

Research Article

Dealing with Nonregular Shapes Packing

**Bonfim Amaro Júnior, Plácido Rogério Pinheiro,
Rommel Dias Saraiva, and Pedro Gabriel Calíope Dantas Pinheiro**

*Graduate Program in Applied Informatics, University of Fortaleza (UNIFOR), Avenue Washington Soares 1321,
Bl J Sl 30, 60811-905 Fortaleza, CE, Brazil*

Correspondence should be addressed to Plácido Rogério Pinheiro; placidrp@uol.com.br

Received 11 April 2014; Accepted 7 June 2014; Published 8 July 2014

Academic Editor: Jer-Guang Hsieh

Copyright © 2014 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.

This paper addresses the irregular strip packing problem, a particular two-dimensional cutting and packing problem in which convex/nonconvex shapes (polygons) have to be packed onto a single rectangular object. We propose an approach that prescribes the integration of a metaheuristic engine (i.e., genetic algorithm) and a placement rule (i.e., greedy bottom-left). Moreover, a shrinking algorithm is encapsulated into the metaheuristic engine to improve good quality solutions. To accomplish this task, we propose a no-fit polygon based heuristic that shifts polygons closer to each other. Computational experiments performed on standard benchmark problems, as well as practical case studies developed in the ambit of a large textile industry, are also reported and discussed here in order to testify the potentialities of proposed approach.

1. Introduction

The constant competitiveness between the modern industries requires that a part of the investments has to be directed to the optimization of the production processes. In garment, glass, paper, sheet metal, textile, and wood industries, for instance, the main concern is to avoid the excessive expenditure of raw material required to meet a particular demand.

In this scenario, the two-dimensional irregular strip packing problem is included. Concisely, the irregular strip packing problem is a combinatorial optimization problem that consists of finding the most efficient design for packing irregular shaped items onto a single rectangular object with minimum waste material. More precisely, the problem can be defined as follows. Assume a rectangular object that has a constant width and infinite length. Consider also a collection of irregular items grouped in m types. For each piece type i , characterized by a set of points, there are an associated number of pieces b_i . The objective function of the problem aims to find an arrangement of items onto the rectangular object such that its length is minimized, and two geometric conditions hold. (1) No two pieces overlap with each other. (2) Each packed piece lies entirely onto the rectangular object.

A specialization of this problem is the placement of irregular figures with characteristics similar to regular cut, but dealing with irregular figures, the nesting problems [1]. They have been known as NP-hard due to their difficulty where few exact methods have been reported in the literature [2], where it is possible to find promising solutions by applying methodologies addressed in [3, 4].

The irregular strip packing problem is known to be NP-hard even without rotation [5], meaning that its globally optimal solution is unlikely to be found by polynomial-time algorithms. Solution techniques range from simple placement heuristics that convert a sequence of pieces into a feasible layout to local optimization techniques involving mathematical programming models. In this paper, we propose a novel approach based on the aggregation between a modified genetic algorithm and a greedy bottom-left procedure for tackling the target problem [6, 7]. A shrinking algorithm is also encapsulated into the metaheuristic engine in order to improve the arrangement of pieces.

To have better analysis, the remainder of this paper is organized as follows. In Section 2, we present state-of-the-art methodologies dedicated to the irregular strip packing problem. Section 3 conveys essential concepts related to the pro-

posed methodology, which is introduced in Section 4. In Section 5, some control parameters are discussed. Computational results on benchmark problem instances and a case study on a textile industry are given in Section 6. Finally, in Section 7, we draw conclusions regarding the quality of the solutions provided by our algorithm and make some considerations concerning future development.

2. Literature Review

Problems involving irregular shapes comprise the most difficult class of packing problems. Whatever the constraints or secondary objectives, there are basically three approaches to find suitable layouts: (1) the polygons may be considered one at a time and packed onto the rectangular object according to the sorting criteria or (2) may be nested either singly or in groups into a set of enclosing polygons which are then packed onto the rectangular object, or (3) an initial allocation is improved iteratively.

Jakobs [9], for instance, presented a genetic algorithm in which each individual is represented by a list of pieces. The way they are disposed of in the chromosome defines the order they are placed by the Bottom-Left heuristic. A shrinking algorithm improves the partial solution by shifting the polygons closer to each other. On the other hand, a new method for implementing a bottom-left-fill packing algorithm which allows shapes that incorporate circular arcs and holes to be nested was presented by Burke and Kendall [10]. The placement rule is combined with hill climbing and tabu local search methods, which determine an efficient nesting sequence. Similarly, Oliveira et al. [11] developed the well-known constructive algorithm known as “*Técnicas de otimização para o posicionamento de figuras irregulares*” (TOPOS). The solution grows from a floating origin and both the next polygon to be packed and its position are defined by two heuristics called *local search* and *initial sort*. Different objective functions are proposed to evaluate and compare partial solutions.

In recent papers, mathematical programming techniques have been adopted for solving one of the following sub-problems: *overlap minimization problem*, whose objective is to place all polygons onto a rectangular object with given width and length so that the total number of overlaps between polygons is made as small as possible; *compaction problem*, which requires a feasible design and relocates many polygons simultaneously so as to minimize the strip length; and *separation problem*, which takes an infeasible layout and performs a set of translations of the polygons that eliminates all overlaps and has a minimum total translation. Gomes and Oliveira [8], for instance, hybridized simulated annealing and linear programming. Firstly, the initial layout is obtained by a greedy bottom-left placement heuristic, being that each polygon is selected according to a *random weighted length* criterion. The simulated annealing algorithm guides the search over the solution space where each neighborhood structure handles linear programming models, which are a compaction algorithm (Figure 1) and a separation algorithm (Figure 2). Likewise, an extended local search algorithm

based on nonlinear programming was conceived by Leung et al. [12]. The algorithm starts with a feasible layout, and its length is saved as the best length. Then, a new design is achieved by randomly swapping two polygons in the current solution. Within a time limit, the strip length is reduced, and local search method solves overlap minimization problems. If the new arrangement is feasible, the best solution is updated, and its length is further reduced to find even better solutions. Otherwise, the strip length is increased, and local search is invoked, which is conducted by a Tabu Search technique in order to escape from local minima. A compaction algorithm is used to improve results.

By other means, a successful approach that combines a local search method with a guided local search to deal with two- and three-dimensional irregular packing problems was proposed by Egeblad et al. [13]. An initial strip length is found by a fast placement heuristic. By reducing this value, overlap situations occur, which are removed by a local search that may apply one of the following four changes: horizontal translation; vertical translation; rotation; or flipping. The guided local search is selected to escape from local minima. Finally, in a recent paper [14], Bennell and Song modified the TOPOS placement heuristic and applied it to a beam search tree, which represents the placement order of polygons onto the rectangular object. Each node in the search tree corresponds to a partial solution, which means that partial solutions can be generated in parallel.

3. Related Concepts

To describe our proposed methodology, we first explain essential concepts related to its behavior, involving genetic algorithm, greedy bottom-left heuristic, and no-fit polygon.

3.1. Genetic Algorithm. Introduced by Holland [15] and perfected by Goldberg [16], *genetic algorithm* is a search and optimization technique inspired in the metaphor of biological evolution. Although a genetic algorithm is a heuristic method, which does not guarantee to find the optimal solutions, it can be applied in many optimization problems.

An implementation of a typical genetic algorithm begins with a population of (generally random) n chromosomes, which are evaluated and associated with a particular reproduction rate in such a way that those chromosomes that represent better solutions to the target problem are given more chances to “reproduce” than those which are poorer solutions. The quality of the solution is measured according to the current population. Then, the crossover and mutation mechanisms are implemented with some probability to the chosen chromosomes. Only the fittest individuals take part in the next generation. Once a new population is generated, the stopping condition of the metaheuristic is checked. If the genetic algorithm is not terminated, each candidate solution is again evaluated and the genetic search process renewed.

3.2. Greedy Bottom-Left Heuristic. A traditional constructive algorithm for solving any two-dimensional cutting or packing problem aims to order pieces and then place them in turn,

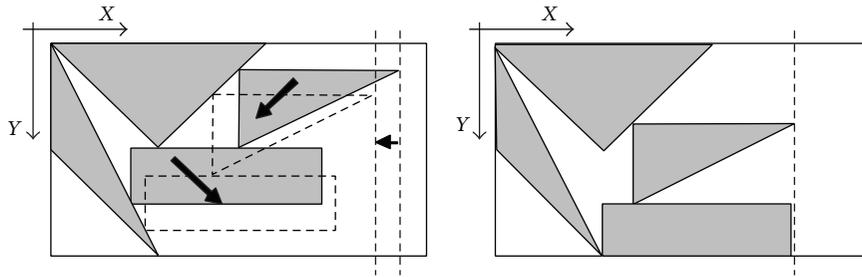


FIGURE 1: Compaction procedure (Gomes and Oliveira [8]).

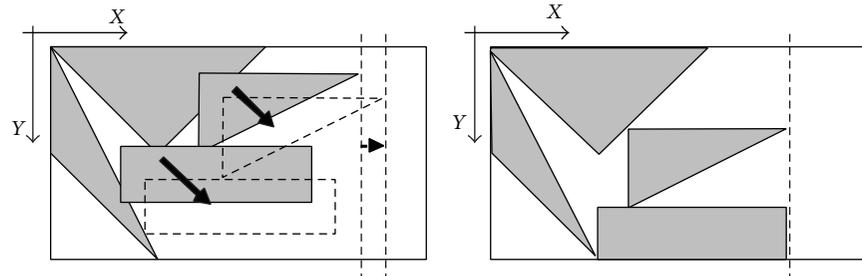


FIGURE 2: Separation procedure (Gomes and Oliveira [8]).

choosing the leftmost feasible position and breaking ties by selecting the lowest, as illustrated by Figure 3. This process, known as *greedy bottom-left heuristic*, was introduced by Baker et al. [17] for packing an arbitrary collection of rectangular pieces into a rectangular bin so as to minimize the layout height. The advantages of this type of approach are its speed and simplicity when compared with more sophisticated methods that may be able to produce solutions of higher quality.

Some papers have considered placement algorithms based on the greedy bottom-left rule in the field of two-dimensional cutting and in this work the aforementioned heuristic was chosen as placement policy.

3.3. No-Fit Polygon. The *no-fit polygon* is a powerful data structure used for fast and efficient handling of geometry in cutting and packing problems involving irregular shapes. The idea behind this trigonometric technique firstly described by Art [18] as *shape envelop* comes as follows. Given two polygons, A (the fixed piece) and B (the orbital piece), and a reference point on B called R_B , the no-fit polygon of A in relation to B , denoted by NFP_{AB} , is the set of points traced by R_B when B slides around the contour of A without overlapping, as displayed in Figure 4. Three situations may arise with respect to the interaction between both shapes. If polygon B is positioned so that its reference point is inside NFP_{AB} , then it overlaps with polygon A ; if the reference point is on the boundary of NFP_{AB} , then polygon B touches polygon A ; finally, if the reference point is outside of NFP_{AB} , then polygons A and B do not overlap or touch. So, the interior of the computed NFP_{AB} represents all intersecting positions of A and B , and the boundary represents all touching positions.

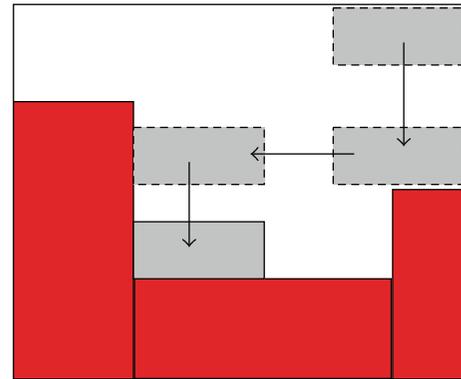


FIGURE 3: Greedy bottom-left procedure for an input piece.

For our implementation, the construction of no-fit polygon was performed by using the Minkowski sum, whose concept involves two arbitrary point sets A and B . The *Minkowski sum* is obtained by adding each point in A to each point in B ; that is, $A \oplus B = \{a + b : a \in A, b \in B\}$. Simple vector algebra can be used to show that $A \oplus -B$, defined as the *Minkowski difference* of A and B , is equivalent to no-fit polygon produced by both shapes. Since we follow the convention that polygons have counter-clockwise orientation, then $-B$ is simply B with clockwise orientation.

4. Proposed Methodology

The proposed methodology prescribes the integration of the distinct components described in Section 3. On what concerns the genetic algorithm, whose control parameters and calibration are set out in Section 5, each individual

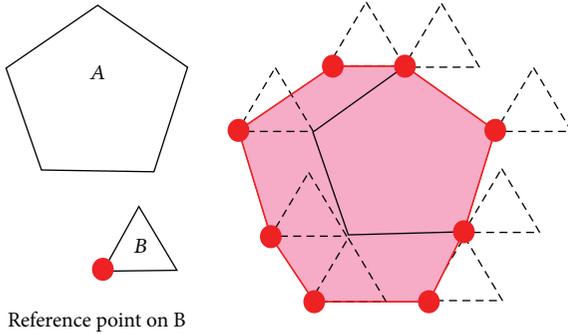


FIGURE 4: No-fit polygon generated by polygons A and B.

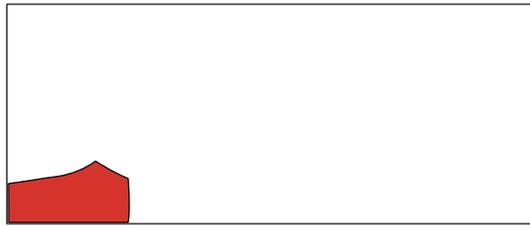


FIGURE 5: Allocation of the first piece.

is encoded by an integer chromosome that represents a placement vector $chrom(i)$, ($i = 1, \dots, m$), which determines the packing sequence of the m polygons onto the rectangular object. For each gene $chrom(i)$, characterized by a piece type t_i , there is an associated rotation variant r_i . The length required to pack all input polygons is assigned as fitness value of the corresponding individual. Having revealed these details, the steps executed by each individual of the current population are presented below.

Step 1. As depicted in Figure 5, polygon $chrom(i)$ is placed onto the rectangular object by the greedy bottom-left rule. Set $i \leftarrow i + 1$.

Step 2. According to the no-fit polygon technique, polygon $chrom(i)$ is positioned on the contour of polygon $chrom(i - 1)$ such that the length of the rectangular object does not increase or is minimized, as presented in Figure 6. In the case of multiple positions, the leftmost one is considered.

Step 3. The greedy bottom-left rule is applied to polygon $chrom(i)$, as illustrated by Figure 7.

Step 4 (termination test). If $i \neq m$, set $i \leftarrow i + 1$ and go to Step 2;

Step 5 (final layout). A shrinking algorithm (Figure 8) is configured as a genetic algorithm operator, working in each generation on the two fittest individuals (i.e., layouts with short lengths) with an occurrence probability. Considering the no-fit polygon technique, this task is characterized by choosing polygons located in P1 (orbital piece) and P2 (stationary piece). After the $NFP_{P1,P2}$ is calculated, the aforementioned packing process is applied for both pieces.

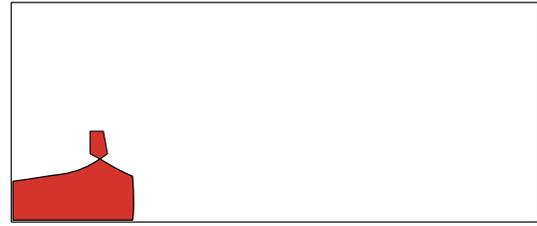


FIGURE 6: Best position computed by no-fit polygon.



FIGURE 7: Final location of pieces.

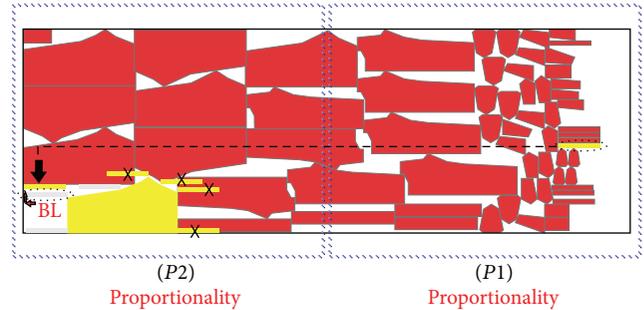


FIGURE 8: Shrinking heuristic.

The main objective of this method is to allocate polygons situated in rightmost spaces in leftmost free spaces in order to decrease the strip length. Furthermore, calculated NFP polygons between all the types of pieces reduce the number of operations during execution, considering that with Polygon processed NFP just a translation to the reference point of the stationary polygon is needed. The preprocessing of the NFP will only run once at the beginning of the algorithm and its complexity depends only on the quantity and types of possible angles.

The methodology flow chart is presented in Figure 9.

5. Control Parameters and Calibration

In this section, we discuss the parameters that control many aspects of the proposed methodology; some of these parameters represent tradeoffs between opposite goals, while others simply allow the fine-tuning of the effectiveness and/or efficiency of the methodology for particular problem instances. However, the set of control parameters discussed here allows the fine tuning of the methodology for a large range of optimization problems. We will see that several parameters were actually set to constant values during the

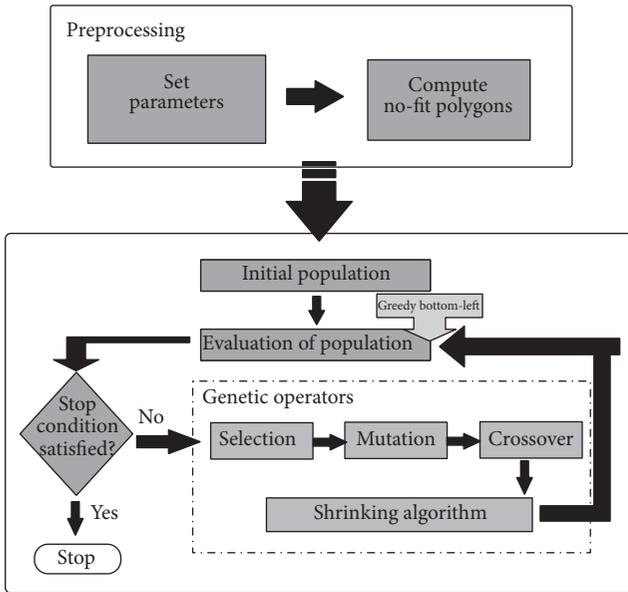


FIGURE 9: Methodology flow chart.

course of our computational experiments, suggesting that certain parameter-value pairs tend to work well on a wide variety of problem instances. On what concerns the meta-heuristic engine, for the sake of computational performance, a simple genetic algorithm instance has been used, which is configured with uniform order-based crossover, swap mutation, fitness proportional parent selection, and age-based population replacement (whereby the parents are replaced by the entire set of offspring). Important control parameters related are described as follows.

(i) *Elitism*. This parameter indicates whether the best current individual is preserved from one generation to the next. Since elitism usually has a positive impact on the search process (as it allows better exploitation of the best individuals generated so far), in our experiments, we enabled this parameter, obtaining gains in convergence without any side effects.

(ii) *Population Size*. A large population size allows for more diversity and, thus, facilitates the exploration of several regions of the search space. On the other hand, a large population size implies slow evolutionary cycles. The best value obtained for this parameter was 400.

(iii) *Crossover Rate*. This parameter determines how often the crossover operator will be applied. If there is no crossover, offspring will be exact copies of parents. If there is a crossover, offspring is generated from parts randomly selected from their parents. To achieve the results reported in Section 6, we adopted the value of 0.90 for this parameter.

(iv) *Mutation Rate*. This parameter controls how often a gene of an individual has its value altered through mutation. While small values for this parameter tend to reduce the genetic algorithm exploration ability, larger values may cause

the premature loss of good individuals from one generation to the next. In our experiments, we have selected the value of 0.2.

(v) *Shrinking Probability*. This parameter controls how often the shrinking method is applied. In our experiments, we have used the value of 0.6.

(vi) *Total Time to Stop*. This parameter (TTS, for short) stipulates the maximum amount of running time allowed for the genetic algorithm to execute. If its value is reached, the execution is interrupted and the best solution found so far is returned. The user according to his/her time availability should set this parameter.

(vii) *Maximum Number of Generations*. This is a secondary parameter used to cease the optimization process and is hereafter referred to as MNG. When the maximum number of generations is reached, the genetic algorithm process is interrupted and the best solution found so far is returned. If desired, this parameter can be set to an undetermined value in order to be disregarded.

6. Computational Experiments

A series of experiments were carried out on a desktop machine with a 3.60 GHz Intel i5 CPU and 4 GB of RAM. The genetic algorithm was implemented in Java and configured with those parameters described in Section 5. Concerning the MNG and TTS, they were set as 200 and 6 hours, respectively.

To evaluate the potentialities behind the proposed methodology, we conducted a series of experiments on benchmark problems available on the EURO Special Interest Group on Cutting and Packing (ESICUP) at <http://www.fe.up.pt/esicup>, which involved five data sets: DIGHE1 and DIGHE2 are jigsaw puzzles with known optimum conceived from Digne and Jakiela [19]; JAKOBS1 and JAKOBS2 are artificial data sets proposed by Jakobs [9]; TROUSERS is an approximation of a real instance taken from the garment industry and it was firstly presented in Oliveira et al. [11]; SHAPES0 and SHAPES1 are artificially created data set, coordinates stated in Oliveira et al. [11]; ALBANO, MAO, and MARQUES are real instances from the textile industry and were presented in [20–22], respectively. Further descriptions of the instances are shown in Table 1.

Table 2 provides the best results achieved by 10 runs of our aggregation method (AM). In this table, for each data set, *Length*, *Utilization* and *Time* denote, respectively, the best length among those produced by all runs, the utilization percentage of the rectangular object of the best solution value, and the time elapsed from the beginning of the run until the best solution found.

In Table 3, we present a comparison of the best results achieved by some state-of-the-art methodologies to solve the irregular strip packing problem, where SAHA by Gomes and Oliveira [8], BLF by Burke and Kendall [10], 2DNest by Egeblad et al. [13], and BS by Bennell and Song [14] are included.

TABLE 1: Data set characteristics.

Data set	Number of different pieces	Total number of pieces	Orientations (degrees)
DIGHE1	16	16	0
DIGHE2	10	10	0
JAKOBS1	25	25	0
JAKOBS2	25	25	0
TROUSERS	17	64	0, 180
SHAPES0	4	43	0
SHAPES1	4	43	0, 180
MARQUES	8	24	90 incremental
MAO	9	20	90 incremental
ALBANO	8	24	0, 180
SWIM	10	48	0, 180

TABLE 2: Computational Results.

Data set	Length	Utilization (%)	Time (seconds)
DIGHE1	100.00	100.00	4113
DIGHE2	100.00	100.00	3751
JAKOBS1	12.22	84.46	13498
JAKOBS2	26.11	79.61	11985
TROUSERS	245.45	88.90	14775
SHAPES0	63.275	65.17	4303
SHAPES1	61.301	69.93	5032
MARQUES	81.89	87.80	6871
MAO	1840.37	82.69	6348
ALBANO	10247.28	88.40	10.230
SWIM	6099.00	72.63	15.204

We denote by Dif the relation, in percentage, between the solution found by AM and the best solution of the row:

$$\text{Dif} = \frac{\text{Solution found by AM (in percentage)}}{\text{Best solution of the row (in percentage)}} \times 100. \quad (1)$$

To sum up, regarding the utilization percentage of the rectangular object, it can be stated that AM has presented promising results. Taking as reference the scores achieved by BLF and SAHA, the proposed methodology presents better solutions in most of cases, as well as the average solutions. Regarding the 2DNest algorithm, we obtained better results in three problem instances, namely, ALBANO, DIGHE1, DIGHE2, and SWIM. Moreover, AM yielded better scores compared to the BS in two data sets, namely, SHAPES0 and ALBANO. Furthermore, even without applying the same rotation variants allowed by other studies (90 incremental), Table 3 shows that AM proved to be very competitive when compared with the reported approaches. We strongly believe that equivalent or better results can be found if we allow more rotations in a future extension of the work.

In Figures 10, 11, and 12, we show the convergence to the best solution found during the search of the genetic algorithm

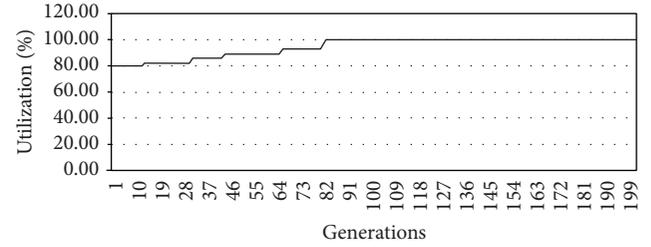


FIGURE 10: Dynamics of the genetic algorithm evolutionary process: a step-by-step improvement-DIGHE1.

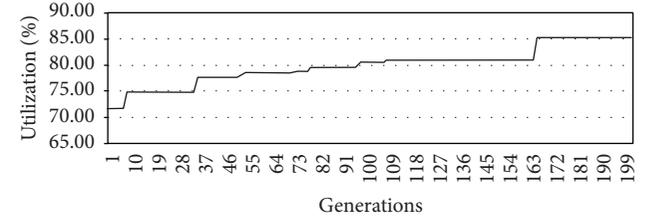


FIGURE 11: Dynamics of the genetic algorithm evolutionary process: a step-by-step improvement-ALBANO.

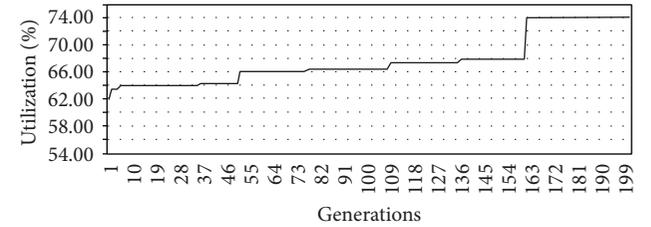


FIGURE 12: Dynamics of the genetic algorithm evolutionary process: a step-by-step improvement-JAKOBS2.

for DIGHE1, ALBANO, and JAKOBS2 data sets, while Figures 13, 14, and 15 display their best cutting configurations.

6.1. Applied Large Textile Industry. Brazil is the sixth largest textile producer country of the world and the main power, according to the Brazilian Textile Industry Association (ABIT). Among the Brazilian poles, we can highlight the state of Ceara. The success of cotton in this state, until the mid-1980s, stimulated the establishment of a solid textile and apparel park, which soon had to adapt to the new reality.

The aggregation method has also been applied in the ambit of a large textile industry. This particular industrial unit prints soccer team logos (Figure 16) on rectangular strips and then performs the cutting of these items for embroidering on caps, coats, shirts, shorts, and socks.

A significant difference in the production of shells is to define a feasible layout and has a prominent advantage. The acquisition of the points of the figures is performed by means of discretization, or some points are selected manually or automatically, in a manner that represent them.

The greater the number of points to describe the polygon is, the better the resolution is; however, all the processing routines consume high computational time. The width of a

TABLE 3: Comparative analysis.

Data set	SAHA	BLF	2DNest	BS	AM	Dif
DIGHE1	100.00	77.40	99.86	100.00	100.00	100
DIGHE2	100.00	79.40	99.95	100.00	100.00	100
JAKOBS1	78.89	82.60	89.07	85.96	84.46	95
JAKOBS2	77.28	74.80	80.41	80.40	79.61	99
TROUSERS	89.96	88.50	89.84	90.38	88.90	98
SHAPES0	66.50	60.50	67.09	64.35	65.17	97
SHAPES1	71.25	66.50	73.83	71.25	69.93	95
MARQUES	88.14	86.50	89.17	88.92	87.80	98
MAO	82.54	79.50	85.15	84.07	82.69	97
ALBANO	89.96	84.60	86.96	87.88	88.40	98
SWIM	74.36	71.60	71.53	75.04	72.63	96
Mean	83.53	77.45	84.81	84.39	83.60	98

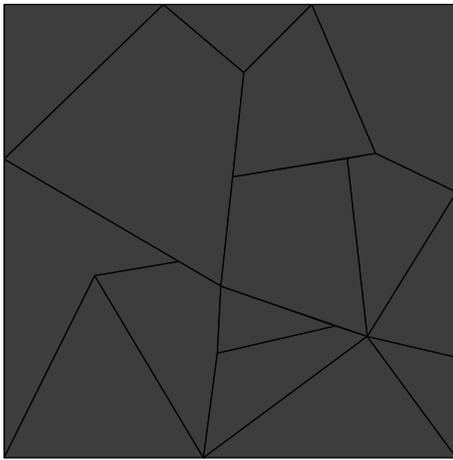


FIGURE 13: Best solution found for DIGHE1 instance.

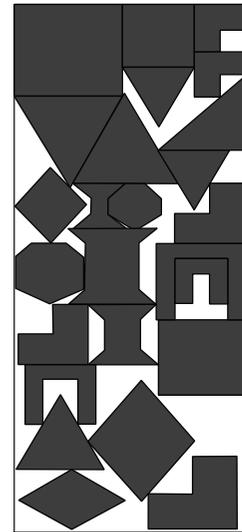


FIGURE 15: Best solution found for JAKOBS2 instance.

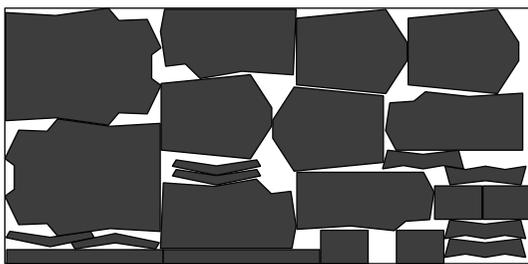


FIGURE 14: Best solution found for ALBANO instance.

rectangular strip is 210 mm and each logo is composed by a set of points (see input data for each instance at <https://www.dropbox.com/sh/qzwnx03lwp8klbs/9ULx0YkFey>).

Moreover, Table 4 contains the instance, the number of kinds, the number of polygons, and width to applications.

In Table 5, we present the results obtained by empirical methods (EM) in solving instances of the case study. We define “empirical” as the knowledge used by employees with extensive experience in textile cutting (without applying any combinatorial optimization technique).

The layout produced by applying the empirical methods (EM) to Soccer Logos-I, Soccer Logos-II, and Soccer Logos-III instances is displayed in Figures 17, 18, and 19.

In addition, the layout produced by applying the aggregation method (AM) to Soccer Logos-I, Soccer Logos-II, and Soccer Logos-III instances is displayed in Figures 20, 21, and 22. Moreover, Table 6 shows the values obtained from the length, improvement, and execution time presented. Comparing Tables 5 and 6, there is a better use of the aggregation method (AM) over the empirical methods (EM).

The highlighted variation in a small increase in the number of generations from the compression method is applied to the methodology. Moreover, it is observed that the application of aggregation method for the experiments of the textile industry presented has made significant progress.

7. Conclusions and Future Work

The main objective is to minimize the length of the layouts, while the width remains fixed. For this, we developed a

TABLE 4: Features of the figures of the case study.

Data set	Number of different pieces	Total number of pieces	Orientations (degrees)	Length
Soccer Logos-I	2	40	45 incremental	210 mm
Soccer Logos-II	1	20	45 incremental	210 mm
Soccer Logos-III	1	30	45 incremental	210 mm



FIGURE 16: Soccer team logos.

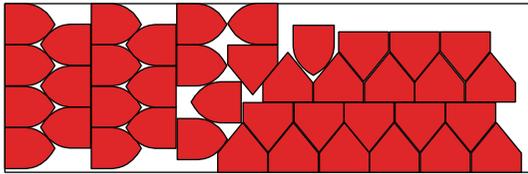


FIGURE 17: Best solution found for Soccer Logos-I instance.

TABLE 5: Results obtained from the practical case studied—empirical methods (EM).

Data set	Length	Improvement (%)	Time (seconds)
Soccer Logos-I	637.617	78.01	5100
Soccer Logos-II	307.474	79.21	1200
Soccer Logos-III	499.758	76.21	4800

strategy based on hybridization of genetic algorithms and a heuristic placement that applies the concepts of calculating the no-fit polygons methodology and bottom-left heuristic. In this initial study, we have introduced an aggregation methodology to cope with the irregular strip packing problem, which is based on a kind of hybridization between a genetic algorithm greedy bottom-left heuristic.

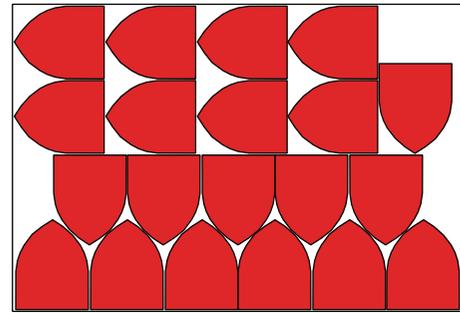


FIGURE 18: Best solution found for Soccer Logos-II instance.

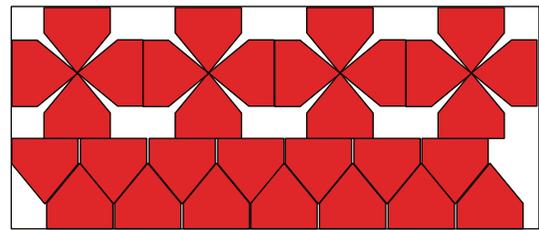


FIGURE 19: Best solution found for Soccer Logos-III instance.

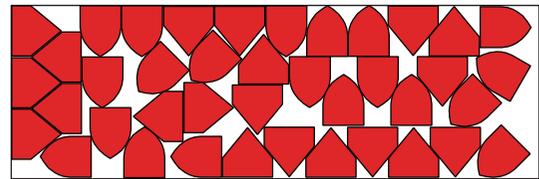


FIGURE 20: Best solution found for Soccer Logos-I instance: (AM).

TABLE 6: Results obtained from the practical case studied—aggregation method (AM).

Data set	Length	Improvement (%)	Time (seconds)
Soccer Logos-I	633.58	78.51	5220
Soccer Logos-II	300.95	80.94	3800
Soccer Logos-III	453.39	84.01	3690

For specific types of figures, certain approaches will produce best computational results. However, depending on the shape of polygons, the same may be a decrease in efficiency. The criteria for selection of various positioning method is an outstanding solution to the problem; however, the computational complexity involved in such methods is high. The computational results were good cheer, compared to other approaches, and especially according to the inherent difficulty of this problem.

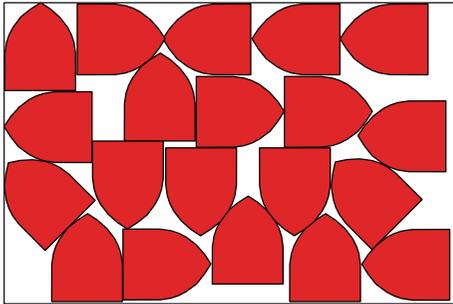


FIGURE 21: Best solution found for Soccer Logos-II instance: (AM).

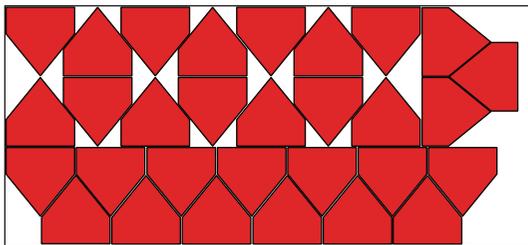


FIGURE 22: Best solution found for Soccer Logos-III instance: (AM).

Overall, the optimization performance achieved with the novel methodology has been promising, taking as reference the results achieved by other approaches and taking into account the inherent difficulties associated with this particular cutting and packing problem.

As future work, we plan to investigate the performance of the biased random-key genetic algorithm [23] with different criteria evaluations for the irregular strip packing problem, to apply processing in distributed genetic algorithm coupling observing criteria for each process, and to investigate other forms of representation of the polygon, so that irregular figures holes are characterized by applying the method of positioning. In addition, investigate other forms of representation of polygons, so that figures with irregular bores are characterized in the application of the method of placement. Another possibility is to investigate the application of Integer Linear Programming models for compacting layouts.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The first and third authors are thankful to Coordination for the Improvement of Higher Level or Education Personnel (CAPES) and the second and fourth authors are thankful to National Counsel of Technological and Scientific Development (CNPq) via Grants 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] 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.
- [3] P. R. Pinheiro and P. R. Oliveira, "A hybrid approach of bundle and Benders applied large mixed linear integer problem," *Journal of Applied Mathematics*, vol. 2013, Article ID 678783, 11 pages, 2013.
- [4] P. R. Pinheiro, A. L. V. Coelho, A. B. de Aguiar, and T. O. Bonates, "On the concept of density control and its application to a hybrid optimization framework: investigation into cutting problems," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 463–472, 2011.
- [5] B. K. Nielsen and A. Odgaard, "Fast neighborhood search for the nesting problem," Tech. Rep. 03/03, Department of Computer Science, University of Copenhagen, 2003.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] S. Jakobs, "On genetic algorithms for the packing of polygons," *European Journal of Operational Research*, vol. 88, no. 1, pp. 165–181, 1996.
- [10] E. Burke and G. Kendall, "Applying evolutionary algorithms and the no fit polygon to the nesting problem," in *Proceedings of the International Conference on Artificial Intelligence*, vol. 1, pp. 51–57, 1999.
- [11] J. F. Oliveira, A. M. Gomes, and J. S. Ferreira, "TOPOS: a new constructive algorithm for nesting problems," *OR Spektrum: Quantitative Approaches in Management*, vol. 22, no. 2, pp. 263–284, 2000.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Oxford, UK, 1975.
- [16] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

- [17] 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.
- [18] R. C. Art, "An approach to the two-dimensional irregular cutting stock problem," Tech. Rep. 36.008, IBM Cambridge Centre, 1966.
- [19] R. Dighe and M. J. Jakiela, "Solving pattern nesting problems with genetic algorithms employing task decomposition and contact detection," *Evolutionary Computation*, vol. 3, no. 3, pp. 239–266, 1996.
- [20] A. Albano and G. Sapuppo, "Optimal allocation of two-dimensional irregular shapes using heuristic search methods," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 10, no. 5, pp. 242–248, 1980.
- [21] C. Bounsaythip and S. Maouche, "Irregular shape nesting and placing with evolutionary approach," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 3425–3430, October 1997.
- [22] V. M. M. Marques, C. F. G. Bispo, and J. J. S. Sentieiro, "A system for the compactation of two-dimensional irregular shapes based on simulated annealing," in *Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation (IECON '91)*, pp. 1911–1916, Kobe, Japan, November 1991.
- [23] 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.



Hindawi

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

