Discrete Optimization

Heuristic approaches to large-scale periodic packing of irregular shapes on a rectangular sheet

M. Teresa Costa a,*, A. Miguel Gomes b, José F. Oliveira b

a Instituto Superior de Engenharia do Porto, Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto, Portugal
b INESC Porto – Instituto de Engenharia de Sistemas e Computadores do Porto, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n, 4200-465, Porto, Portugal

Received 2 June 2006; accepted 5 September 2007
Available online 14 September 2007

Abstract

The nesting problem is a two-dimensional cutting and packing problem where the small pieces to cut have irregular shapes. A particular case of the nesting problem occurs when congruent copies of one single shape have to fill, as much as possible, a limited sheet. Traditional approaches to the nesting problem have difficulty to tackle with high number of pieces to place. Additionally, if the orientation of the given shape is not a constraint, the general nesting approaches are not particularly successful. This problem arises in practice in several industrial contexts such as footwear, metalware and furniture. A possible approach is the periodic placement of the shapes, in a lattice way. In this paper, we propose three heuristic approaches to solve this particular case of nesting problems. Experimental results are compared with published results in literature and additional results obtained from new instances are also provided.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Packing; Periodic packing; Lattice packing; Heuristics

1. Introduction

Nesting has been widely studied (Dowsland and Dowsland, 1992), as it is a common problem in several industrial production processes (textile, garment, footwear, metalware, furniture, etc.). Most of the published papers impose orientation constraints: the shape can only be rotated a predetermined number of fixed angles (the most common is 180° and, less commonly, 90°) or it cannot be rotated at all. However, in certain particular cases, orientation constraints do not arise. Additionally, some industries frequently need to cut big quantities of each type of shape. In these two situations, the periodic packing of the shapes (lattice packing) proved to be a very competitive approach, because:

– good cutting patterns, with very small waste, can be obtained;
– the automatic cutting process is more efficient when a periodic placement is considered;
– reduced computational time (when compared with general nesting approaches, where each piece has to be treated individually), which allows to tackle problems of greater dimension.

In this section, we start by describing the problem and refer to related works in a brief review. In Section 2, we point out the geometric considerations necessary to understand this work. Section 3 describes the underlying rules and criteria used to build the three heuristic approaches, which are proposed in the fourth section. In Section 5, the instances used to evaluate the quality of the proposed approaches are defined and the results obtained with all...
the three heuristics are presented and compared. Finally, in the last section, we draw conclusions and make some comments about future work.

1.1. Problem description

The periodic placement of congruent copies of one shape can be found in literature under the keywords “periodic packing”, “regular packing” or “lattice packing”. The lattice packing may be defined over an infinite plane or a limited sheet. If it is the plane, the main objective is to regularly place the shapes so that the density obtained is maximised. When lattice packing on a limited stock sheet, the goal is to maximise the percentage of stock sheet utilisation, which is equivalent to maximising the number of shapes placed inside the plate. As will be shown later, these two “apparently” similar objectives are not quite equivalent. However, the definition of a lattice is identical in the two situations. A lattice is generated by two linearly independent vectors, \( \mathbf{v}_0 \) and \( \mathbf{v}_1 \), whose linear combinations will produce a grid that guarantees that pieces do not overlap. The shapes are located in each node of the grid.

The problem we propose to solve does not consider orientation constraints, so, any rotation of the given shape can be allowed. However, after a specific orientation is chosen, all the shapes must follow the same orientation (Single-Lattice – Fig. 1a) or be alternately rotated 180° (Double-Lattice – Fig. 1b). According to the typology by Wäscher et al. (2007), the problem described in this paper is a 2-dimensional irregular IIPP (Identical Item Packing Problem).

The orientation of the shapes is not relevant when lattice packing on the plane, as maximum density remains unchanged. However, when stock sheet borders need to be taken into account, the orientation of the shapes may be extremely important as will be shown in the following sections.

In this work, we propose to solve the problem of Periodic Packing of Irregular Shapes (PPIS) on a limited stock sheet (a rectangle) by heuristic methods.

1.2. Related work

There are references in the literature to the Lattice Packing Problem since the fifties, the first results concerning only theoretical aspects of the lattice packing of a single convex polygon on the plane (Rogers, 1951). However, only a few publications during the next 30 years can be found, and some of them are just marginally related to lattice packing. O’Rourke et al. (1986) have considered the problem of finding minimal enclosing triangles and Dorf and Ben Bassat (1983) have dealt with the problem of finding minimal enclosing convex polygons with fewer sides. Those approaches can be seen as a pre-processing to lattice packing, as regular convex polygons can be easily packed than the original ones.

Between 1990 and 2000, essentially theoretical results on lattice packing have been published, mostly considering unlimited sheets and convex shapes (Mount and Silverman, 1990; Mount, 1991; Kuperberg and Kuperberg, 1990; Stoyan and Patsuk, 2000).

More recently, results concerning real industrial problems, with limited sheets, irregular shapes and no orientation constraints, have been published. Prasad et al. (1995) describe a nesting problem of irregular shaped metal blanks and propose a heuristic search, considering the variation in the orientation of the blanks on the stock sheet. In Cheng and Rao (1999), Cheng and Rao (2000), Milenkovic (2002), heuristic algorithms have also been proposed, to solve problems concerning real industrial irregular shapes and limited sheets. Simplifications of a mathematical model are described and solved to optimality by Stoyan and Pankratov (1999), being the presented results related to an application in the footwear industry.

When lattice packing on the plane, the vectors \( \mathbf{v}_0 \) and \( \mathbf{v}_1 \) must be chosen (Fig. 2) so that the density of the packing is maximised (Milenkovic, 2002), what corresponds to minimising the area of the parallelogram defined by the two vectors, \( \min|\mathbf{v}_0 \times \mathbf{v}_1| \). When lattice packing on a limited stock sheet, the vectors \( \mathbf{v}_0 \) and \( \mathbf{v}_1 \) must be chosen so that the number of shapes entirely placed inside the stock sheet is maximised. However, maximising the percentage of stock sheet utilisation does not always correspond to a maximum density lattice (as if there were no borders). This can be shown with the simple example given in Fig. 2. It can be easily seen that the minimum area of the parallelogram criterion allows only to pack three triangles (the fourth is only partially inside the stock sheet), while a bigger area of the parallelogram allows to pack a fourth triangle within the same stock sheet. Certainly this example shows an extreme situation, but it perfectly illustrates this problematic situation.
In general, published papers address either the maximisation of the percentage of the stock sheet utilisation (in the case of a limited stock sheet) or the maximisation of the density of the lattice (in the unlimited situation or, approximately, in the case of very large sheets). The major difference innovation in this paper is that proposes heuristics to tackle both objectives when solving the PPIS problem on a limited stock sheet.

2. Geometric considerations

All the shapes are represented by polygons. Arcs are approximated by a set of exterior tangents to the curve and holes inside the shapes are not considered. A polygon is represented by a clockwise sorted list of vertices and each polygon has its own reference point. Polygons must not overlap and must be entirely placed inside the stock sheet. To guarantee that two polygons do not overlap and are positioned as close as possible, the concept of no-fit-polygon (NFP) (Bennell et al., 2001) is used. To generate the NFP between two polygons \( P_1 \) and \( P_2 \) (NFP\(_{12} \)), the first polygon \( (P_1) \) is firstly positioned on the plane and is considered as the fixed piece. The polygon \( (P_2) \) is called the tracing polygon and its movement, while sliding around the boundary of the fixed piece and marking the locus of the reference point, forms a closed path: the no-fit polygon NFP\(_{12} \) (Fig. 3). The relative orientation of the two pieces is kept fix and they are always in touch without overlapping.

To ensure that any shape is entirely placed inside the stock sheet, we use the concept of inner-fit-polygon (IFP) (Gomes and Oliveira, 2002). The IFP is the geometric place of all the points where the reference point of the piece to place can be positioned, so that the piece can be completely placed inside the stock sheet. When \( P_2 \) slides along the internal contour of \( P_1 \), the IFP\(_{12} \) is obtained. As the stock sheet is a rectangle, the IFP is also a rectangle.

The complete description of how those polygons are generated can be found in the above cited papers. The computation of NFPs is a very time consuming operation for non-convex polygons. As a new NFP needs to be recomputed whenever a new relative orientation between two polygons is considered, the complexity of this operation must be taken into account when developing lattice heuristics.

The objective of the PPIS problem is to place the largest number of congruent irregular shapes inside a limited rectangular stock sheet, where all the copies of the shape have the same orientation and the generated layout forms a replication scheme, i.e. a lattice. In its most general formulation, there are no constraints on the selection of the shape orientation used to build the lattice layout and, in fact, different orientations should be tested in order to find the one that takes more advantage of the stock sheet’s borders.

To build a lattice layout for the PPIS problem, it is necessary to define two parameters:

- the orientation of the shape \( (\alpha) \), i.e. the rotation angle relative to the original orientation;
- the pair of vectors that generate the lattice \( (v_0 \text{ and } v_1) \). When considering a limited stock sheet, a third parameter must be introduced;
- the placement point of the first shape positioned \( (P_0) \).

The lattice is then defined by \( L(v_0, v_1) = \{i_0 v_0 + i_1 v_1 : i_0, i_1 \in Z\} \) such that copies of the shape will be expanded to fill the entire stock sheet. Only copies that are completely placed inside the stock sheet are considered. The vectors \( v_0 \) and \( v_1 \) are chosen so that shapes do not overlap and the stock sheet utilisation is the best possible, or, identically, the number of placed pieces is maximised. Two examples of different lattice layouts obtained with the same shape, by changing the rotation angle and the placement point of the first shape, are presented in Fig. 4.

One of the problems arising in all types of nesting problems is the intrinsic difficulty to deal with geometry, as usually shapes are not regular and even not convex. Another problem, specific of the PPIS problem, is that the best lattice scheme on the plane, i.e. with an unlimited stock sheet, may not be the best choice when we have to fill a stock sheet (with borders). But even the consideration of the best lattice on the plane is not an easy problem, as it involves the determination of the vectors \( (v_0, v_1) \) that minimise the area of the parallelogram formed by the four vertices: origin, \( v_0 \), \( v_1 \) and \( v_0 + v_1 \), known as the fundamental parallelogram. The objective function \( \min|v_0 \times v_1| \) of this problem is neither linear nor convex and so its solution is not trivial. Besides this difficulty, the necessity of considering all the
non-overlapping constraints naturally leads to the consideration of heuristic procedures.

3. The common framework

Solving the PPIS problem on a limited stock sheet implies the determination of the parameters that completely define a lattice layout. Once different orientations are allowed, a rotation angle \( \alpha \) is first applied to the piece to be replicated. After that, the lattice heuristic iteratively calculates the other parameters that define a lattice layout: the pair of vectors \((v_0, v_1)\) and the location of the initial point \(P_0\). This means that the main problem is approached under a hierarchical framework. Therefore, in each iteration (for each angle \( \alpha \)) the lattice heuristic starts by determining the generating vectors and only afterwards the initial point, so that the highest number of shapes that can be packed inside the stock sheet is obtained. In the Double-Lattice case, there is an additional step that is to decide the best way of merging the two shapes, so that a merged-piece is created. After that, all the iterative process is identical to the Single-Lattice case, but applied to the new merged-piece. The general structure of the lattice heuristic is represented in Fig. 5. Its main components will be described in the following sections.

Several rules and criteria to deal with each one of those sub-problems were extensively tested, so that one could be chosen for the new heuristic final implementation.

3.1. Merge rules

These rules are only used for the Double-Lattice case. A Double-Lattice is a lattice of two shapes, where the second shape is the first one rotated by 180°. If rotating one given shape of 180° is trivial, it is not so obvious the relative position of the two shapes, so that the “merge” of the two shapes (the merged-piece) produces a good layout when replicated in a lattice way. Usually, in real situations, a Double-Lattice leads to good layouts as the two shapes, the one with the original rotation and the one rotated by 180°, fit very well together, i.e. when merged they produce a compact shape with little waste inside. In this case, it is therefore necessary to create a new merged-piece and to ensure that the best relative position of the two pieces has to be found.

A set of criteria, introduced by Oliveira et al. (2000) were considered as possible merge rules:

- **MIN_AREA (MA)** – minimisation of the area of the merged-piece enclosing rectangle;

![Fig. 4. Different orientations of a given shape lead to different utilisations of the stock sheet.](image)

![Fig. 5. Lattice heuristic general structure.](image)
• MIN_LENGTH (ML) – minimisation of the length of the merged-piece enclosing rectangle;
• MAX_OVERLAP (MO) – maximisation of the overlap between the two enclosing rectangles of the two individual shapes, while not overlapping the pieces themselves;

The application of each criterion is illustrated in Fig. 6, where the resulting merged-pieces are also presented.

The merge rule is a time consuming operation since it implies the recomputing of the NFPs. It should also be mentioned that the merged-piece depends on the rotation considered for the base polygon.

3.2. Lattice criteria

Considering the existence of a limited stock sheet, different criteria can be used to calculate the vectors that will generate the lattice, called the Lattice Criteria. Three lattice criteria have been tested:

• MIN_AREA_PARALLELOGRAM (Ø) – the choice of the pair \((v_0, v_1)\) is made as if the stock sheet was the plane, that is, the borders of the stock sheet are not taken into account;
• X_PARALLEL (X∥) – \(v_0\) is fixed parallel to the horizontal border of the stock sheet; \(v_1\) is determined so that the third piece placed stays in contact with the first and the second ones already placed;
• Y_PARALLEL (Y∥) – \(v_0\) is fixed parallel to the vertical border of the stock sheet; \(v_1\) is determined so that the third piece placed stays in contact with the first and the second ones already placed.

In the first criterion (Ø), the non-linear objective of minimising the area of the fundamental parallelogram is achieved by a greedy iterative heuristic. This heuristic uses the concept of NFP and iterates until a pre-defined number of iterations without any improvement is reached. The reader should refer to Fig. 7 in the following explanation. The first replica \((P_1)\) is placed and taking a second replica \((P_2)\) as the tracing polygon, the NFP12 is computed. Then, one edge of the NFP12 is chosen to be the one which \(P_2\) will be in touch in some point. Once \(P_2\) is positioned (\(v_0\) is fixed), the third replica \((P_3)\) is easily placed, as it has to touch simultaneously \(P_1\) and \(P_2\) (\(v_1\) is fixed), which guarantees that the rest of the pieces do not overlap and the area of the parallelogram formed by the two vectors \((v_0\) and \(v_1)\) is no greater than what is strictly necessary, given \(v_0\). This is achieved by arbitrarily choosing one of the intersection points of the two NFPs involved: the first one (NFP13) defining the feasible positions for the third piece, when compared against the first one and the second one (NFP23) defining the feasible position of the third piece, when compared against the second one. In Fig. 7 it is possible to see the position of the three pieces that determines the vectors \((v_0, v_1)\). The area of the fundamental parallelogram is calculated and, if it is lesser than the best value kept until that moment, it is updated. Otherwise, the number of iterations without improvement is incremented and another point of contact in the edge under analysis is chosen. NFP12 is equal to NFP13 as \(P_2\) is equal to \(P_3\). This process is repeated over the edges of NFP12. It should be noticed that only half of the edges has to be searched given the NFPs symmetry, which is due to the congruency of the pieces involved.

Two other criteria were considered to take into account the existence of a rectangular stock sheet and aiming to take advantage from the borders. For instance, in the X∥ criterion, after placing the first piece, the second piece is placed with the same value of \(y\) coordinate and touching the first piece (\(v_0\) is determined), while the third piece, as
in the previous criterion, is positioned so that it touches simultaneously the two already positioned pieces ($v_1$ is determined). The same occurs in $Y// criterion, exchanging $x$ and $y$ coordinate roles.

3.3. The choice of the initial point $P_0$

The admissible values for the lattice start position $P_0$ are all the points that allow the first piece to be placed completely inside the board. To generate such set of points, we use the concept of IFP. However, we can restrict the search to a restricted subset of points, because, after the first piece, everything will be repeated. The search has therefore been restricted to an area that corresponds to the rectangular enclosure of the irregular shape already rotated by a particular angle $z$, as it is obvious that, at least one piece, must be as closer as possible to a corner of the board. We place that rectangular enclosure, divided in four parts, each one being in touch with one corner of the IFP (Fig. 8). For each corner, the search changes iteratively and randomly the value of $P_0$ coordinates.

3.4. The angle list

The first and second heuristics proposed in this paper include searching, in a list of angles, the angle that leads to a good lattice layout. The way the list of angles is created is what distinguishes this two heuristics. This will be described in detail in the next section.

4. The three heuristic approaches

4.1. The parallel angles heuristic (PAH)

This heuristic is based on a finite list of candidate rotation angles. This list is constructed with the most promising rotation angles, in what concerns the good usage of the board edges. To obtain those angles, the shape is rotated so that one edge of its convex hull is parallel to one of the borders of the stock sheet. The angle list is created and the lattice heuristic is applied to all angles in the list (Fig. 9).

After the list is exhaustively searched, a post-optimisation phase is applied. This post-optimisation is no more than a local search procedure, which is only applied to the $n\%$ angles with better results of the list. It consists of searching around each one of the base angles, by successively adding very small random increments. It ends after a predefined number of consecutive non-improvement movements occur. The same search process is repeated in decrementing direction. The structure of the algorithm is presented in Fig. 10. It should be noted that, for each small change in the value of the angle, all NFPs need to be recalculated and the lattice heuristic applied.

4.2. The equally spaced angles heuristic (EAH)

This approach was developed to overcome what seemed to be some disadvantages of the PAH, namely:

- the dimension of the list is not known beforehand, as it depends on the number of edges of the shape’s convex hull;
- the computational time depends strongly on the dimension of the list, as the search is exhaustive;
- there is no guarantee that such a list covers all the range of angles.

Equally spaced rotation angles allows to generate an angle list that covers all range, as the search is exhaustive. Additionally, the computational time is easily controlled by an adequate choice of the rotation angle step.

The same local search described in the PAH is also performed, as a post-optimisation phase applied to the $n\%$ better angles of the list. The structure of the EAH algorithm is identical to the structure of the PAH.

4.3. The free angles heuristic (FAH)

This approach uses a high level search procedure – Iterated Local Search (ILS) – to guide the search over the solu-
4.3.1. Initial solution

The initial solution $s_0$ is the result of the lattice heuristic run with an initial rotation angle $\alpha_0$ randomly chosen.

4.3.2. Neighbourhood structure

The search for the value of the rotation angle $\alpha$, guided by an ILS algorithm, relies on the low-level iterative heuristic – the lattice heuristic – to calculate the other parameters: the vectors $(v_0, v_1)$ and the initial point $P_0$. The neighbourhood structure is based on changing the rotation angle $\alpha$ by two types of movements: a first one that introduces a big change on the rotation angle $\alpha$, used in the perturbation phase, and a second one that applies a small variation to the current rotation angle, which is used in the local search phase. However, the implementation of these movements needs to be carefully designed due to the time consuming operations that have to be performed, to ensure the geometric feasibility of the layouts, each time a new rotation angle $\alpha$ is considered. To reduce the computational time, not all time consuming operations are performed in the local search phase for the Double-Lattice approach: the merged-piece is only computed after a perturbation takes place, when the current solution is updated, instead of being computed each time a small variation of the rotation angle is applied. In these situations, the new merged-piece is not calculated, it is approximated by the merged-piece already computed for the current main rotation angle, rotated by the small variation. After a perturbation, a big variation of the rotation angle is applied and therefore the complete lattice heuristic needs to be applied, including the determination of the best way of merging together one piece and its $180^\circ$ rotated counterpart.

4.3.3. Acceptance criterion and stop condition

In this ILS implementation, any solution generated by the local search phase is accepted, i.e. the perturbation is always applied to the most recently visited local optimum. This acceptance criterion clearly favours diversification, allowing a convenient freedom on the selection of the crucial rotation angle $\alpha$. The algorithm stops after a fixed amount of iterations or after a predefined number of iterations without improvements.

5. Computational experiments

5.1. Instances used

Seven test instances were used to evaluate and compare the performance of the three new heuristic approaches. The description of the first two test instances, Cheng and Stoyan, is available respectively in Cheng and Rao (1999) and Stoyan and Pankratov (1999). Pieces coordinates and stock sizes are completely described in the papers referred. However, it is convenient to specify that, in Cheng instance case, the stock sheet considered is a square of edge length equal to 200. Though both instances have non-convex shapes, they present big differences in other two important characteristics: the number of vertices of the irregular shapes (35 for the Cheng, 17 for the Stoyan) and, specially, the ratio between the stock sheet’s area and the shapes’ area (534 for the Cheng, 93 for the Stoyan).

Unfortunately, we could not find other results published in literature concerning this problem. So, and for future result comparison, five other instances were created to test the new heuristics. These instances (Swim1 to Swim5) were based on the nesting data set SWIM (Oliveira et al., 2000) available in ESICUP (the EURO Special Interest Group on Cutting and Packing) web site http://www.fe.up.pt/esi-cup/. The boards dimensions have been inspired in the strips used by Oliveira et al. (2000), that have their origin in real-world industrial processes, which height, measured in inches, leads to the value of 5752 mm.

The computational tests were run on a personal computer with a Pentium 4 processor at 3.0 GHz and 1 GB of RAM.

5.2. Results

Preliminary tests were run for each variant of the algorithm, to adjust the values of some parameters and to compare the performance of the proposed criteria. The number of shapes placed inside the stock sheet is used to measure the quality of the layouts.

5.2.1. Preliminary tests

The main objective of the preliminary tests was to assess the relative performance of the several rules and criteria presented in the previous sections, so that a few of them could be chosen for the more intensive computational experiments. Simultaneously, the tuning of some parameters embedded in each heuristic procedure, inherent to any iterative process, was also made. For these preliminary tests, the three heuristic approaches were run but only instances Cheng and Stoyan were considered. After testing a thorough combination of approaches, rules, criteria and values of embedded parameters (several number of
iterations without improvement have been tried, several steps between fixed angles, in EAH approach, have been tested), one combination has been chosen to present some comparative results and to take conclusions concerning the performance of the several rules and criteria.

As the relative performance of the different rules and criteria has shown to be the same, independently of the heuristic approach used (PAH, EAH or FAH), the computational results of the preliminary tests will only be presented for the EAH heuristic. The comparative results are summarised in Table 1. The tests that lead to these results were obtained by applying EAH approach, with the number of iterations without improvement equal to 10, a local search phase applied to the best 10% of main angles, a smallest angle value of 0.1° and an angle step of 5°. In Table 1, the second column presents the best result published. The best result obtained from 10 runs of each instance, with Single-Lattice and Double-Lattice, is marked in boldface. Since the stock sheet considered for the Cheng instance is a square, we did not include the Y// lattice criterion as it results equivalent to the X//.

By analysing Table 1, we can reach some interesting conclusions. First, comparing the results obtained with the three different lattice criteria (◇, X// and Y//), it is obvious that our initial intuition was right: the PPIS problem on a limited sheet should be treated differently from the traditional lattice problem. In fact, ◇ criterion (minimisation of the area of the fundamental parallelogram) never produces better results than the other two criteria and it is more time consuming. That conclusion is more evident for Stoyan instance, where the ratio stock sheet area/piece area is small, and is less visible for Cheng instance, where that ratio is big. When the stock sheet is too big, compared with the dimension of the piece, there is the tendency to behave as if the borders were not there.

Due to the iterative nature of the positioning of the second piece, the ◇ criterion is very time consuming, while for the X// and Y// criteria, the positioning of the second piece is immediately determined. This happens because the lattice criterion is repeatedly called, so, it has a significant influence on the final computational time. The difference between the performance of X// and Y// mainly depends on the dimensions of the stock sheet, i.e., if the length is greater than the width, the X// seems allowing to reach better results and the opposite also holds.

When looking for differences between results obtained with the three merge rules proposed for the Double-Lattice case, we can also conclude that MA (minimising the area of the enclosing rectangle) is the criterion with the best behaviour, obtaining (almost) always the best layouts.

5.2.2. Computational results

After the conclusions drawn from the preliminary tests we have selected some criteria and discarded others. There-

<table>
<thead>
<tr>
<th>Lattice type</th>
<th>Instance name</th>
<th>Published result</th>
<th>Lattice criteria</th>
<th>Merge rule</th>
<th>Number of placed shapes</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best</td>
</tr>
<tr>
<td>Single</td>
<td>Cheng</td>
<td>399a</td>
<td>◇</td>
<td>–</td>
<td>411 408.2 404 2.9</td>
<td>493.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X//</td>
<td>–</td>
<td>417 411.1 404 4.7</td>
<td>434.3</td>
</tr>
<tr>
<td></td>
<td>Stoyan</td>
<td>75b</td>
<td>◇</td>
<td>–</td>
<td>72 70.2 69 1.3</td>
<td>71.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X//</td>
<td>–</td>
<td>75 75.0 75 0.0</td>
<td>18.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y//</td>
<td>–</td>
<td>75 74.4 72 1.3</td>
<td>18.4</td>
</tr>
<tr>
<td>Double</td>
<td>Cheng</td>
<td>–</td>
<td>◇</td>
<td>MA</td>
<td>450 449.0 444 2.2</td>
<td>580.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ML</td>
<td>450 448.0 440 3.0</td>
<td>595.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MO</td>
<td>438 436.0 426 4.3</td>
<td>537.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X//</td>
<td>MA</td>
<td>450 450.0 450 0.0</td>
<td>426.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ML</td>
<td>448 440.0 436 3.3</td>
<td>445.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MO</td>
<td>428 426.2 424 1.8</td>
<td>402.2</td>
</tr>
<tr>
<td></td>
<td>Stoyan</td>
<td>80b</td>
<td>◇</td>
<td>MA</td>
<td>76 76.0 76 0.0</td>
<td>181.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ML</td>
<td>70 70.0 70 0.0</td>
<td>185.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MO</td>
<td>76 76.0 76 0.0</td>
<td>192.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X//</td>
<td>MA</td>
<td>80 77.8 76 2.0</td>
<td>48.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ML</td>
<td>76 74.4 72 2.1</td>
<td>47.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MO</td>
<td>80 76.2 72 2.6</td>
<td>50.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y//</td>
<td>MA</td>
<td>80 79.2 72 2.5</td>
<td>48.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ML</td>
<td>72 72.0 72 0.0</td>
<td>53.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MO</td>
<td>80 77.4 70 4.2</td>
<td>49.8</td>
</tr>
</tbody>
</table>

Note: results obtained with EAH approach and an angle step of 5°.

b Stoyan and Pankratov (1999).
fore, the final tests were run applying only lattice criteria $X//\text{ and } Y//$, as the $e$ criterion has proved to behave worse. Also, in Double-Lattice case, the MA merge criterion has been chosen.

Tables 2 and 3 resume the computational results obtained, presenting both the best and the worst result, as well as the average of the results for each instance and for each heuristic approach used. Those results were obtained from 10 runs of each combination of criteria. The step between the angles, for the EAH approach, has been set equal to $\frac{5}{216}$. In the iterative procedures, the number of iterations without improvement has been set equal to 15. The best result obtained for each instance, Single-Lattice or Double-Lattice, is marked in boldface. In PAH and EAH approaches, all the tests have included a post-optimisation local search around the 10% best angles. In some cases, the local search did not improve the results previously obtained, and those cases are identified in Tables 2 and 3 with (*). The percentage of total computational time spent in the local search phase is high, between 50% and 80% of total time, depending on the instance, but it improved the results in most of the cases.

A comment is necessary on the best result for Cheng instance (Cheng and Rao, 1999). The value 464 was obtained only once, with a special rotation, a particular seed and a local search procedure around that angle value. The most common best value was only 452.

In the Appendix, the best layouts obtained for all the instances are presented.

The tables show computational results for instances Cheng and Stoyan. The tables include columns for the instance name, lattice type, published result, heuristic approach, lattice criteria, number of placed shapes, and time (seconds).

### Table 2

<table>
<thead>
<tr>
<th>Instance name</th>
<th>Lattice type</th>
<th>Published result</th>
<th>Heuristic approach</th>
<th>Lattice criteria</th>
<th>Number of placed shapes</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best</td>
<td>Average</td>
</tr>
<tr>
<td>Cheng</td>
<td>Single</td>
<td>399</td>
<td>PAH</td>
<td>$X//$</td>
<td>418</td>
<td>414.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EAH</td>
<td>$X//$</td>
<td>418</td>
<td>417.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAH</td>
<td>$X//$</td>
<td>418</td>
<td>415.3</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>–</td>
<td>PAH</td>
<td>$X//$</td>
<td>464</td>
<td>452.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EAH</td>
<td>$X//$</td>
<td>452</td>
<td>450.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAH</td>
<td>$X//$</td>
<td>450</td>
<td>448.1</td>
</tr>
<tr>
<td></td>
<td>(200 $\times$ 200)$^a$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stoyan</td>
<td>Single</td>
<td>75</td>
<td>PAH</td>
<td>$Y//$</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EAH</td>
<td>$X//$</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAH</td>
<td>$X//$</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>80</td>
<td>PAH</td>
<td>$X//$</td>
<td>80</td>
<td>79.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EAH</td>
<td>$X//$</td>
<td>80</td>
<td>79.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAH</td>
<td>$X//$</td>
<td>80</td>
<td>76.0</td>
</tr>
<tr>
<td></td>
<td>(900 $\times$ 700)$^b$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


$^b$ Stoyan and Pankratov (1999).

$^c$ Board dimensions.

$^d$ Local search did not improve the solution from which has started.

The evolution of EAH’s performance against stock dimensions

To better evaluate the performance of any lattice algorithm, Cheng and Rao (1999) have established a Universal Compact Yield (UCY). This unit of measure indicates the maximum compactness that is possible to obtain with the chosen shape, when there are no constraints on the stock sheet size. In other words, it measures the yield that corresponds to the maximum density obtainable with a particular shape on the plane. UCY provides a bound to assess the quality of the results obtained with any algorithm. Taking into account their proposed value of UCY for the Cheng instance, when it is replicated in a Single-Lattice way (83.07%), we have varied the edge length of the squared stock sheet from 25 until 500. The comparative results are illustrated in Fig. 12.

Unfortunately, we could not include in the same graphics the curve presented by Cheng and Rao (1999), as we have not the exact values of the point coordinates along that curve. However, as the authors have published (Cheng and Rao (2000)) the yield obtained with the application of their Compact Neighbourhood Algorithm (CNA) for a square stock sheet of edge length of 200 (which is 73.9% against our value of 77.98%), it is easy to conclude, by visual inspection (Cheng and Rao (1999, Figure 10)), that the curve presented in Fig. 12 is always above the published curve. Below the value of 200, the difference is higher but above 200 the two curves become closer when the dimensions increase. The biggest difference is for small dimensions of the stock sheet. For instance, considering a square of edge length of 25, our value is almost 48 and the published value is under 40. For an edge of 50, our value is over 71 and published value is under 60. As the stock size increases, the difference becomes smaller and looks worthless near 500, which means that both approaches converge to UCY.

After this study, we are able to conclude that EAH approach outperforms the results obtained by the application of CNA algorithm proposed by Cheng and Rao (1999), independently of the stock sheet dimension. This
is achieved because our approach deals simultaneously with the maximisation of the density of the lattice and the maximisation of the percentage of stock sheet utilisation.

Table 3
Computational results: instances Swim1 to Swim5

<table>
<thead>
<tr>
<th>Instance name</th>
<th>Lattice type</th>
<th>Heuristic approach</th>
<th>Lattice criteria</th>
<th>Number of placed shapes</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Best</td>
<td>Average</td>
<td>Worst</td>
<td>Std. dev.</td>
</tr>
<tr>
<td>Swim1</td>
<td>Single</td>
<td>PAH</td>
<td>X/</td>
<td></td>
<td>192</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>X/</td>
<td></td>
<td>195</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>195</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>PAH</td>
<td>X/</td>
<td></td>
<td>247</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>X/</td>
<td></td>
<td>245b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>244</td>
</tr>
<tr>
<td>Swim2</td>
<td>Single</td>
<td>PAH</td>
<td>Y/</td>
<td></td>
<td>96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>Y/</td>
<td></td>
<td>96b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>PAH</td>
<td>X/</td>
<td></td>
<td>107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>Y/</td>
<td></td>
<td>106</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>116</td>
</tr>
<tr>
<td>Swim3</td>
<td>Single</td>
<td>PAH</td>
<td>X/</td>
<td></td>
<td>100b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>X/</td>
<td></td>
<td>100b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>PAH</td>
<td>X/</td>
<td></td>
<td>116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>X/</td>
<td></td>
<td>116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>116</td>
</tr>
<tr>
<td>Swim4</td>
<td>Single</td>
<td>PAH</td>
<td>X/</td>
<td></td>
<td>202</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>X/</td>
<td></td>
<td>202</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>202</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>PAH</td>
<td>X/</td>
<td></td>
<td>245</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>X/</td>
<td></td>
<td>247</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>246</td>
</tr>
<tr>
<td>Swim5</td>
<td>Single</td>
<td>PAH</td>
<td>Y/</td>
<td></td>
<td>234b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>X/</td>
<td></td>
<td>234b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>240</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>PAH</td>
<td>Y/</td>
<td></td>
<td>240</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAH</td>
<td>Y/</td>
<td></td>
<td>240</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAH</td>
<td>X/</td>
<td></td>
<td>240</td>
</tr>
</tbody>
</table>

a Board dimensions.

b Local search did not improve the solution from which has started.

6. Conclusions

All three heuristic approaches proposed in this paper outperform the results presented in Cheng and Rao (1999), Cheng and Rao (2000) and match the results presented in Stoyan and Pankratov (1999).

As would be expected, the variants based on the Double-Lattice approach achieve better results than the ones based on the Single-Lattice approach, as one given non-convex shape usually fits well with itself, when rotated 180°. However they also need significantly more time, due to the time consuming merge operation. Concerning the evaluation of the different lattice criteria, the criterion is completely dominated by the others: less placed shapes and bigger computational times, which is explained by the more complex operations needed to minimise the area of the fundamental parallelogram. A similar observation can be made about the merge rules in the Double-Lattice case: the vari-

Fig. 12. Evolution of stock sheet utilisation when changing the dimensions of the square stock sheet, using Cheng instance, Single-Lattice and EAH approach.
ant which uses MA as merge rule always produces better solutions (or sometimes identical) than all the other merge rules.

Comparing the three heuristic approaches, we can conclude that, in general, despite the FAH approach does present good best results, its average results are usually worse. Also, when comparing computational times, this heuristic becomes even less interesting. Although PAH and EAH are similar in structure, EAH is most of the times preferable as it is independent on the number of vertices of the piece, presenting almost always the best results of the computational experiments. Moreover, its average results are also better, what means that it is more robust.

In general, published papers address either the maximisation of the percentage of stock sheet utilisation (in the case of a limited stock sheet) or the maximisation of the density of the lattice (in the unlimited situation or, approximately, in the case of very large sheets). The major contribution of this study is to propose heuristics that handle simultaneously the maximisation of density and percentage of stock sheet utilisation. Computational results show a good performance when solving the PPIS problem on a limited stock sheet, independently of its dimensions.

(a) Single-Lattice (418 shapes).  
(b) Double-Lattice (464 shapes).

Fig. 13. Best lattice layouts obtained for Cheng instance.

Fig. 14. Best Double-Lattice layout obtained for Swim1 instance (247 shapes).

Fig. 15. Best Double-Lattice layout obtained for Swim2 instance (116 shapes).

Fig. 16. Best Double-Lattice layout obtained for Swim3 instance (116 shapes).

Fig. 17. Best Double-Lattice layout obtained for Swim4 instance (247 shapes).

Fig. 18. Best Double-Lattice layout obtained for Swim5 instance (240 shapes).
Appendix

The best layouts obtained in the computational experiments are presented in this appendix. Figs. 13a and b illustrate the best layouts obtained for the Cheng instance: Single-Lattice (418 shapes/77.98% Utilisation) and Double-Lattice (464 shapes/86.56% Utilisation), respectively. The best layouts obtained for the Stoyan instance are illustrated on Figs. 1a and b: Single-Lattice (75 shapes/80.53% Utilisation) and Double-Lattice (80 shapes/85.9% Utilisation), respectively. The last five (Figs. 14–18) concern Double-Lattice layouts for each of the five instances, Swim1 to Swim5: Fig. 14 – Swim1 instance (247 shapes/67.87% Utilisation); Fig. 15 – Swim2 instance (116 shapes/86.28% Utilisation); Fig. 16 – Swim3 instance (116 shapes/87.26% Utilisation); Fig. 17 – Swim4 instance (247 shapes/79.53% Utilisation); Fig. 18 – Swim5 instance (240 shapes/88.19% Utilisation).

References


