents negative values to be assigned to the best length.

The dotted board model for ISPP is formulated as follows.

$$\text{Min } z \quad (2)$$

Subject to:

$$c \times g_x \times X_t^M \times \delta_t^d \leq z, \forall d \in IFP_t, \forall t \in T \quad (3)$$

$$\sum_{d \in IFP_t} \delta_t^d = q_t, \forall t \in T \quad (4)$$

$$\delta_u^e + \delta_t^d \leq 1, \forall e \in NFP_{t,u}^d, \forall t, u \in T, \forall d \in IFP_t \quad (5)$$

$$\delta_t^d \in \{0, 1\}, \forall d \in IFP_t, \forall t \in T \quad (6)$$

$$z \geq 0 \quad (7)$$

## III. A BIASED RANDOM-KEY GENETIC ALGORITHM USING DOTTED BOARD MODEL

This study implements and extends some of the computational techniques presented in section II. First, the algorithm used in this study extends the BRKGA variant presented by Bonfim et al. [23] for selecting a promising sequence of items and their rotations. This BRKGA variant produces each generation by selecting not only the fittest but also randomly selected individuals from the previous iteration. Two main reasons justify this choice. First, as shown by [13], BRKGA evolves faster than other random key algorithms. In addition, this heuristic is based on random keys, a strategy that allows mapping a problem on a set of keys (chromosomes) so that it becomes practical to perform operations on them and thus generate new solutions.

The implemented algorithm also uses the dotted board model for assessing individuals’ fitness values, one of the most promising exact methods for resolving the ISPP and which has been the basis for several outstanding works. Together, the NFP, IFP and CFR techniques ensure layout satisfaction. The next sections discuss the encoding or representation of solutions, implementation of the BRKGA and feasibility validation.

### A. Encoding and decoding process

A coding and decoding process is required by the algorithm to construct the solutions. The coding process is done at the beginning of each iteration to allow the application of the

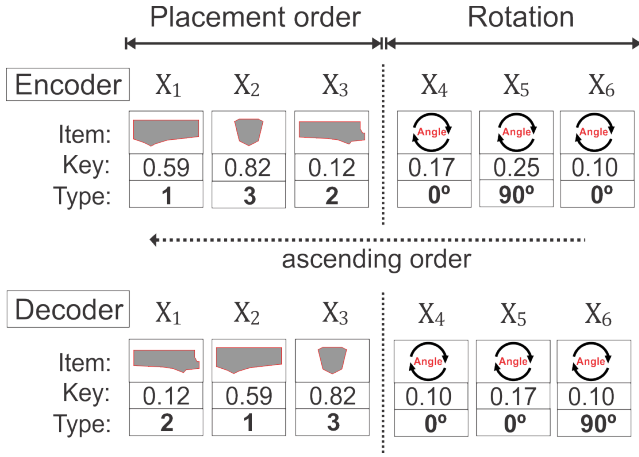


Fig. 4. Encoder and Decoder processes.

genetic operators. In this process, each solution is represented as a vector of real values in the  $[0, 1)$  interval. Each position of the vector that encodes a solution encompasses information on the place in the sequence and rotation of an item to be packed. In the decoding process, these values are sorted by the real value of the positioning chromosome part, so that their position in the rearranged vector defines the sequence in which each corresponding piece will be inserted on the surface, as illustrated in Figure 4.

The decimal part of the value of each allele describes the rotation of the designated item. Decimal parts in  $[0, 0.25)$  denote that the item is not rotated. For values in  $[0.25, 0.5)$ ,  $[0.5, 0.75)$  and  $[0.75, 1)$ , the item is rotated by  $90^\circ$ ,  $180^\circ$  and  $270^\circ$  respectively.

### B. Evolution of generations

BRKGA receives as an input the size of the population ( $p$ ), the size of the elite ( $e$ ), probability of crossover ( $p_c$ ) and mutation ( $p_m$ ), probability of choice  $p_{ch}$ , and the number of generations without improvement ( $n$ ) used as the stop criterion. The proposed BRKGA interactively evolves an initial randomly generated population of individuals until there is no improvement in the fitness of the overall best solution for  $n$  generations. This pseudocode is shown in Algorithm 1.

In each iteration, the elite and non-elite are combined as explained in section II-B. In the crossover operator of BRKGA, the allele of the elite parent is transmitted to the offspring with probability  $p_c$ . Otherwise, the allele from the non-elite parent is transmitted, illustrated in Figure 5. While the elite population is selected as a whole to compose the next generation, the mutant population is obtained from the generation of a vector of random keys.

### C. Layout Generation

In the last stage, all pieces are positioned. As you can see in Figure 6, each piece has a positioning order and rotation defined according to an associated chromosome. Following this order, the parts are rotated and their positioning possibilities are calculated by CFR. Among them, the piece is positioned in

### Algorithm 1 Biased random-key genetic algorithm

**Input:**  $p, e, p_c, p_m, p_{ch}, n$   
**Output:** bestSolution

- 1: numberMutated  $\leftarrow p * p_m$
- 2: crossSize  $\leftarrow p - \text{tamElite} - \text{numberMutated}$
- 3: generation  $\leftarrow 1$
- 4: POPULATION  $\leftarrow \text{generateRandomPopulation}(p)$
- 5: bestSolution  $\leftarrow \text{min}(\text{POPULATION}, \text{by} = \text{layoutLength})$
- 6: **while** (*stopcriteria*) **do**
- 7:   sort(POPULATION, by = layoutLength)
- 8:   ELITEPOP  $\leftarrow \text{POPULATION}[:\text{eliteSize}]$
- 9:   RESIDUALPOP  $\leftarrow \text{POPULATION}[\text{eliteSize}:]$
- 10:   CROSSOVERPOP  $\leftarrow \text{crossover}(\text{ELITEPOP}, \text{RESIDUALPOP}, c, \text{crossSize})$
- 11:   MUTANTPOP  $\leftarrow \text{mutate}(\text{numberMutated})$
- 12:   POPULATION  $\leftarrow \text{join}(\text{ELITEPOP}, \text{CROSSOVERPOP}, \text{MUTANTPOP})$
- 13:   newSolution  $\leftarrow \text{min}(\text{POPULATION}, \text{by} = \text{layoutLength})$
- 14:   **if** (*newSolution* < *bestSolution*) **then**
- 15:     bestSolution  $\leftarrow \text{newSolution}$
- 16:     generation  $\leftarrow 0$
- 17:   **end if**
- 18:   generation  $\leftarrow \text{generation} + 1$
- 19: **end while**
- 20: **return** bestSolution

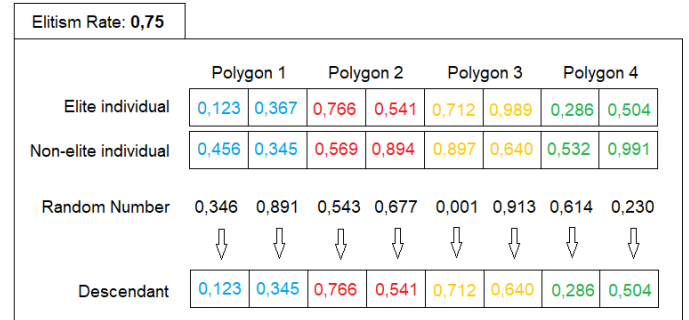


Fig. 5. Generating a new individual with the crossover operator

the with the smallest coordinate  $x$ . In case of ties, the smallest coordinate  $y$  is selected.

In Figure 6a is shown the CFR for the first piece, equivalent to its IFP, since there are no other pieces positioned to overlap. The following pieces are positioned towards the upper left corner, as illustrated in Figure 6b,c. After all the items are packed, the value of the total length is calculated and assigned to the candidate solution.

## IV. COMPUTATIONAL EXPERIMENTS

Computational experiments to assess the effectiveness of the proposed heuristic were performed on an Intel i3 2.2GHz with 4 GB of RAM. The Gurobi and Cplex were used for solving the dotted board model. The BRKGA was implemented in the programming language Julia to leverage of the JuMP

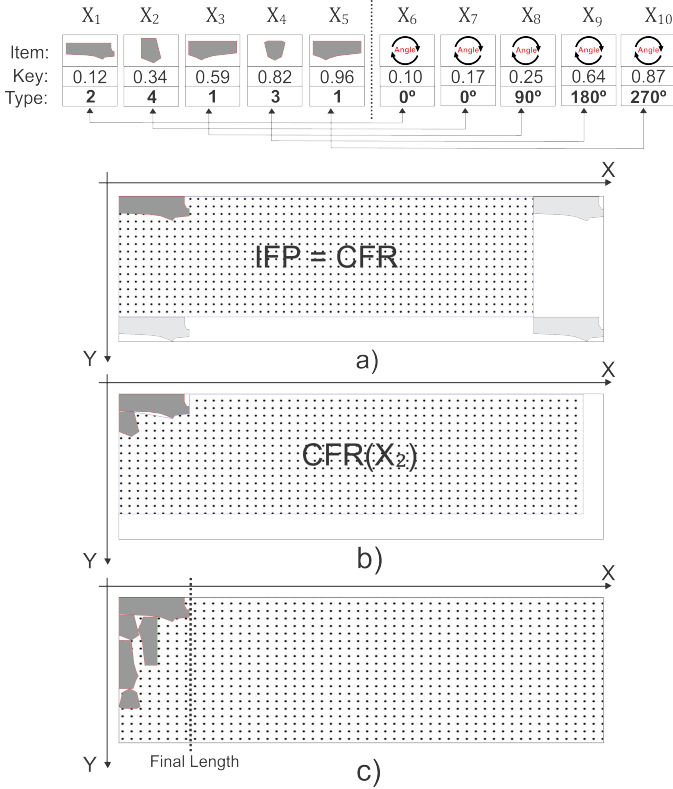


Fig. 6. The layout generation procedure.

package, which provides a level of abstraction to represent a mathematical model and solvers.

The following instances available in the literature have been selected: RCO [14], BLAZ [33]. For these datasets, a fitting surface of width 15 and length 60 was used. For the datasets SHAPES2 and SHAPES4 [34], a fitting surface of 40 (width) x 100 (length) was used. These values are considered in [14]. The execution is interrupted after it reaches a timeout of 36,000 seconds (TL).

Hence, we divide the computational experiments into two parts. First, a comparison of the proposed algorithm and the original dotted board model, aiming to ensure that the optimal solution is also found. In the second part, the algorithm is performed for all the instances, including those which the dotted board model could not find optimal results. The results of each run have been publicly available [35].

Firstly, the dotted board model was executed against the selected datasets to provide baseline results for the introduced packing approach. For each used solver (Gurobi and Cplex), Table I presents the mean running time, the best length found and the gap (the difference between solution found and the known optimal value) of each execution. Table I shows only the instances for which the solver could obtain the optimal value in less than 36,000 seconds. In the original dotted board model [14], only the Cplex solver was used. In this study, we also consider the gurobi solver to check the resolution speed. Changing the adopted solver can be easily implemented by

TABLE I  
MEAN TIME, BEST LENGTH AND GAP PER SOLVED INSTANCE PER SOLVER USED IN THE DOTTED BOARD MODEL.

| Instance | Gurobi   |    |         | CPLEX    |    |         |
|----------|----------|----|---------|----------|----|---------|
|          | Time (s) | L  | Gap (%) | Time (s) | L  | Gap (%) |
| RCO1     | 1.74     | 8  | 0       | 1.22     | 8  | 0       |
| RCO2     | 37.83    | 15 | 0       | 2.74     | 15 | 0       |
| RCO3     | 857.09   | 22 | 0       | 778.18   | 22 | 0       |
| BLAZ1    | 1.38     | 8  | 0       | 1.52     | 8  | 0       |
| BLAZ2    | 41.85    | 14 | 0       | 36.20    | 14 | 0       |
| BLAZ3    | 1657.20  | 20 | 0       | 23452.03 | 20 | 0       |
| SHAPES2  | 6.74     | 14 | 0       | 2.17     | 14 | 0       |
| SHAPES4  | 36000    | 27 | 44      | 36000    | 25 | 4       |

the directive *change* in the *JuMP* library for Julia Language. Table I provides baselines results for the proposed algorithm, and it can be observed that for all the tested instances (except SHAPES4), the optima values could be found.

Next, the BRKGA with dotted board model was executed against the datasets. In turn, more instances are considered even assuming that the model did not find optimal value in its execution. Based on preliminary tests, the following parameters are selected as they lead to overall better results: population size of 40 individuals; elite population percentage of 20% (8 individuals); the probability of mutation of 30% (12 possible individuals). The remaining individuals (20), for instance, make up the population, possibly, with genetic information exchanged. The algorithm ends after 150 generations without improvement of the best solution or after a timeout of 36,000 seconds (TL).

Table II show all chosen instances, the required lead time, length and, gap for dotted board model in comparison with our heuristic. It can be observed that the time shown is that when the optimal solution was first found. Because of this, the SHAPES9 instance has a longer time than SHAPES15, even though it is more complex to solve because of its larger size. Also, the results of RCO<sub>3,4,5</sub> and BLAZ<sub>3,4,5</sub> exposed refer to the second test performed, with the population size doubled.

These results confirm that some solutions were found just for heuristic procedure. Interestingly, for RCO<sub>4</sub>, the proposed algorithm could find a result very close to the optimum in reduced running time. In RCO<sub>5</sub> the best length was found; on the other hand, the dotted board model stop after time limit situation with 8.1% of the gap.

Figures 7 and 8 present the resultant layouts generated for instances SHAPES4 and SHAPES9, respectively. The evolution of the solutions with the genetic algorithm can be observed in the graph illustrated in Fig 9. Solution populations for some instances have evolved over the 150 established generations and for others even longer, as finalization only occurs when that number is reached without improvement. However, the instances SHAPES<sub>9,15</sub> did not reach this mark, which is due to the complexity of the polygons that compose it and the number of pieces demanded. Because of this, layout generation takes longer, so the time limit is reached after fewer iterations.

TABLE II  
COMPARISON BETWEEN HEURISTIC AND DOTTED BOARD MODEL

| Instance | dotted board model |    |         | heuristic |    |
|----------|--------------------|----|---------|-----------|----|
|          | Time (s)           | L  | GAP (%) | Time (s)  | L  |
| RCO1     | 0.62               | 8  | 0       | 0.76      | 8  |
| RCO2     | 6.28               | 15 | 0       | 292.48    | 15 |
| RCO3     | 2393.42            | 22 | 0       | 666.11    | 22 |
| RCO4     | TL                 | 29 | 3.5     | 277.79    | 30 |
| RCO5     | TL                 | 37 | 8.1     | 420.39    | 37 |
| BLAZ1    | 0.69               | 8  | 0       | 0.98      | 8  |
| BLAZ2    | 15.98              | 14 | 0       | 486.52    | 14 |
| BLAZ3    | 5583.82            | 20 | 0       | 35.86     | 22 |
| BLAZ4    | TL                 | 28 | 10.7    | 545.19    | 29 |
| BLAZ5    | TL                 | 35 | 14.3    | 55.20     | 37 |
| SHAPES2  | 0.45               | 14 | 0       | 10.28     | 14 |
| SHAPES4  | 17951.33           | 25 | 0       | 6265.11   | 25 |
| SHAPES5  | TL                 | 30 | 13.3    | 2506.72   | 30 |
| SHAPES7  | TL                 | 45 | 39.4    | 10523.56  | 41 |
| SHAPES9  | TL                 | 54 | 40.4    | 16599.28  | 50 |
| SHAPES15 | TL                 | 67 | 40.8    | 9271.36   | 64 |

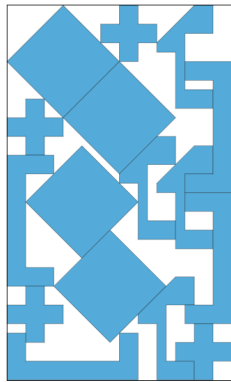


Fig. 7. Layout for SHAPES4

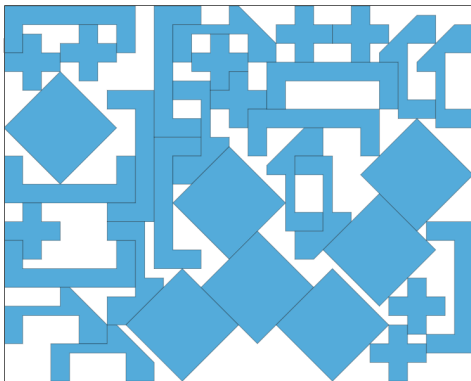


Fig. 8. Layout for SHAPES9

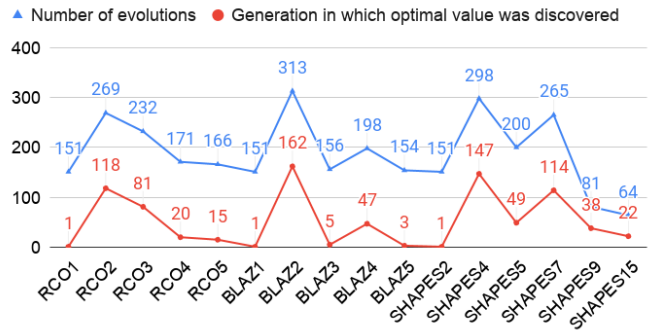


Fig. 9. Evolution of solutions by the heuristic

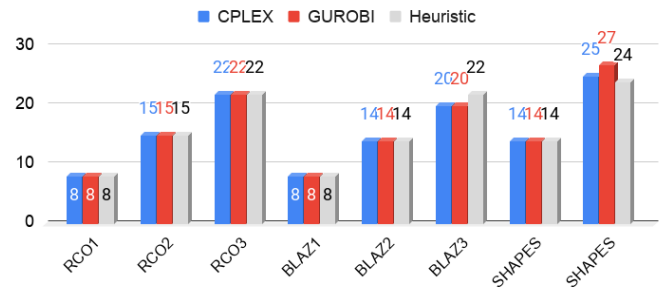


Fig. 10. Optimal values found by each approach

The results show that the CPLEX solver was the only one able to find the optimal value indicated by [14] for all instances tested. The Gurobi solver did not find the optimal value for the SHAPES4 instance, due to the time limit set. The heuristic could not find the optimal value for the instance BLAZ3. The graph in Fig 10 presents a comparison between the optimal values obtained with the three approaches, considering only the instances with proven optimal value.

Regarding time, CPLEX achieved the best performance in half of the cases and the heuristic in the rest. The graph in Fig 11 illustrates the time taken by CPLEX and Gurobi solvers and heuristics to find the optimal value, with a log scale. Only instances with known optimal value are considered.

Figure 12 presents a comparison between the optimal values obtained from the heuristic and the results given by [14]. It is possible to observe that the heuristic found the same or a better solution as [14] in 75% of instances. Moreover, 25% of instances (SHAPES<sub>4,7,9,15</sub>) presented better results. The results of the remaining instances were very close, exceeding a maximum of 2 units.

Despite the disadvantage of computational power, heuristics found optimal values with a competitive time advantage, in the case of some instances. In Fig 13, a log-scale graph shows a comparison between heuristic and model times for instances whose heuristic optimal values were equal or better. In it, it can be observed that in most of these instances (7 of 12), the heuristic was faster.

The results indicate that the heuristic performs better with

## V. DISCUSSION

C&P are practical combinatorial optimisation problems faced by several types of companies such as, for example, in the construction sector, furniture manufacturing, clothing and shipbuilding business. Hence algorithmic solutions aiming the reduction of waste, increased profit and the automation of existing complex human task-driven manufacturing have gained strategical relevance.

The choice of the most suitable packing algorithm among the many introduced in the academic literature depends on aspects such as time constraints and the availability to in-house resources. Exact approaches such as mathematical formulations for C&P problems have demonstrated limited feasibility, especially in the presence of a large number of required items. Hence, recent packing algorithms employ metaheuristics, which are optimisation techniques for delivering satisfactory solutions in a reasonable time.

The algorithm approach for the ISSP showed in this study combines the BRKGA with the dotted board model. The motivation is to leverage from search mechanisms in the BRKGA that exploit promising areas of the search space with constrained global optimisation methods like the dotted board model. The results achieved in this study demonstrates the efficiency of the proposed method, especially for solving larger solutions that resemble more closely to demands observed in real-world demands.

## VI. CONCLUSIONS AND FUTURE WORK

C&P are hard combinatorial optimisation problems that can be found in several manufacturing processes such as, for example, clothing manufacturing [3] and ship container loading [6]. The spectrum of C&P algorithms in the literature is diverse, often sharing the trade-off between computationally expensive computation for finding global optima solutions and efficient running time. Hence most of the literature employs metaheuristics capable of selecting good-enough solutions in a reasonable time.

This paper presented a variant for the BRKGA which uses the dotted board model combined with the CFR technique for solving the two-dimensional ISSP problem. Computational experiments against datasets in the literature demonstrated the efficiency of the proposed method, which presented competitive results for most of the instances. The proposed method obtained equal or better results than other methods for 75% of the tested instances. For the remaining instances, it obtained near-optimal solutions in significantly less running time.

In future research, the authors expect to exploit the results from different packing algorithms combined with features of the input data. Such information enables the use of data science techniques and statistical models (e.g. deep learning) that can predict the most suitable packing algorithm, packing performance and relevant features in C&P problems.

## REFERENCES

- [1] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European journal of operational research*, vol. 183, no. 3, pp. 1109–1130, 2007.

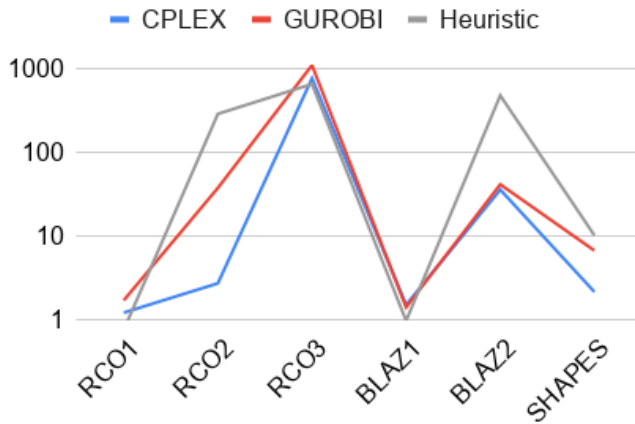


Fig. 11. Solvent and heuristic times to find the optimal value

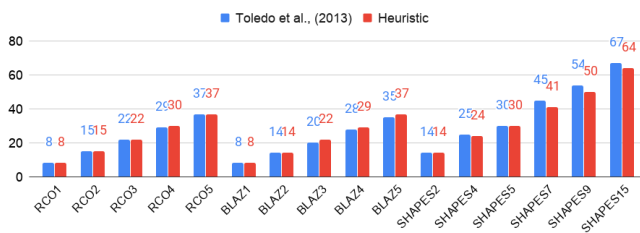


Fig. 12. Optimal values found by each approach

the SHAPES instance group since all of them obtained equal or better results and in less time than [14]. The BLAZ group presented low performance, since only 40% of the group instances were resolved optimally and in a longer time. In the RCO group, only one instance did not reach optimality and time performance was better for larger instances and worse for smaller ones.

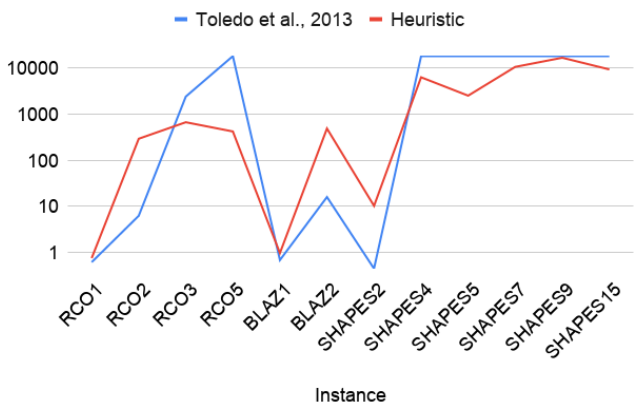


Fig. 13. dotted board model and heuristic times to find the optimal value

- [2] I. Elamvazuthi, S. Kamaruddin, and M. S. Azmi, "Automation of nesting and cutting processes of leather furniture production: a case study," *International Journal of Mechanical & Mechatronics Engineering*, vol. 9, no. 10, pp. 25–29, 2008.
- [3] J. Heistermann and T. Lengauer, "The nesting problem in the leather manufacturing industry," *Annals of Operations Research*, vol. 57, no. 1, pp. 147–173, 1995.
- [4] K. Hamada, Y. Ikeda, H. Tokumoto, and S. Hase, "Development of automatic nesting system for shipbuilding using the branch-and-bound method," *Journal of Marine Science and Technology*, vol. 24, no. 2, pp. 398–409, 2019.
- [5] Z. Jin, K. Ohno, and J. Du, "An efficient approach for the three-dimensional container packing problem with practical constraints," *Asia-Pacific Journal of Operational Research*, vol. 21, no. 03, pp. 279–295, 2004.
- [6] L. J. P. de Araújo and P. R. Pinheiro, "Combining heuristics backtracking and genetic algorithm to solve the container loading problem with weight distribution," in *Soft Computing Models in Industrial and Environmental Applications, 5th International Workshop (SOCO 2010)*, E. Corchado, P. Novais, C. Analide, and J. Sedano, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 95–102.
- [7] K. A. Dowsland, S. Vaid, and W. B. Dowsland, "An algorithm for polygon placement using a bottom-left strategy," *European Journal of Operational Research*, vol. 141, no. 2, pp. 371–381, 2002.
- [8] B. K. Nielsen and A. Odgaard, "Fast neighborhood search for the nesting problem," *Sci. Report, University of Copenhagen*, 2003.
- [9] A. A. Leao, F. M. Toledo, J. F. Oliveira, and M. A. Carravilla, "A semi-continuous mip model for the irregular strip packing problem," *International Journal of Production Research*, vol. 54, no. 3, pp. 712–721, 2016.
- [10] L. H. Cherri, L. R. Mundim, M. Andretta, F. M. Toledo, J. F. Oliveira, and M. A. Carravilla, "Robust mixed-integer linear programming models for the irregular strip packing problem," *European Journal of Operational Research*, vol. 253, no. 3, pp. 570–583, 2016.
- [11] T. de Castro Martins and M. d. S. G. Tsuzuki, "Simulated annealing applied to the rotational polygon packing," *IFAC Proceedings Volumes*, vol. 39, no. 3, pp. 475–480, 2006.
- [12] L. J. P. de Araújo, E. Özcan, J. A. Atkin, and M. Baumers, "Analysis of irregular three-dimensional packing problems in additive manufacturing: a new taxonomy and dataset," *International Journal of Production Research*, vol. 57, no. 18, pp. 5920–5934, 2019. [Online]. Available: <https://doi.org/10.1080/00207543.2018.1534016>
- [13] J. F. Gonçalves and M. G. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *Journal of Heuristics*, vol. 17, no. 5, pp. 487–525, 2011.
- [14] F. M. Toledo, M. A. Carravilla, C. Ribeiro, J. F. Oliveira, and A. M. Gomes, "The dotted-board model: a new mip model for nesting irregular shapes," *International Journal of Production Economics*, vol. 145, no. 2, pp. 478–487, 2013.
- [15] B. A. Júnior and P. R. Pinheiro, "Approaches to tackle the nesting problems," in *Artificial Intelligence Perspectives in Intelligent Systems*. Springer, 2016, pp. 285–295.
- [16] B. Chazelle, "The bottomn-left bin-packing heuristic: An efficient implementation," *IEEE Transactions on Computers*, no. 8, pp. 697–707, 1983.
- [17] J. A. Bennell, K. A. Dowsland, and W. B. Dowsland, "The irregular cutting-stock problem—a new procedure for deriving the no-fit polygon," *Computers & Operations Research*, vol. 28, no. 3, pp. 271–287, 2001.
- [18] E. K. Burke, R. S. Hellier, G. Kendall, and G. Whitwell, "Complete and robust no-fit polygon generation for the irregular stock cutting problem," *European Journal of Operational Research*, vol. 179, no. 1, pp. 27–49, 2007.
- [19] A. M. Gomes and J. F. Oliveira, "Solving irregular strip packing problems by hybridising simulated annealing and linear programming," *European Journal of Operational Research*, vol. 171, no. 3, pp. 811–829, 2006.
- [20] J. A. Bennell and X. Song, "A beam search implementation for the irregular shape packing problem," *Journal of Heuristics*, vol. 16, no. 2, pp. 167–188, 2010.
- [21] S. Jakobs, "On genetic algorithms for the packing of polygons," *European journal of operational research*, vol. 88, no. 1, pp. 165–181, 1996.
- [22] A. K. Sato, T. C. Martins, and M. S. G. Tsuzuki, "An algorithm for the strip packing problem using collision free region and exact fitting placement," *Computer-Aided Design*, vol. 44, no. 8, pp. 766–777, 2012.
- [23] B. Amaro Júnior, P. R. Pinheiro, and P. V. Coelho, "A parallel biased random-key genetic algorithm with multiple populations applied to irregular strip packing problems," *Mathematical Problems in Engineering*, vol. 2017, 2017.
- [24] M. Fischetti and I. Luzzi, "Mixed-integer programming models for nesting problems," *Journal of Heuristics*, vol. 15, no. 3, pp. 201–226, 2009.
- [25] M. O. Rodrigues and F. M. Toledo, "A clique covering mip model for the irregular strip packing problem," *Computers & Operations Research*, vol. 87, pp. 221–234, 2017.
- [26] S. C. Leung, Y. Lin, and D. Zhang, "Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem," *Computers & Operations Research*, vol. 39, no. 3, pp. 678–686, 2012.
- [27] A. Elkeran, "A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering," *European Journal of Operational Research*, vol. 231, no. 3, pp. 757–769, 2013.
- [28] L. R. Mundim, M. Andretta, and T. A. de Queiroz, "A biased random key genetic algorithm for open dimension nesting problems using no-fit raster," *Expert Systems with Applications*, vol. 81, pp. 358–371, 2017.
- [29] L. J. P. de Araújo and P. R. Pinheiro, "Applying backtracking heuristics for constrained two-dimensional guillotine cutting problems," in *Information Computing and Applications*, B. Liu and C. Chai, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 113–120.
- [30] H. John, "Holland. adaptation in natural and art cial systems," 1975.
- [31] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA journal on computing*, vol. 6, no. 2, pp. 154–160, 1994.
- [32] W. M. Spears and K. D. De Jong, "On the virtues of parameterized uniform crossover," Naval Research Lab Washington DC, Tech. Rep., 1995.
- [33] J. Błażewicz, P. Hawryluk, and R. Walkowiak, "Using a tabu search approach for solving the two-dimensional irregular cutting problem," *Annals of Operations Research*, vol. 41, no. 4, pp. 313–325, 1993.
- [34] J. F. Oliveira, A. M. Gomes, and J. S. Ferreira, "Topos – a new constructive algorithm for nesting problems," *OR-Spektrum*, vol. 22, pp. 263–284, 2000.
- [35] B. A. Junior. Experiment data. [Online]. Available: [https://drive.google.com/open?id=14RbabDOsOCw\\_TsXILJcLHRAN5gIB\\_vSV](https://drive.google.com/open?id=14RbabDOsOCw_TsXILJcLHRAN5gIB_vSV)