An efficient solution method for relaxed variants of the nesting problem

Benny K. Nielsen

Department of Computer Science
University of Copenhagen,
DK-2100 Copenhagen Ø, Denmark
Email: benny@diku.dk

Abstract

Given a set of irregular shapes, the strip nesting problem is the problem of packing the shapes within a rectangular strip of material such that the strip length is minimized, or equivalently the utilization of material is maximized. If the packing found is to be repeated, e.g., on a roll of fabric or a coil of metal, then the separation between repeats is going to be a straight line. This constraint can be relaxed by only requiring that the packing produced can be repeated without overlap. Instead of minimizing strip length one minimizes the periodicity of these repeats.

We describe how to extend a previously published solution method (Egeblad, Nielsen & Odgaard 2006) for the nesting problem such that it can also handle the relaxation above. Furthermore, we examine the potential of the relaxed variant of the strip packing problem by making computational experiments on a set of benchmark instances from the garment industry. These experiments show that considerable improvements in utilization can be obtained.

Keywords: cutting, packing, irregular strip packing, lattice packing, nesting

1 Introduction

The nesting problem generally refers to the problem of placing a number of shapes within in the bounds of some material, typically rectangular, such that no pair of shapes overlap. Moste often, it is also an objective to minimize the size of the material which is equivalent to maximizing the utilization of the material. This problem is also known as the *strip nesting problem* or the *irregular strip packing problem.* Both two- and three-dimensional applications of the nesting problem can be found within a large number of industries. For example, the two-dimensional problem arises when cutting parts of clothes from a roll of fabric. In this case one needs to maximize the utilization of the fabric or equivalently minimize the waste of fabric. This is done by finding the shortest strip of fabric necessary to cut all involved parts of clothes. An example of a solution to such a nesting problem is given in Figure 1. Other two-dimensional applications include sheet metal, glass and animal hide cutting. An example of an application of three-dimensional nesting can be found in the industry of *rapid prototyping.*

Knapsack or bin packing variants of the nesting problem can also be formulated, but in this paper as well as in the majority of related publications, the strip packing variant is the main focus. We are also going to focus on the two-dimensional problem although the problems and the solution method described can be generalized to three dimensions. In the typology of Wäscher, Haussner & Schumann (2006) for cutting and packing problems, we are dealing with a two-dimensional irregular open dimension problem (ODP). The problem is $\mathcal{NP}$-hard even when the shapes and the material involved are rectangles.

In recent years, the strip nesting problem has received a considerable amount of interest. A multitude of solution methods which use various different geometric, heuristic and meta heuristic techniques have been described. Recent surveys are given in the introductions of the papers by Burke, Hellier, Kendall & Whitwell (2006), Gomes & Oliveira (2006), and Egeblad et al. (2006), and these papers also represent some of the most recent efforts to obtain good solutions for the nesting problem.

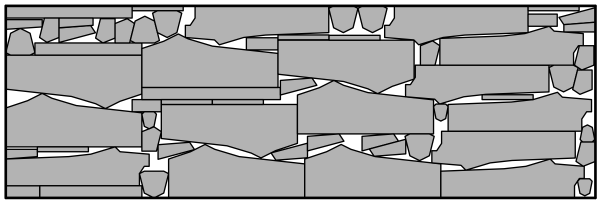


Figure 1: A nesting example of clothing parts for trousers.

This paper concerns a simple variant of the nesting problem based on the following observation. In some cases the solution to a nesting problem is going to be repeated continuously on the material used. For example, consider a nesting solution for a subset of the shapes in Figure 1. If this solution is repeated continuously on a roll of fabric as illustrated in Figure 2a then it becomes clear that the vertical bounds of the rectangles are not really necessary constraints. Relaxing this constraint enlarges the search space and it might be possible to find solutions with better utilization. An example is given in Figure 2b. Relaxing both the vertical and horizontal bounds of the material (see Figure 3) makes the search space even larger. We label these relaxed problem variants *repeated nesting problems.* Note that we picked a subset of the shapes in Figure 1 to better illustrate the idea of repeated patterns, but it might also be a good idea in practice if the nesting problem contains duplicates.

A formal description of repeated nesting problems is given in Section 2 and this is followed by Section 3 which reviews related existing literature. In Sections 4 and 5 we describe our approach to the problem which is an extension of the solution method described by Egeblad et al. (2006). Computational

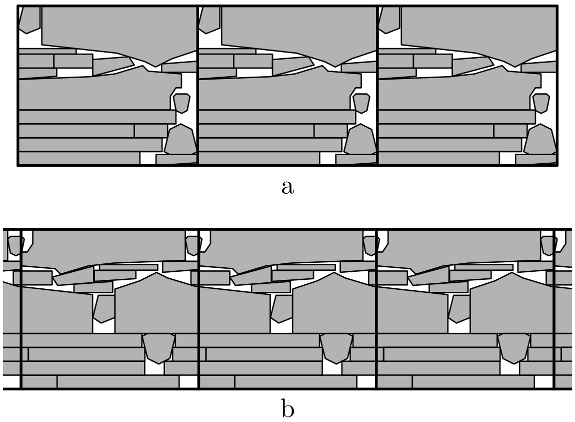


Figure 2: a) A solution to a strip nesting problem might be repeated continuously on a long strip, e.g., a roll of fabric. b) A better solution could be obtained if the straight vertical bounds between repeats are relaxed. In this example utilization went up from 88.68% to 89.74% (based on 10 minute runs with the implementation described in Section 6).

experiments on benchmark problems from the literature are presented in Section 6. Finally, some concluding remarks are given in Section 7.

2 Problem descriptions

The decision variant of the nesting problem can be formulated as follows.

Nesting Decision Problem. *Given a set of shapes and a piece of material with fixed size, determine whether a non-overlapping placement of the shapes within the bounds of the material exists.*

Now, assuming that the material has a rectangular shape with fixed width and variable length then we can state the following problem.

Strip Nesting Problem. *Given a set of shapes and a strip of material with width w, find the minimum length l of the material for which the nesting decision problem has a positive solution.*

If the sum of areas of all shapes is denoted A then the utilization of material for a given solution is $A/(l \cdot w)$. Of course, one is not only interested in the minimum value of l (or the maximum utilization value), but also in a corresponding placement of the shapes.

Note that the use of the words width and length is based on the tradition in the (strip) nesting literature. In some cases it can be confusing, e.g., in the drawings, the visual width of a solution is its length and the visual height of a solution is its width.

The problem definitions above use very general terms. Numerous additional constraints can often be added depending on the specific application of the nesting problem. This includes the type of shapes allowed and to what extent rotation or flipping of shapes is allowed. Note that the decision problem is $\mathcal{NP}$-hard even for rectangular shapes (and material) with no rotation allowed. Thus in most cases one has to settle for heuristic solutions.

In this paper, the shapes are polygons with no self intersections. Holes in the polygons are allowed and although in theory rotation could also be allowed (Egeblad et al. 2006), the current implementation of the solution method presented cannot handle more than a few fixed rotation angles.

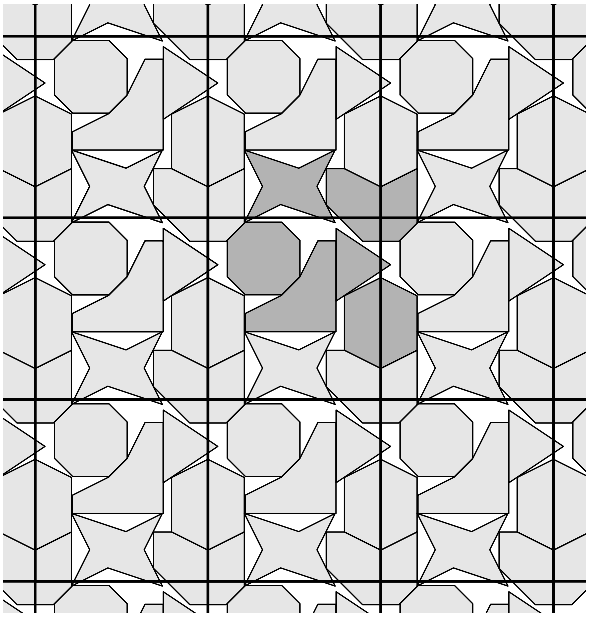


Figure 3: Six polygonal shapes are nested such that they can be efficiently repeated both horizontally and vertically.

The problems dealt with in this paper can be viewed as relaxations of the strip nesting problem. The first relaxation is illustrated in Figure 2b. Here a nesting solution is repeated in one orientation assuming an infinite length of the strip of material. A formal definition of a more general decision problem is as follows.

Repeated Nesting Decision Problem. *Given a set of shapes, a width w and a length l, determine whether a non-overlapping placement of the shapes exists for which any translational copy with offset $(il, jw), i, j \in \mathbb{Z}, (i, j) \neq (0, 0)$, can be made without introducing overlap.*

The values w and l now describe the periods of an infinitely repeated nesting. It is easy to see that such a nesting also has a utilization of $A/(l \cdot w)$. We denote the problem of minimizing $w \cdot l$, the *repeated (pattern) nesting problem*, and if w is a fixed value and repeats in the orientation of w is not allowed, we denote it the *repeated strip nesting problem* (as in Figure 2b).

An interesting alternative visual interpretation of the repeated strip nesting problem is as follows. Given a set of polygons and a cylinder with fixed height, find the minimum diameter for the cylinder such that a non-overlapping placement of the polygons on the cylinder wall exists.

3 Related work

To the best of our knowledge, the problems described above have not previously been described or solved in the existing literature. Therefore, the following is a survey of papers describing solution methods for problems involving some kind of repeated patterns. Note that this includes a more general problem, known as *the densest translational lattice packing of k polygons*, which is described at the end of this section.

Existing literature on nesting problems with repeated patterns has mainly been focused on *point lattice* packings with only one or very few polygons involved. A (point) lattice in two dimensions is the infinite set of points $au + bv, a, b \in \mathbb{Z}$ for two linearly independent vectors u and v. The parallelogram spanned by the vectors u and v is denoted the *fundamental parallelogram*. Now, the *densest translational lattice packing* of a given polygon is defined by a lattice in which a given polygon P can be repeated at every point of the lattice such that no overlap occurs and the area of the fundamental parallelogram, $|u \times v|$, is minimized. This corresponds to maximum utilization of an infinite material.

Given an n-sided *convex* polygon, Mount & Silverman (1990) showed that this problem can be solved to optimality by an algorithm with linear running time. Note that a solution to this problem is also a densest translational packing since Rogers (1964) has shown that such a packing is always a lattice packing. Given a non-convex polygon, Stoyan & Patsuk (2000) solved the problem using a mathematical model, but the result is an algorithm which runs in $O(n^6)$ time (reduced to $O(n^3)$ given a convex polygon). A mathematical model including the constraints of a fixed size rectangular material is presented by Stoyan & Pankratov (1999), but it is a simplified mathematical model and the solutions found for this model are not guaranteed to be optimal. Other heuristic approaches for a non-convex polygon which include the constraints of a rectangular material are described by Cheng & Rao (2000) and Gomes, Costa & Oliveira (2005).

A slightly more general problem is to find the *densest double-lattice packing*. A double-lattice packing is the union of two lattice packings where one lattice is used for the polygon P and the other one is used for a 180 degree rotation of P. Kuperberg & Kuperberg (1990) showed that a densest double-lattice packing of a convex polygon can be constructed by finding a minimum area *extensive parallelogram* within P. Mount (1991) then used this fact to describe an algorithm which can solve the problem in linear time. Kuperberg & Kuperberg (1990) also proved that such a packing would have a density of at least $\sqrt{3}/2 \approx 86.6\%$. Double-lattice packings for a non-convex polygon are also handled by the mathematical model presented by Stoyan & Pankratov (1999) (which includes a rectangular material), but again, optimal solutions are not guaranteed. Gomes et al. (2005) handle double-lattice packings by first pairing P and a 180 degree copy, and then use their lattice packing heuristic on the merged polygon.

Much earlier work by Dori & Ben-Bassat (1984) represents an interesting heuristic approach to the problem of packing a single convex polygon in the plane. They search for a small convex polygon which can both be used to *pave* the plane and to circumscribe P. Numerous possible pavers exist (corresponding to tilings of the plane) and they could include various rotations of P — unfortunately the authors restrict their search to a single type of paver which would make any solution found equivalent to a double-lattice packing. For this problem, the linear time algorithm by Mount (1991) is guaranteed to find optimal solutions. An approach allowing several rotation angles of a non-convex polygon is described by Jain, Fenyes & Richter (1992). A fixed number of copies are rotated and translated in a stepwise search of good solutions which can be repeated in one direction (packing on a strip). During the search, overlap is allowed and to avoid local minima the search is guided by the meta heuristic technique *simulated annealing*. Experiments are only done with 2 and 3 copies of the given polygon. The best solution is with 2 copies where one is a 180 degree variant of the other. This is equivalent to a one-dimensional double-lattice.

Few papers handle problems (with repeated patterns) which include several different polygons and some handle such problems by reducing them to problems only involving one polygon, e.g., Cheng & Rao (2000). The first problem is then to cluster the polygons and to describe this cluster by a single polygon. Afterwards any of the solution methods described above (handling a non-convex polygon) can be used. The hard part is to make a cluster which is both tightly packed and well-suited for repetition.

Milenkovic (2002) has a more direct approach to the problem of handling several polygons in which a mathematical model is used to search for the densest translational lattice packing of k polygons. A solution to this problem can be described as the union of k lattice packings, one for each polygon, all using the same fundamental parallelogram. Note that unlike translational double-lattice packing, no rotation is involved. Milenkovic first reduces the problem by fixing the area of the fundamental parallelogram. A series of decision problems with decreasing area is then solved when trying to minimize the area. A decision problem can then be stated as follows. Given a fixed area α, find a lattice defined by vectors u and v and a set of translation vectors for all polygons, such that $|u \times v| = \alpha$ and no polygons overlap when copied according to the lattice. Milenkovic has also implemented his algorithm and results are presented which handle up to 4 polygons.

The densest translational lattice packing of k polygons (Milenkovic 2002) is the problem that best resembles the problems examined in this paper. The repeated nesting problem as described in the previous section requires the two vectors u and v to be parallel to the x- and y-axes and the repeated strip nesting problem fixes the length of one of the vectors, i.e., the one corresponding to the strip width.

4 Solution method

Some of the best results obtained for the nesting problem have been reported by Egeblad et al. (2006). In the following we give an outline of their approach and discuss how to extend it such that it can also handle repeated patterns.

First of all, the problem of minimizing the strip length is handled separately from the problem of nesting the shapes given a fixed strip length. An initial strip length is found by using a fast placement heuristic of the shapes. The length is then reduced such that the remaining area of the strip is reduced by some fixed percentage. Now the search for a nesting solution is initiated, and if one is found, the strip length is again reduced. If it takes too long to find a solution then a smaller reduction is attempted.

The strip nesting problem has now been reduced to a nesting decision problem. The shapes are either given as polygons or alternatively approximated by polygons and thus, geometrically, the problem is to nest (or pack) a set of polygons within the bounds of a rectangle.

Initially, all polygons are simply placed within the bounds of the rectangle. Most likely, several pairs of polygons overlap and thus the goal is to reduce this overlap iteratively until no overlap exists — corresponding to a solution to the nesting decision problem. The objective function value for this minimization problem is the total amount of overlap in the placement. A local search scheme is used with a neighborhood consisting of horizontal and vertical translations of each polygon. Now, the major strength of this approach is the existence of a very efficient algorithm for finding a minimum overlap horizontal or vertical translation of a given polygon.

The local search reaches a local minimum when no polygon can be translated vertically or horizontally with a decrease in the total amount of overlap. Whenever such a minimum is not zero (which is the known

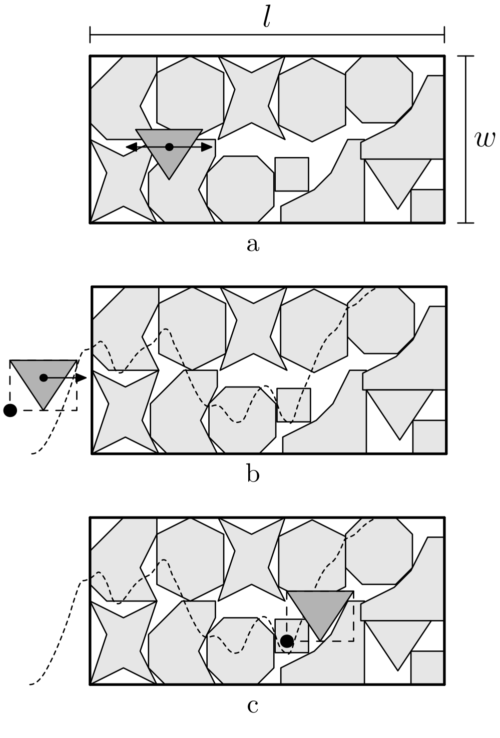


Figure 4: Translation of a polygon (triangle) in a bounded rectangle. a) The triangle has been selected for translation and should be moved horizontally to minimize its overlap with the other polygons. b) The triangle is moved outside the container. The dashed curve illustrates the area of overlap function for which we need to find the minimum. c) Finally, the triangle is moved to the position corresponding to minimum overlap.

global minimum that we are looking for) the search needs to be diversified. The meta-heuristic *guided local search* is very well suited for this purpose. In short, the strategy is as follows. Given a local minimum placement of polygons, the "worst" overlap between a pair of polygons is punished (possibly more than one pair is punished) by adding a penalty to the objective function whenever these polygons overlap. This increases the value of the objective function at the local minimum. At some point it is no longer going to be a local minimum and the local search can continue. More details are described by Egeblad et al. (2006).

To be able to handle repeated patterns we have to extend this solution method. It turns out that the only major change needed is a generalization of the translation algorithm. It needs to be able to take into account that the current placement wraps around at the ends. This is the subject of the following section.

5 Polygon translation with repeats

We need to extend the bounded translation algorithm given by Egeblad et al. (2006) to handle repeated patterns. Without loss of generality we can assume that we are looking for a minimum overlap *horizontal* translation of a given polygon. In the following, we first describe the bounded translation algorithm (with a high-level view) and afterwards we show how it can easily be extended to handle repeated patterns. Note that no changes are needed concerning the low-level details of the algorithm.

In the following we use a very general notion of a *polygon*. Shortly, it can be described as a union of non-intersecting simple polygons for which holes are allowed.

Formally, the bounded (horizontal) translation algorithm solves the following problem.

Bounded Horizontal Translation Problem. *Given a polygon P with a fixed vertical offset y and a polygon Q with a fixed position, find a horizontal offset x for P within some given range $[x_1, x_2]$, such that the area of overlap between P and Q is minimized.*

An example is given in Figure 4a. The emphasized triangle is the polygon P and the union of the remaining polygons is Q. The goal is to find a horizontal position of P which minimizes its area of overlap with all of the other polygons. The lower left corner of a bounding box for P is the reference point when offsetting the polygon, thus the left-most position of P is $x = 0$. If w_P is the width of the polygon P then the range of values for x, that places P inside the rectangle, is $[x_1, x_2] = [0, l - w_P]$.

Intuitively, the translation algorithm first places P to the left of Q (see Figure 4b) and then iteratively moves it to the right while calculating a function which measures the overlap between P and Q. This is a piecewise quadratic function and it is drawn in Figure 4b where the x-coordinate is the position of the reference point of P and the y-coordinate is the amount of overlap. The latter value is scaled such that the maximum is at the top of the rectangle. In Figure 4c the triangle has been moved horizontally to its minimum overlap position.

If P contains n edges and Q contains m edges then this problem can be solved in $O(nm \log nm)$ time (Egeblad et al. 2006). Note that most often $n \ll m$. If the number of edges in P is bounded by some constant then the running time is $O(m \log m)$.

The problems examined in this paper are slightly different and we need to solve a slight variation of the bounded translation problem.

Periodic Horizontal Translation Problem. *Given a polygon P with a fixed vertical offset y and an infinite set of copies of a polygon Q at offsets $(i \cdot l, y), i \in \mathbb{Z}$, find a horizontal offset x within the range $[0, l[$ such that the area of overlap between P and the copies of Q is minimized.*

The range given is just one period of the repeated polygon Q. Given a solution x^*, identical solutions are given at any offset $x^* + i \cdot l, i \in \mathbb{Z}$. Note that P is also going to be repeated at each $x^* + i \cdot l$ and thus it is important that it cannot overlap itself. A simple check can ensure this (proof not given): If there is no overlap between P at offset 0 and a copy at horizontal offset l then no pair of copies of P at offsets $i \cdot l$ and $j \cdot l, i \neq j$, can overlap.

Now, the key observation to be able to extend our bounded translation algorithm is that to correctly evaluate the overlap between P and the copies of Q we only need a finite subset of the copies. In most cases only one or two copies. Assume that the width of Q is w_Q and that its minimum positive offset is at o_Q (horizontally). Given that the width of P is w_P then the following copies of Q are sufficient (maybe not necessary) to guarantee correct overlap calculations in the range $[0, l[$.

$$(i \cdot l, y) \text{ for all } i < 0 : i \cdot l + o_Q + w_Q > 0 \quad (1)$$

$$\text{for all } i \geq 0 : l + w_P > i \cdot l + o_Q \quad (2)$$

Condition (1) includes any copies of Q which has an offset to the left of the range $[0, l[$ while still overlapping the range and condition (2) includes any

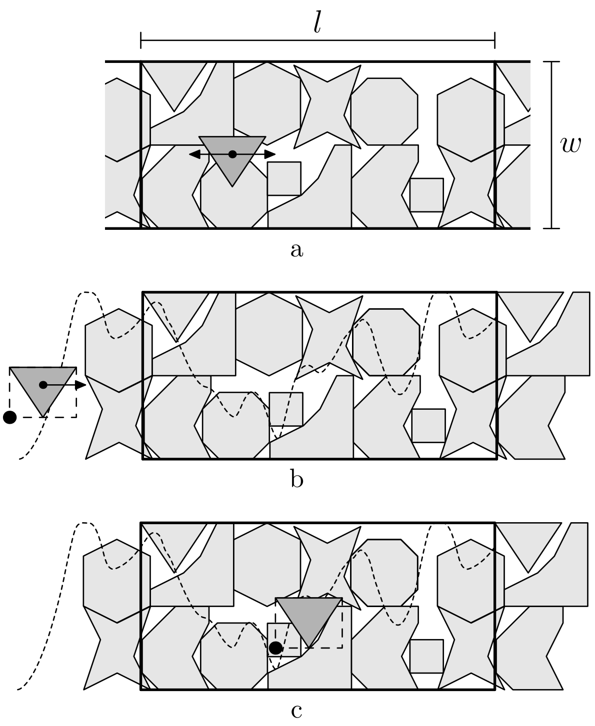


Figure 5: Illustration of the translation of a polygon within a horizontally repeated pattern. a) The triangle has been selected for translation. b) The triangle is moved to the left of the material *and* to the left of any polygons needed to get correct overlap calculations inside the material. Polygons are also included to the right of the container. c) The triangle is moved to the position corresponding to minimum overlap

copies of Q with a positive offset which P might overlap with an offset inside the range. Clearly, the copy of Q with an offset within the range $[0, l[$ is always included by condition (2).

In the above, only one fixed polygon was handled, but if it represents a set of polygons then they could be handled individually when checking the conditions. An example of the periodic translation problem is given in Figure 5a. Again the triangle is moved to the left of the fixed polygons, but it is also moved to the left of any copies of the fixed polygons which can affect overlap calculations while inside the interval $[0, l]$ (condition (1) above). The overlap area function is also calculated for the full range which can require extra copies to the right of the maximum x-coordinate l (condition (2) above). Ensuring these copies is more or less all that is required to handle repeated patterns. In Figure 5b the triangle is moved to the left and the area of overlap function is calculated as before (and drawn in the figure). Finally, in Figure 5c, the polygon is moved to its new minimum overlap position.

The conditions for copies of Q may look overly complicated, but it is necessary since one could be dealing with horizontally very long (and tilted) polygons both regarding P and Q. An extreme example is given in Figure 6 in which a single polygon stretches four repeats.

A few issues remain to be discussed. 1) The description above does not include the penalties used by the guided local search. These are constants which are added to the quadratic overlap function whenever penalized pairs of polygons overlap. This requires some bookkeeping, but it does not affect the worst case running time and no special changes are needed to handle repeated patterns. 2) If both horizontal and vertical repeats are allowed then the copies of Q described are not sufficient since a horizontally translated polygon could overlap with several vertical copies of Q. This is simply handled by adding more copies of Q before translation using conditions similar to those already described. 3) We have not considered what to do with the ends of a roll (or the borders of a plate). It is assumed that the number of repetitions makes this additional waste negligible, but it is possible to solve a special nesting problem to minimize this additional waste. For example, one could treat the final placement as one big polygon (possibly simplifying it) and then solve the normal strip nesting problem using this polygon and all of the original polygons. The big polygon would only be able to move horizontally and the rest would be placed to its left and to its right side within the bounds of the rectangular material.

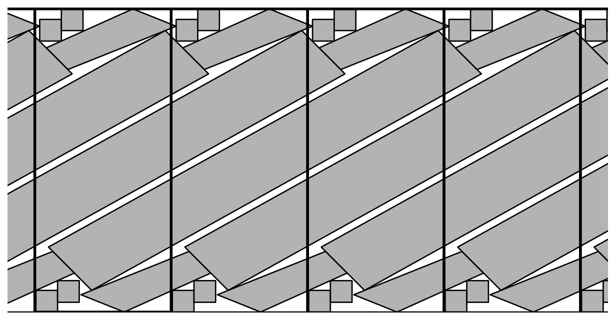


Figure 6: An example showing that a large number of copies of a given polygon can be necessary. Note the single large polygon which stretches four repeats.

6 Computational experiments

The changes needed to handle repeated patterns have been implemented in C++ and incorporated in the nesting program 2DNest (Egeblad et al. 2006). All experiments are done on a machine with a 3GHz Pentium 4 processor.

A set of problem instances has been selected from the literature and downloaded from the ESICUP homepage[1]. All of these instances are taken from the garment industry and thus the results presented should indicate the potential of repeated patterns in realistic settings. Characteristics of these instances can be seen in Table 1. The number of polygons vary between 20 and 99, but the number of unique polygons only vary between 8 and 17. The width ratio is the ratio between the width and the length of a solution with 100% utilization (which most likely does not exist). This value indicates how the rectangle of a solution is going to be shaped. Note that one could easily construct problem instances which would see considerable improvements when allowed to repeat patterns. The problem shown in Figure 6 is one example.

Although 2DNest can work with rotation angles of 90, 180 and 270 degrees, all experiments have been done without rotation. This also means that the results are not directly comparable with results from papers on the strip nesting problem. For this purpose, see the results of 2DNest reported by Egeblad et al. (2006). Here we focus on the improvements obtained by repeated patterns only.

The main results are presented in Table 2. Each instance has first been solved as a standard nesting problem. The average utilization of 10 runs with ran-

[1] `http://www.fe.up.pt/esicup`

Problem instance	Number of shapes	Number of unique shapes	Average number of vertices	Strip width	Width ratio	Origin
Albano	24	8	7.25	4900	0.56	Albano & Sappupo (1980)
Dagli	30	10	6.30	60	1.18	Ratanapan & Dagli (1997)
Mao	20	9	9.22	2550	1.73	Bounsaythip & Maouche (1997)
Marques	24	8	7.37	104	1.50	Marques, Bispo & Sentieiro (1991)
Shirts	99	8	6.63	40	0.74	Oliveira, Gomes & Ferreira (2000)
Trousers	48	17	5.06	79	0.36	Oliveira et al. (2000)

Table 1: Problem instances used in the experiments. The width ratio is the ratio between the width of the problem instance and the length of a solution with 100% utilization.

Instance	No repeats		1D repeat			2D repeat			Waste reduction	
	Average	Max.	Average	Impr.	Max.	Average	Impr.	Max.	1D	2D
Albano	85.02 (0.6)	86.43	85.80 (0.4)	0.78	86.48	86.20 (0.4)	1.18	86.73	5.2	7.9
Dagli	82.05 (0.5)	82.88	83.04 (0.6)	0.99	84.13	83.06 (0.3)	1.01	83.56	5.5	5.6
Mao	78.91 (0.6)	79.54	80.11 (0.5)	1.20	81.02	80.73 (0.4)	1.82	81.54	5.7	8.6
Marques	86.90 (0.4)	87.51	87.05 (0.4)	0.15	87.50	87.38 (0.2)	0.48	87.50	1.1	3.7
Shirts	84.21 (0.3)	84.88	84.94 (0.1)	0.73	85.14	85.19 (0.2)	0.98	85.74	4.6	6.2
Trousers	85.21 (0.3)	85.84	85.60 (0.2)	0.39	86.02	85.98 (0.3)	0.77	86.26	2.6	5.2

Table 2: Results of experiments with three different problems. All values are based on 10 runs of 10 minutes with different seeds. The three problems are strip nesting with no repeats, strip nesting with horizontal repeats (1D) and strip nesting with both horizontal and vertical repeats (2D). Average utilization and maximum utilization are reported for all problems. The improvement in percentage is reported for the two latter problems in comparison with the first problem. The two rightmost columns are the waste reduction in percent (based on the same comparison).

dom seeds are given and these are supplemented with the standard deviation and the maximum utilization found. The same instances have then been solved allowing either horizontal repeats only or allowing both horizontal and vertical repeats. Again, average utilization, standard deviation and maximum utilization are presented in the table. Furthermore, the improvement in percentage points is given. In all cases only the strip length is minimized, i.e., even when allowing both horizontal and vertical repeats the width is still fixed. The best solutions found for the problem with horizontal repeats are shown in Figure 7.

Focusing on the results for horizontal repeats, the improvements vary from only 0.15 up to 1.20 percentage points. This may not seem like a lot, but if one considers the amount of waste in the solutions then these numbers correspond to a 1.1% and 5.7% decrease in waste (second to last column in Table 2). Note also that the best improvements are obtained for the instances with the lowest utilization and vice versa. This indicates that the effect of repeated patterns is greater for “hard” instances. When repeating the pattern both horizontally and vertically the waste reduction is between 3.7% and 8.6%. It would be a reasonably conjecture that small problem instances are better candidates for improvements. This conjecture is supported by the large improvements for the small instances, Albano (24), Mao (30) and Dagli (20), but it is not supported by Marques (24) which is not really improved and Shirts (99) which is more or less improved just as much as the small instances. Nevertheless, given instances that are sufficiently large in the number of polygons involved, the advantage of repeated patterns should decrease.

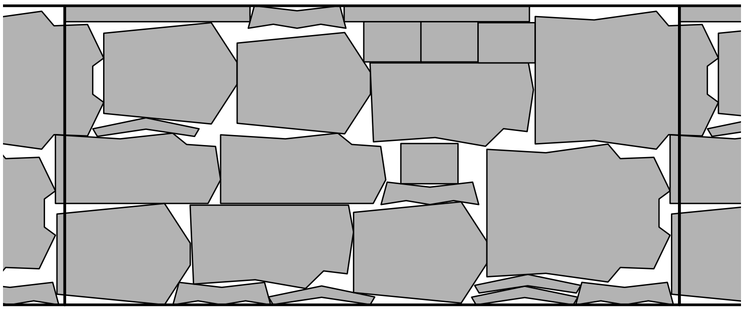
Since we do not find the optimal solutions we cannot really know if the improvements are caused by the existence of better solutions in the larger search spaces, or are caused by a more efficient search for good solutions. It is worth noting though that the search for repeated patterns involves fewer steps than it does for non-repeated patterns. The worst case running time of the translation algorithm did not change when applied to repeated patterns, but it is slower by some constant factor. This is emphasized by the results presented in Table 3. The average number of translations performed per second is given for all three nesting problems. Small instances are punished more than large instances (most likely because relatively more copies of polygons are needed in each translation), but the results are quite similar. On average the number of translations performed decreases by 21% when making horizontal repeats only and by 33% when also making vertical repeats. This decrease in the number of translations is important to consider when analyzing the results in Table 2.

Problem instance	Translations per second			Decrease in percent	
	No	1D	2D	1D	2D
Albano	2847	2178	1891	23%	34%
Dagli	2808	2235	1856	20%	34%
Mao	1622	1218	982	25%	39%
Marques	2957	2213	1882	25%	36%
Shirts	1613	1385	1223	14%	24%
Trousers	2010	1658	1443	18%	28%
Average	2310	1815	1546	21%	33%

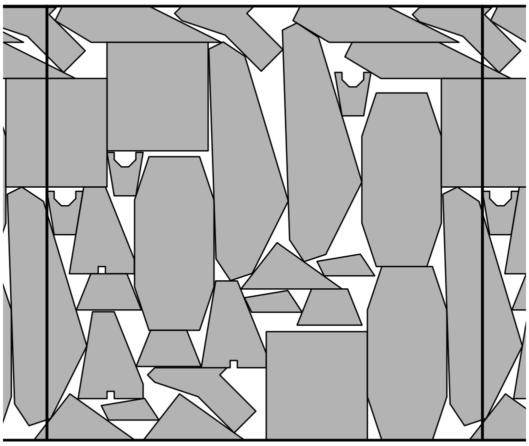
Table 3: The average number of translations performed within the running time of 10 minutes. Fewer translations are done when searching for solutions with repeated patterns. The two rightmost columns describe the decrease in percent.

7 Conclusion

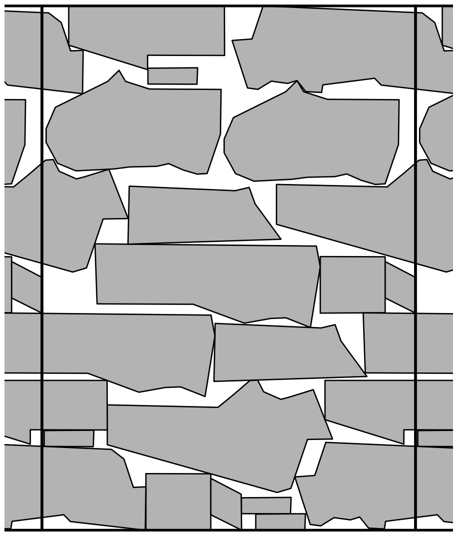
We have presented an efficient heuristic solution method which can construct very good solutions for strip nesting problems in which the solution is going to be repeated either horizontally or both hori-



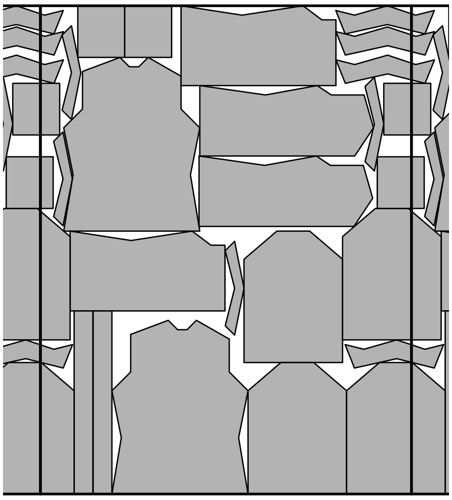
Albano (utilization: 86.48%, length: 10066.06)



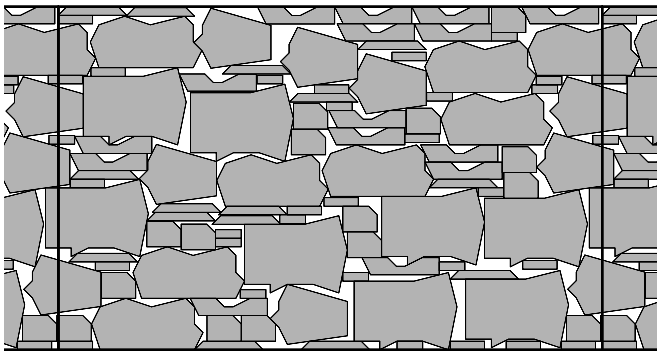
Dagli (utilization: 84.13%, length: 60.28)



Mao (utilization: 81.02%, length: 1819.20)



Marques (utilization: 87.50%, length: 79.05)



Shirts (utilization: 85.14%, length: 63.42)

Trousers (utilization: 86.02%, length: 253.19)

Figure 7: Best results for the repeated strip nesting problem.

zontally and vertically. Results are given for fairly large instances and they strongly indicate that this can give a considerable reduction of waste for problem instances in the garment industry. The solution method is an extension of the work of Egeblad et al. (2006) and the main difference is an altered algorithm for finding a minimum overlap horizontal or vertical position of a given polygon.

Numerous problems remain for future research including free rotation, border filling and a generalization to three dimensions. It should also be possible to alter the solution method such that it applies to the more general translational lattice packing problem for k polygons (Milenkovic 2002). This would involve varying both width and length of the solution, but more importantly it would be necessary to repeat the solution in arbitrary directions. Another interesting idea is to extend the problem to other pavers than the parallelogram used in lattice packings. For example, it might be possible to produce solutions using a hexagonal paver.

Acknowledgments

The author would like to thank Jens Egeblad and Allan Odgaard for allowing the changes to 2DNest which were needed to produce the results presented in this paper.

References

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

Bounsaythip, C. & Maouche, S. (1997), Irregular shape nesting and placing with evolutionary approach, *in* 'IEEE International Conference on Systems, Man, and Cybernetics', Vol. 4, IEEE, New York, pp. 3425–3430.

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

Cheng, S. K. & Rao, K. P. (2000), 'Large-scale nesting of irregular patterns using compact neighborhood algorithm', *Journal of Materials Processing Technology* **103**(1), 135–140.

Dori, D. & Ben-Bassat, M. (1984), 'Efficient nesting of congruent convex figures', *Commun. ACM* **27**(3), 228–235.

Egeblad, J., Nielsen, B. K. & Odgaard, A. (2006), 'Fast neighborhood search for two- and three-dimensional nesting problems', *European Journal of Operational Research* . In Press.

Gomes, A. M., Costa, M. T. & Oliveira, J. F. (2005), An iterated local search approach to the large-scale regular packing of irregular shapes in a limited sheet, *in* 'The 6th Metaheuristics International Conference', Vienna, Austria.

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

Jain, P., Fenyes, P. & Richter, R. (1992), 'Optimal blank nesting using simulated annealing', *Journal of Mechanical Design* **114**(1), 160–165.

Kuperberg, G. & Kuperberg, W. (1990), 'Double-lattice packings of convex bodies in the plane', *Discrete and Computational Geometry* **5**(1), 389–397.

Marques, V. M. M., Bispo, C. F. G. & Sentieiro, J. J. S. (1991), A system for the compaction of two-dimensional irregular shapes based on simulated annealing, *in* 'International Conference on Industrial Electronics, Control and Instrumentation, IECON '91', Kobe, Japan, pp. 1911–1916.

Milenkovic, V. J. (2002), 'Densest translational lattice packing of non-convex polygons', *Computational Geometry* **22**, 205–222.

Mount, D. M. (1991), The densest double-lattice packing of a convex polygon, *in* J. E. Goodman, R. Pollack & W. Steiger, eds, 'Discrete and Computational Geometry: Papers from the DIMACS Special Year', Vol. 6 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Amer. Math. Soc., pp. 245–262.

Mount, D. M. & Silverman, R. (1990), 'Packing and covering the plane with translates of a convex polygon', *Journal of Algorithms* **11**(4), 564–580.

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

Ratanapan, K. & Dagli, C. H. (1997), An object-based evolutionary algorithm for solving irregular nesting problems, *in* 'Proc. Artificial Neural Networks in Engrg. (ANNIE'97)', Vol. 7, ASME Press, New York, pp. 383–388.

Rogers, C. (1964), 'The closest packing of convex two-dimensional domains', *Acta Mathematica* **86**(1), 309–321.

Stoyan, Y. G. & Pankratov, A. V. (1999), 'Regular packing of congruent polygons on the rectangular sheet', *European Journal of Operational Research* **113**(3), 653–675.

Stoyan, Y. G. & Patsuk, V. N. (2000), 'A method of optimal lattice packing of congruent oriented polygons in the plane', *European Journal of Operational Research* **124**(1), 204–216.

Wäscher, G., Haussner, H. & Schumann, H. (2006), 'An improved typology of cutting and packing problems', *European Journal of Operational Research* . In Press.