A HYBRID APPROACH FOR THE NESTING PROBLEM: CASE STUDY ON A TEXTILE INDUSTRY

Bonfim Amaro Júnior
Universidade de Fortaleza
Av. Washington Soares, 1321-Edson Queiroz, Fortaleza - CE, 60811-905
baj007@gmail.com

Plácido Rogério Pinheiro
Universidade de Fortaleza
Av. Washington Soares, 1321-Edson Queiroz, Fortaleza - CE, 60811-905
placido@unifor.br

Rommel Dias Saraiva
Universidade de Fortaleza
Av. Washington Soares, 1321-Edson Queiroz, Fortaleza - CE, 60811-905
rommeldiassaraiva@gmail.com

Pedro Caliope Pinheiro
Universidade de Fortaleza
Av. Washington Soares, 1321-Edson Queiroz, Fortaleza - CE, 60811-905
pedrogcdpinheiro@hotmail.com

RESUMO

O objetivo deste trabalho é descrever uma metodologia híbrida que combina um Algoritmo Genético e uma heurística *Bottom-Left Greedy* para resolver o Problema de Posicionamento de Formas Irregulares bidimensional. A ferramenta *No-Fit Polygon* é aplicada para obtenção de ótimos locais. Além disso, um algoritmo de encolhimento é incorporado à meta-heurística para identificar soluções de melhor qualidade. Experimentos computacionais realizados em uma biblioteca de testes, bem como um estudo de caso em uma indústria têxtil, são realizados objetivando testar as potencialidades por trás da nova abordagem.

PALAVARAS CHAVE. Corte e Empacotamento, Problema de Posicionamento de Formas Irregulares, Métodos Híbridos.

ABSTRACT

The aim of this paper is to describe a hybrid methodology that combines a Genetic Algorithm and a Bottom-Left Greedy procedure for solving the two-dimensional Nesting Problem. The No-Fit Polygon construct is applied for obtaining local optima. Furthermore, a shrinking algorithm is incorporated to the meta-heuristic engine to identify good quality solutions. Computational experiments performed on standard benchmark problems, as well as a practical case study developed in a textile industry, are also reported and discussed here in a manner as to testify the potentialities behind the novel approach.

KEYWORDS. Cutting and Packing. Nesting Problem. Hybrid Methods.

1. Introduction

Roughly speaking, the Nesting Problem is a classic cutting and packing problem which consists of finding the most efficient layout for cutting irregular shapes out of a given strip with minimum waste material. More specifically, given the dimensions of the rectangular strip - a constant width and an infinite length - and the pieces which need to be cut, the problem can be defined as to find such an arrangement of pieces (with no overlapping) that minimizes the length of the rectangular strip. According to the typology of Wäscher et al. (2007), this is a typical case of Open Dimension Problem and it is of particular interest to industries with mass-production (e.g. garment, glass, steel, textile, wood, etc.), where the reduction of production cost is one of the major issues in an ever more competitive world. Other Nesting Problem variations exist, but in the following the focus is on the strip-packing variant.

Although easily understood, revealing a simplicity that contrasts directly with the real complexity, the Nesting Problem has received some attention in the domains of Computer Science and Operations Research due to NP-Hardness, as noted by Nielsen and Odgaard (2003). Therefore, there do not exist deterministic polynomial-time algorithms to solve this problem. Several approaches have been proposed for the resolution of the Nesting Problem. 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 a kind of hybridization between a modified Genetic Algorithm and a Bottom-Left Greedy procedure for tackling the Nesting Problem. A shrinking algorithm is also incorporated to the meta-heuristic engine in order to improve the arrangement of pieces.

To have better analysis, the remainder of this paper is organized as follows. In Sect. 2, we present some state-of-the-art methodologies dedicated to the two-dimensional Nesting Problem; Sect. 3 conveys fundamental concepts related to the proposed methodology, which is introduced in Sect. 4; discussions on computational results on benchmark problem instances and a case study on a textile industry are given in Sect. 5; Finally, in Sect. 6, 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 pieces comprise the most difficult class of packing problems. According to Dowsland and Dowsland (1995), whatever the constraints or secondary objectives, there are basically three approaches to find suitable layouts. The pieces may be considered one at a time and packed onto the stock sheet according to a sorting criteria, or may be nested either singly or in groups into a set of enclosing polygons which are then packed onto the stock sheet, or an initial allocation is improved iteratively.

A Genetic Algorithm for placing polygons on a rectangular board is proposed by Jakobs (1996) and also implemented by Amaro Jr et al. (2013, 2013). The ordered list of pieces forms the chromosome of each individual, which is decoded by the Bottom-Left placement rule. A shrinking algorithm improves partial solutions by shifting polygons closer to each other. Oliveira et al. (2000) implemented a constructive algorithm called "Técnicas de Optimização para o Posicionamento de Figuras Irregulares" (TOPOS). The solution grows from a floating origin and both the next piece to be packed and its position are defined by two heuristics named, respectively, *local search* and *initial sort*. Different objective functions are proposed to evaluate and compare partial solutions. Burke et al. (2006) introduced a new method for implementing the Bottom-Left-Fill packing algorithm which allows shapes that incorporate circular arcs and holes to be nested. Hill Climbing and Tabu Search methods find efficient nesting sequences for packing shapes.

A few authors have adopted mathematical programming techniques for solving one of the following sub-problems: Overlap Minimization Problem, whose objective is to place all polygons into a container with given width and length so that the total amount of overlap between polygons is made as small as possible; Compaction Problem, which requires a feasible layout 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 which eliminates all overlaps and has the minimum total translation. Gomes and Oliveira (2006), for instance, hybridized Simulated Annealing and Linear Programming. Firstly, an initial layout is obtained by the Bottom-Left Greedy placement heuristic, being each piece 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 (see Fig. 1) and a separation algorithm. An extended local search algorithm based on Nonlinear Programming is conceived by Leung et al. (2012). The algorithm starts with a feasible layout and its length is saved as the best length. Then, a new layout is achieved by randomly swapping two polygons in the current solution. Within a time limit, the strip length is reduced and a local search method solves Overlap Minimization Problems. If the new placement 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 a local search is invoked, which is guided by Tabu Search techniques in order to escape from local minima. A compaction algorithm is used to improve the results.

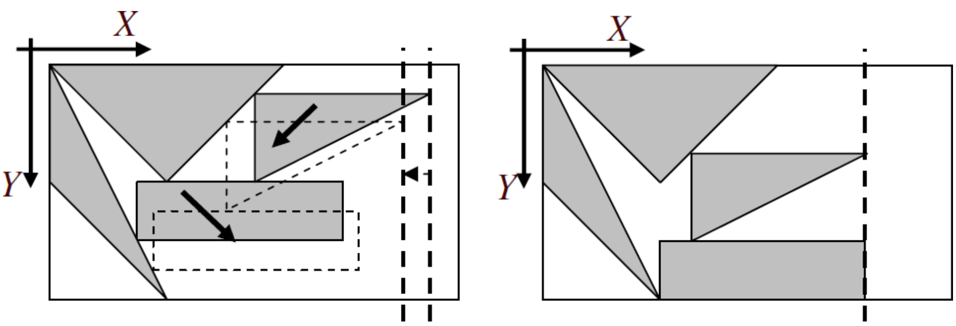

Figure 1. Compaction example

By other means, a successful approach that combines a local search method with a Guided Local Search to deal with two- and three-dimensional Nesting Problems is proposed Egeblad et al. (2007). An initial strip length is found by a fast placement heuristic (e.g. Bottom-Left). 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 meta-heuristic is adopted to escape from local minima. Finally, in a recent paper, Bennell and Song (2010) modified the TOPOS placement heuristic presented by Oliveira et al. (2000) and applied it to a Beam Search tree, which represents the placement order of the pieces onto the stock sheet. Each node in the *search* tree corresponds to a partial solution, which means that partial solutions can be generated in parallel.

Some additional important related works were produced by Albano and Sapuppo (1980) and Imamichi et al. (2009).

3. Related Concepts

In order to explain our proposed methodology, we need to describe essential concepts related to its behavior, which are Genetic Algorithm, Bottom-Left Greedy heuristic and No-Fit Polygon.

3.1 Genetic Algorithm

Introduced by Holland (1975) and perfected by De Jong (1975) and Goldberg (1989), Genetic Algorithm is a search heuristic that mimics the process of natural evolution.

This meta-heuristic is routinely used to generate useful solutions to optimization and search problems and belongs to the larger class of evolutionary algorithms, which generate solutions to optimization problems using techniques inspired by natural evolution.

An implementation of Genetic Algorithm begins with a population of (typically random) n chromosomes, which are evaluated and associated with a particular reproduction rate in such a way that those chromosomes which represent a better solution to the target problem are given more chances to "reproduce" than those chromosomes which are poorer solutions. The quality of a solution is defined with respect to the current population. Then the crossover and mutation mechanisms are applied with some probability T_{CROSS} and T_{MUT}, previously defined, to the chosen chromosomes. Only the fittest individuals take part in the next iteration. Once a new population is generated, the stopping condition of the meta-heuristic is checked. If the Genetic Algorithm is not terminated, it starts working with a new set of chromosomes.

Due to the flexibility and intelligent search, which explain why Genetic Algorithm can efficiently cope with hard optimization problems, a significant amount of research has investigated its performance in the field of irregular cutting, where we can mention the studies provided Fujita et al. (1993), who hybridized it with a local minimization algorithm; and Burke and Kendall (1999), who chose the concept of convex hull to place polygons. In our methodology, a modified Genetic Algorithm dictates the order in which polygons are cut by the placement policy described below.

3.2 Bottom-Left Greedy Heuristic

A popular constructive algorithm to any two-dimensional cutting or packing problem aims to order the pieces and allocate them at feasible positions to a rectangular object, more precisely into its lowest possible location and then closest to its left without overlapping with any placed item, as illustrated by Fig. 2. This process, known as Bottom-Left Greedy heuristic, was introduced by Baker et al. (1980) for packing an arbitrary collection of rectangular pieces into a rectangular bin so as to minimize the height achieved by any piece. 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.

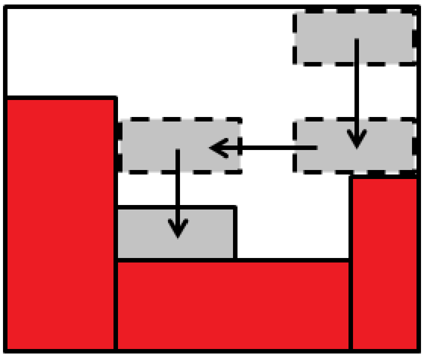

Figure 2. Bottom-Left Greedy procedure for an input piece

In the case of two-dimensional cutting, the papers yielded by Hopper and Turton (1999), Burke et al. (2006) and Gomes and Oliveira (2006), for instance, considered placement algorithms based on the Bottom-Left Greedy rule and here the aforementioned heuristic was also chosen as placement policy.

3.3 No-Fit Polygon

As pointed by Burke et al. (2007), 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 (1966) as "shape envelop" come 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 NFP_{AB}, is the set of points traced by R_B when B slides around the contour of A without overlapping, as displayed in Fig. 3. 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. Unlike what occurs in some applications reported in the literature, where authors usually adopt the concept of No-Fit Polygon for detecting whether polygon B overlaps polygon A, in this work this geometric tool is combined with the Bottom-Left Greedy heuristic in order to achieve local optima.

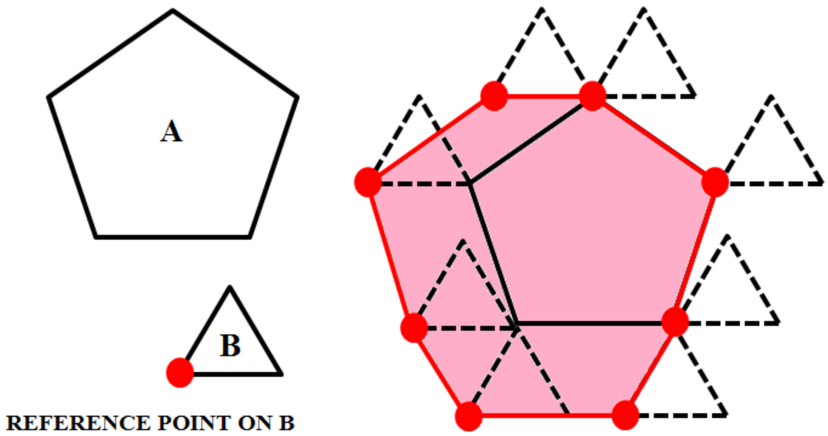

Figure 3. No-Fit Polygon generated by polygons A and B

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, i.e. the set: $A \oplus B = \{a + b : a \in A, b \in B\}$. 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.

Some useful algorithms for building the No-Fit Polygon have been recently produced Bennell et al. (2001), Burke et al. (2007) and Zhang et al. (2009).

4. Proposed Methodology

Our hybrid methodology prescribes the integration of the distinct components described in Sect. 3. On what concerns the Genetic Algorithm, whose control parameters and calibration are set out in Sect. 5, each individual is encoded by an integer chromosome that is represented as a placement vector $chrom(i)$, (i = 1, ..., m), which determines the cutting sequence of the m polygons. For each gene $chrom(i)$, characterized by a polygon of certain type t_i, there is an associated rotation variant r_i. The length required to cut all input pieces is assigned as fitness value of the corresponding individual. Revealed these details, the steps followed by all chromosomes of the current population of the Genetic Algorithm are presented below.

Step 1: As seen in Fig. 4, polygon $chrom(i)$ is placed within the rectangular object by the Bottom-Left Greedy rule. Set $i \leftarrow i + 1$;

Figure 4. Allocation of the first piece

Step 2: According to the No-Fit Polygon construct, polygon *chrom*(i) is placed on a vertex of polygon *chrom*(i-1) such that the length of the rectangular object does not increase or is minimized, as presented in Fig. 5. In the case of multiple positions, it is considered the leftmost one;

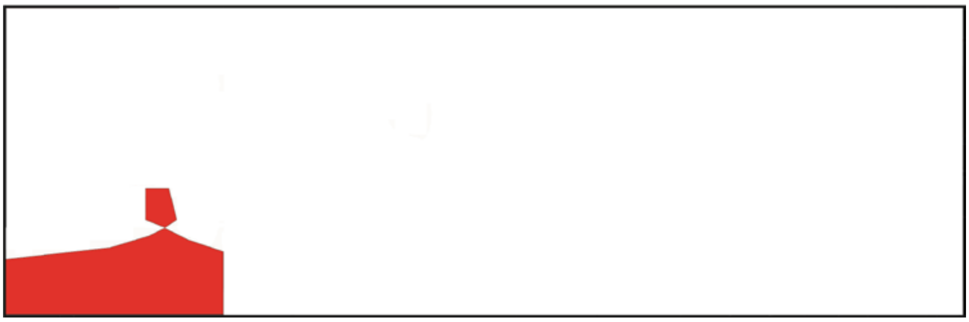

Figure 5. Best position computed by No-Fit Polygon

Step 3: The Bottom-Left Greedy rule is applied to polygon *chrom*(i), as illustrated by Fig. 6;

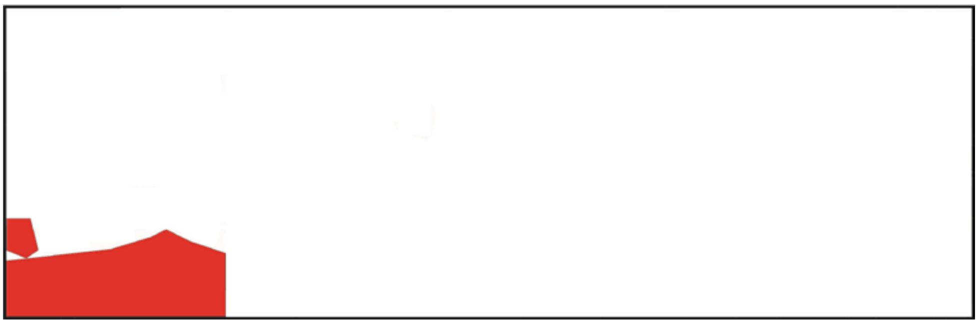

Figure 6. Final location of pieces

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 is configured as Genetic Algorithm operator, working in each generation on the two fittest individuals (i.e. the layouts with the shorter lengths). Considering the No-Fit Polygon technique, this task is characterized by collecting a piece located in the rightmost position of the layout and attempting to place it on a vertex of a polygon situated in the leftmost position in such a way that the length of the rectangular strip is decreased. Then, in a second moment the Bottom-Left Greedy routine is applied to the translated shape. The technique is clarified in Fig. 7.

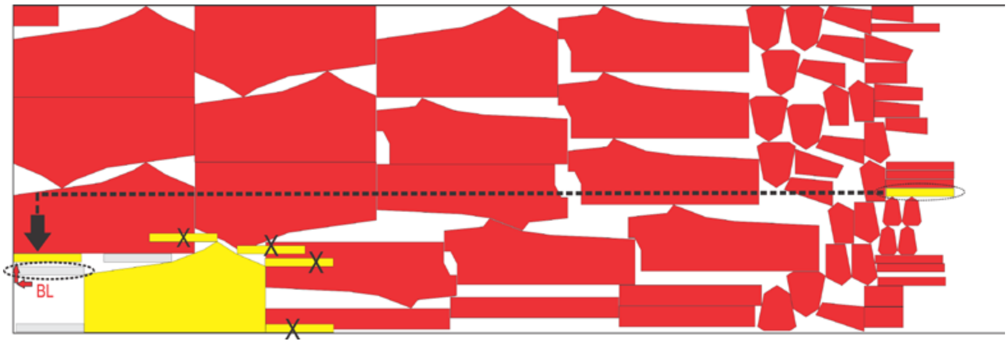

Figure 7. Shrinking heuristic

5. Computational Experiments

A series of experiments were carried out on a desktop machine with a 3.60 GHz Intel i5 CPU and 4GB of RAM. The Genetic Algorithm was implemented in Java and configured with Roulette Wheel Selection; Uniform Order-Based Crossover, whose application has recently achieved promising results in a particular cutting and packing problem in Saraiva and Pinheiro (2012); and Swap Mutation. After preliminary experiments, we set the parameters discussed in Sect. 3.1 as follows: n = 400 individuals, T_{CROSS} = 0.9 and T_{MUT} = 0.2. Concerning the maximum number of generations and the limit on the execution time values, 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) (http://www.apdio.pt/esicup), which involved five data sets: DIGHE1 and DIGHE2 are jigsaw puzzles with known optimum taken from Dighe and Jakiela (1996); JAKOBS1 and JAKOBS2 are artificial data sets taken from Jakobs (1996); and TROUSERS is an approximation of a real instance taken from the garment industry and it was firstly presented Oliveira et al. (2000). Some additional descriptions of the instances are shown in Table 1, whereas Table 2 provides the best results achieved by 10 runs of the hybrid methodology. 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.

Table 1. Data sets 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

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	80.22	13498
JAKOBS2	26.11	73.92	11985
TROUSERS	245.45	88.74	14775

In Table 3, we present a comparison of the best results achieved by some state-of-the-art methodologies to solve the Nesting Problem, where are included SAHA by Gomes and Oliveira (2006), BLF by Burke et al. (2006), 2DNest by Egeblad et al. (2007) and BS by Bennell and Song (2010). Our proposed methodology is referred to as HM.

Table 3. Comparative analysis

Data set	SAHA	BLF	2DNest	BS	HM
DIGHE1	100.00	77.40	99.86	100.00	100.00
DIGHE2	100.00	79.40	99.95	100.00	100.00
JAKOBS1	78.89	82.60	89.07	85.96	80.22
JAKOBS2	77.28	74.80	80.41	80.40	73.92
TROUSERS	89.96	88.50	89.84	90.38	88.74

To sum up, with regard to the utilization percentage of the rectangular object, it can be stated that the HM has presented satisfactory results. Taking as reference the scores achieved by BLF, the proposed methodology presents better solutions in most of cases. Furthermore, even without applying the same rotation variants allowed by other researches (90 incremental), the results show that HM proved to be very competitive when compared with the other approaches. We strongly believe that equivalent or better results can be found if we allow more rotations in a future extension of the work.

Fig. 8 shows the convergence to the best solution found during the search of the Genetic Algorithm for TROUSERS problem instance, and Fig. 9 displays its best cutting configuration.

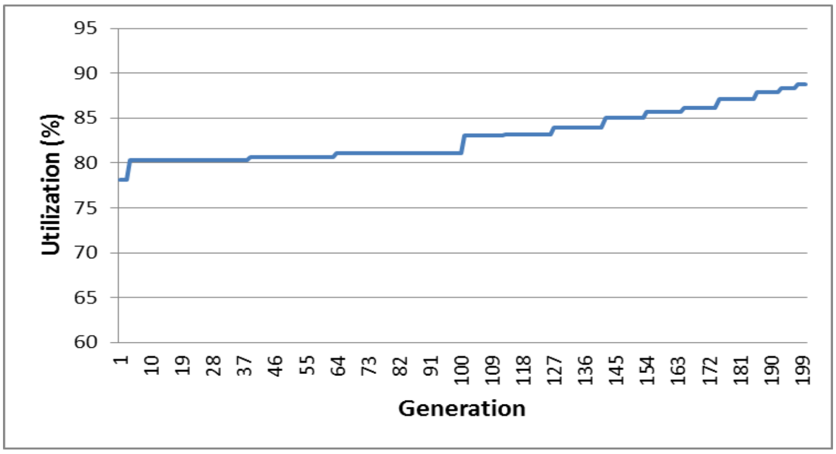

Figure 8. Dynamics of the genetic algorithm evolutionary process: a step-by-step improvement

Figure 9. Best solution found for TROUSERS instance

5.1 Case Study

The hybrid methodology has also been applied in the ambit of a large textile industry. This particular industrial unit prints soccer team logos on rectangular strips and then performs the cutting of these items for embroidering on caps, coats, shirts, shorts and socks. The width of the rectangular strip is 210 and points of the soccer logos are acquired as shown in picture below. All data sets, whose pieces configuration are depicted in Fig. 10, can be obtained in this link https://www.dropbox.com/sh/qzwnx03lwp8klbs/9ULx0YkFey

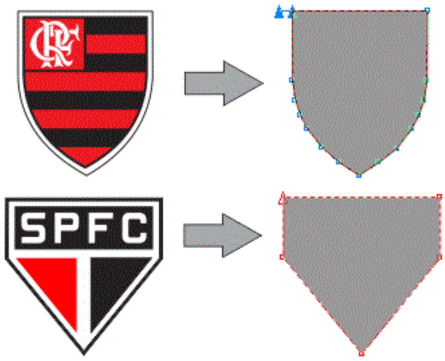

Figure 10. Obtainment of polygon points.

Results for the case study are displayed in Table 4. Moreover, in Fig. 11-13, it is illustrated the layout produced by HM related to the application.

Table 4. Results obtained for the practical case studied

Data set	Length	Utilization (%)	Time (seconds)
Soccer Logos – Teste1	654.39	76.02	7920
Soccer Logo – CRF	313.37	77.72	10200
Soccer Logo –SPFC	496.28	76.74	6540

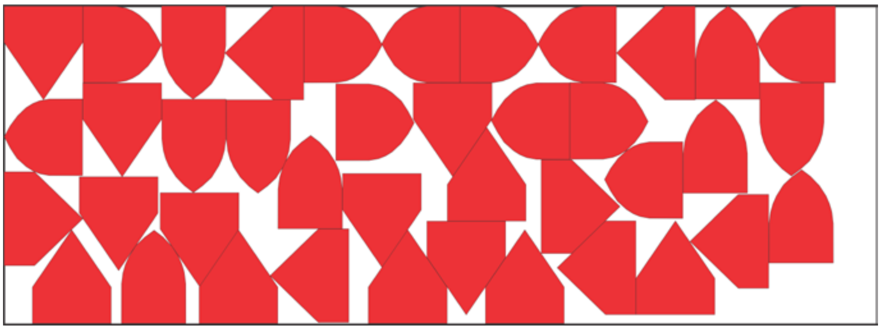

Figure 11. Cutting pattern produced by HM for Soccer Logos instance "Teste1".

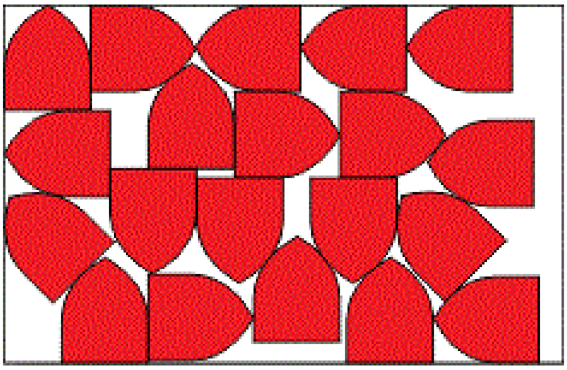

Figure 12. Cutting pattern produced by HM for Soccer Logos instance "CRF".

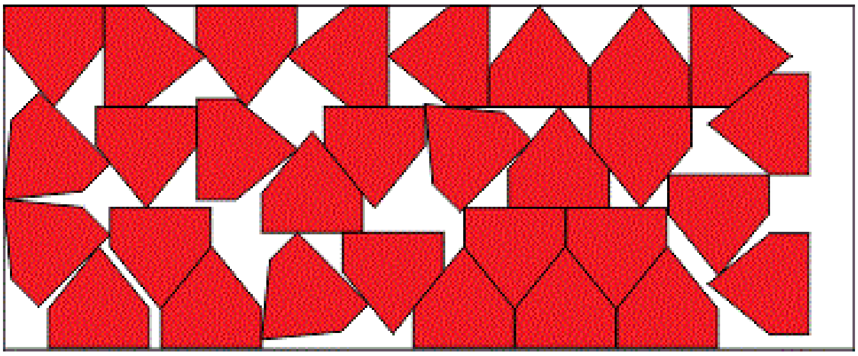

Figure 13. Cutting pattern produced by HM for Soccer Logos instance "SPFC".

6. Conclusion and Future Work

In this initial study, we have introduced a novel hybrid methodology for coping with the Nesting Problem, which is based on a particular type of hybridization between a meta-heuristic search method and a well-known placement policy (i.e. Genetic Algorithm and Bottom-Left Greedy). Overall, the optimization performance achieved with the new 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 application of more rotation variants to the polygons of the analyzed data sets and examine the performance of the proposed methodology in more instances available on ESICUP website. Another possibility is to investigate the application of linear programming models for compacting layouts, which have been used by Li and Milenkovic (1995), Stoyan et al. (1996) and Bennell and Dowsland (2001).

Acknowledgements

The first author is thankful to Coordination for the Improvement of Higher - Level or Education - Personnel (CAPES) and the second and third authors are thankful to National Counsel of Technological and Scientific Development (CNPq).

References

Albano, A. and Sapuppo, G. (1980), Optimal allocation of two-dimensional irregular shapes using heuristic search methods. *IEEE Transactions on Systems, Man and Cybernetics*, 10(5), 242-248.

Amaro Jr, B., Pinheiro, P.R. and Saraiva, R.D. (2013), A hybrid methodology for tackling the irregular strip packing problem. *In Proc. 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS'13)*, To appear.

Amaro Jr, B., Pinheiro, P.R. and Saraiva, R.D. (2013), Tackling the Irregular Strip Packing Problem by Hybridizing Genetic Algorithm and Bottom-Left Heuristic. *In Proc. 2013 IEEE Congress on Evolutionary Computation (CEC 2013)*, To appear.

Art, R.C. An approach to the two-dimensional irregular cutting stock problem. *Technical Report* 36.008, IBM Cambridge Centre, 1966.

Baker, B.S., Coman, E.G. and Rivest, R.L. (1980), Orthogonal packing in two dimensions. *SIAM Journal on Computing*, 9(4), 846-855.

Bennell, J.A. and Dowsland, K.A. (2001), Hybridising tabu search with optimisation techniques for irregular stock cutting. *Management Science*, 47(8), 1160-1172.

Bennell, J.A., Dowsland, K.A. and Dowsland, W.B. (2001), The irregular cutting-stock problem - a new procedure for deriving the no-fit polygon. *Computers & Operations Research*, 28(3), 271-287.

Bennell, J.A. and Song, X. (2010), A beam search implementation for the irregular shape packing problem. *Journal of Heuristics*, 16(2), 167-188.

Burke, E. and Kendall, G. (1999), Applying evolutionary algorithms and the no fit polygon to the nesting problem. *In Proc. 1999 International Conference on Artificial Intelligence*, 1, 51-57.

Burke, E., Hellier, R., Kendall, G. and Whitwell, G. (2006), A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. *Operations Research*, 54(3), 587-601.

Burke, E., Hellier, R.S.R., Kendall, G. and Whitwell, G. (2007), Complete and robust no-fit polygon generation for the irregular stock cutting problem. *European Journal of Operational Research*, 179(1), 27-49.

De Jong, K.A. An analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, University of Michigan, Ann Arbor, USA, 1975.

Dighe, R. and Jakiela, M.J. (1996), Solving pattern nesting problems with genetic algorithms employing task decomposition and contact detection. *Evolutionary Computation*, 3(3), 239-266.

Dowsland, K.A. and Dowsland, W.B. (1995), Solution approaches to irregular nesting problems. *European Journal of Operational Research*, 84(3), 506-521.

Egeblad, J., Nielsen, B.K. and Odgaard, A. (2007), Fast neighborhood search for two- and three-dimensional nesting problems. *European Journal of Operational Research*, 183(3), 1249-1266.

Fujita, K., Akagji, S. and Kirokawa, N. (1993), Hybrid approach for optimal nesting using a genetic algorithm and a local minimisation algorithm. *In Proc. 19th Annual ASME Design Automation Conference*, 1, 477-484.

Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA, 1989.

Gomes, A.M. and Oliveira, J.F. (2006), Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research*, 171(3), 811-829.

Holland, J.H. Adaptation in natural and artificial systems: an introductory analysis with Applications to biology, control, and artificial intelligence. University of Michigan Press, Oxford, England, 1975.

Hopper, E. and Turton, B. (1999), A genetic algorithm for a 2d industrial packing problem. *Computers & Industrial Engineering*, 37(1-2), 375-378.

Imamichi, T., Yagiura, M. and Nagamochi, H. (2009), An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. *Discrete Optimization*, 6(4), 345-361.

Jakobs, S. (1996), On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88(1), 165-181.

Leung, S.C.H., Lin, Y. and Zhang, D. (2012), Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem. *Computers & Operations Research*, 39(3), 678-686.

Li, Z. and Milenkovic, V. (1995), Compaction and separation algorithms for non-convex polygons and their applications. *European Journal of Operational Research*, 84(3), 539-561.

Nielsen, B.K. and Odgaard, A. Fast neighborhood search for the nesting problem. *Technical Report* 03/03, Department of Computer Science, University of Copenhagen, 2003.

Oliveira, J.F., Gomes, A.M. and Ferreira, J.S. (2000), TOPOS - a new constructive algorithm for nesting problems. *OR Spektrum*, 22(2), 263-284.

Saraiva, R.D. and Pinheiro, P.R. (2012), A novel application of crossover operator to a hybrid optimization framework: Investigation into cutting problems. *In Proc. 2012 IEEE Congress on Evolutionary Computation*, 633-639.

Stoyan, Y.G., Novozhilova, M.V. and Kartashov, A.V. (1996), Mathematical model and method of searching for a local extremum for the non-convex oriented polygons allocation problem. *European Journal of Operational Research*, 92(1), 193-210.

Wäscher, G., Hauβner, H. and Schumann, H. (2007), An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3), 1109-1130.

Zhang, D.F., Chen, J.C., Liu, Y.K. and Chen, H.W. (2009), Discrete no-fit polygon, a simple structure for the 2-d irregular packing problem. *Journal of Software*, 20(6), 1511-1520.