

Research Article

A Parallel Biased Random-Key Genetic Algorithm with Multiple Populations Applied to Irregular Strip Packing Problems

Bonfim Amaro Júnior, Plácido Rogério Pinheiro, and Pedro Veras Coelho

Graduate Program in Applied Informatics, University of Fortaleza (UNIFOR), Av. Washington Soares, 1321 Bl J Sl 30, 60.811-905 Fortaleza, CE, Brazil

Correspondence should be addressed to Plácido Rogério Pinheiro; placido@unifor.br

Received 2 April 2017; Revised 6 July 2017; Accepted 1 August 2017; Published 17 September 2017

Academic Editor: Jorge Magalhaes-Mendes

Copyright © 2017 Bonfim Amaro Júnior et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The irregular strip packing problem (ISPP) is a class of cutting and packing problem (C&P) in which a set of items with arbitrary formats must be placed in a container with a variable length. The aim of this work is to minimize the area needed to accommodate the given demand. ISPP is present in various types of industries from manufacturers to exporters (e.g., shipbuilding, clothes, and glass). In this paper, we propose a parallel Biased Random-Key Genetic Algorithm (μ -BRKGA) with multiple populations for the ISPP by applying a collision-free region (CFR) concept as the positioning method, in order to obtain an efficient and fast layout solution. The layout problem for the proposed algorithm is represented by the placement order into the container and the corresponding orientation. In order to evaluate the proposed (μ -BRKGA) algorithm, computational tests using benchmark problems were applied, analyzed, and compared with different approaches.

1. Introduction

The large market dispute between manufacturing and exporters, coupled with the scarcity of some items that make up the raw material for a product manufacturing, has motivated research around the world to find answers that reproduce efficient solutions at a low cost.

Cutting and packing (C&P) problems belong to NP-hard [1]. A dataset of items must be packed in a two-dimensional stage and an objective function can be analyzed through the minimum area needed to place pieces or the maximum number of items allowed in a current layout. Our research focus is on a particular C&P problem commonly referred to as the irregular strip packing problem (ISPP).

Furthermore, according to the typology of [1], the ISPP has an open arbitrary dimension classification. A variation of C&P can be described as a container (C) with a constant width (W) and a variable length (L) and a dataset of irregular polygons, where the objective is to minimize L and place (with no overlaps) the entire demand into C . In some cases all angles to rotate polygons are accepted; in other cases only some angles are allowed to rotate the polygons. Figure 1

illustrates a classical instance of this problem known as Trousers [2].

Various studies have been devoted to obtain solutions (layouts) that minimize surface area and computational time. However, despite the many proposals that have been presented, the analysis of further strategies to obtain improved layouts is still pertinent. Some examples of strategies are as follows:

- (i) A positioning rule must be applied to the list of parts
- (ii) A precise layout is accepted as the initial solution, and overlapping of items is valid, but a penalty is applied to the objective function
- (iii) A set of clusters between parts is performed in the preprocessing phase to reduce the complexity of the fit among the parts
- (iv) Other mathematical programming models.

The approach proposed here was a parallel computing implementation of the Biased Random-Key Genetic Algorithm (BRKGA) [3] using multiple populations applied to an ISPP. Each solution develops a sequence in which the

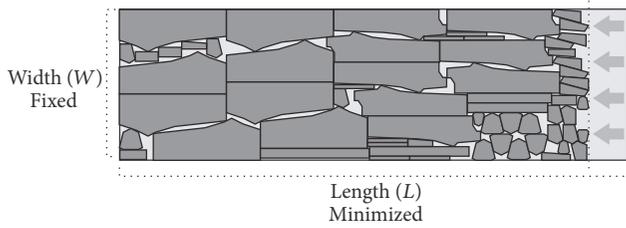


FIGURE 1: An example of a solution to an irregular strip packing problem.

pieces are positioned in the container. The current angle and the method of placement applied was based on a collision-free region [4]. Computational tests carried out on dataset examples of the target problem are discussed, and the method presented here is shown to be compatible with or better than some other important studies.

This paper is structured as follows. Section 2 presents some studies about ISPPs and their different approaches to solve them. Section 3 introduces important concepts associated with the suggested methodology, which is presented in Section 4. Documentation concerning computational tests on well-known literature datasets is given in Section 5. Finally, in Section 6 the conclusions concerning the proposed methodology are presented along with some suggestions to develop this work further.

2. Approaches to Tackle Irregular Strip Packing Problems

ISPPs are a challenge since they present obstacles for the construction of feasible and optimized solutions. Among these are the geometric aspects of the polygons and the combinatorial characteristics for quality layouts. These two obstacles must be managed concomitantly. In addition, an estimate of the computational complexity of each approach must be measured to serve as a criterion to select the methods for the planning phase in algorithmic projects.

Various mathematical programming models have been developed to provide feasible solutions for ISPPs [5, 6]. Although there are still constraints for industrial scale applications, due to combinatorial characteristic optimization, among other factors, the entire computational concept arising from this type of strategy introduces a theoretical foundation of great importance to research in this area.

For instance, in [7], the authors presented a model based on constraint logic programming (CLP) adapted to ISPPs. Also, this strategy was applied to convex and nonconvex polygons. The concepts of NFP and IFP were used to satisfy the geometric constraint of overlap and to show the domain of the positioning of the first item. The composition of the layouts, using the fit patterns, adopts a CLP model with variables within a finite domain. The width, the upper boundary of the stage place, and the geometry of the pieces from the dataset compose the initial variables of the proposed model. Once the initial settings are defined, the approach has a layer that manages the constant changes in the variable domains. Moreover, it handles the consistency of the process.

On the other hand, the authors in [8] presented a mixed integer linear programming (MILP) approach to ISPPs based on the work of [9]. The aim was to maximize efficiency while minimizing the length needed for the layout. Another aim of the objective function was that all the pieces must still be as near as possible to their origin position, as if compacted. This process is carried out by minimizing the coordinates of the reference points of each polygon. The objective function calculation does not consider the wastage and holes between the items. All tests were made for cases in the garment industry, and the model runtime was just a few seconds.

Likewise, in [10], the authors also proposed an MILP; however, the decision variables, in this case, are binary and are associated with each discrete point present on board, a nomenclature for the container. An outstanding advantage of this model is its flexibility to tackle the geometry of the items and the container and therefore allowing an extension to more complex problems. For instance, this approach can be applied to nonconvex boards and items with holes. On the other hand, there is a limitation on the number of parts to run in a feasible time.

Another model based on CLP was presented by [11]. The researchers reported constraints aiming to ensure all feasibility requirements to achieve a solution. By changing the definition of the variables, which have a binary domain, this approach diverges from the models referred to by [7, 12] which can be applied to the problem with fixed containers.

Nevertheless, methods from various fields of computational science, such as the concepts of metaheuristics and dynamic programming, have been applied to produce feasible solutions for ISPPs and to reduce their computational times. Several approaches in the literature have considered these concepts in order to obtain estimates for their use in real cases [13–15]. However, this type of strategy leads us to a thorough investigation of the calibration parameters.

In the field of metaheuristics, Jakobs [16] exposed an approach that combines a genetic algorithm and a Bottom-Left (BL) strategy as a constructive heuristic for the positioning of items. The chromosome coding was mapped as an ordered sequence of parts, also applied by [17, 18]. On the other hand, in [19], the authors used the same idea, but a raster representation for the pieces.

Incidentally, in [20], the authors proposed a probabilistic heuristic known as 2-exchange as a parameter Δ to control the change between neighbors items. This feature coordinates the size of the search between the neighborhoods by assigning the maximum amount of permutations between items in a complete sequence. Also, the authors used an NFP-based mechanism to position pieces in spaces left empty by the solutions.

Furthermore, in [21] these authors presented a new constructive heuristic based on the BL with the characteristic to fill empty spaces in the layouts, innovating one of the biggest limitations of the traditional method. The translation of the items begins on the Bottom-Left side, and the horizontal sliding is done discretely, from a point inside a grid, while vertical movement is performed in continuous space. So, it determines the valid positions; the NFP algorithm was applied. This technique is combined with hill climbing and a tabu search.

Another strategy was developed by [4]. In this paper, the approach was divided into two steps, aiming to obtain efficient solutions for ISPPs. The initial stage consists of the Simulated Annealing algorithm, considering the dimensions of a container as being fixed. The initial solution is randomly generated, the sequence of positioning items is represented by integer values as well as the permissible rotations for each case, and the location point is an ordered pair of real numbers within the interval $[0, 1]$.

At the last level, the control over two attributes of the approach is performed: the value of the size of the container dimensions and the initial temperature applied Simulated Annealing (SA). As soon as a layout is found, at the end of the first stage, this level reduces the area and restarts the SA process. If no solution is found, the container length is increased and the initial stage restarted. Also, two parameters are used to manage compaction and expansion, respectively, of the container. The authors of [22–24] applied the same idea as this approach but used a method of relaxed positioning method with the aim of minimizing the amount of overlap between the pieces.

Techniques that combine models in mathematical programming and artificial intelligence strategies are also possible. These approaches are known as hybrids and, similarly, several studies have investigated their results and limitations.

For instance, in [7], the authors proposed an approach that differs from others by not requiring a positioning rule. A discrete container and a CLP model was adopted, and the objective function was to find the best point of placing, besides the current fixed solution.

In [25], a Simulated Annealing (SA) was combined with a linear programming model. First an initial layout is obtained by applying the Bottom-Left method, and each piece is selected according to a length-based criteria. The SA guides the search over the solution space where each neighborhood structure handles linear programming models, which is composed of a compaction and a separation algorithm.

Another combination between local and guided search applied to two- and three-dimensional items was proposed in [24]. The Bottom-Left method finds an initial layout length. By reducing this value, there are overlapping situations, which are removed by a local search through one of the following changes: horizontal and vertical translations, rotation, or flipping. The guided local search is adopted to try escape of solutions in local minima. Another outstanding aspect is the computational complexity. Given a simple polygon with m vertices and a set of n edges, the algorithm can find the translation with the minimum number of overlays in $O(mn \log(mn))$.

Also, an extended local search algorithm based on nonlinear programming is designed in [23]. The procedure starts from a feasible layout, and its length is set as best. Then a new layout is obtained by random exchanges between two polygons at the current solution. Through a time limit, the length is reduced, and a local search manages the overlap subproblem.

The approaches presented in this section were divided into three distinct categories as [26] mathematical programming, metaheuristics, and mixed strategies. For our

research, we adopted a Biased Random-Key Genetic Algorithm (BRKGA) applied with multiple parallel populations to tackle ISPP.

3. Supporting Concepts

Before discussing the proposed algorithm, we need to describe some supporting concepts. These concepts are geometric representations; no-fit polygon (NFP), and Inner-Fit Polygon (IFP); collision-free region (CFR); and Biased Random-Key Genetic Algorithm (BRKGA).

3.1. Geometric Representations. The geometric representation of nesting problems is an important choice to obtain feasible solutions and will impact directly on the availability of geometric tools that can be used to perform the overlap detection in the layout. The problem aspect depends on the adopted representation. The impact of choice depends on the accuracy of the overlap detection, on the memory requirements needed, and on the computational performance to execute the approach. There are two basic types of representation for polygons: raster or geometric (Figure 2).

Raster representation uses a matrix to describe the geometry of an item. This matrix is obtained from the definition of a default size grid. A grid position that has some internal region of the item is marked with a different value. The rasterized representation has the advantage of being able to represent generic geometry items, and there are simple geometric tools to deal with overlaps. However, if the grid resolution is low, the shape of the object may be different and change a solution. But if the resolution of the grid is high, the computational cost to find overlaps is expensive.

Moreover, polygonal representation uses polygons to represent the items. A list of bidimensional vectors containing the coordinates of vertices is applied to compose the shape of the item. Sometimes, the items cannot be represented in their exact form; in these cases, an approximation of vertices is applied. The geometric calculations to overlap detection are complex. Due to their precision, several approaches in literature use polygonal representation.

There are other representations. However, there has been little research with them. To typify the pieces that need better points precision, linear or nonlinear functions can be applied (ϕ -functions). This representation strategy is the mathematical expression obtained from the distance equation between two objects. They were first presented and applied by [27]. The combinations of primary objects serve to compose complex ones.

The ISPP demands a suitable geometrical representation when continuous rotations are necessary for a set of pieces. In these cases, pieces can be represented by circles (Circle Covering) [28]. Due to the irregular pattern of pieces, the circle representation, of the problem in case, needs to agree with an approximation of polygons.

3.2. No-Fit Polygon (NFP) and Inner-Fit Polygon (IFP). This trigonometric technique was presented in [29] as a shape envelop and changed to NFP by [30]. The idea behind this

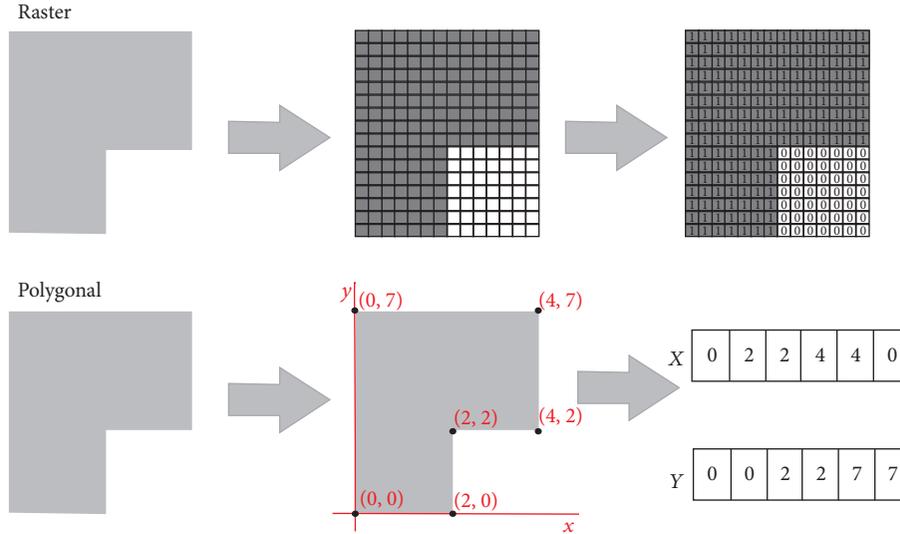


FIGURE 2: A raster and polygonal representation.

trigonometric technique is as follows: given two polygons, P_i (the fixed piece) and P_j (the orbital piece), and a reference point on P_j called r_j , the no-fit polygon of P_i in relation to P_j denoted as $NFP_{(i,j)}$ is the locus of points traced by r_j when P_j slides around the external contour of P_i as shown in Figure 3.

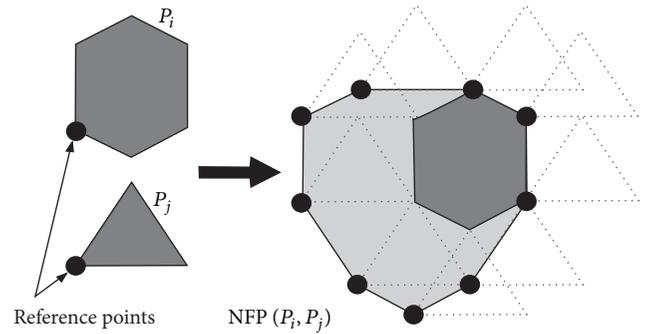
From this definition, three situations may arise:

- (i) If r_j stays inside $NFP_{(i,j)}$ (grey area) then P_j intersects P_i .
- (ii) If r_j strays over the boundary of $NFP_{(i,j)}$ then P_j touches P_i .
- (iii) If r_j stays outside $NFP_{(i,j)}$ then P_j does not intersect or touch P_i .

In our implementation, all possible NFPs are computed in a preprocessing level applying the Minkowski sum [31] method and polygon decomposition is based on [32]. Given two arbitrary point sets, A and B , the Minkowski sum of A and B is represented as $A \oplus B = \{a + b: a \in A, b \in B\}$. A simple vector algebra can be applied to show that $A \oplus (-B)$, known as the Minkowski difference of A and B , is equivalent to $NFP_{(A,B)}$ since a stipulation that polygons stay in counter-clockwise orientation, $-i$, is simply B in a clockwise direction.

The concept of NFP as described is applied only for convex polygons. However, the application of decomposition in nonconvex polygons normally constructs several subpolygons, and consequently many no-fit polygons must be created. In computational times, a unique no-fit polygon intersection is fast, but an additional calculation must be considered to recombine all decomposed parts. Therefore, if a large number of parts are needed or the polygon contains holes, the time to complete this operation is expensive.

In [33], the authors analyzed different decomposition and applied recombination operations to construct the Minkowski sums of nonconvex polygons. The first inference

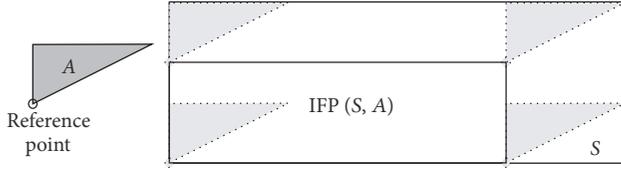
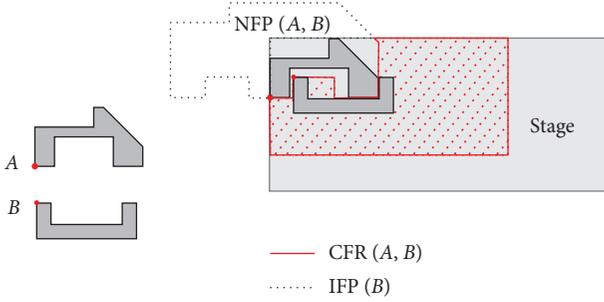
FIGURE 3: The no-fit polygon $NFP_{(i,j)}$ of two convex polygons P_i and P_j .

is that the computation times to find optimal decomposition does not compensate the possible benefits that may occur in the recombination process.

In [34] the author presented an approximation algorithm which takes $O(n \log n)$ time and $O(n)$ space, where n is the vertices number of the input polygon and outputs a decomposition whose size is guaranteed to be no more than four times the size of the optimal decomposition.

As observed in [32], the authors proposed an efficient heuristic for decomposing a polygon into convex subpolygons. The algorithm works this way. Initially, it tries to subdivide the polygon by connecting two reflex vertices with an edge. When this is not possible anymore, it eliminates all reflex vertices by connecting them to other convex vertices, such that the new edge best approximates the angle bisector of the reflex vertex. The algorithm operates in $O(n^2)$ time and takes $O(n)$ space in the worst case, where n is the number of vertices of the input polygon.

The Inner-Fit Polygon (IFP) [20] is similar to the NFP except that the orbiting polygon, in this case, slides around


 FIGURE 4: The Inner-Fit Polygon $IFP_{(S,A)}$ of stage S and piece A .

 FIGURE 5: The collision-free region $CFR_{(S,B)}$ of stage S and piece B .

inside the stationary polygon. The IFP represents a set of points that allow the placement of a polygon inside a hole of another polygon, usually the container. Approaches to tackle NFP can be adjusted to handle IFP. If the outline of the IFP is a rectangle, the IFP is called Inner-Fit-Rectangle (IFR). This happens when generating the IFP from the container, since it is, most of the time, a rectangle. An IFR can be seen in Figure 4.

3.3. Collision-Free Region (CFR). The collision-free region (CFR) is a geometric procedure that allows the feasible position points of a piece inside the strip to be found, efficiently. This technique was developed to plan robot motion in order to present the shortest path without hitting any obstacle. In [4], the CFR was adapted to tackle the irregular strip packing problem using a polygonal Boolean operation.

Consider a stage S and a next piece in sequence B . The $CFR_{(S,B)}$ is composed of the polygon difference (Figure 5) between (1) and (2) described next:

$$(i) \quad IFP_{(S,B)} \quad (1)$$

$$(ii) \quad \bigcup_{i=1}^P NFP(P_i, P_B). \quad (2)$$

In (1), the Inner-Fit Polygon between S and piece B is computed to find a polygon that represents all available points inside S . In the other equation (2), the resultant polygon is composed of a union between the NFPs of B and each piece already placed P_i . The basic case occurs when the stage is empty; then $CFR_{(S,B)} = IFP_{(S,B)}$.

3.4. Biased Random-Key Genetic Algorithm (BRKGA). The Biased Random-Key Genetic Algorithm (BRKGA) is an evolutionary metaheuristic for discrete and global optimization

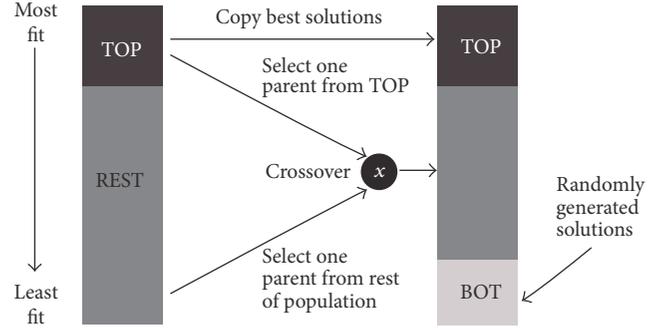


FIGURE 6: Population evolution process to next generations of a BRKGA.

problems based on [35]. Each solution is considered as a list of float numbers in the interval $[0, 1]$, called random keys.

The initial population is generated with p vectors of random keys. This procedure is the starting point of the BRKGA method [3]. In all iterations, two subvectors are created, the set containing high-valued solutions (elite) and the residual individuals (nonelite). The best elements (elite) are transcribed, similarly, to the next population. A small number of random-key vectors (the mutants) are added to the population of the next iteration. The remaining elements of the population of the next iteration are generated by combining with the parameterized uniform crossover of [36], pairs of solutions, where one is elite and the other not.

The main difference between BRKGA and RKGA is the way that the operator of choice and crossover are implemented. In the RKGA procedure both parents are chosen from the entire population and in BRKGA one parent is always selected from the elite set, while the other is chosen from the nonelite (or, in some cases, the rest of the population). Both algorithms combine parents with uniform crossover based on [36], but in BRKGA, one parent belongs to the elite while the other is in the nonelite group. Figure 6 illustrates the evolutionary process for each new population with TOP, REST, and BOT parts of found individuals.

In [37], the authors proposed a RKGA for ISPP and for the positioning method applied three constructive procedures to the polygonal representation of the pieces. A solution is coded as a vector $X = (x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_{2n}, \dots, x_{3n})$, where n represents the number of items of instance. All chromosomes have $3n$ random keys in a real interval $[0, 1]$. The n 's first decide the positioning sequence of items, then x_{n+1}, \dots, x_{2n} shows the angles of rotations and, finally, x_{2n+1}, \dots, x_{3n} stands for the placing rules BL, BL-fill [21], and some heuristics developed in [38].

Furthermore, a BRKGA to ISPP was developed by [39], but in this work, the researches applied a BL heuristic to the positioning items and presented an NFP-raster at a grid model introduced by [10]. Each individual is mapped by a chromosome with n alleles, containing keys with real values in the interval $[0, 1]$. The rotation of the pieces is verified by its corresponding key in the initial vector. Considering that each polygon has k allowable angles, the domain gap of the variables is divided into the same k subintervals.

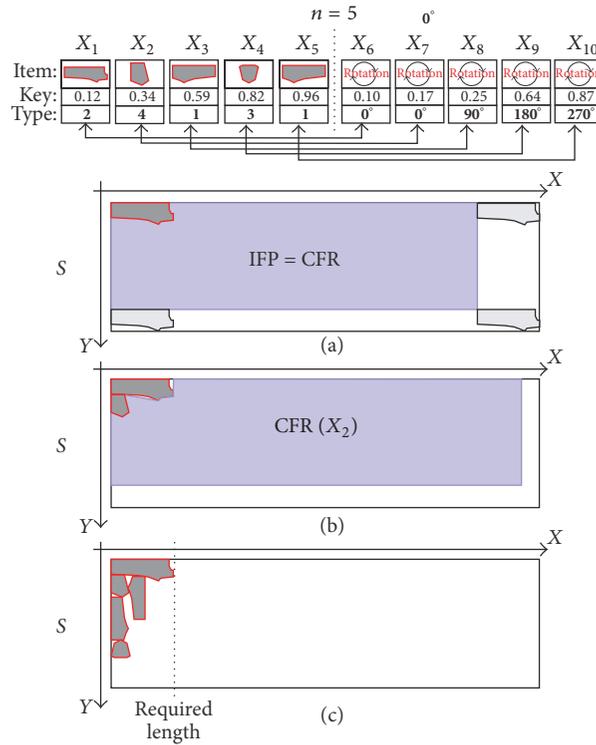


FIGURE 7: Collision-free region positioning method.

4. Proposed Methodology

In the μ -BRKGA heuristic, a chromosome is mapped as a list $X = \{X_1, X_2, \dots, X_{2n}\}$, where n is the total number of pieces to be placed representing an initial solution. A dataset piece number and orientation stand for each gene. Any chromosome contains n genes, which are the total number of items to be packed. The variables X_1, X_2, \dots, X_n determine the Item Packing Sequence (IPS) and $X_{n+1}, X_2, \dots, X_{2n}$ correspond to item packing angle (IPA). All alleles are set in ascending order.

The positioning rule applied to μ -BRKGA is based on the concept of the collision-free region [4] and, therefore, the evaluation considers the minimization of the length for the layout produced for each individual, as well as the maximization of the utilization. For instance, Figure 7 illustrates the procedure applied in a dataset with 5 items.

The positioning rule process receives a decoded vector as input in an ascending order according to the allele key. The CFR polygon is computed for all pieces at IPS rotated by their corresponding IPA. In Figure 7(a) the first piece, stage (S) is empty, so $CFR(X_1)$ equals $IFP(S, X_1)$. Also, in Figure 7(b) the remaining items are placed at the leftmost coordinate, choosing smaller x (left) and minor y (bottom), from the CFR polygon. Then for the last step of the process, the layout length can be assigned and the corresponding chromosome is evaluated (Figure 7(c)).

A BRKGA procedure is applied to a single population. However, this algorithm can be changed to accept more than one population. The method here uses μ populations, where

μ depends on many of processor kernels on a target test computer. All initial populations are assigned some random solutions in an independent way. The information exchange is carried out according to the attribute P_{change} that defines the periodicity of this process. If $P_{\text{change}} = 1$, in every new generation, the population evolution process will be modified using the following rule:

- (i) The “TOP” solutions P^c for each population P_i in $\text{Generation}_{(i)}$ are changed for worst layouts of “REST” solutions space of population P_{i+1} in $\text{Generation}_{(i+1)}$, considering P as a circular list, where the last element P_μ sends information to P_1 ; see Figure 8.

The main goal of this strategy is to improve the variability of good solutions between all populations, left in charge of the crossover process (REST), mutation solutions (BOT), elitism (TOP), and, at CFR positioning method, the work to construct new solutions to be evaluated and translated to next generations until stop criteria are verified. Figure 9 shows a flowchart of the proposed approach, discussing all levels: preprocessing, algorithm, and final solution.

In the preprocessing stage, all control parameters are defined (see Section 5) and assigned. For each instance selected, the no-fit polygons are computed for every pair of polygons type. At the next level, we considered our modified BRKGA (μ -BRKGA) with the information exchange rules and μ populations. In the last step, the best solution is picked as the one with the shortest length between all the layouts of each population adopted.

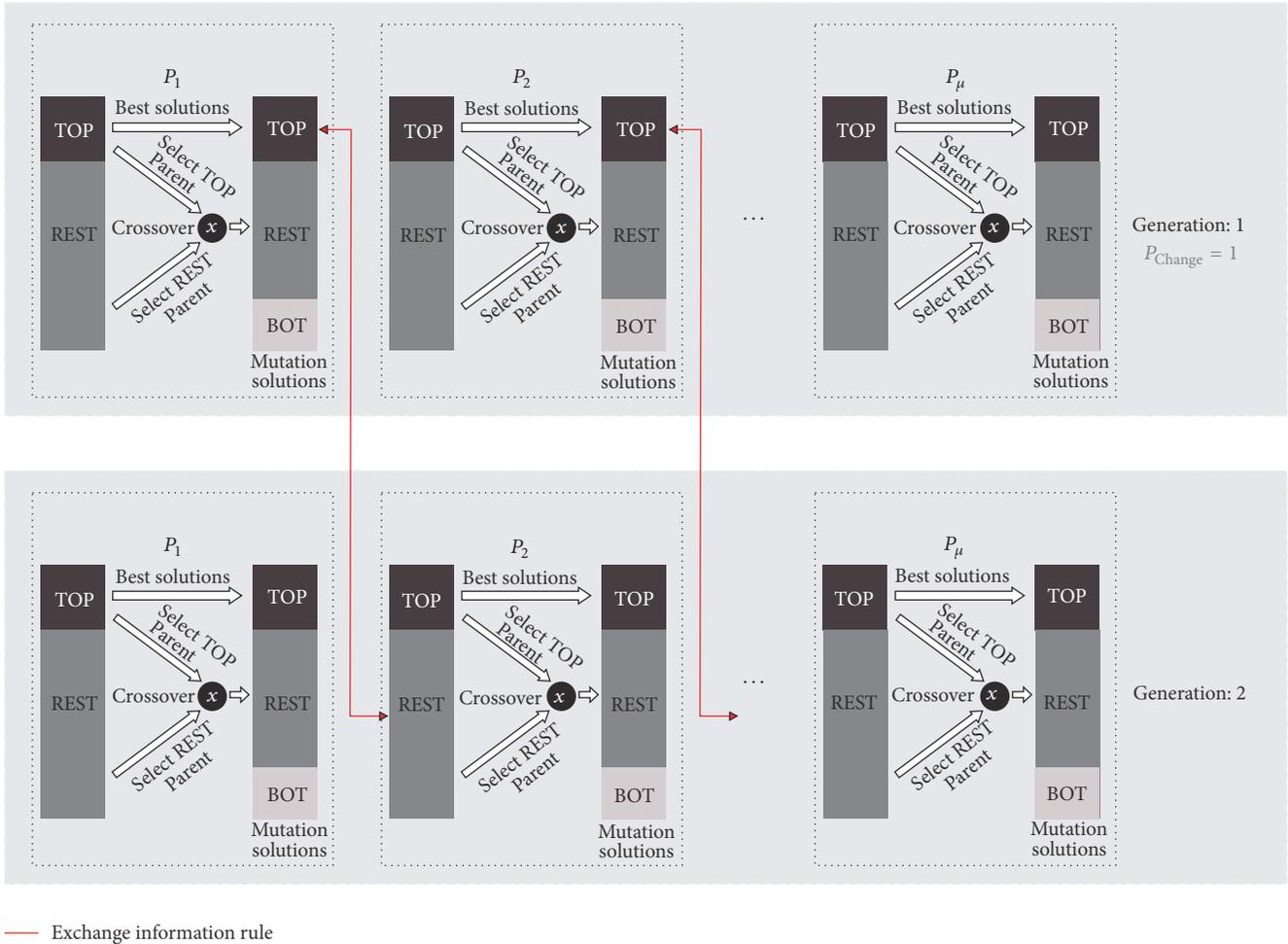


FIGURE 8: Information exchange rule for the next generations of a BRKGA with multiple populations.

5. Computational Experiments

The algorithm was written in Java Language. Computations tests were carried out on a machine with a 3.2 GHz Intel i7-960 CPU with 4 kernels ($\mu = 4$) and 8 GB of RAM memory. To measure the results of the μ -BRKGA, a set of tests was carried out on well-known problems extracted from the EURO Special Interest Group on Cutting and Packing (ESICUP) [40]. Also jigsaw puzzles (Dighe 1 and Dighe 2), clothes examples (Marques, Albano, Trousers, and Swim), and datasets created artificially (Jakobs 1, Jakobs 2, Shapes 0, and Shapes 1) were available. Table 1 presents some information of these examples. The following parameters for the parallel genetic algorithm with multiple populations were chosen: total number of individuals $P = 200 * \mu$ (200 per population), maximum number of generations of 200, $P^e = 0.3 * P$ (Elite parameter), $P^m = 0.2 * P$ (mutants parameter), $P^a = 0.85$ [36], $P_{change} = 5$ (information exchange processes), and the stop criteria were defined by P_{stop} . If the best solution has the same fitness as the P_{stop} generations, the algorithm ends. The value set for this parameter is 30.

For each instance, the μ -BRKGA algorithm was executed 10 times and the best results are presented in Table 2 with

the focus on utilization percentage found in best solution and average of tests. Also, the same table shows a comparative analysis of the best use among the algorithms SAHA [25], BLF [21], 2DNest [21], BS [31], CFREFP [4], and GCSPC [41]. All these approaches have a prominent importance in the literature for problems of positioning irregular figures.

By analyzing Table 2, the μ -BRKGA application presents significant results in several datasets. In [37], a RKGA procedure for ISPP was implemented and the same tests were executed. Table 3 demonstrates that μ -BRKGA had a faster and better solution than RKGA-NP. The dataset named FU, which presented a prominent result in μ -BRKGA, was not applied in RKGA-NP.

In the RKGA-NP a different process of coding to represent a chromosome was applied and a group of random keys is responsible for choosing the respective positioning rule: Bottom-Left, Bottom-Left-Fill [21], and the heuristics developed in [38]. These characteristics contributed to the longer time required for the results shown in Table 3. However, the positioning method applied in μ -BRKGA reached better solutions in some cases. Figure 10 illustrates the final best layout for Albano, FU, Jakobs.1, and Trousers.

TABLE 1: Problem instances characteristics.

Instance	Quantity of different pieces	Quantity of pieces	Width	Orientations admitted
Albano	8	24	4900	$\{0^\circ, 180^\circ\}$
Dighe 1	16	16	100	$\{0^\circ\}$
Dighe 2	16	16	100	$\{0^\circ\}$
Jakobs 1	25	25	40	$\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$
Jakobs 2	25	25	70	$\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$
FU	12	12	38	$\{0^\circ\}$
Marques	8	24	104	$\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$
Shapes 0	4	43	40	$\{0^\circ\}$
Shapes 1	4	43	40	$\{0^\circ, 180^\circ\}$
Swim	10	48	5752	$\{0^\circ, 180^\circ\}$
Trousers	17	64	79	$\{0^\circ, 180^\circ\}$

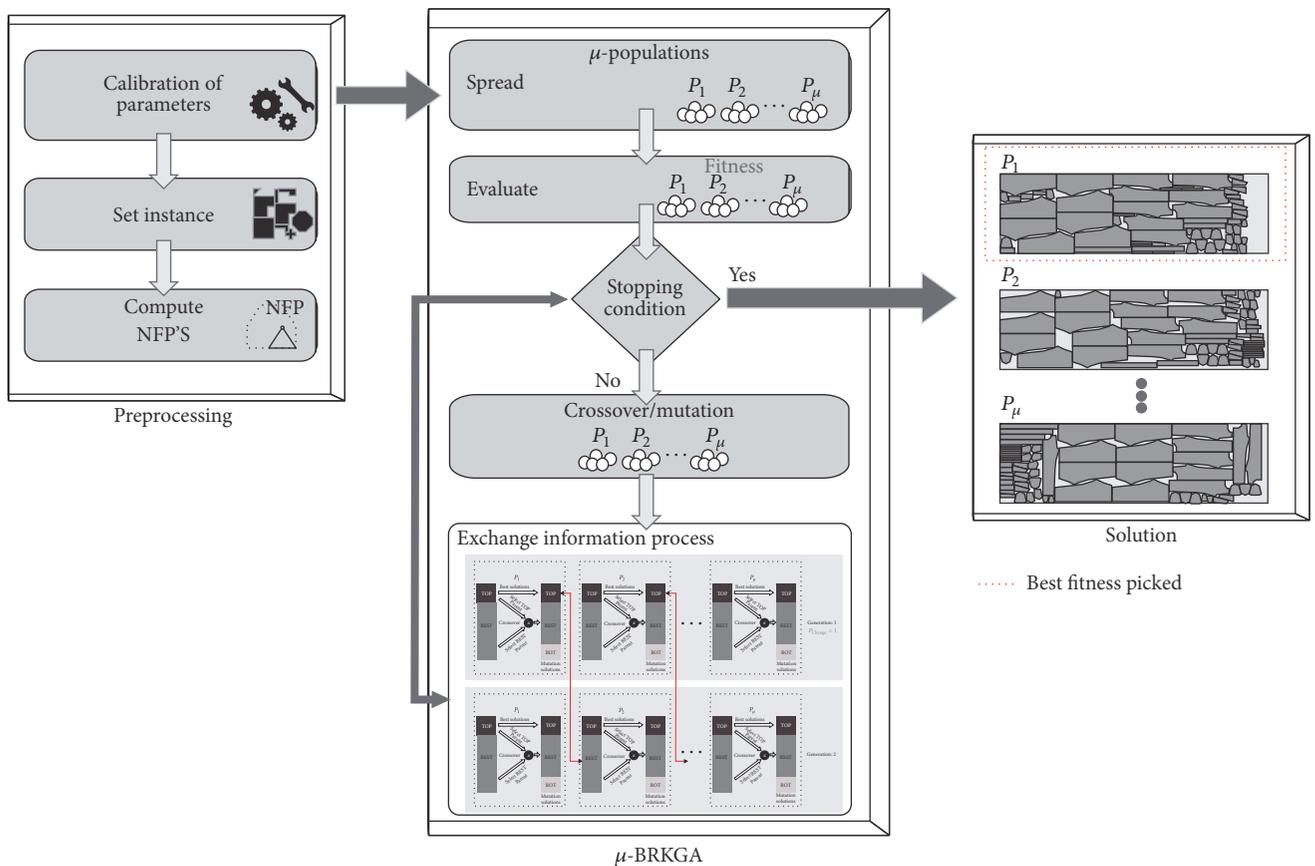


FIGURE 9: Flowchart of entire approach.

6. Conclusions and Future Work

An irregular strip packing problem with a rectangular stage, a fixed width, and an unlimited length was discussed in this paper. To tackle it, we combined a collision-free region placement procedure with a parallel Biased Random-Key Genetic Algorithm with multiple populations, which were also used to develop the order in which the polygons were placed within the stage and their corresponding orientations in order to find the most effective layouts.

ISPPs are present in many production processes in factories, and the approaches to overcome these problems need to find good solutions in a relative short time. In the method proposed here, we chose an effective metaheuristic combined with a convincing placement algorithm to carry this out. The parallel aspect helps to find the best solutions in the space available.

Computational results on datasets of ISPP show that the approach outperformed and/or matched well-known algorithms. As a future work, we aim to modify and implement

TABLE 2: μ -BRKGA \times reference approaches.

Instance	SAHA	BLF	2DNest	BS	CFREFP	GCSPC	μ -BRKGA		
							Best (%)	Avg. (%)	Time (sec.)
Albano	87.43	84.60	87.88	87.88	89.21	89.58	89.11	84.69	4200
Dighe 1	100.00	77.40	99.84	100.00	100.00	100.00	100.00	100.00	1920
Dighe 2	100.00	79.40	93.02	100.00	100.00	100.00	100.00	100.00	2050
Jakobs 1	75.89	82.60	89.03	85.96	89.07	89.10	89.09	78.79	6300
Jakobs 2	77.28	74.80	81.07	80.40	84.83	87.73	79.75	77.90	7150
FU	90.96	86.9	92.03	88.48	91.96	91.96	91.96	88.34	1815
Marques	88.14	86.50	89.82	88.92	90.01	90.59	89.14	84.31	3240
Shapes 0	66.50	60.50	66.42	65.41	67.59	68.79	66.50	63.50	1200
Shapes 1	71.25	66.50	73.23	72.55	72.52	76.73	72.55	65.98	1860
Swim	74.37	68.40	72.49	75.04	71.78	75.94	73.59	70.78	7400
Trousers	89.96	88.50	90.46	90.38	90.07	91.00	90.52	88.67	7250

TABLE 3: Results of RKGA-NP [37] versus μ -BRKGA.

Instance	RKGA-NP			μ -BRKGA		
	Best (%)	Avg. (%)	Time (sec.)	Best (%)	Avg. (%)	Time (sec.)
Albano	84.74	84.69	10230	89.11	84.69	4200
Dighe 1	100.00	99.96	2053	100.00	100.00	1920
Dighe 2	100.00	99.99	2291	100.00	100.00	2050
Jakobs 1	81.67	78.79	10268	89.09	78.79	6300
Jakobs 2	79.75	77.90	12205	79.75	77.90	7150
FU	—	—	—	91.96	88.34	1815
Marques	84.47	84.31	6871	89.14	84.31	3240
Shapes 0	63.74	63.50	1826	66.50	63.50	1200
Shapes 1	66.37	65.98	3025	72.55	65.98	1860
Swim	73.59	70.78	14352	73.59	70.78	7400
Trousers	90.07	88.67	16025	90.52	88.67	7250

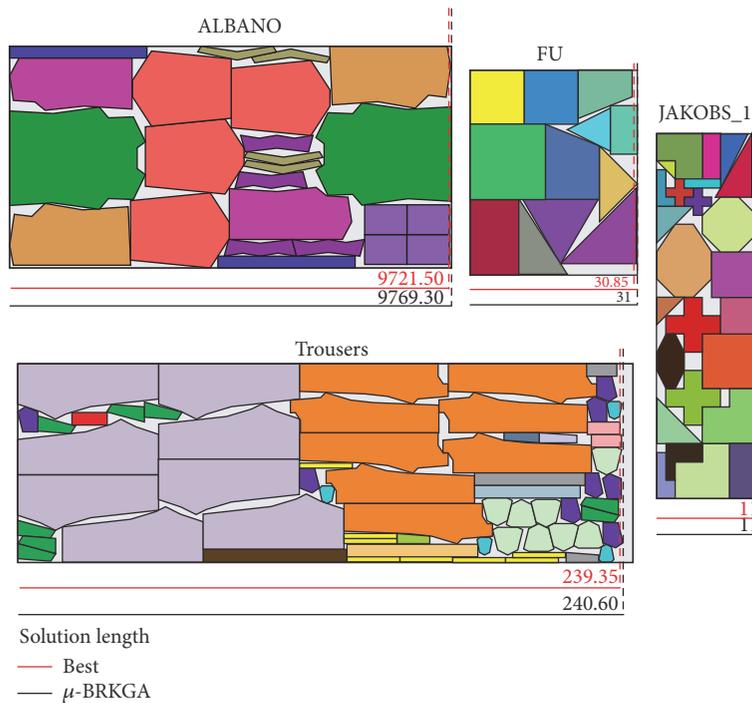


FIGURE 10: Results for datasets FU, Dighe 1, Dighe 2, and Trousers.

different placement procedures; these can be TOPOS [2], Bottom-Left-fill [21], best-fit algorithms, and a more effective method to change information between all populations to be combined with the parallel random-key genetic algorithm. Another possibility is to investigate two-dimensional block building algorithms to place the polygons in pairs and thus reduce the computational time since that approach is fundamentally based on a metaheuristic process.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors are thankful to Coordination for the Improvement of Higher Level Education Personnel (CAPES), National Counsel of Technological and Scientific Development (CNPq), via Grant no. 475239/2012-1.

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.
- [2] J. F. Oliveira, A. M. Gomes, and J. . Ferreira, "Topos—a new constructive algorithm for nesting problems," *OR Spektrum. Quantitative Approaches in Management*, vol. 22, no. 2, pp. 263–284, 2000.
- [3] J. F. Gonçalves and M. G. C. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *Journal of Heuristics*, vol. 17, no. 5, pp. 487–525, 2011.
- [4] 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," *CAD Computer Aided Design*, vol. 44, no. 8, pp. 766–777, 2012.
- [5] R. Alvarez-Valdes, A. Martinez, and J. M. Tamarit, "A branch & bound algorithm for cutting and packing irregularly shaped pieces," *International Journal of Production Economics*, vol. 145, no. 2, pp. 463–477, 2013.
- [6] B. S. Baker, E. G. Coffman, and R. L. Rivest, "Orthogonal packings in two dimensions," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 846–855, 1980.
- [7] M. A. Carravilla, C. Ribeiro, and J. F. Oliveira, "Solving nesting problems with non-convex polygons by constraint logic programming," *International Transactions in Operational Research*, vol. 10, no. 6, pp. 651–663, 2003.
- [8] M. Fischetti and I. Luzzi, "Mixed-integer programming models for nesting problems," *Journal of Heuristics*, vol. 15, no. 3, pp. 201–226, 2009.
- [9] K. K. Daniels, V. J. Milenkovic, and Z. Li, "Multiple containment methods," Tech. Rep., Center for Research in Computing Technology, Harvard University, Cambridge, UK, 1994.
- [10] F. M. B. 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.
- [11] L. H. Cherri, M. A. Carravilla, and F. M. B. Toledo, "A model-based heuristic for the irregular strip packing problem," *Pesquisa Operacional*, vol. 36, no. 3, pp. 447–468, 2016.
- [12] C. Ribeiro and M. A. Carravilla, "A global constraint for nesting problems," *Artificial Intelligence Review*, vol. 30, no. 1-4, pp. 99–118, 2008.
- [13] B. K. Nielsen and A. Odgaard, "Fast neighborhood search for the nesting problem," *Sci. Report, University of Copenhagen*, 2003.
- [14] K. A. Dowsland and W. B. Dowsland, "Solution approaches to irregular nesting problems," *European Journal of Operational Research*, vol. 84, no. 3, pp. 506–521, 1995.
- [15] B. A. Junior, P. R. Pinheiro, and R. D. Saraiva, "A hybrid methodology for nesting irregular shapes: case study on a textile industry?" in *Proceedings of the 6th IFAC/ACM Conference on Management and Control of Production and Logistics (MCPL '13)*, pp. 15–20, 2013.
- [16] S. Jakobs, "On genetic algorithms for the packing of polygons," *European Journal of Operational Research*, vol. 88, no. 1, pp. 165–181, 1996.
- [17] B. Amaro Jr., P. R. Pinheiro, and R. D. Saraiva, "A hybrid methodology for tackling the irregular strip packing problem," in *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS '13)*, 11, pp. 396–401, May 2013.
- [18] B. A. Junior, P. R. Pinheiro, R. D. Saraiva, and P. G. Dantas Pinheiro, "Dealing with nonregular shapes packing," *Mathematical Problems in Engineering*, Article ID 548957, 10 pages, 2014.
- [19] A. Ramesh Babu and N. Ramesh Babu, "A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms," *CAD Computer Aided Design*, vol. 33, no. 12, pp. 879–891, 2001.
- [20] A. M. Gomes and J. F. Oliveira, "A 2-exchange heuristic for nesting problems," *European Journal of Operational Research*, vol. 141, no. 2, pp. 359–370, 2002.
- [21] E. Burke, R. Hellier, G. Kendall, and G. Whitwell, "A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem," *Operations Research*, vol. 54, no. 3, pp. 587–601, 2006.
- [22] T. Imamichi, M. Yagiura, and H. Nagamochi, "An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem," *Discrete Optimization*, vol. 6, no. 4, pp. 345–361, 2009.
- [23] S. C. H. Leung, Y. Lin, and D. Zhang, "Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem," *Computers and Operations Research*, vol. 39, no. 3, pp. 678–686, 2012.
- [24] J. Egeblad, B. K. Nielsen, and A. Odgaard, "Fast neighborhood search for two- and three-dimensional nesting problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1249–1266, 2007.
- [25] 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.
- [26] B. A. Júnior and P. R. Pinheiro, "Approaches to tackle the nesting problems," *Advances in Intelligent Systems and Computing*, vol. 464, pp. 285–295, 2016.
- [27] Y. Stoyan and L. Ponomarenko, "Minkowski sum and hodograph of the dense placement vector function," *Reports of the SSR Academy of Science, SER. A*, vol. 10, 1977.

- [28] P. Rocha, R. Rodrigues, F. M. B. Toledo, and A. M. Gomes, "Circle covering using medial axis," in *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS '13)*, pp. 402–407, 2013.
- [29] R. C. Art, "An approach to the two-dimensional irregular cutting stock problem," Technical Report 36.008, IBM Cambridge Centre, 1966.
- [30] M. Adamowicz and A. Albano, "A Solution of the Rectangular Cutting-Stock Problem," *IEEE Trans. Systems, Man, Cybernetics*, vol. SMC-6, pp. 302–310, 1976.
- [31] 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.
- [32] E. Flato and Dan Halperin, "Robust and efficient construction of planar Minkowski sums," in *In Abstracts 16th European Workshop Comput. Geom.*, pp. 85–88, Ben-Gurion University of the Negev, 2000.
- [33] P. K. Agarwal, E. Flato, and D. Halperin, "Polygon decomposition for efficient construction of Minkowski sums," *Computational Geometry: Theory and Applications*, vol. 21, no. 1-2, pp. 39–61, 2002.
- [34] D. H. Greene, "Computational Geometry," in *Adv. Comput. Res.*, F. P. Preparata, Ed., vol. 1, pp. 235–259, JAI Press, Greenwich, 1983.
- [35] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.
- [36] W. M. Spears and K. A. De Jong, "On the virtues of parameterized uniform crossover," in *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 230–237, 1991.
- [37] P. R. Pinheiro, A. J. Saraiva, and R. A. Saraiva, "A random-key genetic algorithm for solving the nesting problem," *International Journal of Computer Integrated Manufacturing*, vol. 28, pp. 1–7, 2015.
- [38] B. A. Junior, P. R. Pinheiro, and R. D. Saraiva, "Tackling the irregular strip packing problem by hybridizing genetic algorithm and bottom-left heuristic," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 3012–3018, Cancun, Mexico, June 2013.
- [39] 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.
- [40] ESICUP, 2016, <http://paginas.fe.up.pt/~esicup/>.
- [41] 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.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

