



**A HYBRID APPROACH USING A RANDOM-KEY GENETIC ALGORITHM AND  
POSITIONING METHODS TO TACKLE THE IRREGULAR STRIP PACKING PROBLEMS**

**Bonfim Amaro Júnior**

University of Fortaleza  
Graduate Program in Applied Informatics  
Av. Washington Soares, 1321 - J30, 60.811-905  
Fortaleza - Brazil  
[baj.unifor@outlook.com.br](mailto:baj.unifor@outlook.com.br)

**Plácido Rogério Pinheiro**

University of Fortaleza  
Graduate Program in Applied Informatics  
Av. Washington Soares, 1321 - J30, 60.811-905  
Fortaleza - Brazil  
[placido@unifor.br](mailto:placido@unifor.br)

**Rommel Dias Saraiva**

University of Fortaleza  
Graduate Program in Applied Informatics  
Av. Washington Soares, 1321 - J30, 60.811-905  
Fortaleza – Brazil  
[rommeldiassaraiva@gmail.com](mailto:rommeldiassaraiva@gmail.com)

**Pedro Gabriel Pinheiro**

University of Fortaleza  
Graduate Program in Applied Informatics  
Av. Washington Soares, 1321 - J30, 60.811-905  
Fortaleza – Brazil  
[pedrogcdpinheiro@hotmail.com](mailto:pedrogcdpinheiro@hotmail.com)



## RESUMO

Este artigo apresenta um algoritmo genético de chaves aleatórias aplicado aos problemas de posicionamento de figuras irregulares, um caso particular dos problemas de corte e empacotamento, cujo uma coleção de itens (ou polígonos) deve ser encaixada em uma chapa retangular, objetivando a redução do comprimento adequado. Em geral, nossa abordagem prescreve a integração entre a metaheurística e conhecidas regras de posicionamento (ex: bottom-left). Além disso, um método de encolhimento que opera com um algoritmo de chaves aleatórias é proposto, com o intuito de melhorar soluções parciais. Para avaliar as potencialidades da metodologia proposta, experimentos computacionais realizados sobre um conjunto de casos de destacadas soluções são discutidos para fins de avaliação, mostrando que algoritmo desenvolvido neste trabalho é capaz de competir com estudos de sucesso apresentados anteriormente em alguns casos particulares.

**PALAVRAS CHAVE:** Corte e Empacotamento, Problema de Posicionamento de Formas Irregulares, Metaheurísticas.

## ABSTRACT

This paper presents a random-key genetic algorithm (RKGA) for the irregular strip packing problem, a particular case of cutting and packing problems in which a collection of items (or polygons) has to be packed onto a rectangular object with the aim at minimizing its length. In general, our approach prescribes the integration of the aforementioned metaheuristic and well-known placement rules (e.g. bottom-left). Furthermore, a shrinking algorithm-that operates within the RKGA-is also proposed to improve partial solutions. To assess the potentials of the proposed methodology, computational experiments performed on a set of difficult benchmark instances of the nesting problem are discussed here for evaluation purposes, showing that our algorithm is able to compete with previous success studies in some particular problem instances.

**Keywords:** Cutting and packing; Irregular packing; Metaheuristics.

## 1. Introduction

In the field of operations research, cutting and packing (C&P) problems are typical combinatorial optimization problems encountered in many industrial segments during the production processes. In general, these problems seek to find the best allocation of a number of small items into some larger ones in order to optimize a given objective function and satisfy prescribed constraints.

Our research focus is on a particular C&P problem commonly referred to as nesting problem (also called irregular strip packing), which impacts upon several manufacturing industries, e.g. furniture, garment, metalware, textile and wood. According to the typology of Wäscher et al. (2007), the nesting problem is classified as a two-dimensional irregular open dimension problem, a problem category in which the set of small items has to be accommodated completely by one or more large objects whose extensions-in at least one dimension-can be considered as a variable. More specifically, the problem at hand consists of packing a collection of irregular items (or polygons) onto a rectangular object with a constant width and an unlimited length. The packing process aims at minimizing the length of the rectangular object such that no overlap between items occurs and each packed item lies entirely onto the rectangular object. Furthermore, three variations appear depending on rotations of items: rotations of any angles are allowed; rotations of finite number of angles are allowed; and no rotation is allowed.

### 1.2 Literature overview and contribution

The nesting problem is a classic NP-hard (Nielsen and Odgaard, 2003) combinatorial optimization problem. Therefore, there is no known polynomial-time algorithm that is able to solve instances of practical interest. The simplest heuristic methods consist of sorting the items and packing them onto the rectangular object through a fast placement policy. Jakobs (1996), for instance, presented a genetic algorithm where each individual is represented by a list of items. The way they are disposed in the chromosome defines the order they are packed by the bottom-left heuristic. A shrinking algorithm improves partial solutions by shifting items closer to each other. Conversely, Burke et al. (2006) implemented a bottom-left-fill packing algorithm which allows shapes that incorporate circular arcs and holes to be nested. The authors combined the new technique with standard hill climbing and tabu search methods that work as sorting criteria of pieces. In another work, Bennell and Song (2010) implemented a beam search tree which governs the placement order of the items to be packed onto the rectangular object. The TOPOS constructive procedure (Oliveira et al. 2000) was revised and used to decode the sequence into a layout. In this approach, the nodes in the search tree correspond to partial solutions while the arcs represent the decision to add items to generate next partial solutions.

Another heuristic approach is to find a suitable layout and resort to mathematical programming based methods that relocates items by solving one of the following subproblems: (1) the overlap minimization problem, whose objective is to minimize the overlap penalty for all pairs of items; (2) the compaction problem, which requires a feasible layout and relocates many of its items simultaneously so as to minimize the length of the rectangular object; and (3) the separation problem, which takes an infeasible layout and performs a set of translations of the items with the aim at eliminating all overlaps occurrences. In this context, Gomes and Oliveira (2006) introduced an algorithm which focus on the hybridization of simulated annealing with linear programming models. An initial solution is obtained by applying the bottom-left heuristic. The meta-heuristic component is used to guide the search over the solution space while compaction and separation algorithms are solved to generate neighborhoods during the search process. Imamichi, M. Yagiura, and H. Nagamochi (2009) propose an algorithm that separates overlapping polygons based on nonlinear programming and an algorithm that swaps two polygons in a layout so as to find their

new positions in the layout with the least overlap. These algorithms are incorporated as components of an iterated local search algorithm for the overlap minimization problem.

In a recent paper, Leung et al. (2012) suggested an extended local search algorithm based on nonlinear programming for the nesting problem. As first step, a feasible solution is founded and the length of the rectangular object is saved as the best length. Then, a new placement is found by swapping items in the current layout. Within a time limit, the length of the rectangular object 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 length is increased and go on to do local search, which is guided by tabu search techniques in order to escape from local minima. A compaction algorithm is used to improve the results.

By other means, Egeblad et al. (2007) conceived a successful approach that combines a local search method with a guided local search to tackle two- and three-dimensional nesting problems. A fast placement heuristic (e.g. bottom-left) is used to achieve an initial solution. By reducing the length value, overlap situations may occur, which are removed by local search procedures (e.g. horizontal translation, vertical translation, rotation and flipping). The guided local search metaheuristic is adopted to escape from local minima.

In this work, we extend the methodology presented by Amaro Junior et al. (2013). We propose a random-key genetic algorithm (RKGA, for short) for packing two-dimensional items with irregular shapes. The approach focus on the hybridization of the aforementioned metaheuristic and well-known placement rules. Furthermore, a shrinking algorithm operates within the RKGA to improve partial solutions. Computational experiments conducted on a set of difficult benchmark instances show that the results achieved by our approach outperform some of those found by previous success studies reported in the literature.

### 1.3 Structure of the paper

To have better analysis, the remainder of the paper is organized as follows. Section 2 describes essential ingredients (i.e. no-fit polygon and random-key genetic algorithms) related to the proposed methodology, which is step-by-step presented in Section 3. Computational experiments on benchmark problem instances are given in Section 4. Finally, Section 5 closes the paper with some conclusions and future perspectives.

## 2. Related concepts

To describe the proposed methodology, we first need to define and explain some structures related to its behavior. These are no-fit polygon and random-key genetic algorithms.

### 2.1 No-fit polygon

The no-fit polygon is a powerful data structure used for fast and efficient handling of geometry in C&P problems involving irregular shapes. The idea behind this trigonometric technique comes as follows: Given two polygons,  $i$  (the fixed piece) and  $j$  (the orbital piece), and a reference point on  $B$  called  $r_j$ , the no-fit polygon of  $i$  in relation to  $j$ , denoted as  $NFP(i, j)$ , is the locus of points traced by  $r_j$  when  $j$  slides around the external

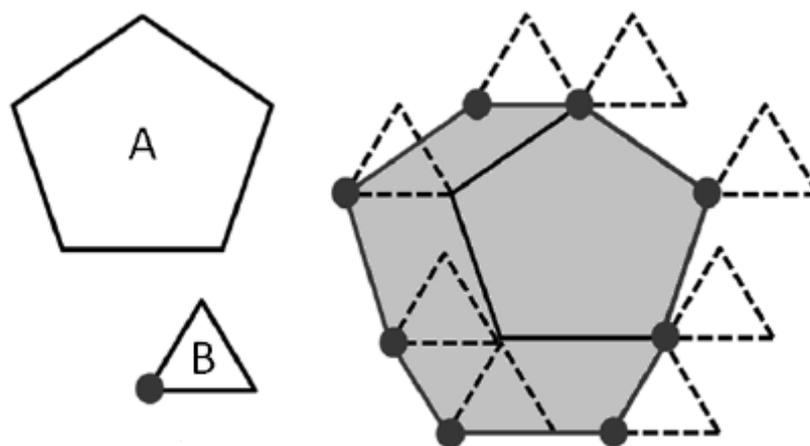


Figure 1. No-fit polygon generated by polygons *A* and *B*.

contour of *i*, as displayed by Figure 1. From this definition, three situations may arise: if  $r_j$  is placed in the interior of  $NFP(i, j)$ , then *j* intersects *i*; if  $r_j$  is placed on the boundary of  $NFP(i, j)$ , then *j* touches *i*; and if  $r_j$  is placed in the exterior of  $NFP(i, j)$ , then *j* does not intersect or touch *i*.

In our implementation, the no-fit polygons of all pairs of shapes are calculated in a pre-processing stage by using the Minkowski sum (Stoyan and Ponomarenko 1977), a form of vector addition. The concept is as follows: Given two arbitrary point sets, *A* and *B*, the

Minkowski sum of *A* and *B* is defined as  $A \oplus B = \{a + b: a \in A; b \in B\}$ . Simple vector algebra

can be used to show that  $A \oplus (-B)$ , known as Minkowski difference of *A* and *B*, is equivalent to  $NFP(A, B)$ . Since we follow the convention that polygons have counter-clockwise orientation,  $-B$  is simply *B* in clockwise orientation.

## 2.2 Random-key genetic algorithms

Genetic algorithms with random keys, or random-key genetic algorithms (RKGA), were first introduced by Bean (1994) to cope with hard combinatorial optimization problems involving sequencing. In a RKGA, each chromosome is represented as a vector of real numbers in the interval  $[0, 1)$ . A deterministic procedure, called a decoder, takes as input a chromosome of random keys and associates with it a solution of the combinatorial optimization problem for which an objective value or fitness can be computed. As any evolutionary technique, a RKGA evolves a population of chromosomes over a number of generations. The initial population is composed of *p* chromosomes, whose alleles are randomly generated in the real interval  $[0, 1)$ . After the fitness of each individual is computed by the decoder, the population is partitioned into two groups of individuals, namely, a small group of  $p_e$  elite individuals, i.e. those individuals with the best fitness values, and the remaining set of  $p - p_e$  non-elite individuals. To evolve the population, a new generation of individuals must be produced. A RKGA uses an elitist strategy, which consists of copying all elite individuals of the population of generation *k* to the population of generation *k* + 1. Mutation is implemented by introducing *pm* mutants into the population. A mutant is a vector of random keys generated in the same way that an element of the initial population. Discounting

$p_e$  elite individuals and  $p_m$  mutants,  $p - p_e - p_m$  additional individuals need to be produced to complete the  $p$  individuals that make up the population

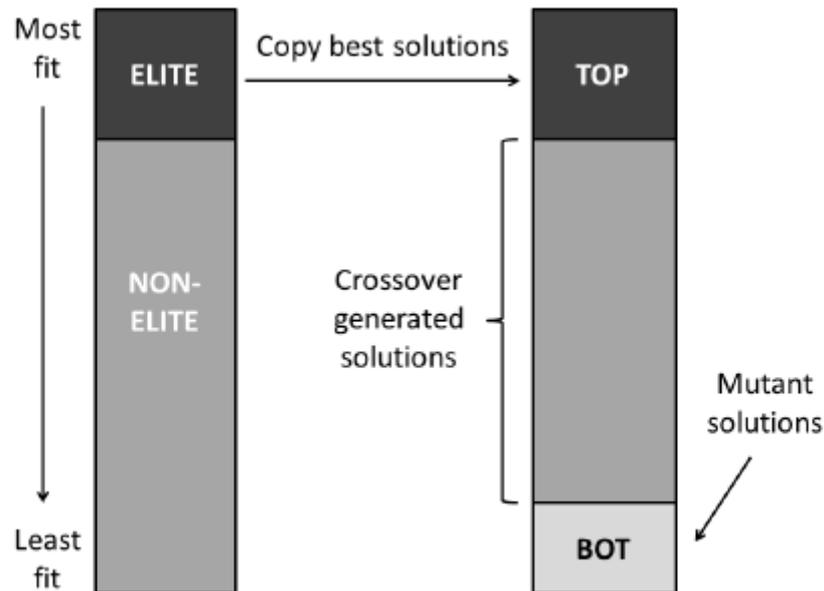


Figure 2. Transition state from generation  $k$  to generation  $k + 1$ .

of generation  $k + 1$ . This is performed by producing  $p - p_e - p_m$  offspring through the process of crossover, which is done by parametrized uniform crossover (Spears and De Jong 1991): Given two parents A and B randomly selected from the current population and a user-chosen probability  $p_A > 0.5$  that an offspring inherits the allele of A. For each allele  $i = 1; 2; 3; \dots; n$ , where  $n$  is the number of genes, the child takes the value of the  $i$ -th allele of A with probability  $p_A$  and the corresponding value of the allele of B with probability  $1 - p_A$ . Figure 2 depicts the evolution process from generation  $k$  to generation  $k + 1$ .

### 3. A RKGA heuristic for the nesting problem

This section describes a RKGA heuristic for the nesting problem. We show how solutions are encoded and how solutions are decoded from a vector of random keys.

#### 3.1 Solution encoding

In our RKGA heuristic, a solution is encoded as a vector  $X = (X_1; \dots; X_{3n})$ , where  $n$  is the total number of items to be packed and  $X_i$  is a number in the interval  $[0, 1)$ , for  $i = 1; \dots; 3n$ . Components  $X_1; \dots; X_n$  determine the item type packing sequence (ITPS); components  $X_{n+1}; \dots; X_{2n}$  design the vector of rotation variant (VRV) of each item; finally, components  $X_{2n+1}; \dots; X_{3n}$  correspond to the vector of placement procedure (VPP), which establish the placement rule that is adopted to pack the items. Three placement rules were chosen: bottom-left, greedy bottom-left and no-fit polygon based heuristic (Amaro Junior et al. 2013).

### 3.2 Solution decoding

A decoder takes as input the encoded solution vector  $X = (X_1; \dots; X_{3n})$  and outputs a solution of the nesting problem, i.e. the length of the rectangular object obtained after

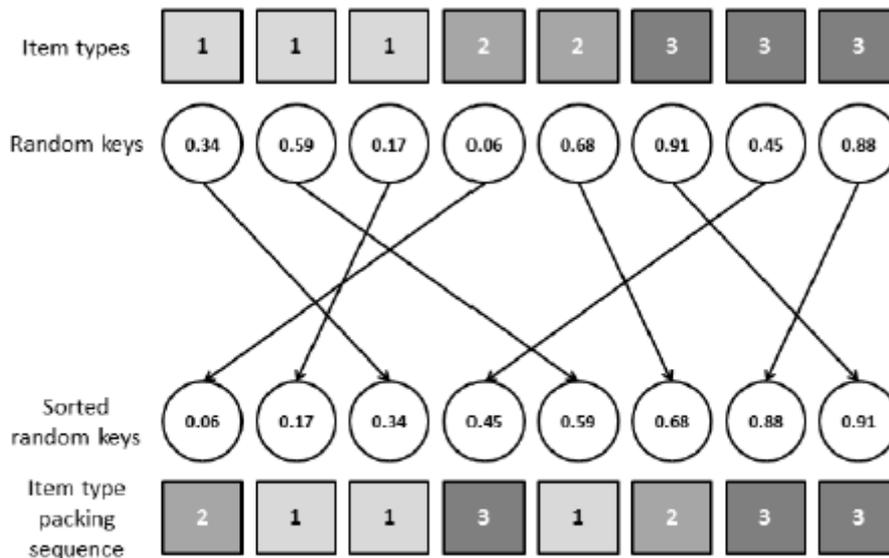


Figure 3. Chromosome decoding procedure.

packing all the input items. Since the vector of random keys has three components, the decoding process has three phases. The decoding of the first  $n$  genes of each chromosome into an ITPS is accomplished by sorting the genes and item types in ascending order, as shown in Figure 3. Regarding the VRV (the second part of the chromosome), the decoding is accomplished using the following expression:

$$VRV(i) \text{ (deg)} = \begin{cases} 0, & \text{if } gene(n+i) < \frac{1}{4} \\ 90, & \text{if } \frac{1}{4} \leq gene(n+i) < \frac{1}{2} \\ 180, & \text{if } \frac{1}{2} \leq gene(n+i) < \frac{3}{4} \\ 270, & \text{if } \frac{3}{4} \leq gene(n+i) < 1 \end{cases}$$

Finally, the decoding of the last  $n$  genes of each chromosome into a VPP is accomplished as follows:

$$VPP(i) = \begin{cases} \text{bottom-left,} & \text{if } gene(2n+i) < \frac{1}{3} \\ \text{greedy bottom-left,} & \text{if } \frac{1}{3} \leq gene(2n+i) < \frac{2}{3} \\ \text{no-fit polygon based heuristic,} & \text{if } \frac{2}{3} \leq gene(2n+i) < 1 \end{cases}$$

In order to improve potential solutions, at each generation of the RKGA a shrinking algorithm is applied to the TOP individuals. Considering the no-fit polygon technique, this task takes items packed on the rightmost position of the layout and packs them on the con-

tour of those items placed on the leftmost position of layout, thus trying to reduce the length of the rectangular object. See (Amaro Junior et al. 2013) for a detailed description of this procedure.

Table 1. Configuration used on the RKGA.

Population size	2000 individuals
Crossover probability	0.9
TOP	The 50% most fit chromosomes
BOT	20% of the next generation is made up of mutants
Fitness	Minimize the length of the rectangular object
Stopping criterion	Stop after 200 generations

Table 2. Problem instances characteristics.

Problem instance	Number of different items	Total number of items	Average vertices of items	Rectangular object width	Allowed orientations (deg)
ALBANO	8	24	7.25	4900	0, 180
DIGHE1	16	16	3.87	100	0
DIGHE2	16	16	4.70	100	0
JAKOBS1	25	25	5.60	40	90 incremental
JAKOBS2	25	25	5.36	70	90 incremental
MARQUES	8	24	7.37	104	90 incremental
SHAPES0	4	43	8.75	40	0
SHAPES1	4	43	8.75	40	0, 180
TROUSERS	17	64	5.06	79	0, 180

#### 4. Computational experiments

Computations were carried out on a desktop machine with a 3.60 GHz Intel i5 CPU and 4GB RAM. The proposed methodology was implemented in Java language and all tests were run using the configuration summarized in Table 1.

To evaluate the potentials behind the RKGA heuristic, we conducted a series of experiments on some benchmark problem instances available on the European Working Group in Cutting and Packing (ESICUP) website (<http://www.fe.up.pt/esicup>). They comprehend artificial data sets (JAKOBS1, JAKOBS2, SHAPES0 and SHAPES1), jigsaw puzzles (DIGHE1 and DIGHE2) and garment pieces (ALBANO, MARQUES and TROUSERS). The main characteristics of these instances are shown in Table 2.

The best results achieved by 10 runs of the RKGA heuristic are provided in Table 3. In this table, for each problem instance, 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 Figure 4, we show the best configuration found by the proposed methodology for TROUSERS problem instance, while in Table 4, 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 (Gomes and Oliveira 2006), BLF (Burke et al. 2006), 2DNest (Egeblad et al. 2007) and BS (Bennell and Song 2010). Our RKGA heuristic is referred to as RKGA-NP.

To sum up, regarding the utilization percentage of the rectangular object, one can state that RKGA-NP presented competitive scores. Taking as reference the results achieved

Table 3. Computational results.

Problem instance	Length	Utilization (%)	Time (sec)
ALBANO	10247.28	84.74	10,230
DIGHE1	100.00	100.00	2,053
DIGHE2	100.00	100.00	2,291
JAKOBS1	12.00	81.67	10,268
JAKOBS2	24.20	79.75	12,205
MARQUES	81.89	84.47	6,871
SHAPES0	62.60	63.74	1,826
SHAPES1	60.12	66.37	3,025
TROUSERS	241.82	90.07	16,025

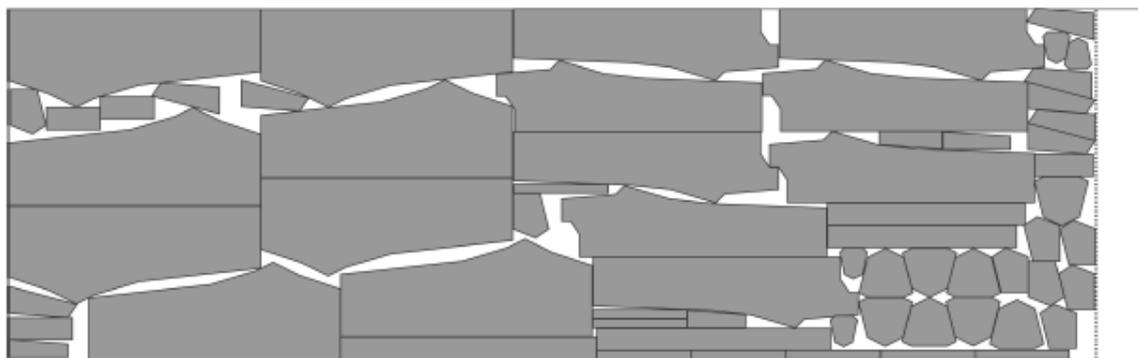


Figure 4. Best configuration found for TROUSERS problem instance.

Table 4. Comparative analysis.

Problem instance	SAHA	BLF	2DNest	BS	RKGA-NP	GCS
ALBANO	87.43	84.60	87.44	87.88	84.74	89.58
DIGHE1	100.00	77.40	99.86	100.00	100.00	100.00
DIGHE2	100.00	79.40	99.95	100.00	100.00	100.00
JAKOBS1	75.89	82.60	89.07	85.96	81.67	89.10
JAKOBS2	77.28	74.80	80.41	80.40	79.75	87.37
MARQUES	88.14	86.50	89.17	88.92	84.47	90.59
SHAPES0	66.50	60.50	67.09	65.41	63.74	68.79
SHAPES1	71.25	66.50	73.84	72.55	66.37	76.73
TROUSERS	89.96	88.50	89.84	90.38	90.07	91.00

by SAHA and BLF, for instance, the proposed methodology presents better solutions in most of cases. In addition, RKGA-NP outperforms 2DNest in three occasions (DIGHE1, DIGHE2 and TROUSERS), reaches the same results of the well-known BS approach in two problem instances and compared with GCS by Ahmed(2013) the same results was obtained just in (DIGHE1,DIGHE2).

Aiming at compare the progress of methodology published by (Amaro Junior et al. 2013) and the RKGA-NP, Table 5 presents the results for the same specifics studies cases. Figure 5 and Figure 6 demonstrate a final placement configuration for data sets “Soccer Logos - SPFC” and “Soccer Logos – Teste1”, respectively, which the RKGA-NP has obtained an improvement in both cases.

Table 5. Comparative analysis.

Problem instance	GA-NP	RKGA-NP
Soccer Logos – Teste1	76.02	78.08
Soccer Logos – CRF	77.72	77.72
Soccer Logos – SPFC	76.74	80.39

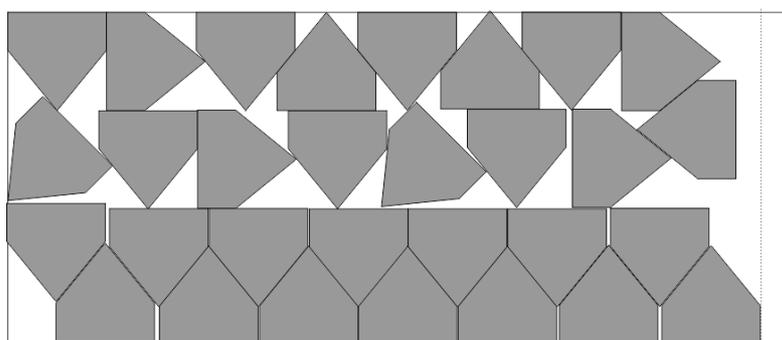


Figure 5. Final Solution for "Soccer Logos - SPFC "

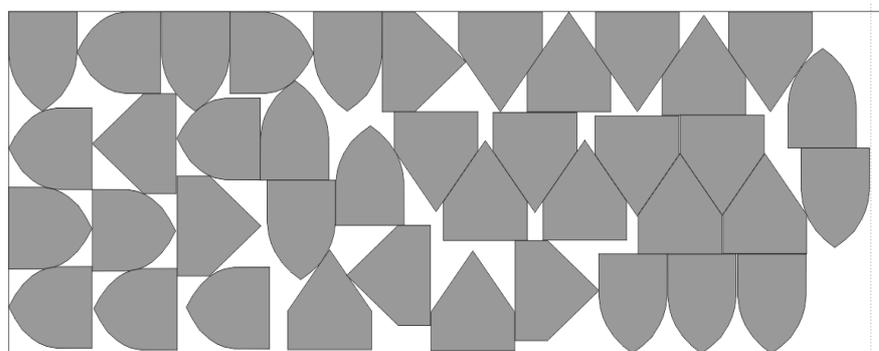


Figure 6. Final Solution for "Soccer Logos - Teste1 "

## 5. Conclusions and future work

The main objective of the nesting problem is to minimize the length of the rectangular object, while the width remains fixed. For this, we extended a recent study (Amaro Junior et al. 2013) and proposed a novel methodology for coping with the target problem, which is based on a particular type of hybridization between a metaheuristic search method and well-known placement rules. Overall, the optimization performance achieved with the new methodology was promising, taking as reference the results achieved by other approaches and taking into account the inherent difficulties associated with this particular C&P problem.

As future work, we plan to examine the performance of the RKGA-NP in more instances available on ESICUP website as well as investigate more robust placement rules en-

coded on the RKGA, e.g. TOPOS (Oliveira et al. 2000). Another possibility is to test the biased random-key genetic algorithm (Gonçalves and Resende 2011a) as metaheuristic component, whose application has already found good quality solutions for constrained two-dimensional orthogonal packing (Gonçalves and Resende 2011b) and container loading (Gonçalves and Resende 2012) problems.

### Acknowledgements

The first is thankful to National Counsel of Technological and Scientific Development (CNPq) via Grants #475239/2012-1, second and third authors are thankful to Coordination for the Improvement of Higher Level or Education Personnel (CAPES).

### References

- Ahmed Elkeran** (2013). A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering, *European Journal of Operational Research*, 231(3), 16, 757-769.
- Amaro Junior, B., Pinheiro, P., and Saraiva, R.** (2013). A hybrid methodology for tackling the irregular strip packing problem. In: *Intelligent Manufacturing Systems*, Vol. 11, 396 – 401.
- Bean, J.C.** (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6 (2), 154 – 160.
- 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., et al.** (2006). A new bottom-left\_ll heuristic algorithm for the two-dimensional irregular packing problem. *Operations Research*, 54 (3), 587 – 601.
- 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.
- Gomes, A.M. and Oliveira, J.F.** (2006). Solving irregular strip packing problems by hybridizing simulated annealing and linear programming. *European Journal of Operational Research*, 171 (3), 811 – 829.
- Gonçalves, J.F. and Resende, M.G.** (2011a). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17 (5), 487 – 525.
- Gonçalves, J.F. and Resende, M.G.** (2011b). A parallel multi-population genetic algorithm for a constrained two-dimensional orthogonal packing problem. *Journal of combinatorial optimization*, 22 (2), 180 – 201.
- Gonçalves, J.F. and Resende, M.G.** (2012). A parallel multi-population biased random-key genetic algorithm for a container loading problem. *Computers & Operations Research*, 39 (2), 179 – 190.
- Jakobs, S.** (1996). On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88 (1), 165 – 181.
- Leung, S.C., Lin, Y., and Zhang, D.** (2012). Extended local search algorithm based on non-linear programming for two-dimensional irregular strip packing problem. *Computers & Operations Research*, 39 (3), 678 – 686.
- Nielsen, B.K. and Odgaard, A.** (2003). Fast neighborhood search for the nesting problem. *Sci. Report*, University of Copenhagen.
- 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.



**Spears, W.M. and De Jong, K.A.** (1991). On the virtues of parameterized uniform crossover. In: Proceedings of the Fourth International Conference on Genetic Algorithms, 230 – 236.

**Stoyan, Y.G. and Ponomarenko, L.D.** (1977). Minkowski sum and hodograph of the dense placement vector function. Reports of the SSR Academy of Science, SER. A, 10.

**T. Imamichi, M. Yagiura, and H. Nagamochi** (2009). An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem, *Discrete Optimization*, vol. 6, no. 4, pp. 345–361.

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