Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Irregular packing overlap minimization using discrete Voronoi mountain

André Kubagawa Sato [*,1] Marcos de Sales Guerra Tsuzuki [*,2]
Thiago de Castro Martins [*,3]
Antonio Miguel da Fonseca Fernandes Gomes [**,4]

[*] *Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil. Mechatronics and Mechanical Systems Engineering Department Computational Geometry Laboratory*
[**] *Universidade do Porto, Porto, Portugal*

Abstract: Cutting and packing problems are of great importance in a myriad of industries such as: wood, textile, glass and shipbuilding. The irregular strip packing problem considers a container with infinite length where irregular items must be inserted into. A group of solutions in the literature solves the problem by allowing overlap between items and then applying a method which minimizes the total overlap value of the layout. The best results in the literature were obtained using overlap minimization techniques and it is the strategy adopted in this work. Fast overlap evaluation is obtained by the employment of the Voronoi mountain concept. It is used to pre-evaluate the overlap value for two items by adopting a raster representation, which is stored prior to the execution of the algorithm. Tests performed using a benchmark case showed that, even with reduced precision, competitive layouts can be obtained using the proposed approach.

Keywords: Irregular strip packing, overlap minimization, raster methods

1. INTRODUCTION

Cutting and packing (C&P) problems are found in a wide array of industries. A classic example of a cutting problem is encountered in textile manufacturing, where specific shapes are cut from rolls of fabric. The cutting patterns dictate the utility rate of the material. A more efficient layout can reduce waste, with both economical and environmental impact. Packing problems consider a container in which items must be inserted into without overlap. A better arrangement results in less unused space.

All C&P problems can be described as: assign a set of small items to a set of big items such that the layout is feasible, i.e., small items do not collide and are all completely contained by a big item. In this work, the terminology for packing problems is used, so small items are referred simply as items and the big items as containers. According to the topology proposed by Wäscher et al. (2007), which consider factors such as dimensionality, kind of assignment, number of containers and shape of items, the problem approached in this paper is the bi-dimensional irregular item open dimension problem, also referred as irregular strip packing problem. In this class of problems, a set of items, whose shape is not restrict to regular polygons, are inserted in a rectangular container with fixed width and variable length. The objective is to find a layout which minimizes the changeable dimension.

For strip packing algorithms, the geometric aspect can be viewed as a separated sub-problem. The analysis of the geometry of items and container is necessary to guarantee that the restrictions are satisfied. This is achieved by employing geometric tools which assure that items do not overlap nor protrude from the container. When dealing with irregular items, a layer of complexity is added on top of the optimization problem.

In this work, we propose a irregular strip packing solution based on raster Voronoi mountain overlap minimization with guided local search. Fast overlap determination is achieved using matrices created in a preprocessing step. Results comparisons with the best approaches in literature showed that the proposed method can obtain competitive solutions with the same time restrictions. Furthermore, the performance can be vastly improved, as it is possible to employ a massive parallel processing to generate a faster implementation.

This paper is structured as follows. Section 2 gives a brief review of the literature of irregular strip packing solutions. In section 3, previous attempts by the authors to solve the discussed problem are discussed. Next, in section 4, the adopted overlap evaluation is described. The proposed overlap minimization algorithm is detailed in section 5 and the results are shown in section 6. Finally, conclusions are drawn in section 7.

2. LITERATURE REVIEW

Strip packing problems are, as several other C&P variations, NP-Hard (Fowler et al., 1981). When complex geometry is also considered, the problem is not trivially

[1] e-mail: andre.kubagawa@usp.br.
[2] e-mail: mtsuzuki@usp.br.
[3] e-mail: thiago@usp.br.
[4] e-mail: agomes@fe.up.pt.

solvable and, as a consequence, there are many different solutions in the literature.

However, it is possible to distinguish two main strategies to represent the solution space: 1) as an ordered list of items; or 2) by a set of translation vectors to be applied to the items. In the former, the adopted placement heuristic guarantees that no overlap between items occurs when placing one item of the list at a time. For the latter, a strategy must be defined to both eliminate the collision and protrusion, and to compact the layout. The first strategy is defined as the search over sequence and the second as search over the layout (Elkeran, 2013).

Search over sequence approaches usually consists of two basic components: a constructive placement heuristic and a search algorithm. In Oliveira et al. (2000), the placement heuristic TOPOS was proposed. It was later used by Gomes and Oliveira (2002) with a 2-exchange heuristic. Sato et al. (2012) proposed a strategy in which a simulated annealing (SA) was used both to search over the sequence and to construct the layout by selecting the orientation of the item and its position, which was limited by the collision free region (CFR).

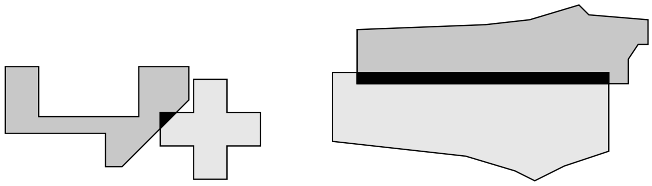
For solutions which perform the search over the layout, one common characteristic is that collision between items are allowed at some point. For this short review, two groups of strategies are analyzed: separation and compaction methods and overlap minimization algorithms. The former group consists of methods which attempt to both eliminate collisions and compact the layout using a combination of two techniques: separation and compaction, respectively. Gomes and Oliveira (2006) used a linear programming based compaction and separation algorithms with SA to guide the local search.

The first overlap minimization approach for the strip packing problem was proposed by Egeblad et al. (2007). Overlap area between items is minimized by the investigation of all horizontal and vertical translations for each item. It uses a fast local search to escape local minima. Umetani et al. (2009) used a similar approach, combining the overlap minimization with guided local search. Leung et al. (2012) improved the solution by applying taboo search metaheuristic. Elkeran (2013) used the guided local search in combination with cuckoo search to search for a minimal overlap layout. Each of these approaches obtained new best solutions for benchmark problems when published.

3. PREVIOUS WORKS

The CFR determines all possible placements for an item in a layout and was defined in Martins and Tsuzuki (2006). Sato et al. (2010) implemented a robust algorithm to determine the CFR using non regularized Boolean operations. Using this implementation, it was possible to detect exactly fitting and exactly sliding positions which, as indicated by the results, are important to obtain more compact layouts. Six best layouts were obtained, however the execution time of the implementation was higher than all other approaches in literature.

As SA was used to control the position along the contour of CFR for each item, fitting placements were rarely discovered. Moreover, due to the employment of finite precision geometric tools, these special positions were not found if the sequence of Boolean operations was changed. In Sato et al. (2013), an algorithm which imposes the creation of exactly fitting or sliding positions for one item in a pair was proposed. Placements heuristics adapted to work with the required pairwise placement were tested but results showed no improvement of the solutions. It was then observed that not all fitting placements are important to obtain an efficient layout.



(a) (b)

Fig. 1. Configurations where the overlap area (in black) in (b) is larger than the overlap area in (a). However, for both cases, only a small translation is needed to separate the items.

The CFR derives from the nofit polygon (NFP) concept, which defines a forbidden placement region. The CFR concept was designed for solutions which perform the search over the sequence. In this work, however, the NFP is used to determine the penetration depth and a search over the layout is executed in order to minimize the overlap.

4. OVERLAP EVALUATION

The proposed approach in this work allows for collision between items. In order to obtain a feasible solution, it is necessary to adopt a strategy which eliminates the overlap and guarantee that all items are inside the container. Overlap minimization techniques assume a value which quantifies the total collision in the layout and attempts to minimize it. The objective is to obtain a solution with zero overlap.

One of the main difficulties in irregular packing problems is to assure that no two items intersect with each other. In this work, the space inside the container is discretized and the items are represented by matrices. An intuitive method of overlap evaluation consists of measuring the area of overlapped regions and is very straightforward when using a raster representation. However, it would be very time consuming to detect the overlapped area for each pair of items at each iteration. Moreover, in some cases a solution with high overlap area can be easily transformed into a feasible one by applying a small translation (see Fig. 1). Overlap minimization is often performed by moving items in the layout. Thus, it is more convenient to adopt the penetration depth, which determines the minimum translation to separate two items, as the overlap function.

So as to assist the description of the methods used in this work, some notations are employed. A layout of a set of items $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ is defined by a translation vector $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$, where $x_i \in \mathbb{R}^2$, and an orientation vector $o = \{o_1, o_2, ..., o_n\}$, $0 \leq o < 2\pi$. A placed item in the layout is described as $P(o)$, where o is its orientation, retrieved from vector o. A translated item

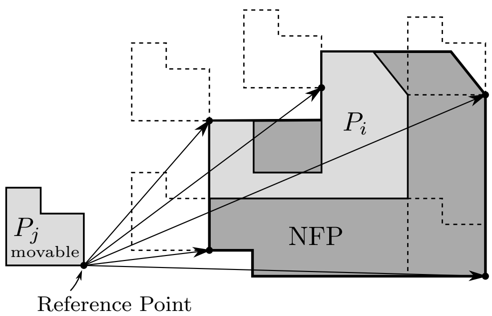
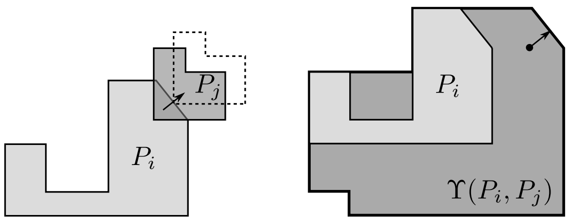


Fig. 2. NFP induced by movable item P_i to item P_j mapped to a polygonal region.



(a) (b)

Fig. 3. The penetration depth of items P_i and P_j, indicated by the size of the arrow.

is denoted by $P(o) \oplus \boldsymbol{x}$, where $\oplus$ is the Minkowski sum. The orientation can be omitted if it is not necessary.

4.1 Nofit Polygon

The NFP is a concept proposed by Art (1966) and it is used to determine whether two items intersect, touch or are separated. Consider two items, the fixed item and the moveable item, both already positioned in the layout and with their orientation set. The NFP determines the set of translation that, when applied to the moveable item, places it in collision with the fixed item. The NFP induced by the fixed item P_i to the moveable item P_j is denoted by $\Upsilon(P_i, P_j)$.

These translations are mapped onto the 2D space using a reference point for the moveable item, as shown in Fig. 2. The NFP can be obtained by sliding the moveable item along the contour of the fixed item.

4.2 Penetration depth

For a pair of items P_i and P_j, the penetration depth of P_j measures the norm of the minimum translation that, when applied to P_j, separates it from P_i. The penetration depth $\delta(P_i, P_j)$ can be described as

$$\delta(P_i, P_j) = \min\left\{\|\boldsymbol{v}\| \mid i(P_j \oplus \boldsymbol{v}) \cap i(P_i) = \emptyset\right\} \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean norm.

Fig. 3(a) shows an example of two colliding items and their penetration depth. Using the NFP, it is possible to determine the penetration depth more efficiently (see Fig. 3(b)). For a NFP $\Upsilon(P_i, P_j)$, the penetration depth of P_j is the minimum distance from its reference point r to a point in the contour of the NFP. Thus,

$$\delta(P_i, P_j) = \min\left\{\|\boldsymbol{v}\| \mid r + \boldsymbol{v} \notin i(\Upsilon(P_i, P_j))\right\}. \quad (2)$$

So as to determine the penetration depth, it is necessary to determine the closest edge or vertex of the NFP from the reference point. This can be achieved using the medial axis concept, which consists of the set of all points with multiple closest points to the contour of the polygon. Fig. 4 shows an example of a medial axis of a convex polygon. It is possible to observe that the medial axis divides the polygon into regions, each corresponding to an element of the contour, edge or vertex. For a given region, the distance from any internal point to its corresponding element is the minimum to any point along the contour of the polygon. Thus, the penetration depth can be easily determined once the region containing the reference point is identified.

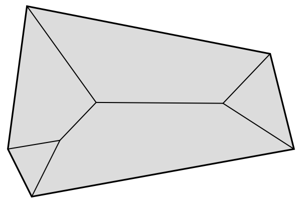
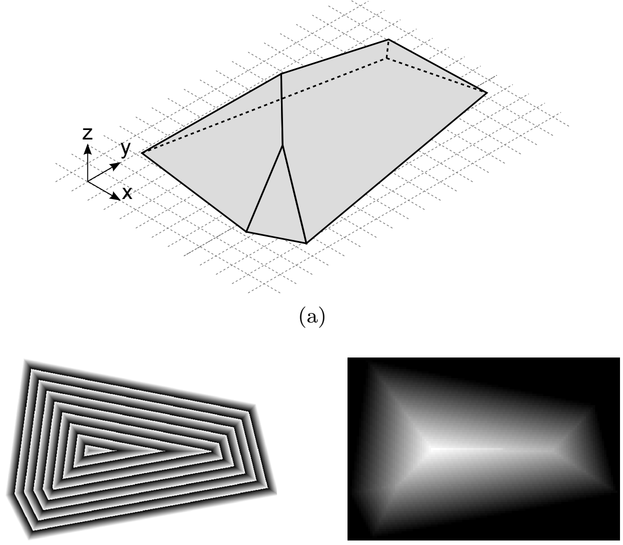


Fig. 4. Medial axis of a convex polygon.



(a)

(b) (c)

Fig. 5. Voronoi mountain of a convex polygon. (a) 3D view. (b) Highlighted iso-contours representation. (c) 2D discrete grayscale transformation.

4.3 Voronoi Mountain

Voronoi diagram for closed polygons can be used to obtain the medial axis. The Voronoi mountain is a concept applied to determine the optimal cutting path in 2.5D pocket machining (Veeramani and Gau, 1997). The Voronoi mountain is obtained by extruding all regions from the Voronoi diagram. After the extrusion, for each region, a Boolean subtraction operation is performed such that the solid height equals to the distance from the corresponding edge or vertex of the region. Fig. 5 shows an example of a Voronoi mountain for the convex polygon of Fig. 4. For any point inside the polygon, the height of the Voronoi mountain has the same value as the penetration depth.

For convex polygons, the Voronoi regions are always associated to an edge of the polygon. In these cases, the top surface of the Voronoi mountain is formed by a set of planes which forms a 45^o angle with the original polygon plane and are delimited by the edges of the Voronoi diagram regions. It is then possible to represent the Voronoi mountain by a set of plane equations. Assume that the original polygon is contained in the xy plane and the z coordinate represents the height of the Voronoi mountain. Also, consider that the points $P_1 = (P_{1x}, P_{1y})$ and $P_2 = (P_{2x}, P_{2y})$ are the endpoints of the edge corresponding to a Voronoi region. Each plane equation is given by

$$z = \frac{1}{\Delta^2}\left[\Delta_y \cdot x + \Delta_x \cdot y - \Delta^2\right] \tag{3}$$

where $\Delta^2 = \sqrt{\Delta_y^2 + \Delta_x^2}$, $\Delta_x = P_{2x} - P_{1x}$ and $\Delta_y = P_{2y} - P_{1y}$. Equation (3) is equivalent to the point-line distance.

In the case of non-convex polygons, Voronoi regions may be associated with a concave vertex. The distance function for such regions is obtained by constructing an upside-down cone with its apex coinciding with the concave vertex and whose generatrix lines makes an 45^o angle with its axis. Consider $P_c = (P_{cx}, P_{cy})$ the concave vertex which generates the Voronoi region. Thus, the function is given by

$$z = \sqrt{(x - P_{cx})^2 + (y - P_{cy})^2} \tag{4}$$

which is equivalent to the distance between points. Eqs. (3) and (4) show that the height of the Voronoi mountain for any given point always matches its penetration depth.

4.4 Discretization and preprocessing

One of the main advantages of the NFP is that it only needs to be computed once for each pair of items. Hence, for the Voronoi mountain approach, it is also only necessary to determine the parameters of the plane and cone surfaces a single time. All this calculations can be performed in a preprocessing step. Using these results, the computation of the penetration depth consists basically of the determination of the relative position of the reference point to the translated NFP and the application of equations (3) or (4).

To further improve the algorithm efficiency through pre-computation, the Voronoi mountain is discretized. Using a matrix representation, it is possible to determine the penetration depth for every element of the matrix. A result of this discretization is exemplified in Fig. 5(c), where lighter shades correspond to higher regions of the Voronoi mountain. This transformation is applied to all NFPs and, as a consequence, the task of obtaining the penetration depth of any two given items in the layout is converted into a simple procedure of retrieving a value of a stored matrix. The big drawback is the high memory requirement to store all NFPs, which increases with finer precision. Nevertheless, amount of necessary memory can be estimated in advance and the precision can be set accordingly.

5. OVERLAP MINIMIZATION

In the previous section, the overlap function for two items was introduced. In order to solve a packing problem, the overlap minimization must also take into account multiple

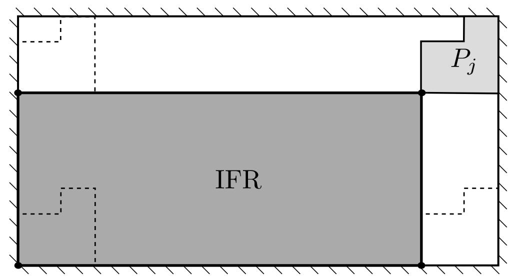


Fig. 6. IFR for a container and a moveable item P_j.

items and the container. The total overlap function F for an item $P_i(o_i)$ in the layout is given by

$$F(P_i(o_i), \boldsymbol{x}, o) = \sum_{j=1, j\neq i}^{n} \delta(P_i(o_i), P_j(o_j)). \tag{5}$$

For the proposed overlap minimization algorithm, only solutions where all items are completely inside the container are considered. The inner fit rectangle, which is derived from the NFP concept, is the geometric tool employed to satisfy this restriction. It represents all translations that, when applied to the item, inserts it into the container without protrusion. Fig. 6 shows an example of the inner fit rectangle (IFR). During the search over the layout, only positions inside the IFR are accepted.

5.1 Total overlap map

For an item in the layout, its total overlap map represents, for each point in the grid inside the IFR, the value of the total overlap. One example is shown in Fig. 7. For an item P_i, the total overlap map can be determined by primarily creating a raster representation of its inner fit triangle with all values set to zero. Next, for each item, its Voronoi mountain processed NFP is translated and cumulatively added to this matrix. After processing all items in the layout excluding P_i, the total overlap map is complete.

The cells with value zero in the map represent feasible placements for item P_i. If a zero threshold is applied to the map, a raster variation of the CFR is obtained.

5.2 Local search

Using the proposed total overlap map, the search for the minimum overlap placement for one item in the layout is obtained by searching for the position of the minimum value in the map. The local search used in this work is adapted from (Elkeran, 2013). Initially, a sequence of items is randomly defined. Then each item is translated to its minimum overlap placement in the layout. When multiple orientations are admissible for the item, all orientations are tested and the one which holds the lowest total overlap value is chosen.

5.3 Guided local search

Local search approaches have a tendency to lock on local minima solutions. It is then imperative to adopt a meta-heuristic in order to progress the search for the optimal solution. The guided local search proposed in (Umetani et al., 2009) was chosen for this work. Weights, initially

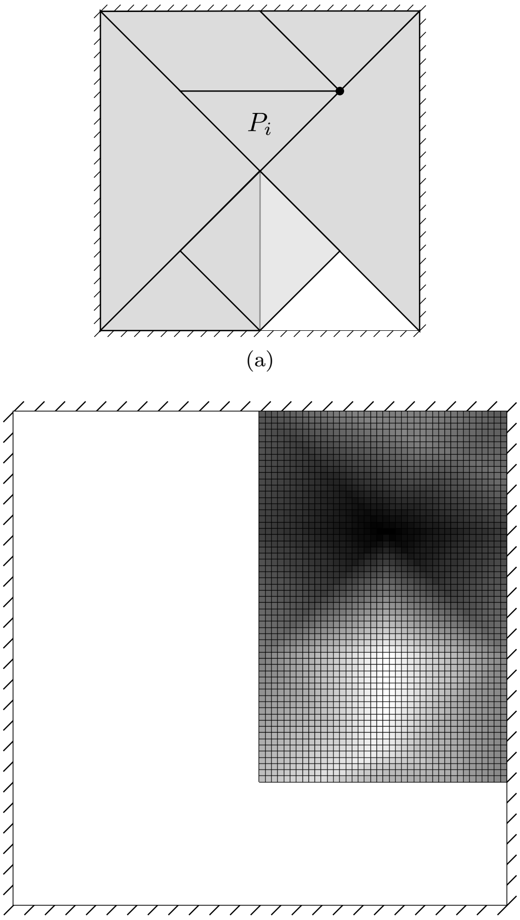


Fig. 7. Overlap map example. (a) Layout for example. Reference point for item P_i is marked with a filled circle. (b) Overlap map for item P_i (scaled). Darker shades of gray indicates smaller overlap values.

assigned the value one, are attributed to each pair of items. The objective function is modified by multiplying each overlap function by its corresponding weight, denoted by w_{ij}, which is associated with items P_i and P_j. The modified function can be expressed as

$$F'(P_i(o_i), \boldsymbol{x}, o) = \sum_{j=1, j\neq i}^{n} w_{ij} \cdot \delta(P_i(o_i), P_j(o_j)). \quad (6)$$

After each iteration, the value of weights are updated. The rule for updating guarantees that the search is not trapped in a local minimum. The adopted implementation, proposed in Umetani et al. (2009), uses the following rule

$$w_{ij} = w_{ij} + \frac{\delta(P_i, P_j)}{\max(\delta(P_k, P_l))} \quad (7)$$

where $1 \leq k < l \leq n$.

First, the maximum penetration depth of all combinations is determined. Then, the weight of each pair of item is incremented by the value of the overlap relative to the maximum value. When a non feasible solution does not change after one local search iteration, the weights of the overlapping items increase and elevate the value of the overlap of colliding items. After some iterations, the total overlap of such items reaches a non minimum value and they are translated to lower overlap positions.

Table 1. Computational times for the Tangram case. Time per iteration values are approximated. All time values are in seconds.

Scale Factor	Preprocessing	Time per iteration
1	31.68	< 0.01
10	197.72	0.01
100	2885.04	0.20

When a number N_{mr} of solutions which does not improve the fitness function are found, the execution of the algorithm is halted. Also, if a feasible solution is found, the algorithm is finished and the layout is returned.

6. RESULTS

Tests for the overlap minimization algorithm were performed with two sets of data. One is the classic puzzle case Tangram, with a container with fixed dimensions, and the other is the case Fu, which is a benchmark data set for strip packing problems. Initial solution was obtained by randomly generating positions for each item inside the container, keeping the default orientations. Parameter N_{mr} was set to 200 iterations. The algorithm was implemented using MATLAB and the tests were performed on a Xeon E5645, 2.40GHz with 48GB.

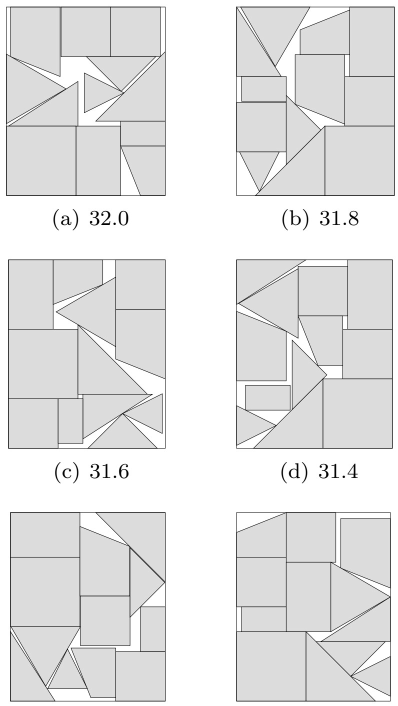
The Tangram puzzle is shown in Fig. 7(a). The original container was set to a 8×8 grid. As the speed of raster methods varies with the adopted precision, three scaled versions of the original Tangram puzzle were tested and the execution times were measured. Table 1 shows the preprocessing times and the times per iteration of the overlap minimization algorithm. It is important to note that the preprocessing step only needs to be performed once. Results show that the preprocessing times is hugely influenced by the scale factor and, thus, it should be chosen carefully. In all tests the algorithm converged to the correct solution.

Fu problem consists of 12 items with 4 admissible orientations. The Fu data set was scaled by a factor of ten for a finer discretization. Table 2 shows the results obtained for the Fu case. To allow a better comparison with other solutions, a execution of the algorithm was defined as a continuous run of 10 min. The algorithm was restarted each time it returned. For each run, a container with a fixed length is defined and the algorithm was executed 36 times for each length. Fig. 8 shows the best layouts for each Fu test.

Most compact layout for the Fu case was obtained in (Elkeran, 2013), with a density of 92.41%. The execution was also limited to 600 seconds. Other best solutions achieved the following utility rates: 92.03% (Egeblad et al., 2007) and 91.94% (Leung et al., 2012). The best compaction obtained by the proposed approach was 91.95% (see Fig. 8(f)) which is not very different from top solutions which ran for 10 minutes. Best layout in previous works was obtained in Sato et al. (2012), with a density of 91.96%, but the execution time was more than 100 times higher. Even with limited precision, the proposed algorithm obtained a competitive solution. The same scenario can be observed for the Tangram problem, a puzzle case, which was successfully solved.

Table 2. Results for the Fu case. **Avg MO**: average minimum overlap of non-valid solutions. $\mathbf{P_{conv}}$: percentage of convergence of executions. Average iteration time was 0.13 seconds.

Length	Density (%)	Avg MO	P_{conv}
32.0	89.07	1.10	70.59
31.9	89.35	1.72	17.86
31.8	89.63	3.83	12.90
31.7	89.92	3.05	15.63
31.6	90.20	3.57	9.68
31.5	90.49	4.74	6.06
31.4	90.77	4.77	3.03
31.3	91.06	5.39	3.03
31.2	91.36	7.29	3.03
31.1	91.65	5.45	0.00
31.0	91.95	6.45	3.03
30.9	92.24	6.69	0.00



(a) 32.0 (b) 31.8 (c) 31.6 (d) 31.4 (e) 31.2 (f) 31.0

Fig. 8. Some layouts for the Fu problem obtained by the proposed approach.

7. CONCLUSION

In this work an overlap minimization algorithm using raster NFPs was proposed. The search over layout strategy was adapted from Elkeran (2013), changed to accommodate the discrete approach. The intention was to transfer the computational load during the execution of the algorithm to a preprocessing step. Computational times confirmed that the performance of the algorithm is highly dependent upon the adopted precision. However, even with relatively low precision, the algorithm was able to achieve good compaction for the benchmark problem Fu when compared to other solutions in the literature.

The work is at an early stage and the tests were performed in order to verify the potential of the algorithm to achieve good layouts in reasonable time. In order to allow a better comparison with literature solutions, the algorithm must be fully converted to a strip packing solution by including an automatic reduction and expansion of the container. Another huge improvement for the algorithm is the parallelization of the discrete Voronoi mountains and overlap maps determination.

ACKNOWLEDGEMENTS

AK Sato was supported by FAPESP (Grant 2010/19646-0). MSG Tsuzuki was partially supported by the CNPq (Grant 309.570/20107). This research was supported by FAPESP (Grants 2008/13127-2 and 2010/18913-4).

REFERENCES

Art, R. (1966). An approach to the two-dimensional irregular cutting stock problem. *Tech Rep 36.008, IBM Cambridge Centre*.

Egeblad, J., Nielsen, B., and Odgaard, A. (2007). Fast neighborhood search for two- and three–dimensional nesting problems. *Eur J Oper Res*, 183, 1249–1266.

Elkeran, A. (2013). A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. *Eur J Oper Res*, 231(3), 757–769.

Fowler, R., Paterson, M., and Tanimoto, S. (1981). Optimal packing and covering in the plane are NP-complete. *Inf Process Lett*, 12(3), 133–137.

Gomes, A. and Oliveira, J. (2002). A 2-exchange heuristic for nesting problems. *Eur J Oper Res*, 141, 359–370.

Gomes, A. and Oliveira, J. (2006). Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *Eur J Oper Res*, 171, 811–829.

Leung, C., Lin, Y., and Zhang, D. (2012). Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem. *Comput Oper Res*, 39(3), 678–686.

Martins, T. and Tsuzuki, M. (2006). Simulated annealing applied to the rotational polygon packing. *Proc 12th IFAC INCOM*, 475–480.

Oliveira, J., Gomes, A., and Ferreira, J. (2000). TOPOS – a new constructive algorithm for nesting problems. *OR Spectrum*, 22, 263–284.

Sato, A., Martins, T., and Tsuzuki, M. (2010). Rotational placement using simulated annealing and collision free region. *Proc 10th IFAC IMS*, 253–258.

Sato, A., Martins, T., and Tsuzuki, M. (2012). An algorithm for the strip packing problem using collision free region and exact fitting placement. *CAD*, 44(8), 766–777.

Sato, A., Martins, T., and Tsuzuki, M. (2013). Placement heuristics for irregular packing to create layouts with exact placements for two moveable items. *Proc 11th IFAC IMS*, 384–389.

Umetani, S., Yagiura, M., Imahori, S., Imamichi, T., Nonobe, K., and Ibaraki, T. (2009). Solving the irregular strip packing problem via guided local search for overlap minimization. *Int T Oper Res*, 16(6), 661–683.

Veeramani, D. and Gau, Y. (1997). Selection of an optimal set of cutting-tool sizes for $21/2D$ pocket machining. *CAD*, 29(12), 869–877.

Wäscher, G., Haussner, H., and Schumann, H. (2007). An improved typology of cutting and packing problems. *Eur J Oper Res*, 183, 1109–1130.