1. INTRODUCTION

Problems involving cutting and packing procedures can be optimized in order to minimize wasted material or to improve space allocation. Many industrial areas are affected by such problems, e.g., glass, wood, textile and shipbuilding industries. An efficient solution for cutting and packing problems can have a significant environmental and economical impact.

The problem considered in this work consists of placing a number of irregular items inside a rectangular container with a variable length, which is defined as 2D irregular open dimension problem (Wäscher et al., 2007) and more commonly referred to as irregular nesting problem (INP). The constraints are: items cannot overlap and must lay entirely inside the container. This problem is considered NP-Hard (Fowler et al., 1981).

Avoiding the geometric constraints is a complex task when irregular items are allowed. Consequently, more advanced techniques need to be employed to solve such problems. One of the most popular tools is the no-fit polygon (NFP), which converts the problem of detecting overlap to a simpler geometric operation. Another solution is to rasterize the items and evaluate the overlap by performing pixel comparisons. Raster methods aim for a fast collision detection, but often suffer from low precision. Many optimization techniques have been applied to solve the INP in the literature. Although several solutions in literature involve using a constructive placement heuristic to avoid collision (Sato et al., 2012, 2013; Oliveira et al., 2000; Bennell and Song, 2010), most successful approaches to date use a search over the layout procedure. It consists in manipulating items coordinates freely, allowing collision during the search, with the objective of minimizing the overlap.

Egeblad et al. (2007) developed one of the first successful overlap minimization algorithm using a guided neighborhood search, detecting minimum overlap for horizontal movements of items. Umetani et al. (2009) searched for horizontal and vertical item translations with a modified guided local search. Imamichi et al. (2009) and Leung et al. (2012) used nonlinear programming to minimize the overlap of the layout. Finally, Elkeran (2013) proposed a guided cuckoo search, which combined guided local search with the cuckoo search metaheuristic to separate the items. Sato et al. (2014) adapted the solution from Elkeran (2013) by exchanging the cuckoo search metaheuristic with a raster method to obtain the minimum overlap position for an item.

In this work, an expanded version of the raster method proposed by Sato et al. (2014) to solve the INP is adopted. A multiresolution approach using the distance transform (DT) is proposed in order to improve the speed of the algorithm. Tests with 4 benchmark problems from literature were performed and competitive results were obtained. Comparing the results obtained using the single and multiresolution algorithms, it was possible to observe
that the latter performed much faster and achieved better results in all cases.

The text is structured as follows: Section 2 describes the overlap minimization problem. The mechanism used to detect overlap between items is explained in Section 3. Section 4 gives a brief overview of the algorithm, whose components, the multiresolution overlap minimization and separation are detailed in section 5 and Section 6, respectively. Results are shown in Section 7 and conclusion are drawn in Section 8.

2. PROBLEM DESCRIPTION

INPs consists of placing irregular shaped items inside a rectangular container, avoiding overlap between items and protrusion of items from the container. Each item can be rotated by a finite set of angles. A nesting problem can be defined by a set of n items \( P = \{P_1, P_2, \ldots, P_n\} \), n sets of admissible orientations \( O_i \), for each item \( P_i \), and a rectangular container \( C \) with an infinite length and a fixed width. A solution can be described by a placement vector \( x = \{x_1, x_2, \ldots, x_n\} \), which consists of translations to be applied to each item, and a orientation vector \( o = \{o_1, o_2, \ldots, o_n\} \), \( o_i \in O_i \), which represents the orientation of each item. Item \( P_i \) with orientation \( o_i \) can be denoted by \( P_i(o_i) \). An item translated by \( x_i \) can be represented by \( P_i(o_i) \oplus x_i \), where \( \oplus \) is the Minkowski sum. The orientation can be omitted when not necessary. Operator \( i(P_i) \) represents only the interior points of \( P_i \). The INP can be described as minimize \( L \) subject to

\[
\begin{align*}
i(P_i(o_i) \oplus x_i) \cap (P_j(o_j) \oplus x_j) &= \emptyset \quad 1 \leq i, j \leq n \\
(P_i(o_i) \oplus x_i) \subseteq C \quad 1 \leq i \leq n \\
L &\in \mathbb{R}_+
\end{align*}
\]

\[
o_i \in O_i \quad 1 \leq i \leq n
\]

\[
x_i \in \mathbb{R}^2 \quad 1 \leq i \leq n
\]

where \( L \) is given by

\[
L = \max \{y_1 \mid (y_1, y_2) \in P_i(o_i) \oplus x_i, P_i \in \mathbb{P} \} - \min \{y_1 \mid (y_1, y_2) \in P_i(o_i) \oplus x_i, P_i \in \mathbb{P} \}
\]  

2.1 Overlap minimization problem

The constraints in the INPs, shown in (1), are very difficult to satisfy. One common strategy is to transform the problem into a series of sub-problems which considers the length of the container fixed. This sub-problem is similar to the irregular bin packing problem. The constraint can then be relaxed through the use of a overlap function \( F(P_i(o_i), P_j(o_j)) \), which is zero only when two items do not collide. The sub-problem can be described as

\[
\begin{align*}
\text{minimize} \quad & \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} F(P_i(o_i), P_j(o_j)) \\
\text{subject to} \quad & (P_i(o_i) \oplus x_i) \subseteq C \quad 1 \leq i \leq n \\
L &\in \mathbb{R}_+ \\
o_i &\in O \quad 1 \leq i \leq n \\
x_i &\in \mathbb{R}^2 \quad 1 \leq i \leq n
\end{align*}
\]

Fig. 1. NFP example.

When \( F \) is zero, a feasible solution is found. In order to solve the nesting problem, it is necessary to employ a procedure to incrementally shrink the container. In this adapted problem the solutions space is a more connected landscape and, thus, optimization techniques are bound to perform better.

3. OVERLAP DETECTION AND EVALUATION

Overlap minimization techniques assign a value for the collision between items and searches for a layout with zero overlap. In this work, the NFP is employed to determine the penetration depth (PD), which obtained using the DT for the rasterized item.

3.1 No-fit polygon

The NFP is a geometric tool proposed by Art (1966) largely used in irregular packing problems. It detects collision between two items. This is achieved by considering one of the items fixed and the other movable. The NFP describes all translations that, when applied to the fixed item, causes it to collide, i.e., overlap, with the fixed item. By defining a reference point for the movable item, the NFP can be represented as a region in space, as shown in Fig. 1.

Previous to the NFP, all geometric tools required that, for each overlap verification, a sequence of complex operations needed to be performed. The main advantage of the NFP is that the most time consuming calculations can be carried out offline. The NFPs are determined in a pre-processing phase with discrete orientations. Using the previously determined NFP, the overlap determination is simplified to detecting whether a point lays inside the polygon.

In this work, the NFP is rasterized using a predefined grid. Therefore, collision detection can be determined by checking the value of a single pixel.

3.2 Penetration depth

In overlap minimization based packing solutions, a value must be assigned to the overlap between two items. If the items do not collide, the overlap value is zero.

The most intuitive value is the overlap area, as employed by Egelblad et al. (2007). However, in some cases, a pair of items may have a large intersection area but are easily separated. Fig. 2 shows examples of such case. Therefore, many solutions adopt the PD as the overlap function. The PD is the norm of the minimum translation that needs
Consider a two items $P_i$ and $P_j$. The PD $\delta(P_i, P_j)$ for the item pair $P_i$ and $P_j$ can be described as

$$\delta(P_i, P_j) = \min \{ \|v\| \mid i(P_j \oplus v) \cap i(P_i) = \emptyset \}$$

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

### 3.3 Distance Transform

DT is a technique used in image processing to perform skeletonizing and blurring effects. It is applied to a binary image, in which obstacle pixels are marked. The result, defined as the distance map, consists of an image in which each pixel contains the value of the minimum distance to an obstacle pixel. This value corresponds, in the raster image, to the PD.

In order to use the distance map to determine the PD value, the rasterized NFP is employed. The PD is the minimum distance between the reference point to a point on the contour of the NFP. Then, the distance map of the rasterized NFP describes the PD for each point inside the NFP. Figure 4 shows an example of the DT applied to the NFP. If the reference point is outside the NFP, the PD value is zero.

### 3.4 Layout and item overlap evaluation

The overlap minimization algorithm aims to eliminate the overlap between items. The objective is to obtain a layout in which the PD for every pair of items is zero. So as to aid the search, an auxiliary function $f$ is defined for each item. It consists of the sum of the PD with all the other items and can be described as

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

$f$ is defined as the total item overlap function.

Global layout overlap $F$ is defined as the sum of all item overlap values and can be described as

$$F(x, o) = \sum_{i=1}^n f(P_i(o_i), x, o).$$

When the global layout overlap value is zero, the layout is feasible.

### 4. ALGORITHM OVERVIEW

The algorithm proposed in this work solves the overlap minimization problem using a raster method based on (Sato et al., 2014). The work is expanded by the use of a multiresolution search. Fig. 5 shows the outline of the algorithm, which solves the overlap minimization problem for a container with fixed dimensions. In order to solve the INP, the length of the container can be set manually or automatically by applying a shrinking strategy. In this work, the length is controlled by the user.
consider a two items separate them (see Fig. 3).

The first step is to load the raster NFPs, which are determined in a preprocessing stage. In a multiresolution approach, two sets of NFPs are loaded, each rasterized with different grid sizes. Then, a random initial solution is generated. The main loop consists of the guided local search metaheuristic. The algorithm runs until a feasible solution is found (overlap is zero) or a time limit is reached.

The local search, which is adapted to a guided local search by employing weights to avoid locally optimum layouts, is the main component of the proposed solution. It consists of translating the items sequentially to minimum overlap positions. The raster method is employed to efficiently determine minimum overlap position for an item in the layout.

5. MULTiresolution OVERlap MINimization

The separation strategy adopted in this work involves translating items to positions of minimum overlap. This procedure can be computationally costly if high resolution rasterization is performed. Thus, a multiresolution overlap minimization strategy is proposed to achieve a faster solution. Guided local search is performed to escape a local optimum solution.

5.1 Total overlap map

In a raster approach, the set of placement vectors for an item \( P_i \) that inserts the item completely inside the container is finite. Moreover, for rectangular containers, it can be mapped to a rasterized rectangular shape, the inner fit rectangle (IFR). As only placements inside the container are allowed in this work, the search for a minimum position is restricted to the raster IFR.

Using (5), it is possible to determine the overlap value for an item \( P_i(o_i) \) placed in the layout. It is then possible to create a map with all overlap values inside the IFR, the total overlap map for item \( P_i \), denoted by \( M(P_i, g) \), where \( g \) is the grid size. Fig. 6 shows an example of a total overlap map.

5.2 Multiresolution search

The minimum overlap position for item \( P_i \) can be obtained by searching the index of the minimum value of the map. The time efficiency of the total overlap map creation and its minimum value search routines dictates the speed of the packing algorithm. Moreover, these routines run in quadratic complexity for different grid size inputs. Hence the grid size and, consequently, the precision of the algorithm is limited by the performance of the overlap minimization routine.

In order to circumvent this limitation, a multiresolution search was proposed. The search is performed on two levels, each with different resolutions, or grid sizes: \( g_r \) and \( g_f \). The first search is performed on the rough map, with higher grid size \( g_r \), \( M(P_i, g_r) \). The second step is to create a smaller square map centered around the minimum overlap position from the first search using the fine grid size \( g_f \). This map is denoted by \( M'(P_i, g_f, x, l) \), where \( x \) is its center and \( l \) is its length. The final placement is determined by the position of minimum value of \( M'(P_i, g_f, x, l) \).

Figure 7 shows an example of the multiresolution search procedure. Fig. 7(b) shows the first step, in which a low resolution map is used to find the approximated minimum position. The fine grid map is shown in Fig. 7(c), in which the final placement is determined.

As the complete map is constructed using the rough resolution, the value \( g_r \) determines the speed of the routine.
which is a side effect of using a coarse grid, whereas the precision is dictated by the value of $g_f$. The value of the square length $l$ is usually small enough that it does not impact the algorithm run time significantly.

6. SEPARATION STRATEGY

Reducing the overlap to zero is the objective of the separation algorithm. It consists of applying a metaheuristic, guided local search, to obtain a layout without overlap. The multiresolution overlap minimization is applied to find the minimum overlap placement for a single item in the local search procedure.

6.1 Local search

The local search was adapted from Elkeran (2013). It consists of translating each item to its minimum overlap position, in a random sequence, using the total overlap map. For each item with multiple admissible orientations, all rotations are tested and the orientation which gives the minimum overlap placement is applied.

6.2 Guided local search

Using local search directly usually leads to local minimum solution. So as to avoid this problem, a metaheuristic approach is often employed. The guided local search has been widely applied to the overlap minimization problem Egeblad et al. (2007); Umetani et al. (2009); Elkeran (2013). It consists of adapting the objective function by adding weights. The modified overlap function is described as

$$ F'(P_i(o_i), x, o) = \sum_{j=1, j\neq i}^n w_{ij} \cdot \delta(P_i(o_i), P_j(o_j)). $$ (7)

After each iteration, the weights are updated according to

$$ w_{ij} = w_{ij} + \frac{\delta(P_i, P_j)}{\max(\delta(P_k, P_l))}, $$ (8)

which was proposed by Umetani et al. (2009). This rule guarantees that the solution will not be trapped in a local minima. After a number of iterations, the weights must be reset, as the objective function may be completely impacted by the weights such that the original overlap values does not contribute to the separation procedure.

7. RESULTS

The proposed algorithm was implemented using Visual Studio 2010 and tests were performed on a Xeon E5645, 2.40GHz with 4GB. Four benchmark cases from the literature were selected to evaluate the algorithm: Albano (24 items), Dighe1 (16 items), Shapes1 (43 items) and Shirts (99 items). Albano and Shirts problems were extracted from garment industry cases, whereas Dighe1 and Shapes1 were artificially created. With the exception of Dighe1, which is a translational problem, $0^\circ$ and $180^\circ$ are the allowed orientations.

As the cases studied were initially INPs, the length of the container is manually determined in order to apply the proposed overlap minimization algorithm. Lengths were gradually reduced in order to obtain the most compact solution. For each length, 12 executions were performed.

<table>
<thead>
<tr>
<th>Case</th>
<th>$g_r$</th>
<th>$g_f$</th>
<th>Length</th>
<th>D (%)</th>
<th>$P_{conv}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albano</td>
<td>–</td>
<td>10</td>
<td>10050</td>
<td>86.62</td>
<td>16.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10000</td>
<td>87.06</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9950</td>
<td>87.49</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>10</td>
<td>10050</td>
<td>86.62</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10000</td>
<td>87.06</td>
<td>50.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9950</td>
<td>87.49</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9900</td>
<td>87.94</td>
<td>16.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9850</td>
<td>88.38</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9800</td>
<td>88.83</td>
<td>0.0</td>
</tr>
<tr>
<td>Dighe1</td>
<td>– 0.1</td>
<td>130</td>
<td>76.92</td>
<td>50.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>120</td>
<td>83.33</td>
<td>33.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>110</td>
<td>90.91</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
<td>100.00</td>
<td>16.7</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.1</td>
<td>130</td>
<td>76.92</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>120</td>
<td>83.33</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>110</td>
<td>90.91</td>
<td>83.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Shapes1</td>
<td>– 0.1</td>
<td>54.8</td>
<td>72.81</td>
<td>25.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>54.6</td>
<td>73.07</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>54.4</td>
<td>73.34</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.1</td>
<td>54.8</td>
<td>72.81</td>
<td>91.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>54.6</td>
<td>73.07</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>54.4</td>
<td>73.34</td>
<td>91.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>54.0</td>
<td>73.88</td>
<td>58.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>53.0</td>
<td>75.28</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>52.8</td>
<td>75.56</td>
<td>0.0</td>
</tr>
<tr>
<td>Shirts</td>
<td>– 0.1</td>
<td>62.8</td>
<td>85.99</td>
<td>8.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62.6</td>
<td>86.27</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62.4</td>
<td>86.54</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.1</td>
<td>62.8</td>
<td>85.99</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62.6</td>
<td>86.27</td>
<td>16.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62.4</td>
<td>86.54</td>
<td>18.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62.2</td>
<td>86.82</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62.0</td>
<td>87.10</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Each execution had a time limit of 600 seconds. The grid size is the main factor which impacts the speed of the algorithm. Therefore, as a limit of 2 seconds per iteration was adopted, the fine grid size $g_f$ was chosen accordingly. For each case, single and multiresolution versions of the proposed solution with the same grid size were executed for comparison. The value of the rough grid size $g_r$, used to accelerate the minimum overlap search, is set to a multiple of $g_f$.

Table 1. Results for cases Albano and Dighe1.

D: density of the layout. $P_{conv}$: percentage of the number of runs which converged to a feasible solutions.

Table 2. Results for cases Shapes1 and Shirts.

Tables 1 and 2 display the results obtained for the execution with different lengths using both the single and multiresolution approaches. It is possible to observe that, in all cases, the minimum length obtained by the solution with the multiresolution approach was lower or equal to the single resolution approach. The Dighe1 case is a broken-glass problem, which has a known optimum solution, and both versions of the algorithm were capable of reaching the best layout. However, comparing the executions with
In order to allow for a better comparison with other solutions in literature, the algorithm must be adapted to solve the INP and the precision error caused by the raster approach must be evaluated. The speed of the algorithm can be further improved by parallelizing the determination of the total overlap map, which is the most time consuming routine of the algorithm.

ACKNOWLEDGEMENTS

AK Sato was supported by FAPESP (Grant 2010/19646-0). MSG Tsuzuki and TC Martins were partially supported by the CNPq (Grants 310.663/2013-0 and 306.415/2012-7). This research was supported by FAPESP (Grants 2008/13127-2, 2010/18913-4 and 2013/26532-9) and CPNq (Grant 456180/2014-1).

REFERENCES


