A guided local search based on a fast neighborhood search for the irregular strip packing problem

Shunji Umetani [1], Mutsunori Yagiura [2], Takashi Imamichi [3], Shinji Imahori [4], Koji Nonobe [5] and Toshihide Ibaraki [6]

[1] Faculty of Electro-Communications, University of Electro-Communications, Chofu, Tokyo 182-8585, Japan
[2] Graduate School of Information Science, Nagoya University, Nagoya 464-8603, Japan
[3] Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
[4] Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan
[5] Faculty of Engineering, Hosei University, Koganei, Tokyo 184-8584, Japan
[6] School of Science and Technology, Kwansei Gakuin University, Sanda, Hyogo 669-1337, Japan
[1] umetani@se.uec.ac.jp, [2] yagiura@nagoya-u.jp, [3] ima@amp.i.kyoto-u.ac.jp,
[4] imahori@simplex.t.u-tokyo.ac.jp [5] nonobe@k.hosei.ac.jp, [6] ibaraki@ksc.kwansei.ac.jp

Abstract: The irregular strip packing problem asks to place a set of polygons within a rectangular strip of fixed height without overlap, so as to minimize the strip width required. We consider an overlap minimization problem which minimizes the amount of overlap penalty for all pairs of polygons within a given bound of strip width. We propose a local search algorithm which translates a polygon in horizontal and vertical directions iteratively, and incorporate it in metaheuristic approaches called the iterated local search and the guided local search. Computational results show that our algorithm is competitive with other existing algorithms.

Keywords: Cutting, Packing, Guided local search, Irregular strip packing, Nesting.

1 Introduction

Many cutting and packing problems appear in various industries such as wood, textile, sheet metal, plastics, glass and leather. The *irregular strip packing problem* is one of cutting and packing problems that asks to place a set of polygons within a rectangular strip of fixed height without overlap, so as to minimize the strip width required (Fig. 1). The problem is known to be NP-hard even without rotation. The problem is much harder than the rectangle strip packing problem because of a burden of geometrical computation. However, based on rapid development of computing power and theory of computational geometry, many approaches have been developed in recent years.

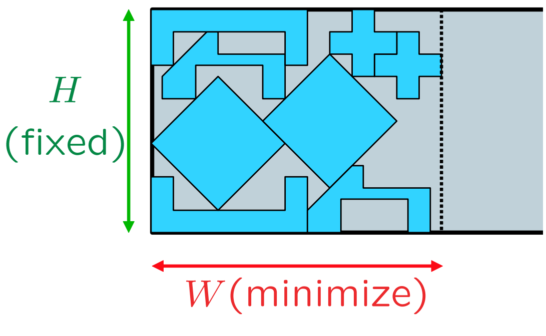

Figure 1: The irregular strip packing problem: a feasible solution

Albano and Sapuppo (1980) proposed a bottom-left heuristic algorithm that places polygons one by one at the bottommost and leftmost position according to a sequence of input polygons, and used to a tree search to obtain a good sequence. Gomes and Oliveira (2002) used a local search algorithm to find a good sequence for the bottom-left heuristics. Li and Milenkovic (1995) proposed separation and compaction algorithms based on linear programming that reduce the strip width and the amount of overlap, respectively. Bennell and Dowsland (2001), and Gomes and Oliveira (2006) developed hybrid approaches of the bottom-left heuristic and linear programming to obtain good solutions. Burke *et al.* (2006) proposed a new bottom-left fill heuristic algorithm which can place shapes that include circular arcs and holes, and incorporated it with tabu search.

Egeblad *et al.* (2006) considered an *overlap minimization problem* which minimizes the amount of overlap penalty for all pairs of polygons within a given bound of strip width. Here, they used the area of intersection for a pair of polygons as the overlap penalty. For the problem, they proposed a local search algorithm in which the neighborhood is any horizontal or vertical translation of a given polygon from its current position. They proposed an efficient neighborhood search that finds a position with the minimum total overlap penalty in its neighborhood, and incorporate it with the guided local search (Voudouris and Tsang, 1999).

In this paper, we first propose another overlap minimization problem, in which we use an *approximate penetration depth* for a pair of polygons as the overlap penalty. We propose an efficient implementation of the neighborhood search by utilizing a data structure called the *no-fit polygon* (NFP). Based on this, we propose a local search algorithm which repeats translating a polygon in horizontal and vertical directions until no better position is found in either direction, and incorporate it with the iterated local search and a variant of guided local search called the weighting method (Selman and Kautz, 1993).

In the following section, we formulate the irregular strip packing problem and the overlap minimization problem. In Section 4, we describe the local search algorithm with details concerning its implementation. Finally, Section 5 presents the computational experiments and some concluding remarks.

2 Formulation

The problem is described as follows: We are given a set of small polygons[1] $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ and a rectangular strip $R = R(W)$ of height H and width W called the stock sheet, where H is a given constant and W is a positive variable. Each polygon $P_i \in \mathcal{P}$ has a set of modes $\mathcal{M}_i = \{1, 2, \ldots, m_i\}$, which specifies its configuration except for its position, e.g., reflection, rotation by a given degrees. We denote polygon $P_i \in \mathcal{P}$ specified by a mode $k_i \in \mathcal{M}_i$ by $P_i(k_i)$.

For convenience, we consider that each of polygons $P_i \in \mathcal{P}$ represents its inner region without its boundary, and the strip R represents its inner region including its boundary. Let $\overline{R} = \overline{R}(W)$ be the complement (i.e., the outer region) of the strip R. We describe translations of polygons by Minkowski sums (de Berg *et al.*, 1998), i.e., we denote the translation of polygon $P_i \in \mathcal{P}$ by a translation vector $\boldsymbol{t}_i = (x_i, y_i)$ by $P_i \oplus \boldsymbol{t}_i = \{\boldsymbol{p} + \boldsymbol{t}_i \mid \boldsymbol{p} \in P_i\}$. We describe a solution of this problem by positions $\boldsymbol{t} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$ and modes $\boldsymbol{k} = (k_1, \ldots, k_n)$ of all polygons $P_i \in \mathcal{P}$. Note that the minimum width W is the x-coordinate interval between the leftmost and rightmost points of the polygons placed by $(\boldsymbol{t}, \boldsymbol{k})$. The irregular strip packing problem is formally described as follows:

$$
\begin{array}{ll}
\text{minimize} & W \\
\text{subject to} & (P_i(k_i) \oplus \boldsymbol{t}_i) \cap (P_j(k_j) \oplus \boldsymbol{t}_j) = \emptyset, (1 \leq i < j \leq n), \\
& (P_i(k_i) \oplus \boldsymbol{t}_i) \cap \overline{R}(W) = \emptyset, (1 \leq i \leq n), \\
& W \in \mathbb{R}_+, \\
& \boldsymbol{t}_i \in \mathbb{R}^2, (1 \leq i \leq n), \\
& k_i \in \mathcal{M}_i, (1 \leq i \leq n).
\end{array}
$$

The first constraint ensures that no pair of polygons overlaps, and the second constraint ensures that no polygon protrudes from the strip.

[1]Polygons $P_i \in \mathcal{P}$ are not necessarily convex.

We now consider a variant of the irregular strip packing problem called the *overlap minimization problem* that minimizes the amount of the overlap penalty $f_{ij}(\boldsymbol{t},\boldsymbol{k})$ for all pairs of polygons $P_i, P_j \in \mathcal{P}$, while constraining the strip width W within a bound W_{UB} given by users.

$$\begin{array}{ll} \text{minimize} & F(\boldsymbol{t},\boldsymbol{k}) = \displaystyle\sum_{1\le i<j\le n} f_{ij}(\boldsymbol{t},\boldsymbol{k}) \\ \text{subject to} & (P_i(k_i) \oplus \boldsymbol{t}_i) \cap \overline{R}(W_{\text{UB}}) = \emptyset, (1 \le i \le n), \\ & \boldsymbol{t}_i \in \mathbb{R}^2, (1 \le i \le n), \\ & k_i \in \mathcal{M}_i, (1 \le i \le n). \end{array}$$

Egeblad *et al.* (2006) used the area of intersection of polygons $P_i, P_j \in \mathcal{P}$ as the overlap penalty $f_{ij}(\boldsymbol{t},\boldsymbol{k})$. In this paper, we use an *approximate penetration depth* of a pair of overlapping polygons P_i and P_j instead. We define the approximate penetration depth $f_{ij}(\boldsymbol{t},\boldsymbol{k},\boldsymbol{v})$ of a pair of overlapping polygons P_i and P_j as the minimum translational distance in a given direction $\boldsymbol{v} = (v_x, v_y)$ ($\boldsymbol{v} \in \mathbb{R}^2$) to separate them. If a pair of polygons do not overlap, their approximate penetration depth is zero. The approximate penetration depth is formally described as:

$$f_{ij}(\boldsymbol{t},\boldsymbol{k},\boldsymbol{v}) = \min\{|z| \mid (P_i(k_i) \oplus \boldsymbol{t}_i) \cap (P_j(k_j) \oplus \boldsymbol{t}_j \oplus z\boldsymbol{v}) = \emptyset, z \in \mathbb{R}\},$$

and the overlap penalty for a pair of polygons P_i and P_j is accordingly given by

$$f_{ij}(\boldsymbol{t},\boldsymbol{k}) = \min\left\{f_{ij}(\boldsymbol{t},\boldsymbol{k},\boldsymbol{v}) \mid \boldsymbol{v} \in \{\boldsymbol{e}_x, \boldsymbol{e}_y\}\right\},$$

where $\boldsymbol{e}_x$ (resp., $\boldsymbol{e}_y$) is a unit vector of horizontal (resp., vertical) direction.

3 No-fit polygon

For fast computation of the overlap penalty $f_{ij}(\cdot)$, we introduce a data structure called the *no-fit polygon* (NFP), which is often used in the irregular strip packing problem. The no-fit polygon $\text{NFP}(P_i, P_j)$ for a pair of polygons P_i and P_j is defined by

$$\text{NFP}(P_i, P_j) = P_i \oplus (-P_j) = \{\boldsymbol{p} - \boldsymbol{q} \mid \boldsymbol{p} \in P_i, \boldsymbol{q} \in P_j\}.$$

We can easily check whether two polygons P_i and P_j overlap or not, by simply checking whether the reference point of P_i is inside $\text{NFP}(P_i, P_j)$ or not.

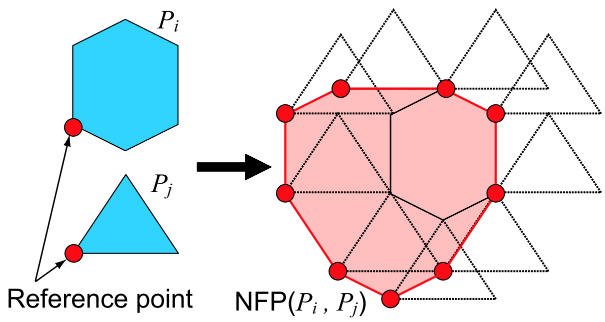

Figure 2: An example of no-fit polygon (NFP)

When P_i and P_j are both convex, we can compute $\text{NFP}(P_i, P_j)$ by the following simple procedure. We first put the reference point of P_i at the origin, and slide P_j around P_i keeping in touch with P_i. The $\text{NFP}(P_i, P_j)$ is the inner region of the trajectory drawn by the reference point of P_j (Fig. 2).

We can also check whether a polygon $P_i \in \mathcal{P}$ protrudes from the strip R by NFP($\overline{R}, P_i$), i.e., a polygon P_i protrudes from R if and only if the reference point of P_i is inside NFP($\overline{R}, P_i$).

Although it takes $O(p_1^2 p_2^2)$ time to compute an NFP of two non-convex polygons with p_1 and p_2 edges (de Berg *et al.*, 1998) in the worst case, practical algorithms to compute it have been proposed, e.g., by Bennell *et al.* (2001) and Burke *et al.* (2006).

penalty $F_i(\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{v})$ by decomposing them into convex elements, i.e., we detect the sets of overlapping positions as intervals for all pairs of convex elements and merge them. We note that our algorithm uses only simple operations instead of difficult geometrical computations such as Minkowski sums of non-convex polygons.

To facilitate the neighborhood search, we detect possibly overlapping polygons $P_j \in \mathcal{P}$ of P_i before computing the overlap penalty function $f_{ij}(\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{v})$. We first project each polygon onto the y-axis (resp., x-axis) when the polygon P_i moves in the horizontal (resp., vertical) direction (Fig. 3). We then check the intersection of two intervals $(y_i^{\min}, y_i^{\max})$ and $(y_j^{\min}, y_j^{\max})$, which are

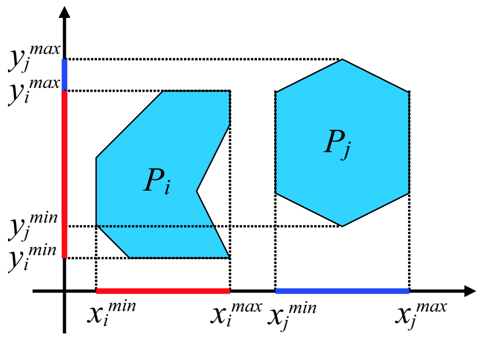

Figure 3: Detecting possibly overlapping polygons by projection onto the x- and y- axes

the projections of two polygons P_i and P_j onto the y-axis. If $(y_i^{\min}, y_i^{\max}) \cap (y_j^{\min}, y_j^{\max}) = \emptyset$ holds, we can skip the computation of the overlap penalty $f_{ij}(\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{v})$ when P_i moves in the horizontal direction.

The computation time of the algorithm is dominated by sorting of the event points of the total overlap penalty function $F_i(\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{v})$. Since Egeblad *et al.* (2006) use the area of intersection of polygons P_i and P_j as the overlap penalty function $f_{ij}(\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{v})$, the number of event points of their total overlap penalty function $F_i(\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{v})$ of P_i is the product of the numbers of edges of the polygon P_i and the other polygons $P_j \in \mathcal{P} \setminus \{P_j\}$. On the other hand, since the number of event points of our overlap penalty function $f_{ij}(\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{v})$ is always three when two polygons P_i and P_j are both convex, the number of event points of our total overlap penalty function $F_i(\boldsymbol{t}, \boldsymbol{k}, \boldsymbol{v})$ of P_i is at most three times as the product of the numbers of convex elements in the polygon P_i and the other polygons $P_j \in \mathcal{P} \setminus \{P_j\}$. This implies that the number of event points of our overlap penalty function becomes much smaller than those of Egeblad's overlap penalty function, when all polygons are convex or possible to divide a few convex elements.

3.1 Metaheuristics

It is often the case that local search (LS) alone may not attain a sufficiently good solution. To improve the situation, many variants of simple LS have been developed, and their frameworks are called *metaheuristics*. The *iterated local search* (ILS) and the *guided local search* (GLS) are representative metaheuristic approaches, which are simple but are known to be quite effective (Glover and Kochenberger, 2003). ILS repeats LS from different initial solutions generated by perturbing the best solution so far. GLS repeats on adaptive evaluation function which is adaptively modified to resume the search from the previous locally optimal solution. For the overlap minimization problem, we develop a hybrid approach of ILS and GLS, called the iterated guided local search (IGLS).

We introduce a variant of GLS called the *weighting method* which has been proposed by Selman and Kautz (1993) for the satisfiability problem (SAT). Based on preliminary computational experiments, we adopt a modified overlap penalty function for a pair of polygons $P_i, P_j \in \mathcal{P}$ as follows:

$$\tilde{f}_{ij}(\boldsymbol{t}, \boldsymbol{k}) = w_{ij} \cdot f_{ij}(\boldsymbol{t}, \boldsymbol{k}).$$

where w_{ij} is the penalty weight for a pair of polygons P_i and P_j. We also adopt the amount of the modified overlap penalty as follows:

$$\widetilde{F}(\boldsymbol{t}, \boldsymbol{k}) = \sum_{1 \le i < j \le n} \tilde{f}_{ij}(\boldsymbol{t}, \boldsymbol{k}).$$

The penalty weights w_{ij} are adaptively modified for every LS, i.e., if two polygons $P_i(k_i) \oplus \boldsymbol{t}_i$ and $P_j(k_j) \oplus \boldsymbol{t}_j$ overlap in the last locally optimal solution $(\boldsymbol{t}, \boldsymbol{k})$, GLS increases the penalty weight w_{ij} by one.

ILS starts from the first initial solution generated by random placement as same as the simple LS, and then the subsequent initial solutions are taken from their last locally optimal solutions except for n iterations of LS. In case of n iterations of LS, the next initial solution is generated by swapping the positions $\tilde{\boldsymbol{t}}_i^*$ and $\tilde{\boldsymbol{t}}_j^*$ of two polygons $P_i, P_j \in \mathcal{P}$ $(i \neq j)$ of the best solution $(\tilde{\boldsymbol{t}}^*, \tilde{\boldsymbol{k}}^*)$ on the adaptive evaluation function $\widetilde{F}(\cdot)$ obtained so far, where P_i and P_j are randomly selected from $\mathcal{P}$. If the new position $\boldsymbol{t}_i = \tilde{\boldsymbol{t}}_j^*$ (resp., $\boldsymbol{t}_j = \tilde{\boldsymbol{t}}_i^*$) of the polygon P_i (resp., P_j) is infeasible, ILS selects another polygon randomly.

The outline of the iterated guided local search (IGLS) is given as follows. Here, *iter* and *kick* denote the current number of iterations of restarting LS from the last improvement and the last perturbation of the initial solution, respectively. *max_iter* (an input parameter given by users) specifies the upper bound on *iter*.

Iterated guided local search

Step1: Set $iter \leftarrow 0$ and $kick \leftarrow 0$, and initialize w_{ij} for all pairs of $P_i, P_j \in \mathcal{P}$. Construct the first initial solution $(\boldsymbol{t}, \boldsymbol{k})$ by random placement, and set $(\boldsymbol{t}^*, \boldsymbol{k}^*) \leftarrow (\boldsymbol{t}, \boldsymbol{k})$ and $(\tilde{\boldsymbol{t}}^*, \tilde{\boldsymbol{k}}^*) \leftarrow (\boldsymbol{t}, \boldsymbol{k})$.

Step2: If $kick \geq n$ holds, apply a random perturbation to the best solution $(\tilde{\boldsymbol{t}}^*, \tilde{\boldsymbol{k}}^*)$ on the adaptive evaluation function $\widetilde{F}(\cdot)$, to obtain the next initial solution $(\boldsymbol{t}, \boldsymbol{k})$, and set $kick \leftarrow 0$.

Step3: Start LS on the adaptive evaluation function $\widetilde{F}(\cdot)$ from the initial solution $(\boldsymbol{t}, \boldsymbol{k})$, to obtain a locally optimal solution $(\boldsymbol{t}', \boldsymbol{k}')$.

Step4: Modify the penalty weights w_{ij} for all pairs of $P_i, P_j \in \mathcal{P}$. If $\widetilde{F}(\boldsymbol{t}', \boldsymbol{k}') < \widetilde{F}(\tilde{\boldsymbol{t}}^*, \tilde{\boldsymbol{k}}^*)$ holds, set $(\tilde{\boldsymbol{t}}^*, \tilde{\boldsymbol{k}}^*) \leftarrow (\boldsymbol{t}', \boldsymbol{k}')$.

Step5: If $F(\boldsymbol{t}', \boldsymbol{k}') < F(\boldsymbol{t}^*, \boldsymbol{k}^*)$ holds, set $(\boldsymbol{t}^*, \boldsymbol{k}^*) \leftarrow (\boldsymbol{t}', \boldsymbol{k}')$ and $iter \leftarrow 0$, and return to Step2.

Step6: If $iter \geq max\_iter$ holds, output $(\boldsymbol{t}^*, \boldsymbol{k}^*)$ and halt; otherwise set $iter \leftarrow iter + 1$ and $kick \leftarrow kick + 1$, and return to Step2.

4 Computational experiments

We conducted computational experiments for six well known benchmark instances (Table 1), which can be downloaded from the ESICUP web site[2].

Table 2 and 3 show that comparison of our iterated guided local search (IGLS) with three existing algorithms, 2DNEST (Egeblad *et al.*, 2006), SAHA (Gomes and Oliveira, 2006), and BLF-tabu (Burke *et al.*), in their best efficiency and computation time in seconds.

Table 1: The benchmark instances of the irregular strip packing problem

Instance	NDS	NTP	NAV	Degrees	Height
Blaz1	7	28	6.29	0,180	15
Shapes0	4	43	8.75	0	40
Shapes1	4	43	8.75	0,180	40
Shirts	8	99	6.63	0,180	40
Swim	10	48	21.90	0,180	5752
Trousers	17	64	5.06	0,180	79

NDS: The number of different shapes
NTP: The total number of polygons
NAV: The average number of vertices of different shapes

Table 2: Computational results of IGLS and other existing algorithms in the best length and efficiency

Instance	IGLS	2DNEST	SAHA	BLF-tabu
Blaz1	26.67	26.60	25.84	†27.20
	81.0%	81.20%	83.6%	†79.41%
Shapes0	60.45	59.47	60.0	65.00
	66.0%	67.09%	66.5%	61.38%
Shapes1	55.42	54.04	56.0	†58.40
	72.0%	73.83%	71.25%	†68.32%
Shirts	63.53	62.55	62.22	63.00
	85.0%	86.33%	86.79%	85.71%
Swim	6319.70	6184.37	5948.37	6462.40
	70.0%	71.53%	74.37%	68.45%
Trousers	250.35	242.44	242.11	243.40
	87.0%	89.83%	89.96%	89.48%

†They used a hill-climbing algorithm instead of the tabu search.

We measure efficiency by both the required length and the ratio of the total area of the polygons to the area of required strip $R(W)$. IGLS was run on each instance 10 times (Pentium IV 2.53GHz, 1GB memory), where the input parameter *max_iter* was set to 100. Egeblad *et al.* (2006) ran 2DNEST on each instance 20 times using 10 minutes for each run (Pentium IV 3GHz). Gomes and Oliveira (2006) ran SAHA on each instance 20 times (Pentium IV 2.4GHz). Burke *et al.* (2006) ran BLF-tabu on each instance 40 times (Pentium IV 2.0GHz). In Table 3, the columns of IGLS and SAHA show the average computation time, the column of 2DNEST shows the time limit of its computation, and the column BLF-tabu shows the computation time to find the best solution in the run found it.

SAHA shows the best results in the four algorithms; however, it spends much more computation time than the other algorithms. IGLS shows better results than those of BLF-tabu in efficiency and computation time except for Shirts and Trousers; however, it shows worse results than 2DNEST even taking account for the computation time. We note that it is not precise comparison of IGLS and the other algorithms, since IGLS solves the overlap minimization problem while the other algorithms solve the irregular strip packing problem. Nevertheless, IGLS finds no overlapping solution within short time for closely best efficiency in the literature. It is the future study to develop an efficient algorithm that minimizes the required strip length without any overlapping

[2] `http://www.apdio.pt/esicup/`

Table 3: Computational time of IGLS and other existing algorithms (in seconds)

instance	IGLS	2DNEST	SAHA	BLF-tabu
Blaz1	174.6	600	2257	†501.91
Shapes0	40.7	600	3914	1515.49
Shapes1	66.9	600	10314	†1810.14
Shirts	56.2	600	10391	806.5
Swim	243.5	600	6937	607.37
Trousers	105.7	600	8588	3611.99

†They used a hill-climbing algorithm instead of the tabu search.

polygon.

Fig. 4–8 show the best solutions for five benchmark instances, Blaz1, Shapes1, Shirts, Swim and Trousers.

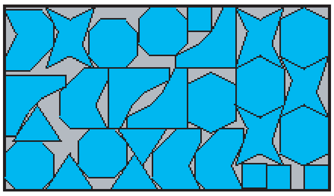

Figure 4: The best solution for the "Blaz1" instance

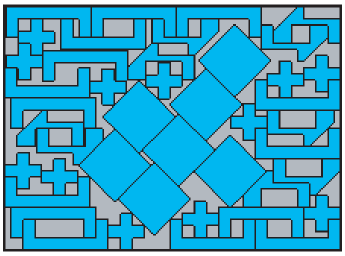

Figure 5: The best solution for the "Shapes1" instance

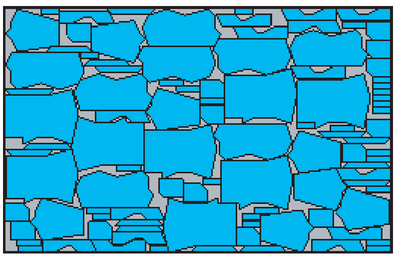

Figure 6: The best solution for the "Shirt" instance

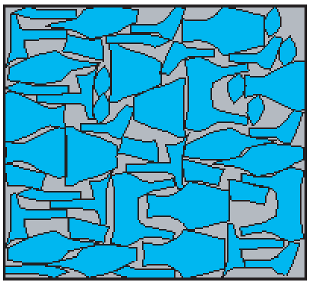

Figure 7: The best solution for the “Swim” instance

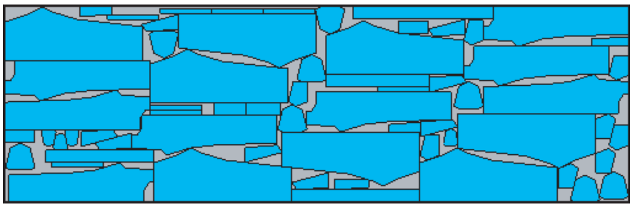

Figure 8: The best solution for the “Trousers” instance

5 Conclusions

In this paper, we present a local search algorithm for the irregular strip packing problem. We consider an overlap minimization problem which minimizes the amount of overlap penalty for all pairs of polygons within a given bound of strip length. We propose a fast neighborhood search which alternately translates a polygon in horizontal and vertical directions, so as to minimize the overlap penalty with the polygon. The neighborhood search can quickly compute a new position of the translating polygon by the projection checking and the no-fit polygon. We incorporate it in the iterated local search and the guided local search approaches. The computational results show that our algorithm attains competitive results to the best results previously published within shorter computation time.

References

Albano, A. and G. Sappupo. (1980). Optimal allocation of two-dimensional irregular shapes using heuristic search methods. *IEEE Transactions on Systems, Man and Cybernetics*, vol.SMC–10, pp.242–248.

Gomes, A.M. and J.F. Oliveira. (2002). A 2-exchange heuristic for nesting problems. *European Journal of Operational Research*, vol.141, pp.359–370.

Li, Z. and V. Milenkovic. (1995). Compaction and separation algorithms for non-convex polygons and their applications. *European Journal of Operational Research*, vol.84, pp.539–561.

Bennell, J.A. and K.A. Dowsland. (2001). Hybridising tabu search with optimisation techniques for irregular stock cutting. *Management Science*, vol.47, pp.1160–1172.

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

Burke, E.K., R.S.R. Hellier, G. Kendall and G. Whitwell. (2006). A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. *Operations Research*, to appear.

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

Voudouris, C. and E. Tsang. (1999). Guided local search and its application to the traveling salesman problem. *European Journal of Operational Research*, vol.113, pp.469–499.

Selman, B., and H. Kautz. (1993). Domain-independent extensions to GSAT: Solving large structured satisfiability problems. In: *Proceedings of 13th International Joint Conference on Artificial Intelligence*, pp.290–295.

de Berg, M. *et al.*. (1998). *Computational geometry — Algorithms and applications*, 2nd ed., Springer, pp.275–281.

Bennell, J.A., K.A. Dowsland and W.B. Dowsland. (2001). The irregular cutting-stock problem — a new procedure for deriving the no-fit polygon. *Computers and Operations Research*, vol.28, pp.271–287.

Burke, E.K., R.S.R. Hellier, G. Kendall and G. Whitwell. (2006). Complete and robust no-fit polygon generation for the irregular stock cutting problem. *European Journal of Operational Research*, to appear.

Glover, F. and G.A. Kochenberger eds. (2003). *Handbook of Metaheuristics*, Kluwer Academic Publishers.