Preprints of the 13th IFAC Symposium on Information Control Problems in Manufacturing, Moscow, Russia, June 3 - 5, 2009
INCOM MOSCOW IFAC 2009
Fr-A3.5

Comparison of Deterministic Heuristics and Simulated Annealing for the Rotational Placement Problem over Containers with Fixed Dimensions

Thiago de Castro Martins *
Marcos de Sales Guerra Tsuzuki *,[1]

* *Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil.*
Mechatronics and Mechanical Systems Engineering Department
Computational Geometry Laboratory
(e-mail: mtsuzuki@usp.br).

Abstract: Two dimensional packing problem arises in the industry whenever one must place multiple items inside a container such that there is no collision between the items, while either minimizing the size of the container or maximizing the area occupied by the items. High material utilization is of particular interest to mass production industries since small improvements of the layout can result in large savings of material and considerably reduce production cost. In this work the Simulated Annealing is combined with deterministic heuristics (larger first (LF), bottom left (BL) and translations only (Tr)) and compared. The rotational generic approach has discrete (sequence of placement) and continuous (angle and position) parameters. It is very important to notice that the cost function (non occupied space) has only discrete values.

Keywords: Computer vision, matched filter, invariant mapping.

1. INTRODUCTION TO PLACEMENT PROBLEMS

Several industrial problems involve placing objects into a container without overlap, with the goal of minimizing a certain objective function. The nesting problem is strongly NP-hard. Furthermore, the geometrical aspects of this problem make it really hard to solve in practice.

The global optimization algorithm Simulated Annealing (SA) has been proposed by Kirkpatrick et al. (1983) in the area of combinatorial optimization, that is, when the cost function is defined in a discrete domain. The SA algorithm was modified in order to apply to the optimization of functions defined in a continuous domain by Corana et al. (1987) using distinct steps according to temperature intervals. Martins and Tsuzuki (2008) proposed a SA algorithm with adaptive neighborhood that can be used with discrete and continuous domain.

The SA was applied to solve the placement problem through different strategies: Gomes and Oliveira (2006) proposed an algorithm where the sequence of placement is controlled by discrete parameters, Heckmann and Lengauer (1995) proposed an algorithm where the translation of items is controlled by continuous parameters. Martins and Tsuzuki (2007) proposed a generic approach to solve the placement problem by controlling simultaneously three types of parameters: sequence of placement, rotation and translation of items.

[1] This author was partially supported by CNPq and FAPESP.

This work turned off some of the controls and combined the SA with deterministic heuristics: larger first (LF), bottom left (BL) and translation only (Tr). A total of seven different strategies can be defined by combining probabilistic and deterministic heuristics.

2. SIMULATED ANNEALING

SA is the probabilistic meta–heuristic adopted in this work. It has been chosen due to its capacity of "escape" from local minima (which are very frequent in this problem). It is also worth of mention that the process of recrystallization, the inspiration for SA, is a natural instance of a placement problem.

SA is a hill-climbing local exploration optimization heuristic, which means it can skip local minima by allowing the exploration of the space in directions that lead to a local increase on the cost function. It sequentially applies random modifications on the evaluation point of the cost function. If a modification yields a point of smaller cost, it is automatically kept. Otherwise, the modification also can be kept with a probability obtained from the Boltzman distribution

$$P(\Delta E) = e^{-\frac{\Delta E}{kT}} \quad (1)$$

where $P(\Delta E)$ is the probability of the optimization process to keep a modification that incurs an increase ΔE of the cost function. k is a parameter of the process (analogous to the Stefan–Boltzman constant) and T is the instantaneous "temperature" of the process. This temperature is defined by a cooling schedule, and it is the main control parameter of the process.

The choice of the cooling schedule and of the next candidate distribution are typically the most important ones in the definition of a SA algorithm. In Bohachevsky et al. (1986), the next candidate point is obtained by first generating a random direction vector the multiplying it by a fixed step size, and summing the resulting vector to the current candidate point. The value of the step size should be chosen so that the percentage of accepted ascent steps with respect to the total number of generated ascent steps is about 60%. It is suggested to split the SA algorithm in two phases: a *global phase* in which the algorithm globally explores the feasible regions and a *local phase* which starts from the best point observed in the global phase (which, hopefully, is close enough to the global optimum). In the case of continuous domain, the local phase employs the same SA algorithm used in the global phase, but with a much lower value for the step size, which reduces the algorithm to a local exploration of the region around the best point observed in the global phase.

The directions are randomly sampled from the uniform distribution over the unit $(n-1)$ dimensional sphere, and the step size is the same in each direction. In this way the feasible region is explored in an isotropic way and it is assumed that the objective function behaves in the same way in every direction. But this is often not the case. The placement problem studied here has an objective function where different directions should have different step sizes, i.e. the space should be searched in an anisotropic way.

In the ASA proposed by Ingber (1989) the temperature is not only employed in the acceptance function, but also in the densities of the distribution of the next candidate point. ASA used the Cauchy distribution, that has a fatter tail allowing easier access to test local minima in the search for the global minimum. Consequently, the algorithm can do global exploration even at the local phase with a low probability.

The SA algorithm used in this work was proposed by Martins and Tsuzuki (2008) and it uses an adaptive neighborhood algorithm to define the densities of the distribution of the next candidate point.

3. DETERMINISTIC HEURISTICS

The great majority of the published articles are related to translation only placement problems (Wäscher et al. (2007)). Deterministic heuristics are mainly used in the definition of the placement order and item translation (Dowsland et al. (2002); Hifi and Hallah (2003); Jakobs (1996); Liu and Teng (1999)). In this section it is discussed the limitations of deterministic heuristics when applied to the placement problem on fixed dimensions container.

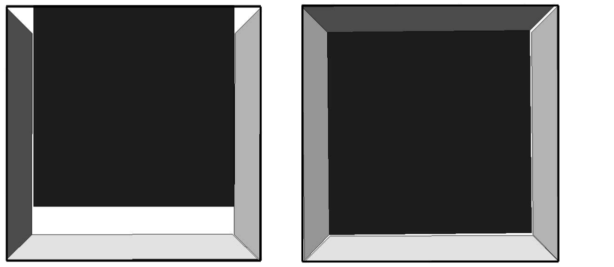
3.1 Placement Order

The main influence of the items sorting is evidently upon the selection of the subset of items that will be placed in the container. That happens because the items to be placed in the container from a certain position in the sequence find it already obstructed, and have a very low possibility of finding a collision–free placement. Still, the influence of the sorting on quality of the final layout itself should not be underestimated. This happens because of the way the sequential placement heuristics work. Those heuristics, in order to avoid dealing with the combinatorial nature of the packing problem, usually consider *only the items already placed*. As such, they try to place new items by estimating the needs of space of the items still unplaced. The consequence of such behavior is that those heuristics generate better layouts when the needs of space for the items to be placed are more predictable. As this prediction is particular to the placement heuristic, there is a strong coupling between the former and the placement order heuristic.

One family of sorting heuristics that frequently appear is composed of heuristics that sort the items in a decreasing size order. Kopardekar and Mital (1999) showed that those heuristics are based in observations of human experts that deal with packing problems. Those experts tend to place first the larger items in the container. Those heuristic are collectively called heuristics of LF. They differ by the particular definition of "size" adopted. This is the most studied problem in the literature. Many works on meta–heuristics applied to the cutting and packing problem apply the meta–heuristic exclusively to the placement order (Dowsland et al. (2002); Gomes and Oliveira (2002); Hallah et al. (2001); Hifi and Hallah (2003); Jakobs (1996)).

This relatively simple sorting heuristic obtains surprisingly good results in the majority of the problem instances. That can be explained by the previously exposed relation between the predictability of the space needs for the latter items and the final layout quality. It is easy to see that a large number of small items can be fitted in a larger variety of regions than a small number of large items of the same total area. As such, it is naturally easier to place small items in an already obstructed container.

As it is common with deterministic heuristics, it is easy to produce problem instances where the LF heuristics produce bad results. Figure 1 shows the best solution for a problem instance that cannot be obtained with a LF heuristic. For this problem, the translation heuristic always try to place the item using the no–fit polygon concept.



(a) (b)

Fig. 1. (a) Problem instance whose optimal solution cannot be reached through LF heuristic combined with the no–fit polygon for defining the translation. (b) Problem instance solved with a placement order produced by the SA.

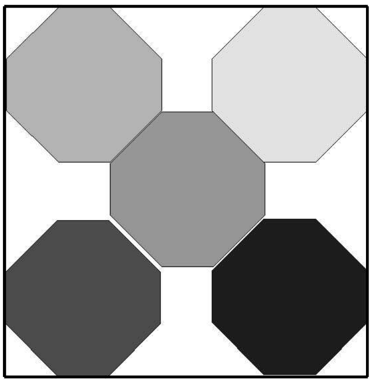


Fig. 2. Simple packing problem whose optimal solution *cannot* be reached through a translational BL heuristic. This problem instance was solved with translations produced by the SA.

3.2 Item Translation

The most commonly adopted deterministic heuristic for the item translation is the BL heuristic. On this approach, the items are placed in the container at the lowest leftmost free position available. The popularity of the BL heuristic can be understood by its low computational cost. Besides that, by naturally nesting the items next to the walls of the container (this is an heuristic applied almost exclusively to rectangular containers), it keeps a single large unobstructed area (in opposition to several small free areas), where a larger variety of items may be fitted. One must notice, that it is easy to produce packing problem instances where the global optimal is unreachable by the BL heuristic (see Fig. 2).

4. THE STUDIED ALGORITHM

Recently, researchers used the no–fit polygon concept to ensure feasible layouts; i.e. layouts where the items do not overlap and fit inside the container. This concept was first introduced by Art (1966). The no–fit polygon is used to efficiently avoid overlapping among the items and to place them inside the container. The algorithm studied in this work is a constructive approach with discrete and continuous parameters controlled by the SA.

The items are sequentially placed, one at a time. When placing an item a rotation is applied and the item is placed on the boundary of its no–fit polygon. The item placement is described in details by Martins and Tsuzuki (2006, 2007). The rotation, the translation and the sequence are controlled by the SA algorithm. The complete algorithm is referenced as NONE in the tables. This placement algorithm can be adapted to use deterministic heuristics. The LF adaptation defines initially an ordered sequence of items that will never change during the optimization. The BL adaptation places the item on the lowest leftmost vertex from its no–fit polygon.

It is important to notice that the placement problem on fixed dimensions containers has a particularity that increases the difficult of its approach with traditional optimization techniques, the fact that its cost function (the non–occupied space) assumes only discrete values, while its parameters are continuous. A related dual problem that is the problem of, given a set of items, find the smallest container where the whole set can be placed has a much larger coverage in the literature (Wäscher et al. (2007)).

To make the cost function smoother, the cost of a given solution can be modified in order to reflect how close this solution is to having a non–placed item fitted in the container. Martins and Tsuzuki (2006, 2007) proposed a solution where for each non–placed item, a limited–depth binary search is performed to find a scale factor (between 0 and 1) that, when applied to the item, would allow it to be fitted in the container.

5. RESULTS

All problem instances studied here have a solution where all items can be fitted in the container. The optimization method was implemented in C++ using a modified version of the *PolyBoolean* library developed by Leonov (1998). The vertices of the polygons can assume only discrete values. The random–number generation uses the Mersene-Twister generator proposed by Matsumoto and Nishimura (1998).

When evaluating the algorithm performance from the obtained results, one must take in account the fact that usually, a solution as good as the final one is found in much less iterations than it takes for the algorithm to converge. Of course, letting the algorithm take its course is the only generic way to know if any previously found solution will be the best found, but this suggests that an algorithm that keeps track of the best found solution may be *interrupted* and still return a satisfactory solution. The adopted convergence condition is that the algorithm stops when during 10 consecutive temperatures the simulated annealing algorithm accepted only solutions equivalent to the best found solution. All tests were executed in a 2.21 GHz Phenom 9550 processor. Every example was executed 30 times for each depth of the binary search.

5.1 LF Fails Puzzle

The LF fails puzzle consists of the placement of five convex items. Figure 1.(b) shows the final solution of this problem. The SA algorithm avoided the LF deterministic heuristic.

One can observe from Table 1 that there is a large difference between scaled and not scaled runs for the rotational placement. The translational placements reached the global minimum at every run and it was reached in approximately 124 times fewer iterations and 350 times smaller execution time when compared to the respective rotational heuristic. The number of runs that reached the global minimum for rotational placement are smaller when compared to translational placement, as a consequence of the strong discretization of the cost function. By modifying the convergence condition to 50 consecutive temperatures (instead of 10), this example showed a much better convergence to the global minimum.

5.2 BL Fails Puzzle

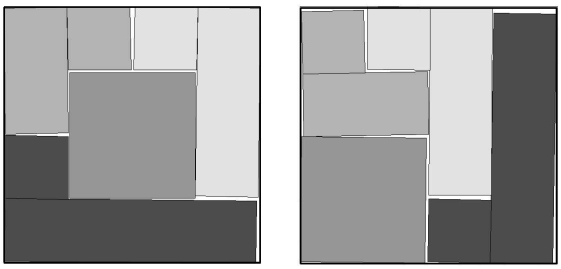
The BL fails puzzle consists of the placement of five identical convex items. Figure 2 shows the final solution of

Table 1. Statistics for the rotational LF fails puzzle. The columns represent respectively the adopted deterministic heuristics, the fixed depth of the binary search, converged energy level, number of iterations to converge, time in seconds to converge, and the percentage of runs that converged to the global optimum. The container area is 2.4% larger than the total area of the items. $\alpha = 0.99$.

	Scale	A_{conv}	N_{conv}	T_{conv}	P_{conv}
BL	0	20.9	34094	17.2	0.0%
	1	2.4	129386	227.3	100.0%
	2	2.8	78123	227.2	93.3%
NONE	0	22.4	23168	9.1	0.0%
	1	9.8	168559	290.6	56.7%
	2	5.1	87328	279.5	80.0%
TrBL	0	2.4	774	0.2	100.0%
	1	2.4	787	0.6	100.0%
	2	2.4	394	0.5	100.0%
Tr	0	2.4	759	0.2	100.0%
	1	2.4	736	0.5	100.0%
	2	2.4	743	0.9	100.0%

Table 2. Statistics for the translational BL fails puzzle. The container area is 29.8% larger than the total area of the items. $\alpha = 0.99$.

	Scale	A_{conv}	N_{conv}	T_{conv}	P_{conv}
Tr	0	29.8	1501	0.5	100.0%
	1	29.8	1517	1.3	100.0%
	2	29.8	1630	2.5	100.0%
TrLF	0	29.8	1462	0.5	100.0%
	1	29.8	1491	1.4	100.0%
	2	29.8	1646	2.5	100.0%



(a) (b)

Fig. 3. The small puzzle solved with SA applied to the rotations of the items. The result shown in (a) used a BL heuristic for the translations and the result shown in (b) used only probabilistic heuristics. The non–convex items were manually decomposed into convex polygons.

this problem. One can observe from Table 2 the execution time increased compared to the value of the depth search. The global minimum was reached at every run.

5.3 Small Puzzle

This example is a fairy simple puzzle with four non–convex non–congruent polygons. The non–convex items are manually decomposed into convex polygons in a pre–processing step. This decomposition does not affect the final solution. Figures 3.(a) and 3.(b) show final solutions of this problem.

Table 3. Statistics for the rotational small puzzle. The container area is 3.4% (2.0%) larger than the total area of the items for rotational (translational) placement. $\alpha = 0.99$.

	Depth	A_{conv}	N_{conv}	T_{conv}	P_{conv}
BL	0	7.6	333724	214.4	80.0%
	1	4.9	376969	873.5	90.3%
	2	3.9	336757	1395.1	96.8%
NONE	0	19.4	310905	151.0	33.3%
	1	3.9	352691	820.3	96.7%
	2	3.4	319794	1246.9	100.0%
LF	0	6.7	300547	132.1	83.3%
	1	3.4	296675	700.5	100.0%
	2	3.4	290491	1147.3	100.0%
BLLF	0	4.8	327544	247.2	93.3%
	1	3.9	315758	842.5	96.8%
	2	3.4	329677	1291.6	100.0%
TrBL	0	2.0	7892	1.3	100.0%
	1	2.0	7783	4.6	100.0%
	2	2.0	8202	1.4	100.0%
Tr	0	2.0	7789	1.2	100.0%
	1	2.0	8135	5.3	100.0%
	2	2.0	8131	9.2	100.0%
TrLF	0	2.0	8218	0.9	100.0%
	1	2.0	7999	4.9	100.0%
	2	2.0	8051	9.0	100.0%

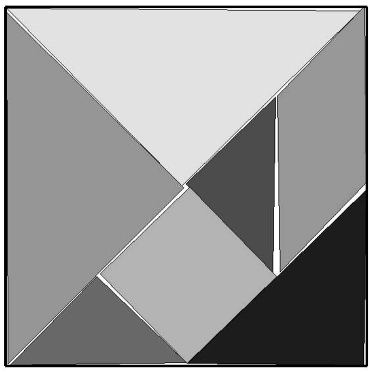


Fig. 4. Final solution of a tangram puzzle with 7 items.

One can observe from Table 3 that deeper depth searches improved the optimization process for the rotational placement, as a higher percentage of runs reached the global minimum. The convergence ratio of the generic rotational with no scaling showed to be very poor, the convergence ratio increased as the value of the depth search increased. The translational placement reached the convergence in 179 times smaller execution time and 40 times fewer iterations when compared to the respective rotational placements.

5.4 Tangram Puzzle

The tangram puzzle consists of the placement of seven convex non–congruent items. Figure 4 shows the final solution of this problem.

One can observe from Table 4 the execution time increased compared to the value of the depth search. The translational placement reached the global minimum at every execution and it was reached in approximately 4290 times fewer iterations and 968 times smaller execution time when compared to the respective rotational placement.

Table 4. Statistics for the rotational tangram. The container area is 3.3% (0.2%) larger than the total area of the items for rotational (translational) placement. $\alpha = 0.99$. It is impossible to reach the global optimum through the BLLF deterministic heuristic.

	Scale	A_{conv}	N_{conv}	T_{conv}	P_{conv}
BL	0	11.6	1222277	1102.7	36.7%
	1	6.5	1855716	4136.6	76.7%
	2	5.9	1566501	5164.2	76.7%
NONE	0	11.8	1224384	1209.6	33.3%
	1	7.1	1648268	3717.4	76.7%
	2	9.8	2024080	6593.5	38.5%
LF	0	8.2	1062867	901.6	70.0%
	1	8.4	1885168	4268.0	56.7%
	2	6.8	1443964	5165.9	76.7%
TrBL	0	0.2	3161	1.7	100.0%
	1	0.2	3258	3.4	100.0%
	2	0.2	3688	5.9	100.0%
Tr	0	0.2	3774	1.9	100.0%
	1	0.2	4000	4.5	100.0%
	2	0.2	4000	6.8	100.0%
TrLF	0	0.2	3260	1.1	100.0%
	1	0.2	3666	2.9	100.0%
	2	0.2	4065	5.2	100.0%

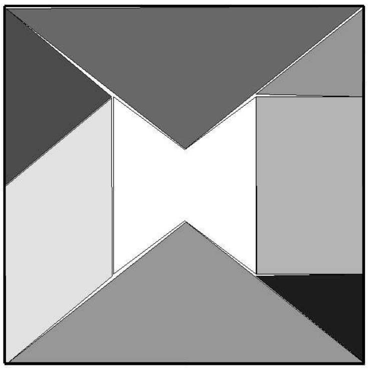


Fig. 5. Container with a hole puzzle.

5.5 Container with a Hole Puzzle

The container with a hole is a variation from the tangram puzzle. The holes are placed as normal polygons that are placed first and are not manipulated by the SA algorithm. The non–convex holes are manually decomposed into convex polygons in a pre–processing step. This decomposition does not affect the final solution. Figure 5 shows the final solution of this problem.

One can observe from Table 5 the execution time increased compared to the value of the depth search. The rotational and translational placements reached the global minimum at almost every execution and it was reached in approximately 65 times fewer iterations and 47 times smaller execution time when compared to the respective rotation heuristic.

5.6 Irregular Shaped Container Puzzle

The problem shown in Fig. 6 illustrates that the algorithm can deal with non–convex containers and items. The non–convex container is represented as a convex container with two holes touching its boundary.

Table 5. Statistics for the rotational container with a hole. The container area is 3.4% larger than the total area of the items. $\alpha = 0.99$.

	Scale	A_{conv}	N_{conv}	T_{conv}	P_{conv}
BL	0	4.0	558786	517.9	90.0%
	1	3.4	100107	309.9	100.0%
	2	3.4	152724	765.5	100.0%
NONE	0	3.4	170089	159.2	100.0%
	1	3.4	93903	262.7	100.0%
	2	3.4	73791	318.9	100.0%
LF	0	3.4	34945	41.2	100.0%
	1	3.4	26200	69.9	100.0%
	2	3.4	30150	124.7	100.0%
BLLF	0	3.4	21490	29.5	100.0%
	1	3.4	22072	70.8	100.0%
	2	3.4	22473	106.7	100.0%
TrBL	0	4.8	1160	2.1	96.7%
	1	4.3	1427	5.0	83.3%
	2	3.7	1447	7.5	90.0%
Tr	0	3.4	800	1.0	100.0%
	1	3.4	987	2.8	100.0%
	2	3.4	768	3.1	100.0%
TrLF	0	3.4	2453	2.4	100.0%
	1	3.4	3250	10.0	100.0%
	2	3.4	2880	10.6	100.0%

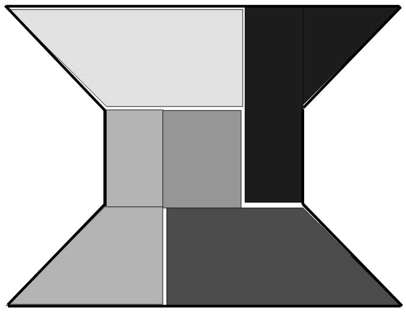


Fig. 6. Puzzle with a irregular shaped container.

Table 6. Statistics for the rotational irregular shaped container puzzle. The container area is 3.9% larger than the total area of the items. $\alpha = 0.99$. It is impossible to reach the global optimum through the BLLF deterministic heuristic.

	Scale	A_{conv}	N_{conv}	T_{conv}	P_{conv}
BL	0	23.7	295871	295.3	26.7%
	1	12.6	409522	1669.4	60.0%
	2	9.7	370431	2536.6	73.3%
NONE	0	17.7	241374	164.4	46.7%
	1	5.3	298855	1196.6	93.3%
	2	5.0	262793	1900.3	96.7%
LF	0	20.5	216075	175.2	46.7%
	1	8.9	322785	1459.3	83.3%
	2	7.7	289439	2091.3	86.7%
TrBL	0	3.9	800	0.2	100.0%
	1	3.9	800	1.9	100.0%
	2	3.9	800	3.5	100.0%
Tr	0	3.9	797	0.2	100.0%
	1	3.9	790	1.3	100.0%
	2	3.9	810	2.4	100.0%
TrLF	0	3.9	769	0.2	100.0%
	1	3.9	796	1.5	100.0%
	2	3.9	814	2.6	100.0%

One can observe from Table 6 the execution time increased compared to the value of the depth search. The translational placement reached the global minimum at every execution and it was reached 850 times fewer iterations and 377 times smaller execution time when compared to the rotational placement.

6. DISCUSSION

On problems without limitations on the nature of the items (convex and non–convex), the rotation establishment becomes a combinatorial problem of difficult solution. As shown by Chazelle (1983) that proposed an algorithm $O\left[p^3q^3(p+q)log(p+q)\right]$ for determining if a given polygon P (with p vertices) can fit into a polygon Q (with q vertices), where translations and/or rotations are allowed. Chazelle (1983) proposed also an algorithm $O\left(p+q\right)$ for determining if a given polygon P (with p vertices) can fit into a polygon Q (with q vertices), where only translations are allowed. This combinatorial difficulty explains that the convergence ratio to the global minimum of the rotational placement was smaller when compared to the translational placement and the greater processing time spend by the rotational placement when compared to the translational placement.

Each example has its own feature: fails with the LF strategy, fails with the BL strategy, non convex items, non convex container and holes inside the container. The majority of the rotational placements have the convergence ratio improved for deeper depth searches.

Considering the convergence ratio for rotational placements, the BL strategy (when possible) had a better performance, following the generic probabilistic approach and the LF strategy. The combination of LF and BL (when possible) converged to the global minimum at every execution.

7. CONCLUSIONS

This work deals with the problem of minimizing the waste of space that occurs on rotational and translational placements of a set of irregular bi–dimensional items inside a bi–dimensional container with fixed dimensions. The placement of an item is controlled by the following SA parameters: the rotation applied, the placement of items and the sequence of placement. Discrete and continuous parameters are present. The rotational placement is a complex combinatorial problem that showed a good convergence ratio when combined with deterministic heuristics.

REFERENCES

Art, R.C. (1966). An approach to the two-dimensional irregular cutting stock problem. Technical report, IBM Cambridge Scientific Centre.

Bohachevsky, I.O., Johnson, M.E., and Stein, M.L. (1986). Generalized simulated annealing for function optimization. *Technometrics*, 28, 209–217.

Chazelle, B. (1983). The polygon containment problem. *Advances in Computing Research*, 1, 1–33.

Corana, A., Marchesi, M., Martini, C., and Ridella, S. (1987). Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Transactions on Mathematical Software*, 13, 262–280.

Dowsland, K.A., Vaid, S., and Dowsland, B.W. (2002). An algorithm for polygon placement using a bottom–left strategy. *European Journal of Operational Research*, 141, 371–381.

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

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

Hallah, R.M., Bouziri, A., and Jilani, W. (2001). Layout of two dimensional irregular shapes using genetic algorithms. *Lecture Notes In Computer Science*, 2070, 403–411.

Heckmann, R. and Lengauer, T. (1995). A simulated annealing approach to the nesting problem in the textile manufacturing industry. *Annals of Operations Research*, 57, 103–133.

Hifi, M. and Hallah, R.M. (2003). Hybrid algorithm for the two-dimensional layout problem: the cases of regular and irregular shapes. *International Transaction in Operational Research*, 10, 195–216.

Ingber, L. (1989). Very fast simulated re-annealing. *Mathematical and Computer Modelling*, 12, 967–973.

Jakobs, S. (1996). On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88, 165–181.

Kirkpatrick, S., Gellat, C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.

Kopardekar, P. and Mital, A. (1999). Cutting stock problem: a heuristic solution based on operator's intuitive strategies. *Journal of Computer Integrated Manufacturing*, 12, 371–382.

Leonov, M.V. (1998). *Implementation of Boolean operations on sets of polygons in the plane*. Master's thesis, Novosibirsk State University.

Liu, D. and Teng, H. (1999). An improved bl–algorithm for genetic algorithms of the orthogonal packing of rectangles. *European Journal of Operational Research*, 112, 413–420.

Martins, T.C. and Tsuzuki, M.S.G. (2006). Simulated annealing applied to the rotational polygon packing. In *Proceedings of the IFAC Symposium Information Control Problems in Manufacturing*, 475–480.

Martins, T.C. and Tsuzuki, M.S.G. (2007). Irregular rotational placement of shapes over non-convex containers with fixed dimensions. In *Proceedings of the IFAC Workshop on Intelligent Manufacturing Systems*.

Martins, T.C. and Tsuzuki, M.S.G. (2008). Simulated annealing with adaptive neighborhood applied to the placement over containers with fixed dimensions. In *Proceedings of the IFAC Workshop on Intelligent Manufacturing Systems*, 159–164.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8, 3–30.

Wäscher, G., Haussner, H., and Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183, 1109–1130.