ELSEVIER

European Journal of Operational Research 141 (2002) 359–370

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

A 2-exchange heuristic for nesting problems ☆

A. Miguel Gomes *, José F. Oliveira

Faculdade de Engenharia, Universidade do Porto – DEEC, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Instituto de Engenharia de Sistemas e Computadores do Porto – UESP, Rua José Falcão 110, 4050–315 Porto, Portugal

Received 15 September 2000; accepted 20 September 2001

Abstract

This paper describes a new heuristic for the nesting problem based on a 2-exchange neighbourhood generation strategy. This mechanism guides the search through the solution space consisting of the sequences of pieces and relies on a low-level placement heuristic to actually convert one sequence in a feasible layout. The placement heuristic is based on a bottom-left greedy procedure with the ability to fill holes in the middle of the layout at a later stage. Several variants of the 2-exchange nesting heuristic were implemented and tested with different initial solution ranking criteria, different strategies for selecting the next solution, and different neighbourhood sizes.

The computational experiments were based on data sets published in the literature. In most of the cases, the 2-exchange nesting algorithm generated better solutions than the best known solutions.

Keywords: Cutting; Packing; Heuristics; Nesting

1. Introduction

The nesting problem is a two-dimensional cutting and packing problem where the small pieces to be cut have irregular shapes. The variant of the problem that we are going to deal with, which naturally arises in several industrial production processes (textile, garment, metalware, etc.), considers only one big rectangular piece, the plate, having fixed width and infinite length. The objective will be the minimisation of the length of the plate that is used to produce a given set of small pieces. The nesting problem is also characterised by the intrinsic difficulty of dealing with geometry when pieces are represented by non-convex polygons. The satisfaction of the "no overlapping" and "containment" constraints involve complex computations.

Several approaches have been proposed for the resolution of nesting problems. Solution techniques range from simple heuristics to local optimisation techniques, including meta-heuristics [2,4–6,9,11–13]. Even for small instances the mathematical programming model of the problem is not solvable in a reasonable amount of time.

☆ Partially supported by FCT, POSI and FEDER (POSI/33757/SRI/2000 (CLPnest)).

* Corresponding author. Tel.: +351-22-508-1888; fax: +351-22-508-1443.

E-mail address: agomes@fe.up.pt (A.M. Gomes).

PII: S0377-2217(02)00130-3

In the following section, the geometric issues involved in the nesting problems resolution are discussed and efficient tools for dealing with geometry are mentioned. In Section 3, the 2-exchange nesting heuristic is described together with details concerning its implementation. Finally, Section 4 presents the computational experiments and some conclusions are drawn.

2. Dealing with geometry

In any cutting or packing problem, the geometric feasibility of the layout has to be tackled, i.e., pieces must not overlap each other and have to be completely inside the material to be cut or the space to be divided. In nesting problems, this geometric problem is particularly difficult due to the natural irregularity and non-convexity of the pieces to be placed.

Since in this problem each solution evaluation implies building a layout, it is very important for the global nesting algorithm efficiency to have an adequate geometry representation and manipulation. Pieces will be represented by polygons. Arcs are approximated by a set of exterior tangents to the curve. Holes inside the interior of the pieces are not allowed. The concept of the *no-fit-polygon* is used to deal with the non-overlapping restrictions. The problem of placing pieces inside the material to be cut (the plate) is, in this particular case, easy to tackle, as the plate is always a rectangle. The concept of *inner-fit-rectangle* will be used to describe the admissible placement region of a piece regarding this constraint.

These two concepts are described in the following sub-sections.

2.1. The no-fit-polygon

The concept of the no-fit-polygon was first introduced by Art [3] and used again by Adamowicz and Albano [1]. Later on, Mahadevan [10] presented a comprehensive description of a no-fit-polygon algorithm implementation which has been followed in the present work.

The no-fit-polygon of piece B relative to piece A ($\mathrm{NFP}_{A,B}$) is the locus of points traced by the reference point associated with B, when this piece slides along the external contour of A. The relative orientations of A and B are maintained during this orbital movement. Piece B (the orbital piece) must never intersect A (the stationary piece) and they must always be in contact (Fig. 1(a)).

From this definition it immediately follows that:

- If the reference point (R_B) of piece B is placed in the *interior* of $\mathrm{NFP}_{A,B}$ then B *intersects* A.
- If the reference point (R_B) of piece B is placed on the *boundary* of $\mathrm{NFP}_{A,B}$ then B *touches* A.
- If the reference point (R_B) of piece B is placed in the *exterior* of $\mathrm{NFP}_{A,B}$ then B does *not intersect or touch* A.

To achieve a feasible (without overlap) and tight layout, each piece should have its reference point on the boundary of at least one no-fit-polygon (relative to another piece) and in the exterior of all the other no-fit-polygons (relative to the remaining pieces).

2.2. The inner-fit-rectangle

The inner-fit-rectangle is derived from the no-fit-polygon concept and it represents the feasible set of points for placing one polygon inside a rectangle.

For the inner-fit-rectangle construction, piece B slides along the internal contour of rectangle A (Fig. 1(b)). This originates a rectangle $\mathrm{IFR}_{A,B}$ whose width is the width of A minus the B bounding-box width and whose height is the height of A minus the B bounding-box height. Consequently:

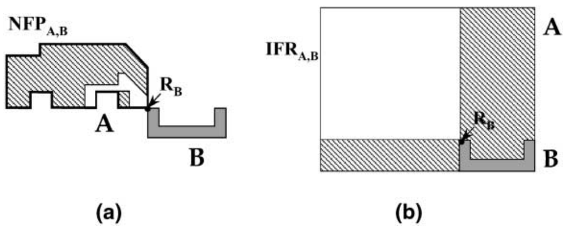

Fig. 1. (a) The no-fit-polygon $\mathrm{NFP}_{A,B}$ and (b) the inner-fit-rectangle $\mathrm{IFR}_{A,B}$.

- If the reference point (R_B) of piece B is placed in the *exterior* of the rectangle $\mathrm{IFR}_{A,B}$ then B is *not contained* in A.
- If the reference point (R_B) of piece B is placed on the *boundary* of the rectangle $\mathrm{IFR}_{A,B}$ then B is *contained* in A and *touches* A.
- If the reference point (R_B) of piece B is placed in the *interior* of the rectangle $\mathrm{IFR}_{A,B}$ then B is *contained* in A but does *not touch* A.

It is assumed that the plate is larger than the biggest piece to place, i.e., that the $\mathrm{IFR}_{A,B}$ always exists.

3. The 2-exchange nesting heuristic

The nesting heuristic presented in this paper deals with piece sequences, utilizing a lower level placement heuristic to place the pieces in the plate. This means that the search is conducted through a solution space of different sequences for the pieces. The search procedure is implemented by a 2-exchange mechanism between pairs of pieces in the current sequence. This sequence is then converted in a layout (cutting pattern) by a greedy bottom-left placement heuristic.

To allow a fairly wide search of the solution space, the placement heuristic must have two characteristics, namely: a layout must be quickly generated; the ability to fill holes at a later stage, thus reducing the dependence from the initial pieces ranking criterion and from the position of the pieces in the sequence.

The main blocks of the 2-exchange nesting heuristic are the following:

- one or more ranking criteria to order the pieces and obtain an initial solution;

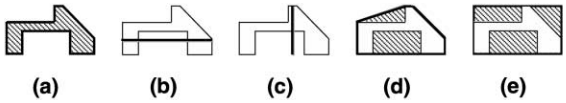

Fig. 2. Initial solution ranking criteria.

- a placement heuristic to transform each sequence of pieces in a layout, allowing its length evaluation;
- a mechanism for guiding the search through the solutions space, the 2-exchange search procedure.

In the following sub-sections, each block of the algorithm (initial solution generation, placement heuristic and 2-exchange search procedure) will be described in more detail.

3.1. The initial solution

The 2-exchange nesting heuristic performs a search over a sequence of pieces, exchanging places between pairs of pieces. To generate the initial solution it is necessary to sort the pieces by some ranking criterion. As in [12], the following criteria were used:

- decreasing area;
- decreasing length;
- decreasing width;
- decreasing irregularity;
- increasing rectangularity;
- random order.

The first three criteria (Fig. 2(a)–(c)) and the last one are obvious and no further explanation is needed. Irregularity is measured as the difference between the piece area and the respective convex hull (Fig. 2(d)). The rectangularity criterion is

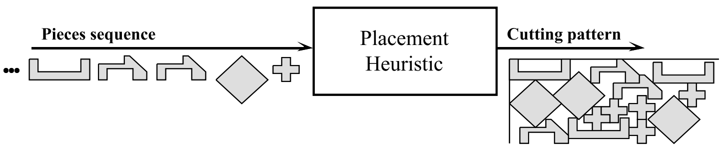

Fig. 3. Placement heuristic.

based on the difference between the piece area and the area of the respective enclosing rectangle (Fig. 2(e)).

The reasoning behind the choice of these criteria is the following:

- bigger/larger pieces are easier to place first;
- holes in the layout can be filled with small pieces at a later stage;
- irregularity seems to play an important role in the nesting algorithms efficiency [12]. So it is natural to use it as a ranking criterion: irregularity and rectangularity are just two different approaches to the same idea.

3.2. Placement heuristic

The placement heuristic has to convert a sequence of pieces in a feasible layout (Fig. 3). A greedy bottom-left heuristic [1] has been used. Dowsland et al. report in [7] a similar bottom-left heuristic, independently developed by those authors. The main idea of the bottom-left heuristic can be stated as follows:

> *Given a plate P and a set of pieces P_i, $i = 1, \dots, m$, already placed in coordinates (x_i, y_i), then the next piece P_k is placed following a bottom-left strategy, i.e., at the point (x_k, y_k) with smaller x and, for equal x with smaller y, so that P_k is inside the plate and does not overlap any piece P_i already placed.*

Using the concepts of no-fit-polygon and inner-fit-rectangle, this reasoning can be instantiated as a set of admissible points for placement of P_k (set A defined by conditions (1) and (2)) and a minimisation objective leading to the most bottom-left placement of piece P_k (conditions (3) and (4)):

$$A = \{(x, y) : (x, y) \in \text{interior}(\text{IFR}_{P,k}) \wedge \quad (1)$$
$$(x, y) \notin \text{interior}(\text{NFP}_{i,k}), \forall i = 1, \dots, m\}, \quad (2)$$

$$B = \{(x_{\min}, y) : x_{\min} \leqslant x \ \forall (x, y) \in A\}, \quad (3)$$

$$(x_{\min}, y_{\min}) : y_{\min} \leqslant y \quad \forall (x_{\min}, y) \in B, \quad (4)$$

where $\text{IFR}_{P,k}$ is the inner-fit-rectangle of piece P_k relative to plate P; $\text{NFP}_{i,k}$ is the no-fit-polygon of piece P_k relative to piece P_i, when piece P_i is placed at coordinates (x_i, y_i).

Set A is an infinite and non-convex admissible region for the placement of piece P_k. However conditions (3) and (4) ensure that piece P_k will be placed not in the interior of the admissible placement region but in one of its vertices. Any vertex of the admissible region is:

- a vertex of the no-fit-polygon $\text{NFP}_{i,k}$ or
- a vertex of the inner-fit-rectangle $\text{IFR}_{P,k}$ or

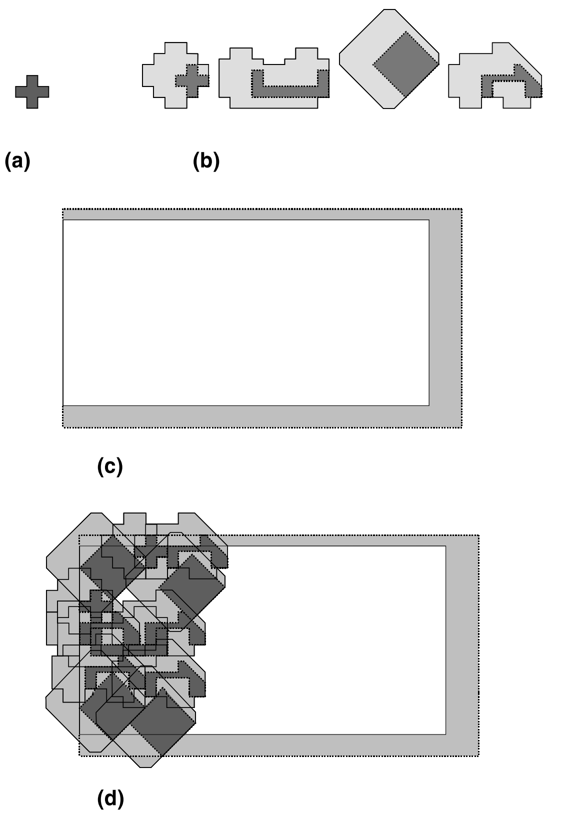

Fig. 4. Placement example: (a) piece P_k; (b) no-fit-polygon $\text{NFP}_{i,k}$; (c) inner-fit-rectangle $\text{IFR}_{P,k}$; (d) admissible placement region.

[1] In fact the placement heuristic looks like an up-left heuristic in the figures because the y-axis origin is on the top of the layout, i.e., the y coordinate increases downward.

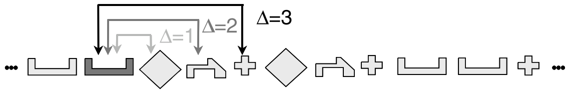

Fig. 6. The neighbourhood size parameter Δ.

- the intersection of two edges of two no-fit-polygons $NFP_{i,k}$ and $NFP_{j,k}$ or
- the intersection of an edge of the inner-fit-rectangle $IFR_{P,k}$ and an edge of a no-fit-polygon $NFP_{i,k}$.

This means that piece P_k will be in contact with the plate, the plate and one or more of the pieces already placed, or two or more of the pieces already placed.

With this procedure, the infinite non-convex set of admissible placement points is reduced to a discrete and finite set of points, were the search for the bottom-left placement point becomes trivial.

In Fig. 4 an example of this placement strategy application can be found. Assuming 12 pieces already placed (in dark gray), piece P_k (Fig. 4(a)) is now going to be placed. In Figs. 4(b) and (c) the no-fit-polygons, between P_k and the already placed pieces, and the inner-fit-rectangle, between P_k and the plate, are represented in light gray. The admissible region for the placement of P_k (corresponding to set A defined by conditions (1) and (2)) is represented in white (Fig. 4(d)). By taking into account conditions (3) and (4) only the vertices of this region are considered for the placement of P_k.

It should also be mentioned that the no-fit-polygon and the inner-fit-rectangle can both be computed off-line, since they only depend on the shape of the pieces. When different orientations area allowed, then for each pair of pieces it is necessary to calculate a no-fit-polygon for each pair of different orientations, and for each piece it is necessary to calculate a inner-fit-rectangle for each orientation. Fortunately, in real cases the admissible orientations are limited to a few possibilities due to technological constraints (drawing patterns, material resistance, etc.). Another important characteristic of this heuristic is that holes, in the middle of the layout, can be filled at any stage of the placement. For instance, in the example presented in Fig. 4 a hole would be chosen for the piece placement.

Finally, in problems where pieces have more than one admissible orientation, each piece is placed by the heuristic in all admissible orientations and the one that produces the most bottom-left placement is chosen (Fig. 5).

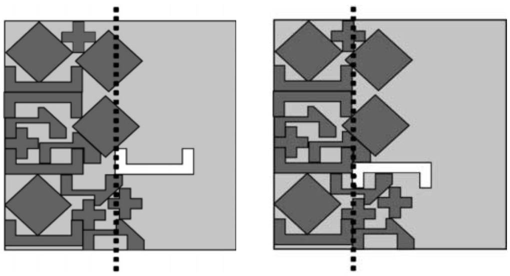

Fig. 5. Placing pieces with several admissible orientations.

3.3. The 2-exchange search procedure

The 2-exchange search procedure is responsible for guiding the search through the solution space. The algorithm moves to a neighbour solution by exchanging a pair of pieces in the current sequence. This is an improvement algorithm that

Table 1
Data sets used in the computational experiments

Data set	Number of different pieces	Total number of pieces	Vertices by piece (average)	Admissible orientations (degrees)	Plate width
SHAPES0	4	43	8.75	0	40
SHAPES1	4	43	8.75	0 and 180	40
SHAPES2	7	28	6.29	0 and 180	15
SHIRTS	8	99	6.63	0 and 180	40
TROUSERS	17	64	5.06	0 and 180	79

Table 2
2-Exchange nesting heuristic results

Data set	Initial solution ranking criteria	*first better*			*best*			*random better*					
		$\Delta = 1$	$\Delta = 2$	$\Delta = 3$	$\Delta = 1$	$\Delta = 2$	$\Delta = 3$	$\Delta = 1$		$\Delta = 2$		$\Delta = 3$	
								Average	Best	Average	Best	Average	Best
SHAPES0	Random (average)	69.09	68.18	67.80	69.26	68.15	68.39	69.05	–	67.78	–	67.84	–
	Random (best)	65.50	66.00	65.50	66.50	**65.00**	66.00	–	67.00	–	**65.00**	–	**65.00**
	Area	70.00	68.00	67.50	70.00	68.50	67.50	70.00	70.00	68.38	68.00	68.20	**65.00**
	Length	67.00	67.00	67.00	67.00	67.00	67.00	67.00	67.00	67.00	67.00	67.00	67.00
	Width	75.00	68.00	68.00	75.00	68.00	68.00	75.00	75.00	68.00	68.00	68.00	68.00
	Irregularity	68.50	68.00	68.00	68.50	68.00	67.00	68.43	68.00	68.08	68.00	67.48	66.00
	Non-rectangularity	71.00	69.00	69.00	71.00	69.00	69.00	71.00	71.00	70.10	69.00	70.10	69.00
SHAPES1	Random (average)	65.88	64.46	64.44	65.10	64.19	63.91	65.15	–	64.20	–	63.59	–
	Random (best)	62.00	60.00	61.00	62.00	60.00	61.00	–	60.00	–	60.00	–	60.00
	Area	65.50	63.00	63.00	65.50	63.00	63.00	65.50	65.50	63.13	63.00	63.13	63.00
	Length	67.00	64.00	63.00	67.00	64.00	63.00	67.00	67.00	64.00	64.00	63.70	63.00
	Width	65.00	65.00	64.00	65.00	63.00	63.00	64.20	63.00	63.00	63.00	63.00	63.00
	Irregularity	69.00	68.00	66.50	69.00	68.00	66.50	69.00	69.00	68.15	66.50	66.25	65.00
	Non-rectangularity	63.00	63.00	62.00	63.00	63.00	60.00	63.00	63.00	63.00	63.00	61.55	**59.00**
SHAPES2	Random (average)	29.90	29.59	29.34	29.88	29.52	29.13	29.78	–	29.52	–	29.34	–
	Random (best)	28.42	28.53	28.33	28.61	28.23	28.22	–	28.90	–	28.38	–	28.50
	Area	29.40	29.40	28.25	29.40	29.40	28.73	29.40	29.40	28.88	28.36	28.45	28.05
	Length	28.40	28.18	27.93	28.70	28.73	27.44	28.30	27.76	28.15	27.60	28.13	**27.30**
	Width	29.45	28.92	29.20	29.45	28.38	28.38	29.45	29.45	28.38	28.38	28.33	27.92
	Irregularity	30.04	30.04	28.50	30.04	30.04	29.06	30.04	30.04	30.04	30.04	29.13	28.83
	Non-rectangularity	28.65	28.30	28.30	28.65	28.30	28.30	28.65	28.65	28.30	28.30	28.27	27.90
SHIRTS	Random (average)	67.97	67.12	67.14	67.69	67.16	66.86	67.05	–	66.99	–	66.79	–
	Random (best)	66.53	65.27	65.63	65.40	65.38	64.63	–	65.40	–	65.32	–	65.40
	Area	65.97	64.75	64.75	65.97	65.80	65.00	65.88	65.80	65.30	64.75	64.91	63.81
	Length	65.74	64.63	64.41	65.74	65.74	64.35	65.74	65.74	64.88	64.32	64.09	63.49
	Width	65.36	65.36	64.82	65.36	63.66	63.66	65.36	65.36	63.64	63.58	63.60	63.39
	Irregularity	65.14	65.03	65.03	65.14	65.03	65.03	68.88	65.80	65.30	64.75	64.91	64.81
	Non-rectangularity	63.73	63.54	63.54	63.73	63.17	63.17	63.73	63.73	63.17	63.17	63.17	**63.13**
TROUSERS	Random (average)	269.49	263.35	259.90	267.59	260.86	260.51	265.33	–	260.88	–	257.13	–
	Random (best)	254.98	249.33	247.93	247.52	247.52	247.30	–	247.52	–	247.52	–	247.52

Area	283.60	247.50	257.46	283.60	249.51	257.46	283.60	283.60	251.42	247.50	253.67	246.60
Length	259.00	259.00	245.94	259.00	259.00	252.78	259.00	259.00	259.00	259.00	251.54	246.67
Width	279.60	280.60	280.60	279.60	280.60	280.60	279.75	279.60	279.86	279.60	279.70	275.73
Irregularity	259.00	250.40	248.22	259.00	250.40	250.40	259.00	259.00	249.06	246.74	249.67	**245.75**
Non-rectangularity	284.93	279.40	278.82	279.60	281.64	279.60	280.62	279.60	280.18	278.82	279.07	275.33

ends when it is impossible to move to a better solution in the current neighbourhood.

Neighbourhoods are build by exchanging pairs of pieces in a sequence. Parameter Δ (Fig. 6) controls the size of the neighbourhood, allowing only exchanges between pieces within a distance of Δ. For each piece, all the exchanges with the Δ following pieces in the sequence area evaluated. Three different neighbourhood parameters were tried: $\Delta = 1, 2, 3$.

Three different strategies were used to select the neighbour that will be center of the neighbourhood in the next iteration, i.e., which pair of pieces will be exchanged:

- the first better solution found (*first better*);
- the best solution found (*best*);
- a randomly chosen solution among the better solutions found (*random better*).

The idea behind the *first better* strategy, when selecting the first better solution found in the neighbourhood, is to rapidly change from one solution to another. On the other hand, the *best* neighbour exhaustively searches the neighbourhood looking for the best solution. The *random better* is a probabilistic approach were a better solution is randomly chosen from a list of candidates solutions, each one with equal probability. This list keeps all the better solutions in the neighbourhood.

4. Computational results

The 2-exchange nesting heuristic was tested with five data sets available in the literature. The main characteristics of these data sets are presented in Table 1. The first three data sets (*SHAPES0*, *SHAPES1* and *SHAPES2*) are artificial, while the other two, *SHIRTS* and *TROUSERS*, are real instances taken from the garment industry. The actual data can be found in [12].

The computational tests were run on a PC with a *Pentium III* processor at 450 MHz. For each data set, nine possible variants of the algorithm were tested:

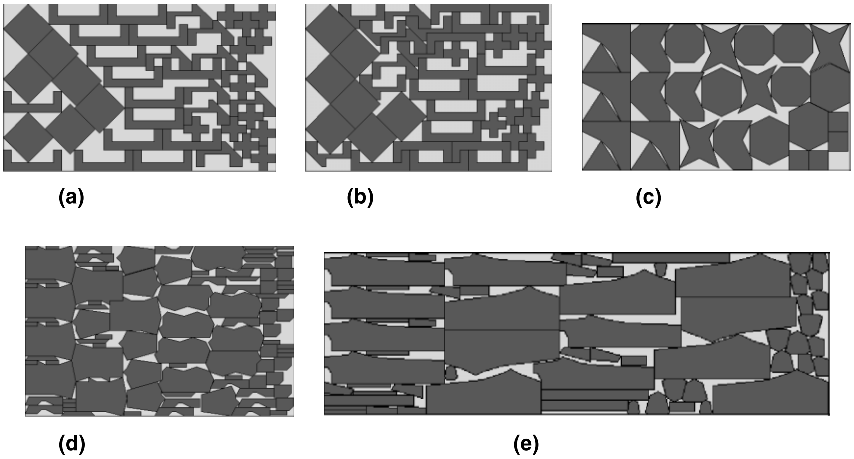

Fig. 7. Best layouts: (a) *SHAPES0*; (b) *SHAPES1*; (c) *SHAPES2*; (d) *SHIRTS*; (e) *TROUSERS*.

- 3 different strategies to select a solution from the candidates list – *first better*, *best* and *random better*;
- size of the neighbourhood – $\Delta \in \{1, 2, 3\}$.

For each algorithm variant, six different ranking criteria to generate initial solutions were applied.

When the random criterion is used to generate the initial solution, or the probabilistic approach is used to select a solution from the candidates list (*random better*), 20 runs with 20 different seeds were carried out for each variant of the heuristic. In those cases, both the average and the best result are presented.

The results for all data sets are summarized in Table 2. The best result for each data set is marked in bold face, and in Fig. 7, the actually best layouts are presented. In Table 3, the best results of the 2-exchange nesting heuristic are compared with the best known results.

The first comment on the results obtained with the 2-exchange nesting heuristic is, obviously, the improvement of the best known results for four of the five data sets (Table 3): 3.28% for *SHAPES1*, 2.85% for *SHAPES2*, 2.88% for *SHIRTS* and 7.00% for *TROUSERS*. Only for data set *SHAPES0*, the best result obtained was 3.17% worse than the best one known from the literature.

A second point of analysis concerns the influence of the initial solution strategy over the results of the algorithm. In Fig. 8 each column of the chart represents, for each data set, the average distance between the results obtained with each variant of the algorithm and the best result obtained, for each data set. As it can be seen, the conclusions strongly depend on the data set:

Table 3
The 2-exchange nesting heuristic best results compared with the best results found in the literature

Data set	2-Exchange nesting heuristic		Best known result in the literature	
	Length	Time (s)	Length	Reference
SHAPES0	65.00	819.32	63.00	Dowsland et al. [6]
SHAPES1	59.00	2019.77	61.00	Oliveira et al. [12]
SHAPES2	27.30	551.73	28.10	Gomes [8]
SHIRTS	63.13	6367.57	65.00	Dowsland et al. [6]
TROUSERS	245.75	13613.67	263.17	Oliveira et al. [12]

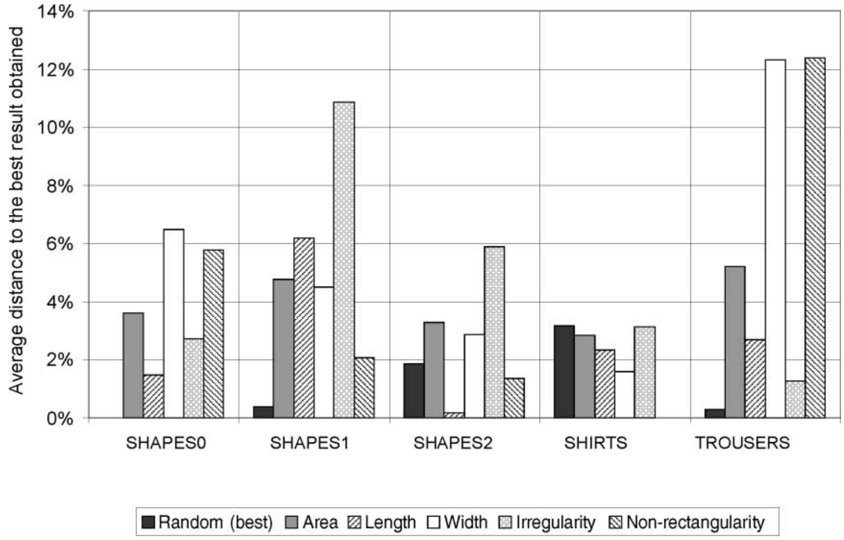

Fig. 8. Initial solution results.

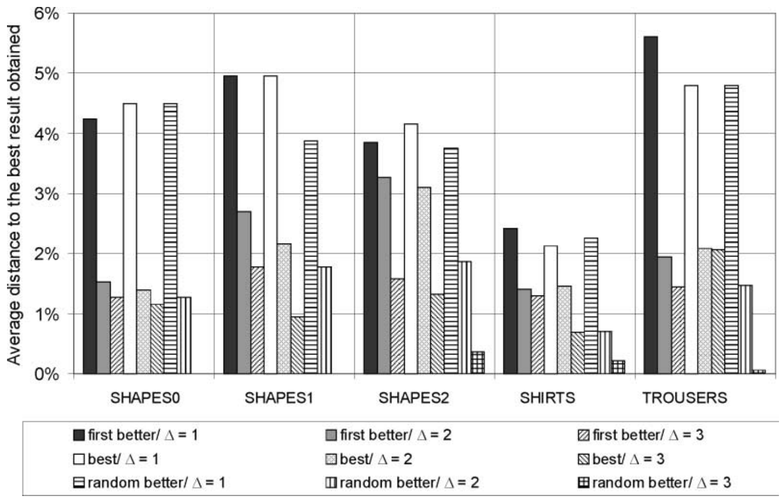

Fig. 9. 2-Exchange nesting heuristic variants results.

- taking the best result from 20 *random* initial solutions proved to be adequate for data sets *SHAPES0* (0.00%), *SHAPES1* (0.37%) and *TROUSERS* (0.29%);
- using *length* as the initial ranking criterion is better for data set *SHAPES2* (0.17%);
- finally, for data set *SHIRTS* the *non-rectangularity* ranking criterion obtained the best results for all the variants (0.00%).

The comparison among the nine variants of the 2-exchange nesting heuristic may be observed in Fig. 9, where the average distance to the best solution is again represented. The variant with the probabilistic approach, when selecting the next neighbour (*random better*) with bigger neighbourhood size ($\Delta = 3$) is clearly the best one: 0.00% for data sets *SHAPES0* and *SHAPES1*, 0.06% for *TROUSERS*, 0.21% for *SHIRTS* and 0.36% for

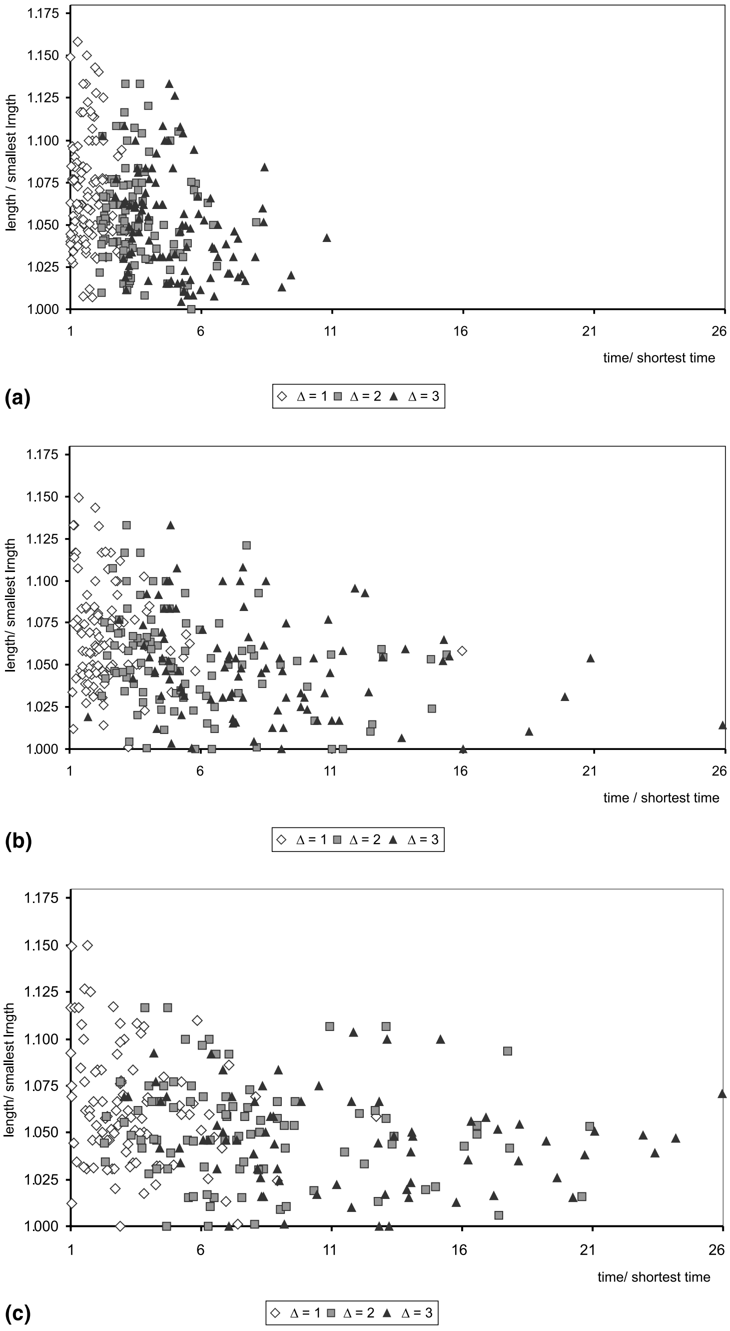

Fig. 10. Tradeoff between solutions quality and computational times: (a) *first better* solution; (b) *best* solution; (c) *random better* solution.

SHAPES2. Given a data set and an initial solution strategy, only in six of a total of thirty combinations, the referred variant was not the best one (Table 2): 1.00% worse for *SHAPES2/random*, 1.16% worse for *SHAPES2/irregularity*, 1.19% worse for *SHIRTS/random*, 0.09% worse for

Table 4
2-Exchange nesting heuristic computational times

Data set		Length	Time (s)	Number of layouts evaluated	$\frac{\text{Time}}{\text{Layout}}$
SHAPES0	*first better*	65.50	161.31	59	2.734
	best	65.00	1145.48	392	2.922
	random better	65.00	819.32	268	3.057
SHAPES1	*first better*	60.00	1191.93	206	5.786
	best	60.00	2402.93	409	5.875
	random better	59.00	2019.77	327	6.177
SHAPES2	*first better*	27.93	161.15	155	1.040
	best	27.44	437.76	423	1.035
	random better	27.30	551.73	539	1.024
SHIRTS	*first better*	63.54	960.97	54	17.796
	best	63.17	1739.00	95	18.305
	random better	63.13	6367.57	344	18.510
TROUSERS	*first better*	245.94	2978.23	357	8.342
	best	247.30	6237.12	664	9.393
	random better	245.75	13613.67	1667	8.167

SHIRTS/irregularity, 0.69% worse for *TROUSERS/random* and 0.30% worse for *TROUSERS/length*.

Another perspective when comparing the nine variants of the 2-exchange nesting heuristic is the tradeoff between solutions quality and computational times. Fig. 10(a)–(c) plot for each variant the results of 300 runs of the algorithm: five data sets $\times$ three Δ values $\times$ 20 random initial solutions. Each point plotted on the graph has as x-coordinate the quotient between the computational time associated with that run and the minimum computational time obtained in all runs with that data set. As y-coordinate it has the quotient between the length and the minimum length obtained in all runs with that data set.

As expected, the computational time increases with the size of the neighbourhood, controlled by the Δ parameter value, and with the complexity of the strategy used in the selection of the next solution, i.e., neighbour. At the same time, the quality of the solutions also increases, i.e., smaller layout lengths are generated.

A final comment about computational times. Table 4 shows the computational times for the best results obtained with the variants of the 2-exchange nesting heuristic, grouped by the next neighbour selection strategy. The neighbour selection strategy clearly affects the computational times. However, we can see that the time needed to generate and evaluate one solution (one layout) is approximately constant for each data set. Moreover, data sets with larger computational times have more pieces to be placed. This leads to the conclusion that, as expected, larger execution times are originated by a wider search and more complex data sets.

5. Conclusions

A 2-exchange nesting heuristic is proposed in this paper. This heuristic is based on a search over sequences of pieces which are placed by a bottom-left placement heuristic. The search is based on a 2-exchange mechanism, where the pieces are initially ordered by different ranking criteria.

Several variants of the 2-exchange nesting heuristic were implemented and tested with different criteria to build initial solutions, different strategies for the selection of the next solution (i.e., neighbour), and different neighbourhoods sizes. Five data sets were used to test and compare the implemented variants, having two of them been taken from the garment industry.

The new 2-exchange nesting heuristic improved the best known result in the literature for four of the five data sets used. Analyzing the results obtained for the different variants, the following conclusions can be drawn:

- the probabilistic approach (*random better*) to select the next solution (neighbour), with larger neighbourhood size ($\Delta = 3$), clearly dominates the other pairs of strategy and neighbourhood size;
- neither of the initial solution ranking criteria tested dominates the others. The results shows that the choice of a initial solution ranking criteria strongly depend on the data set;
- computational times are directly proportional to the number of layouts generated and evaluated, i.e., the "*extent of the search*";
- the placement heuristic rapidly generates and evaluates one layout in less than 20 seconds, taking between 0.01 and 0.09 seconds to place a piece in a given orientation;
- the placement heuristic is quite effective in filling holes in the layout at any stage of the placement.

References

[1] M. Adamowicz, A. Albano, A two-stage solution of the cutting-stock problem, Information Processing 71 (1972) 1086–1091.

[2] A. Albano, G. Sapuppo, Optimal allocation of two-dimensional irregular shapes using heuristic search methods, IEEE Transactions on Systems, Man and Cybernetics SMC-10 (5) (1980) 242–248.

[3] R.C. Art Jr., An approach to the two-dimensional irregular cutting stock problem, Technical Report 36.Y08, IBM Cambridge Centre, 1966.

[4] J.A. Bennel, K.A. Dowsland, Hybridising tabu search with optimisation techniques for irregular stock-cutting, in: Extend Abstracts of MIC'99, Angra Dos Reis, Rio de Janeiro, Brazil, 1999.

[5] J. Błażewicz, P. Hawryluk, R. Walkowiak, Using tabu search approach for solving the two-dimensional irregular cutting problem in tabu search, in: F. Glover, M. Laguna, E. Taillard, D. de Werra (Eds.), Tabu Search – volume 41 of Annals of Operations Research, J.C. Baltzer AG., 1993.

[6] K.A. Dowsland, W.B. Dowsland, J.A. Bennell, Jostling for position: Local improvement for irregular cutting patterns, Journal of the Operational Research Society 49 (1998) 647–658.

[7] K.A. Dowsland, S. Vaid, W.B. Dowsland, An algorithm for polygon placement using a bottom-left strategy, European Journal of Operational Research 141(2), this issue.

[8] A.M. Gomes, Novas contribuições para o problema de posicionamento de figuras irregulares, Master's Thesis, Faculdade de Engenharia da Universidade do Porto, 1995.

[9] Z. Li, V. Milenkovic, A compaction and separation algorithms for non-convex polygons and their applications, European Journal of Operational Research 84 (1995) 539–561.

[10] A. Mahadevan, Optimization in Computer-Aided Pattern Packing, PhD Thesis, North Carolina State University, 1984.

[11] J.F. Oliveira, J.S. Ferreira, Algorithms for nesting problems, in: R.V.V. Vidal (Ed.), Applied Simulated Annealing, Springer, Berlin, 1993, pp. 255–273.

[12] J.F. Oliveira, A.M. Gomes, J.S. Ferreira, TOPOS – A new constructive algorithm for nesting problems, OR Spektrum 22 (2000) 263–284.

[13] Y.G. Stoyan, Precise and approximate methods of problems in irregular allocation of non-convex polygons in a strip of minimal length, in: IFORS'93 Conference, Lisbon, 1993.