BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 57, No. 3, 2009

Placement over containers with fixed dimensions solved with adaptive neighborhood simulated annealing

T.C. MARTINS and M.S.G. TSUZUKI*

Mechatronics and Mechanical Systems Engineering Department, Computational Geometry Laboratory, Escola Politécnica da Universidade de São Paulo, 2231 Prof. Mello Moraes Av., Cidade Universitária, São Paulo, Brazil

Abstract. This work deals with the problem of minimizing the waste of space that occurs on a rotational placement of a set of irregular bi-dimensional items inside a bi-dimensional container. This problem is approached with a heuristic based on Simulated Annealing (SA) with adaptive neighborhood. The objective function is evaluated in a constructive approach, where the items are placed sequentially. The placement is governed by three different types of parameters: sequence of placement, the rotation angle and the translation. The rotation applied and the translation of the polygon are cyclic continuous parameters, and the sequence of placement defines a combinatorial problem. This way, it is necessary to control cyclic continuous and discrete parameters. The approaches described in the literature deal with only type of parameter (sequence of placement or translation). In the proposed SA algorithm, the sensibility of each continuous parameter is evaluated at each iteration increasing the number of accepted solutions. The sensibility of each parameter is associated to its probability distribution in the definition of the next candidate.

Key words: simulated annealing, placement problems, optimization, probabilistic heuristics.

1. Introduction to placement problems

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. Low 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.

The global optimization heuristic Simulated Annealing (SA) has been proposed in the area of combinatorial optimization [1], that is, when the objective function is defined in a discrete domain. The method is reported to perform well in the presence of a very high number of variables (even tens of thousands). The SA heuristic was modified in order to apply to the optimization of multimodal functions defined on continuous domain [2]. The choice of the cooling schedule and of the next candidate distribution are the most important decisions in the definition of a SA algorithm [3]. Corana et al. [2] proposed a self tuning SA algorithm which step size is configured in order to maintain a number of accepted solutions. The probability distribution used by Corana et al. [2] is flat. Ingber [4] proposed to use a Cauchy distribution and a faster temperature decreasing. The very fast simulated reannealing proposed by Ingber and Rosen [5] searches for different sensitivities on parameters by processing the first derivatives of the objective function. The SA algorithm proposed in this work uses a Gaussian probability distribution with a self- controlled standard deviation in order to maintain the number of accepted solutions.

The SA has been applied to solve the placement problem through different strategies: sequence of placement controlled by discrete parameters [6], and translation of items controlled by continuous parameters [7]. The SA algorithm from each application, controlled only one kind of parameter; however as shown in [8] a generic approach to solve the placement problem must control simultaneously three different types of parameters: sequence of placement, rotation and translation of items.

2. The problem

According to [9], the packing problems are mainly characterized by the number of relevant dimensions, the regularity and irregularity of the shapes of the items and containers and the problem assignment. Considering assignment, it is possible to identify two situations: output maximization and input minimization. In the input minimization, the set of containers is sufficient to accommodate all items, and there is no selection regarding items. In the case of output maximization, the set of containers is not sufficient to accommodate all items and a set of items has to be assigned to a given set of containers. The typology of cutting and packing problems proposed in [9] was improved by [10]. According to [10], the problem studied in this work is classified as two dimensional irregular single knapsack problem. In this survey, they identified 413 papers containing material relevant to cutting and packing. Only fourteen papers thereof were classified as dealing with two dimensional irregular single knapsack problem. This fact shows that the literature related to this specific kind of problem remains relatively scarce.

*e-mail: mtsuzuki@usp.br

The problem studied here is the rotational placement of multiple items inside a unique container with fixed dimensions (problems dealing with fixed container space will be labeled here as "primal problems"). It can be defined as the problem of, given a container (a convex or concave polygon) and a set of n items (convex or concave polygons), determining the subset of items and transformations (translations and rotations) to be applied to the respective items. The final layout must fit inside the container without overlapped items, and such that the occupied area is maximized.

2.1. Objective function behavior and duality. The placement problem on fixed dimensions containers has 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 (this problem is often labeled as the cutting stock problem). As pointed by [10], this dual problem has a much larger coverage in the literature. Examples of studies related to this dual problem are [11] and [12]. Jakobs [11] studied the orthogonal packing based on a bottom-left strategy to position the items, the placement sequence is determined using a genetic algorithm. He extended the algorithm to process irregular items. The main idea is the determination of the embedding rectangles with minimum area for all items. The rotation of the items is determined in this local search. Hifi and Hallah [12] proposed a hybrid algorithm to solve the irregular problem. The hybrid algorithm searches for an optimal ordering of the items using a genetic algorithm and identifies the best packing using a constructive approach, which consists of sequentially positioning a set of ordered items. Each item is tested for a set of potential positions defined with respect to already positioned items. They studied an exclusively translational problem and considered the items as rectangles for positioning.

When compared to its dual variant, 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 objective function (the non-occupied space) assumes only discrete values, while its variables (the translations and rotations for each item) are continuous. Figure 1 illustrates a dual optimization problem with a single variable (the rotation applied to the leftmost item), on which the objective is to minimize the width of the container $f(\theta)$. As one can see, the width varies continuously with the optimization variable θ.

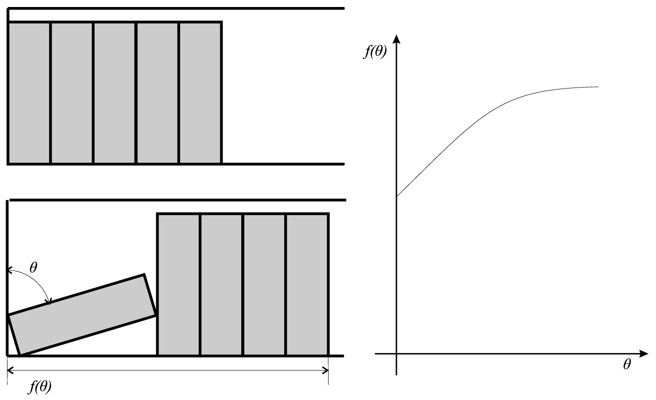

Fig. 1. Objective function behavior on the dual problem

For the primal problem shown in Fig. 2, the objective function is the non- occupied space inside the container. As this space can change only by adding or removing areas of items, the objective function can assume only a finite set of values, becoming discontinuous. This particularity of the primal problem makes it difficult to evaluate the sensibility of the objective function related to the optimization variables.

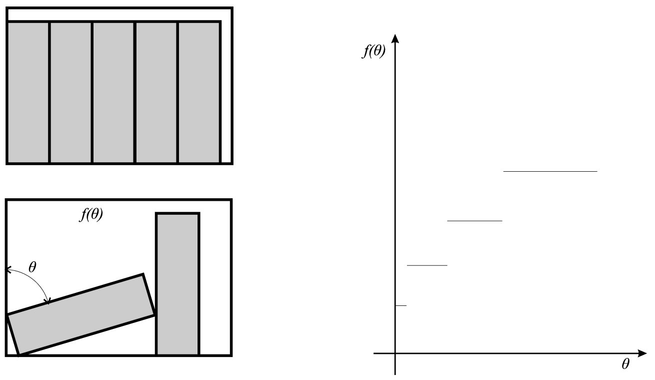

Fig. 2. Objective function behavior on the primal problem

2.2. No-fit polygon. The application of meta-heuristics to the placement problem usually encounters one severe difficulty, the enormous complexity of the space of feasible solutions. A technique usually employed to mitigate such difficulty is the external penalisation. It consists of relaxing constraints on the original problem and adapt a modification to the cost function in order to penalize (increase the cost of) non-feasible solutions. While such technique has been applied to the placement problem with some degree of success, it often leads to invalid layouts that require post-processing [7].

The non-fit polygon has been recently applied by researchers to ensure the generation of collision-free layouts. This concept was first introduced by [13]. Given two polygons, A and B, the no-fit polygon uses a reference point for the moving polygon. The no-fit polygon is the set of points where positioning the reference point imply that polygon B collides with polygon A. The no-fit polygon can be found by tracing one polygon around the boundary of another. One of the polygons remains fixed in position and the other traverses around the fixed polygon edges whilst ensuring that the polygons always touch but never intersect [14]. The no-fit polygon has been used exclusively in the dual problem [6, 15]. Dowsland et al. [15] used the no-fit polygon and a bottom-left strategy as an application of deterministic heuristic. An item is placed in the boundary of the no-fit polygon according to the bottom-left strategy. The positioning sequence is defined by a sorting criteria: decreasing area, decreasing length, decreasing width and others. They studied an exclusively translational problem. Gomes and Oliveira [6] used the no-fit polygon concept to eliminate the overlap in the definition of the initial solution. SA is used to define which item exchange position and with which orientation. After exchanging position two items in the layout, overlap usually occurs, which is removed by applying a separation model, i.e., a set of translations is applied to the overlapping items. Afterwards, the layout is compacted by a set of translations applied to the placed items, achieving layouts that are local minima. However, it is possible that the separation model fails in achieving a feasible layout. In this case, the proposed algorithm ignores the failed swap operation and attempts exchanging two different items.

3. 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 objective function. It sequentially applies random modifications on the evaluation point of the objective function. If a modification yields a point of smaller objective function value, it is automatically kept. Otherwise, the modification also can be kept with a probability obtained from the Boltzman distribution

$$P(\Delta E) = e^{-\Delta E/kT}, \tag{1}$$

where $P(\Delta E)$ is the probability of the optimization process to keep a modification that incurs an increase ΔE (ascendent steps) of the objective 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 probability of a given state decreases with its energy, but as the temperature rises, this decrease (the slope of the curve $P(\Delta E)$) diminishes. Kirkpatrick et al. [1] have applied the SA to a variety of combinatorial problems, such as the traveling salesman and computer circuit design problems. In these cases, the parameters are taken on discrete values; the next point candidate $\mathbf{x}_{k+1}$ corresponds to a permutation in the list of cities to be visited, interchanges of circuit elements, or other discrete operation. The principal complication introduced in going from the discrete to the continuous application of SA is that the choice of random steps becomes more subtle. In general, the optimal magnitude and directions of the steps are not known in advance. In [16], the next point candidate $\mathbf{x}_{k+1}$ is obtained by first generating a random direction vector $\mathbf{u}$, with $||\mathbf{u}|| = 1$, then multiplying it by a fixed step size Δr, and summing the resulting vector to the current candidate point $\mathbf{x}_k$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta r \cdot \mathbf{u}. \tag{2}$$

In [17], the same next candidate distribution as in [16] is employed, but it is shown that the selection of Δr is a critical choice. An appropriate choice of this parameter is strictly dependent on the objective function $F(\mathbf{x})$, and that bad choices may lead to a deterioration of the performance of the algorithm. In [17] it is suggested to choose appropriate values by presampling the objective function. It is suggested to split the 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) and employs the same SA algorithm but with a much lower value for the step size Δr, which reduces the algorithm to a local exploration of the region around the best point observed in the global phase. The first phase should return a rough approximation of the global optimum, while in the second phase the approximation is refined. In [18] the importance of an appropriate choice for the step size Δr is further discussed, but instead of choosing it by presampling the objective function, the step size is adaptively updated according to the distance from the global optimum value (or an estimate of it), and it is defined as

$$\Delta r_k = \beta \cdot (F(\mathbf{x}_k) - F^*)^{g1}, \tag{3}$$

where $\beta > 0$ and $g1 > 0$ are constants. For the cases in which the global optimum value F^* is not know, it is suggested to employ an estimate $\widehat{F}$ of it. It is also proposed to generate the next candidate point according to a distribution which is not uniform over the hypersphere with radius equal to the step size Δr_k. It is suggested to take into account the last point $\mathbf{x}_h$, $h < k$, generated by the algorithm and different from $\mathbf{x}_k$. If $F(\mathbf{x}_h) < F(\mathbf{x}_k)$ the distribution favors the direction $\mathbf{x}_h - \mathbf{x}_k$ with respect to the other directions, while if $F(\mathbf{x}_h) > F(\mathbf{x}_k)$ the opposite direction is favored.

The directions in [16, 17] are randomly sampled from the uniform distribution over the unit $(n-1)$ dimensional sphere, and the step size if 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 not often the case. By decreasing the step size Δr_k, the probability of accepting the new candidate point is increased. But, in order to bring this probability to a reasonable value, one may be obliged to considerably reduce the step size, especially in an ill conditioning situation. In this way each iteration of the algorithm can only perform painfully small steps, even in directions along which the function varies very slowly and larger steps would allow a faster approach to the global optimum. All this suggests that the step sizes to define the next candidate point $\mathbf{x}_{k+1}$ should not be all equal for all the directions, but different directions should have different step sizes, i.e. the space should be searched in an anisotropic way. In particular, the support of the next candidate point should reflect as much as possible the shape of the objective function in the region currently explored. In [5] the concept of anisotropic search through the feasible region is introduced. At each iteration k a single variable of $\mathbf{x}_k$ is modified in order to obtain a new candidate point $\mathbf{x}_{k+1}$, and iterations are subdivided in cycles of n iterations during which each variable is modified. More formally, let $i \in \{0, ..., n-1\}$ be such that

$$i = h \cdot n + i, \tag{4}$$

for some nonnegative integer h. Then $\mathbf{x}_{k+1}$ is obtained from $\mathbf{x}_k$ by changing only the i-th variable of $\mathbf{x}_k$, i.e.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + u \cdot \Delta r_i \cdot \mathbf{e}_i, \tag{5}$$

where u is a uniform random number in $[-1, 1]$, and Δr_i is the maximum allowed step size along the direction $\mathbf{e}_i$ of the i-th axis. The anisotropy is obtained by choosing different values v_j for all the n directions $\mathbf{e}_j$, $j = 1, ..., n$. The values v_j are not kept fixed for N_s cycles of variables (i.e. for $n \cdot N_s$ iterations), where N_s is some positive integer. After the N_s cycles the values are updated. In particular, if the fraction of accepted moves in the direction $\mathbf{e}_j$ with respect to the total number N_s of moves generated in the same direction is below $0 : 4$, then the step size Δr_j along $\mathbf{e}_j$ is decreased (too large steps along $\mathbf{e}_j$ have been performed thus causing most steps to be rejected); if the fraction is above $0 : 6$ then Δr_j is increased (too small steps have been performed in the direction $\mathbf{e}j$ thus causing most steps to be accepted); if the fraction is between $0 : 4$ and $0 : 6$ the step size is left unchanged. In this way, if F slowly varies with respect to changes in the j-th variable, the step size Δr_j becomes large, while if F quickly varies with respect to changes in the j-th variable, the step size becomes small. In this way some anisotropy is introduced in the search through the feasible region. The procedure described above may not be the best possible to keep into account the different behavior of the objective function along different directions. Indeed, a cycle of variables returns a point inside an axis aligned hyperrectangle with center in the current point at the beginning of the cycle, while in some situations, in order to resemble the level curves of F, we should sample points from an ellipsoid which is not axis aligned.

In [4] the so called ASA has been developed and discussed. In the ASA algorithm the temperature is not only employed in the acceptance function, Eq. (1), but also in the densities g_k of the distribution of the next candidate point $\mathbf{x}_{k+1}$. A possible choice is the Gaussian distribution as following:

$$g_k\left(\Delta r, s_k\right) = \frac{1}{\sqrt{2\pi s_k}} \cdot e^{-\frac{\|\Delta r\|}{2s_k}}, \tag{6}$$

where Δr is the maximum step size from the current point $\mathbf{x}_k$ and s_k is the standard deviation. Therefore, the new candidate point $\mathbf{x}_{k+1}$ is obtained by sampling a vector $\mathbf{u}$ from the density, and then by setting

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u} \cdot \Delta r. \tag{7}$$

An alternative is the so called Cauchy distribution [5] whose density is defined as follows

$$g_k\left(\Delta r, s_k\right) = \frac{s_k}{\left(\|\Delta r\|^2 + s_k^2\right)}. \tag{8}$$

The Cauchy distribution has a fatter tail with respect to Eq. (6). This means that if one employs the same temperature, the Cauchy distribution will more likely generate a smaller deviation, thus favoring a local exploration of the feasible region but penalizing the global exploration. Whenever doing a multi-dimensional search in the course of a real-world nonlinear physical problem, inevitably on must deal with different sensitivities of the parameters in the search [4]. If the objective function is very sensitive to changes in the i-th parameter, then the value s_k should decrease quickly, thus allowing the algorithm to perform small steps in the i-th direction; if the objective function is not very sensitive to changes in the i-th variable, then the value s_k should decrease slowly, thus allowing large steps in the i-th direction. According to the already mentioned concept of local adaptivity, the decrease of the temperature is not a priori fixed but is adaptively revised. The revision takes place periodically through the so called reannealing operation. The revision is done by computing objective function derivatives.

4. The proposed algorithm

This work uses the no-fit polygon in order to efficiently generate overlap-free layouts. As the no-fit polygon is related with the interference between two polygons, this requires an approach of sequential placements of the items, where each placement takes in consideration the area obstructed by the items already placed. Also, evidently for a rotational placement problem, the rotation of both the item to be placed and the obstacles must be known in order to calculate the collision-free area.

As such, the item placement is controlled by three different parameters: its position in the placement sequence, its rotation and finally its translation [8, 19]. The placement sequence is a combinatorial parameter, as described by Kirkpatrick et al. [1]. The rotation and translation are continuous parameters belonging to the interval $[0,\ 1[$. While the rotation parameter represents straightforwardly the final rotation of the item, the translation is necessarily more complex, as the item must be placed inside a region (the collision-free region) which is not determinated at the time when the parameters are generated. The translation parameter corresponds to an uniform mapping of the *perimeter* of the collision-free region to the interval $[0;\ 1[$. When the collisionfree region is determined for a given item (by removing all no-fit polygons), a point is selected on its perimeter using the translation parameter to place the item [20]. The translation parameter uniquely determines on which point of the no-fit polygon where the item will be translated to. As such, the solution is an hybrid discrete/continuous object composed of the following elements:

- Discrete:
  Order of placement: A permutation of the item indexes $(s_0;\ s_1, ..., s_n)$. This permutation dictates the order of placements.
- Continuous:
  - Rotations: A sequence $(\theta_0,\ \theta_1,\ ...,\ \theta_n)$ of rotations to be applied to each item.
  - Translations: A sequence $(t_0;\ t_1,\ ...,\ t_n)$ of numbers in $[0;\ 1[$. Each one of this numbers is converted to a placement point at the perimeter of the collision-free region for its respective item.

Those parameters are the product of the SA algorithm described above. Notice that they always produce a feasible solution. When for a given item the collision-free area is empty, it is simply not placed on the container.

4.1. Crystallization heuristic. Rejected solutions do not contribute to the progress of the optimization process. Therefore, the distribution of the step size for each individual parameter is adapted in order to increase the number of accepted solutions. This is accomplished by the adoption of a feedback on the SA. The next candidate point is generated by the following equation:

$$x_{k+1} = x_k + \frac{1}{c_j} \sum_{1}^{c_i} u \cdot \Delta r \cdot e_i, \tag{9}$$

where u is a uniform random number in $[-1; 1]$, Δr is the maximum allowed step size along the direction $\mathbf{e}_i$ of the i-th axis, and c_i is the crystallization factor associated to the i-th parameter. The maximum step size Δr remains always constant, and the probability distribution is adjusted to increase the number of accepted solutions. The crystallization factor controls the standard deviation of the Gaussian probability distribution. This is similar to the algorithm proposed by Corana et al. [2] where the maximum step size Δr_i of each parameter was adjusted in order to increase the number of accepted solutions.

When at a given iteration the modification applied to a parameter leads to a rejected solution, the probability distribution (crystallization factor) for that specific parameter is modified in order to have its standard deviation reduced (resulting in a lower modification amplitude). When the modification leads to an accepted solution, the distribution (crystallization factor) for that parameter is modified to increase its standard deviation (resulting in a larger modification amplitude).

As can be seen from above, the higher the crystallization factor for a given parameter, the smaller the modifications this parameter will receive during the SA. It is said that the item is crystallized.

The crystallization factors are initialized with value 1. When a solution generated by a modification on a parameter is rejected, its related crystallization parameter is increased by one. When the solution is accepted, the correspondent crystallization parameter is reset to its original value.

4.2. The modifed SA algorithm. The proposed SA algorithm is shown in Fig. 3. The main modification is shown in the inner loop, where it is chosen to swap two items in the placement sequence (discrete parameters) or to modify the rotation or translation of an item (continuous parameters).

The objective function $F(\mathbf{x})$ to be minimized is the space wasted inside the container with an adjustment to reduce its discrete behavior (see Subsec. 2.1). For every non-fitted shape, a fixed-depth binary search is performed in order to determinate the largest scale-factor that, applied to it, would allow its placement inside the container with its current rotation [20]. Then, the area of the shape with the largest obtained scale-factor is used to modify the wasted space. As such, the heuristic can be sensible to differences between solutions that have the same set of fitted shapes, but with different possibilities of fitting a new shape.

```
$\mathbf{x}_0 \leftarrow$ <Initial random solution>;
$k \leftarrow 0$;
while <Global stop condition not satisfied> do
  $T \leftarrow CoolingSchedule(k)$;
  $k \leftarrow k + 1$;
  while <Local stop condition not satisfied> do
    $u \leftarrow random(0, 1)$;
    if $u < \frac{1}{3}$ then
      < modify placement sequence >;
      $flag \leftarrow DiscreteParameter$;
    else
      $i \leftarrow random(0, 1) \cdot n$;
      $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \frac{1}{c_i} \sum_1^{c_i} random(-1, 1) \cdot \Delta r \cdot \mathbf{e}_i$;
      $flag \leftarrow ContinuousParameter$;
    end if
    $\Delta E = F(\mathbf{x}_{k+1}) - F(\mathbf{x}_k)$;
    if $\Delta E < 0$ then
      $\mathbf{x}_k \leftarrow \mathbf{x}_{k+1}$;
      if $flag = ContinuousParameter$ then
        $c_i \leftarrow 1$;
      end if
    else
      if $random(0, 1) < e^{-\Delta E/kT}$ then
        $\mathbf{x}_k \leftarrow \mathbf{x}_{k+1}$;
        if $flag = ContinuousParameter$ then
          $c_i \leftarrow 1$;
        end if
      else
        if $flag = ContinuousParameter$ then
          $c_i \leftarrow c_i + 1$;
        end if
      end if
    end if
  end while
end while
<Display solution $x$>
```

Fig. 3. The proposed SA algorithm, where n is the total number of cyclic continuous parameters

5. Results

As presented in the introduction, the literature related to the kind of problem studied here is scarce [10]. The majority of the benchmarks proposed in the literature have an open container (one dimension is not fixed) and allow only translations with few admissible orientations (0°, 90° and 180°). As a consequence, there is no appropriate benchmark in the literature for the kind of problem studied in this research. Firstly, some results showing the features of the proposed algorithm are presented. Secondly, some benchmarks proposed in the literature were adapted to be processed by the proposed algorithm. The advantage of studying puzzle like problems, is that the global minimum is known in advance.

5.1. Proposed problem instances. All problem instances studied here have a solution where all items can befitted on the containers. On all problems, the container area is less than 10% larger than the total items area. A simple geometric cooling with α =0:95 was adopted. The binary search was executed with fixed depth equal four (leading to 16 possible scale levels).

The behavior of the optimization process is illustrated through objective function histograms of the search while the temperature diminishes. For a given temperature, a gray-level histogram of the distribution of the objective function at that temperature is plotted. They are combined in a sequence of histograms (horizontal bars) and temperature (dots) versus iterations. A darker horizontal bar on the histogram, indicates a higher frequency of occurrence of a particular level of energy at a given temperature (see Fig. 4).

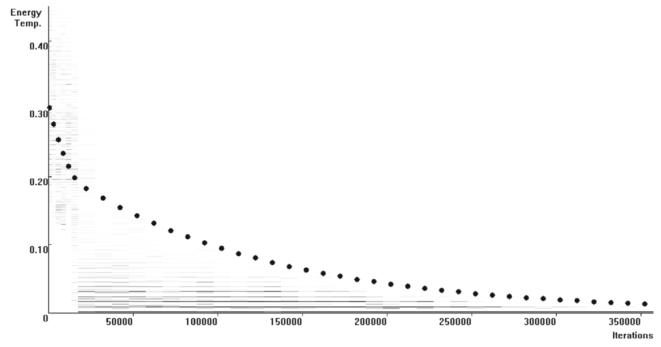

Fig. 4. Cost-function histograms for the four-items puzzle

As can be seen on the Fig. 4, as the cooling schedule progresses, the temperature lowers, leading the system to lower energy states. The transition between states is not smooth. The example in Fig. 4 has two clear "phases", what often corresponds to a macro-organization. Another interesting effect can be noticed on the Fig. 4. As the temperature decreases, the number of accepted solutions per iteration decreases (as more solutions are rejected), which leads to an effective slower cooling. It is not necessary to show the number of rejected solutions as the local stop condition is satisfied when a maximum number of accepted solutions or a maximum number of iterations is reached.

The problem shown in Fig. 5a) is a fairly simple puzzle with four non-convex non-congruent items. Figure 4 shows the objective function histograms for the solution found in Fig. 5a). Based in the objective function histograms, one can observe that there exists two phases. The final solution was found in average after 415:046 iterations. The convergence ratio to the global minimum was 87:1%.

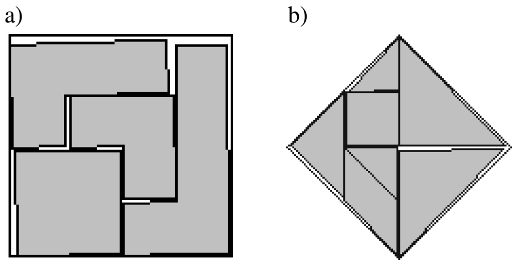

Fig. 5. a) Final solution of a small puzzle with four items; b) final solution of a tangram puzzle with seven items

The problem shown in Fig. 5b) is the placement of seven convex non- congruent items. Figure 6 shows the objective function histograms. On this nesting problem, the SA encounters a phase transition when the two greater triangles settle at their final position. The final solution was found in average after 1:958:401 iterations. The convergence ratio to the global minimum was 64:2%. When compared to previous implementations [15–17], the proposed SA algorithm with crystallization factor showed an improved convergence ratio.

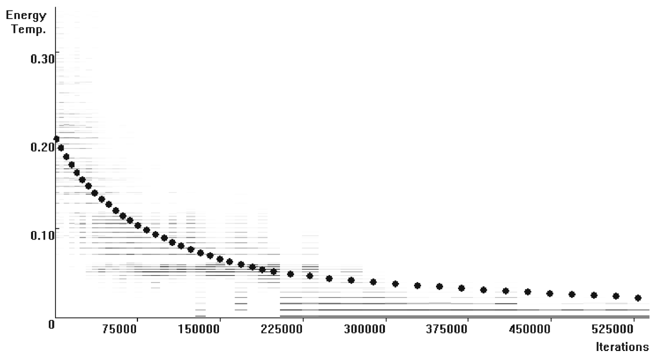

Fig. 6. Cost-function histograms for the tangram puzzle

5.2. Benchmarks adapted from the literature. All problems instances studied here were solved by [6]. They have an open container in the sense that one dimension is not fixed and allow only translations with few admissible orientations. To allow a possibility of comparison, the rotation heuristic was turned off and the container dimensions were considered exactly equal to the ones obtained by Gomes and Oliveira. All results obtained in [15] could be reproduced by the present heuristic. Each problem was run only two times, and the final configuration was reached.

The Albano is an instance problem with 24 items, it is a garment problem with two admissible orientations: 0° and 180°. The result shown in Fig. 7a) was produced by Gomes and Oliveira. The proposed algorithm found a different layout as shown in Fig. 7b). Figure 8 shows the associated objective function histograms. One can observe that the SA encounters a phase transition when the two larger items are placed.

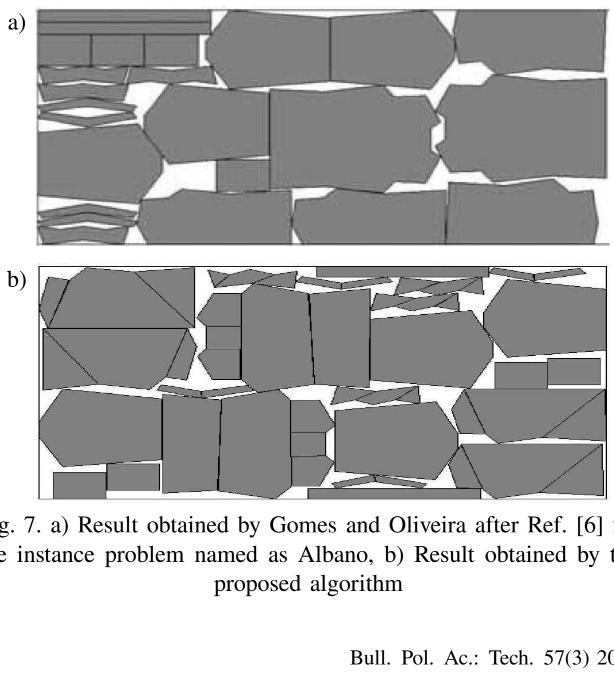

Fig. 7. a) Result obtained by Gomes and Oliveira after Ref. [6] for the instance problem named as Albano, b) Result obtained by the proposed algorithm

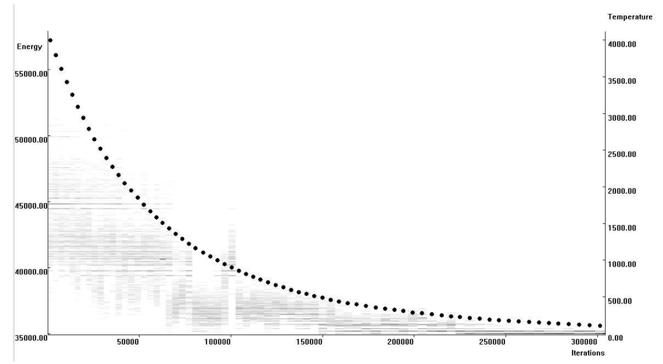

Fig. 8. Cost-function histograms for the instance problem shown in Fig. 7 (Albano)

The Fu is an instance problem with 12 items, it is an artificial problem with three admissible orientations: 0°, 90° and 180°. The result shown in Fig. 9a) was produced by Gomes and Oliveira. The proposed algorithm found different layouts as shown in Fig. 9b) and c). Figure 10 shows the associated objective function histograms. One can observe that the solution is found in a later stage and possible local minima can be seen at the darker lines. The Shapes0 is an instance problem with 43 items, it is an artificial problem with three admissible orientations: 0°, 90° and 180°. The result shown in Fig. 11a) was produced by Gomes and Oliveira. The proposed algorithm found different layouts as shown in Fig. 11b).

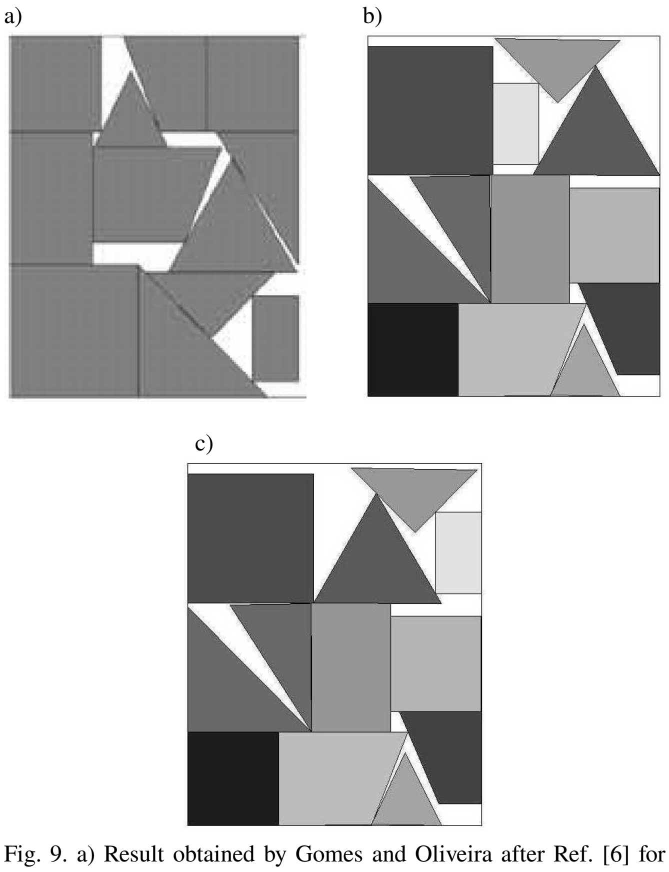

Fig. 9. a) Result obtained by Gomes and Oliveira after Ref. [6] for the instance problem named as Fu. b) and c) Results obtained by the proposed algorithm

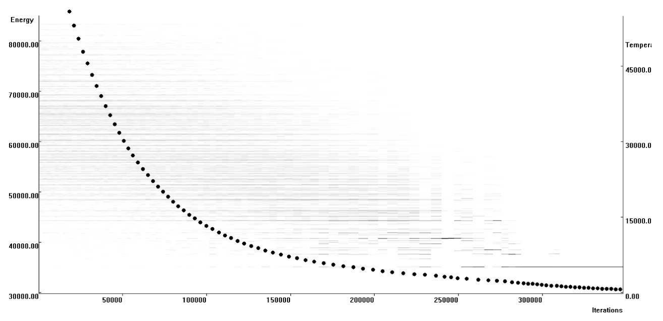

Fig. 10. Cost-function histograms for the instance problem shown in Fig. 9 (Fu)

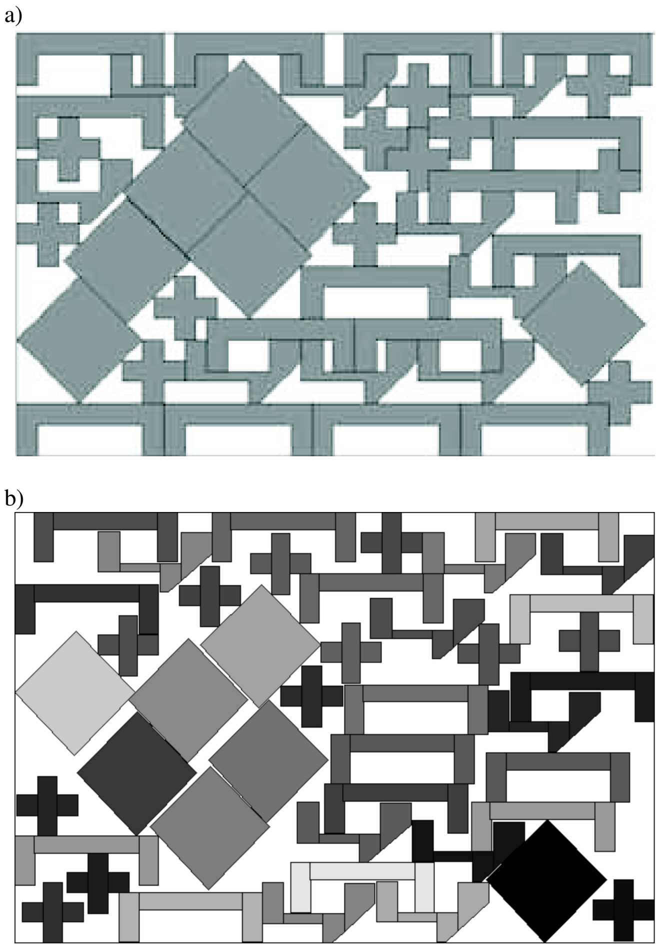

Fig. 11. a) Result obtained by Gomes and Oliveira after Ref. [6] for the instance problem named as Fu. b) and c) Results obtained by the proposed algorithm

6. Conclusions

This work proposed a new SA algorithm with adaptive neighborhood, where the sensibility of each continuous parameter is evaluated at each iteration increasing the number of accepted solutions. We have applied this new algorithm to other type of problems with success: robot path planning and curve fitting. However, the placement problem is by far the most complex problem solved using the proposed algorithm. The placement of an item is controlled by the following SA parameters: rotation, translation and sequence of placement. The cost function has discrete characteristics.We proposed some problems that are puzzles, the advantage of studying such problems is that the global minimum is known in advance. There are no benchmarks proposed in the literature for this kind of problem, then we adapted some problems from the literature (mainly by allowing only translations) and the results has been reproduced.

Acknowledgements. Marcos de Sales Guerra Tsuzuki was partially supported by CNPq and FAPESP.

REFERENCES

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

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

[3] M. Miki, T. Hiroyasu, and K. Ono, "Simulated annealing with advanced adaptive neighborhood", *Proc. Computational Intelligence and Applications* 2, 113–118 (2002).

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

[5] L. Ingber and B. Rosen, "Genetic algorithms and very fast simulated reannealing: a comparison", *Mathematical and Computer Modelling* 16, 87–100 (1992).

[6] A.M. Gomesand and J.F. Oliveira, "Solving irregular strip packing problems by hybridising simulated annealing and linear programming", *Eur. J. Operational Research* 171, 811–829 (2006).

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

[8] T.C. Martins and M.S.G. Tsuzuki, "Irregular rotational placement of shapes over non-convex containers with fixed dimensions", *Proc. IFAC Workshop on Intelligent Manufacturing Systems* 8, 182–187 (2007).

[9] H. Dyckhoff, "A typology of cutting and packing problems", *Eur. J. Operational Research* 44, 145–159 (1990).

[10] G. Wäscher, H. Haussner, and H. Schumann, "An improved typology of cutting and packing problems", *Eur. J. Operational Research* 183, 1109–1130 (2007).

[11] S. Jakobs, "On genetic algorithms for the packing of polygons", *Eur. J. Operational Research* 88, 165–181 (1996).

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

[13] R.C Art, *An Approach to the Two-Dimensional Irregular Cutting Stock Problem. Technical Report,* IBM Cambridge Scientific Centre, Cambridge, 1966.

[14] E.K. Burke, R.S.R. Hellier, G. Kendall, and G. Whitwell, "Complete and robust no-fit polygon generation for the irregular stock cutting problem", *Eur. J. Operational Research* 179, 27–49 (2007).

[15] K.A. Dowsland, S. Vaid, and B.W. Dowsland, "An algorithm for polygon placement using a bottom-left strategy", *Eur. J. Operational Research* 141, 371–381 (2002).

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

[17] D.G. Brooks and W.A. Verdini, "Computational experience with generalized simulated annealing over continuous variables", *American J. Mathematical and Management Sciences* 8, 425–449 (1988).

[18] P.P. Wang and D.S Chen, "Continuous optimization by a variant of simulated annealing", *Computational Optimization and Applications* 6, 59–71 (1996).

[19] T.C. Martins and M.S.G. Tsuzuki, "Simulated annealing applied to the rotational polygon packing", *Proc. IFAC Symposium Information Control Problems in Manufacturing* 12, 475–480 (2006).

[20] T.C. Martins and M.S.G. Tsuzuki, "Rotational placement of irregular polygons over containers with fixed dimensions using simulated annealing and no-fit polygons", *J. Brazilian Society of Mechanical Sciences and Engineering* 30, 196–203 (2008).