
Adaptive Neighborhood Heuristics for Simulated Annealing over Continuous Variables

T.C. Martins, A.K.Sato and M.S.G. Tsuzuki

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/50302>

1. Introduction

Simulated annealing has been applied to a wide range of problems: combinatorial and continuous optimizations. This work approaches a new class of problems in which the objective function is discrete but the parameters are continuous. This type of problem arises in rotational irregular packing problems. It is necessary to place multiple items inside a container such that there is no collision between the items, while minimizing the items occupied area. A feedback is proposed to control the next candidate probability distribution, in order to increase the number of accepted solutions. The probability distribution is controlled by the so called crystallization factor. The proposed algorithm modifies only one parameter at a time. If the new configuration is accepted then a positive feedback is executed to result in larger modifications. Different types of positive feedbacks are studied herein. If the new configuration is rejected, then a negative feedback is executed to result in smaller modifications. For each non-placed item, a limited depth binary search is performed to find a scale factor that, when applied to the item, allows it to be fitted in the layout. The proposed algorithm was used to solve two different rotational puzzles. A geometrical cooling schedule is used. Consequently, the proposed algorithm can be classified as simulated quenching.

This work is structured as follows. Section 2 presents some simulated annealing and simulated quenching key concepts. In section 3 the objective function with discrete values and continuous parameters is explained. Section 4 explains the proposed adaptive neighborhood based on the crystallization factor. Section 5 explains the computational experiments and section 6 presents the results. Finally, section 7 rounds up the work with the conclusions.

2. Background

Simulated annealing is a probabilistic meta-heuristic with a capacity of escape from local minima. It came from the Metropolis algorithm and it was originally proposed in the area of combinatorial optimization [9], that is, when the objective function is defined in a discrete

domain. The simulated annealing was modified in order to apply to the optimization of multimodal functions defined on continuous domain [4]. The choices of the cooling schedule and of the next candidate distribution are the most important decisions in the definition of a simulated annealing algorithm [13]. The next candidate distribution for continuous variables is discussed herein.

In the discrete domain, such as the traveling salesman and computer circuit design problems, the parameters must have 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. In the continuous application of simulated annealing a new choice of the next point candidate must be executed. Bohachevsky et al. [1] proposed that the next candidate \mathbf{x}_{k+1} can be 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 .

Brooks & Verdiní [2] showed that the selection of Δr is a critical choice. They observed that an appropriate choice of this parameter is strictly dependent on the objective function $F(\mathbf{x})$, and the appropriate value can be determined by presampling the objective function.

The directions in [1] are randomly sampled from the uniform distribution and the step size is the same in each direction. In this way, the feasible region is explored in an isotropic way and the objective function is assumed to behave in the same way in each direction. But this is not often the case. The step size to define the next candidate point \mathbf{x}_{k+1} should not be equal for all the directions, but different directions should have different step sizes; i.e. the space should be searched in an anisotropic way. Corana et al. [4] explored the concept of anisotropic search; they proposed a self-tuning simulated annealing algorithm in which the step size is configured in order to maintain a number of accepted solutions. 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 into cycles of n iterations during which each variable is modified. The new candidate point is obtained from \mathbf{x}_k in the following form $\mathbf{x}_{k+1} = \mathbf{x}_k + v \cdot \Delta r_i \cdot \mathbf{e}_i$. Where v is a uniform random number in $[-1, 1]$, and Δr_i is the step size along direction \mathbf{e}_i of the i -th axis. The anisotropy is obtained by choosing different values of Δr_i for all the directions. The step size is kept fixed for a certain number of cycles of variables, and the fraction of accepted moves in direction \mathbf{e}_i is calculated. If the fraction of accepted moves generated in the same direction is below 0.4, then the step size Δr_i along \mathbf{e}_i is decreased. It is assumed that the algorithm is using too large steps along \mathbf{e}_i thus causing many moves to be rejected. If the fraction is between 0.4 and 0.6 the step size is left unchanged. If the fraction is above 0.6 then Δr_i is increased. It is assumed that the step size is too small thus causing many moves to be accepted.

This procedure may not be the best possible to process the different behavior of the objective function along different axes. Ingber [7] proposed that the random variable should follow a Cauchy distribution with different sensitivities at different temperatures. The maximum step size is kept constant during the algorithm and it allows escaping from local minima even at low temperatures. The parameter space can have completely different sensitivities for each dimension, therefore the use of different temperatures for each dimension is suggested. This method is often referred to as very fast simulated re-annealing (VFSR) or adaptive simulated annealing (ASA). The sensitivity of each parameter is given by the partial derivative of the function with relation to the i -th dimension [3].

3. Integer objective function with float parameters

Irregular packing problems arise in the industry whenever one must place multiple items inside a container such that there is no collision between the items, while minimizing the area occupied by the items. It can be shown that even restricted versions of this problem (for instance, limiting the polygon shape to rectangles only) are NP complete, which means that all algorithms currently known for optimal solutions require a number of computational steps that grow exponentially with the problem size rather than according to a polynomial function [5]. Usually probabilistic heuristics relax the original constraints of the problem, allowing the search to go through points outside the space of valid solutions and applying penalization to their cost. This technique is known as external penalization. The most adopted penalization heuristic for external solutions of packing problems is to apply a penalization based on the overlapping area of colliding items. While this heuristic leads to very computationally efficient iterations of the optimization process, the layout with objective function in minimum value may have overlapped items [6].

Fig. 1 shows an example in which the cost function is the non-occupied space inside the container. As this space can change only by adding or removing areas of items, the cost 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 cost function related to the optimization variables.

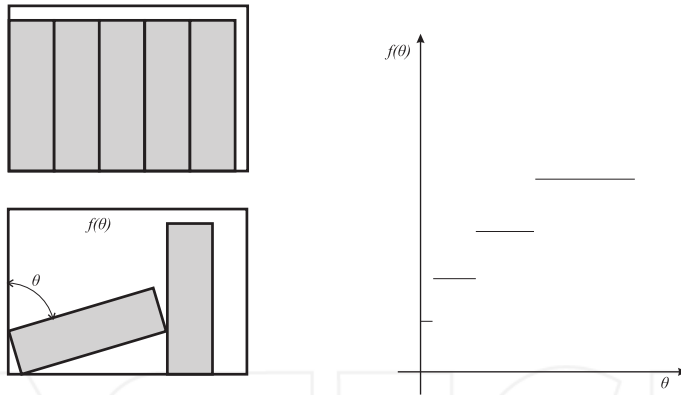


Figure 1. Objective function behavior.

Recently, researchers used the collision free region (CFR) concept to ensure feasible layouts; i.e. layouts in which the items do not overlap and fit inside the container [11]. This way, the solution has discrete and continuous components. The discrete part represents the order of placement (a permutation of the items indexes - this permutation dictates the order of placements) and the translation that is a vertex from the CFR perimeter. The continuous part represents the rotations (a sequence of angles of rotations to be applied to each item). The translation parameter is converted to a placement point at the perimeter of the CFR for its respective item. Fig. 2 shows the connection between the CFR and the translation parameter. Notice that the rotation parameter is cyclic in nature. All arithmetic operations concerning this parameter is performed in modulus 1 (so they always remain inside the interval $[0, 1]$).

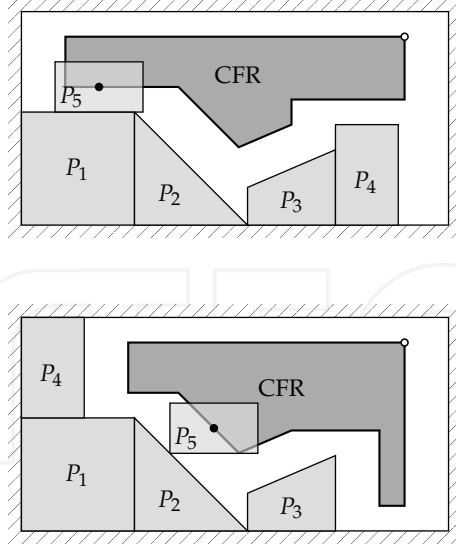


Figure 2. Consider that the container is rectangular and items P_1 , P_2 , P_3 and P_4 are already placed. Item P_5 is the next to be placed and to avoid collisions; it is placed at its CFR boundary. Its translation parameter has a value of 0.5. Both figures have P_4 placed at different positions, and consequently P_5 is also placed in different positions although the translation parameter is the same.

The wasted area that represents the cost function assumes only discrete values, while its variables (the rotations for each item) are continuous. To solve this type of problem, Martins & Tsuzuki [10] proposed a simulated quenching with a new heuristic to determine the next candidate that managed to solve this type of problem.

3.1. Scale factor

The objective function is the wasted space in the container and is discrete, depending on which items have been placed. In order to improve the sensibility of the cost function, intermediate levels can be generated by scaling one of the unplaced items, and attempting to insert the reduced version of the item into the layout. Hence, for each unplaced item, a scale factor between $[0, 1]$ is applied, and the algorithm attempts to place the item, if it fits, the scaled area of the item is subtracted from the objective function. Scale factor was determined by a finite fixed depth binary search, restricted to the interval $[0, 1]$.

4. Adaptive neighborhood

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

The main concept is that rejected solutions do not contribute to the progress of the optimization process. Therefore, the distribution of the step size for each individual continuous parameter is adapted in order to increase the number of accepted solutions. This

```

1  $\mathbf{x}_0 \leftarrow \langle \text{Initial random solution} \rangle;$ 
2  $k \leftarrow 0;$ 
3 while  $\langle \text{Global stop condition not satisfied} \rangle$  do
4    $T_{k+1} \leftarrow T_k * \alpha;$ 
5    $k \leftarrow k + 1;$ 
6   while  $\langle \text{Local stop condition not satisfied} \rangle$  do
7      $u \leftarrow \text{random}(0, 1);$ 
8     if  $u < \frac{1}{3}$  then
9        $\langle \text{Modify placement sequence} \rangle;$ 
10       $flag \leftarrow \text{DiscreteParameter};$ 
11    else
12      if  $u < \frac{2}{3}$  then
13         $\langle \text{Select one vertex from the CFR to place the item} \rangle;$ 
14         $flag \leftarrow \text{DiscreteParameter};$ 
15      else
16         $i \leftarrow \text{random}(0, 1) \cdot n;$ 
17         $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \frac{1}{c_i} \sum_1^{c_i} \text{random}(-1/2, 1/2) \cdot \mathbf{e}_i;$ 
18         $flag \leftarrow \text{ContinuousParameter};$ 
19       $\Delta E = F(\mathbf{x}_{k+1}) - F(\mathbf{x}_k);$ 
20      if  $\Delta E < 0$  then
21         $\mathbf{x}_k \leftarrow \mathbf{x}_{k+1};$ 
22        if  $flag = \text{ContinuousParameter}$  then
23           $c_i \leftarrow \langle \text{positive feedback} \rangle;$ 
24      else
25        if  $\text{random}(0, 1) < e^{-\Delta E/kT}$  then
26           $\mathbf{x}_k \leftarrow \mathbf{x}_{k+1};$ 
27          if  $flag = \text{ContinuousParameter}$  then
28             $c_i \leftarrow \langle \text{positive feedback} \rangle;$ 
29        else
30          if  $flag = \text{ContinuousParameter}$  then
31             $c_i \leftarrow c_i + 1;$ 

```

Figure 3. The proposed algorithm. Different types of positive feedbacks are studied in this work.

is accomplished by the adoption of a feedback on the proposed algorithm. The next candidate is generated by the modification of a *single parameter*, adding to it a summation of c_i random numbers with a uniform distribution.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{c_i} \sum_1^{c_i} \text{random}(-1/2, 1/2) \cdot \mathbf{e}_i \quad (1)$$

where i is the index of the modified parameter and c_i is its crystallization factor. The resulted modification follows a Bates distribution [8, sec. 26.9] centered on 0 with amplitude 1/2. Its standard deviation is given by $1/2\sqrt{3c_i}$.

For $c_i = 1$, as all operations on parameters are performed in modulus 1; the modification is the equivalent of taking a completely new parameter uniformly distributed in the interval $[0, 1[$. As c_i increases, the expected amplitude of the modification decreases. 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 lower modification amplitude), this is the negative feedback. When the modification leads to an accepted solution, the distribution (crystallization factor) for that parameter is modified to increase its standard deviation (resulting in larger modification amplitude), this is the positive feedback. Different positive feedbacks are studied in this work (see Table 1). As can be seen, the higher the crystallization factor for a given parameter, the smaller the modification this parameter will receive during the proposed algorithm. The parameter is said to be crystallized.

5. Computational experiments

Crystallization factor c_i controls the standard deviation of the Bates distribution. When a solution is rejected, a negative feedback is applied and the corresponding c_i is increased, causing a decrease in the parameter standard deviation. Accordingly, positive feedback is applied when a solution is accepted, increasing c_i . In the studied problems, placement was restricted to vertexes of the CFR and thus the only continuous parameter is the rotation. Adopted negative feedback consists of incrementing the crystallization factor. For the positive feedback, the four different strategies in Table 1 were tested.

Feedback Method			Positive Feedback	Negative Feedback
A	B	C	D	$CF_i \rightarrow CF_i - 1$
				$CF_i \rightarrow CF_i + 1$
				$CF_i \rightarrow CF_i / 2$
				$CF_i \rightarrow CF_i + 1$
C	D	A	B	$CF_i \rightarrow CF_i / 4$
				$CF_i \rightarrow CF_i + 1$
D	A	B	C	$CF_i \rightarrow 1$
				$CF_i \rightarrow CF_i + 1$

Table 1. Feedback strategies. CF_i : Crystallization factor for item i .

The convergence of the algorithm is reached when, at a given temperature, all accepted solutions are equivalent to the best found. This is the global stop condition of the algorithm in Fig. 3. Although a solution as good as the final one is found in less iterations, allowing the algorithm to reach the global convergence is the only generic way to ensure that a solution is the best. The local stop condition shown in Fig. 3 is reached when a predefined number of solutions are accepted.

5.1. Problem instances

All problem instances studied here have a solution in which all items can be fitted in the container. Two puzzles cases were considered: Tangram and Larger First Fails (LF Fails). Tangram is a classic problem and LF Fails consists of a problem which cannot be solved using the larger first heuristic. This heuristic determines that the larger items are placed always ahead of the smaller ones. Fig. 4 shows possible solutions to these problems.

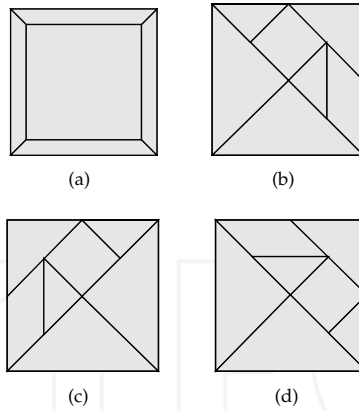


Figure 4. (a) Unique solution for problem LF Fails. (b)-(d) Solutions for the Tangram problem.

6. Results and discussion

The algorithm was implemented in C++ and compiled by GCC 4.4.4. Computational tests were performed using an i7 860 processor with 4GB RAM. Each case was evaluated 100 times. The proposed algorithm is a simulated quenching algorithm which has the following parameters:

- T_0 : Initial temperature.
- α : geometric cooling schedule factor.
- N_{acc} : Number of accepted solutions at a given temperature.

Value of T_0 is calculated such that the number of rejected solutions at initial temperature is approximately 10% of the total number of generated solutions. Parameter α is set to 0.99 and N_{acc} is 800.

6.1. Influence of the feedback strategy

Table 2 shows results obtained using each of the proposed feedback strategy, for each problem instance. For the Tangram problem, it can be observed that strategy A has a low convergence percentage, when compared to other feedback strategies, 0.09 less than the rate obtained using the feedback C method. In the case of the LF fails puzzle, results showed similar performance and convergence rate.

Fig. 5 shows the minimum, maximum and average costs explored by the proposed algorithm loop for the LF Fails, for all feedback strategies. The cost function discrete behavior is observable, and it is possible to notice that the global minimum is reached only at low temperatures. In all graphics, the optimum layout was found. One can note that, in Fig. 5.(b) and Fig. 5.(c), the best solution (cost equals zero) was found before reaching convergence. Variation of cost is shown in Fig. 6 and all graphs are very similar independently of the used positive feedback. The rotation crystallization factor for the largest item is displayed in Fig. 7. Possibility of accepting a higher cost solution is lower at low temperatures. As temperature

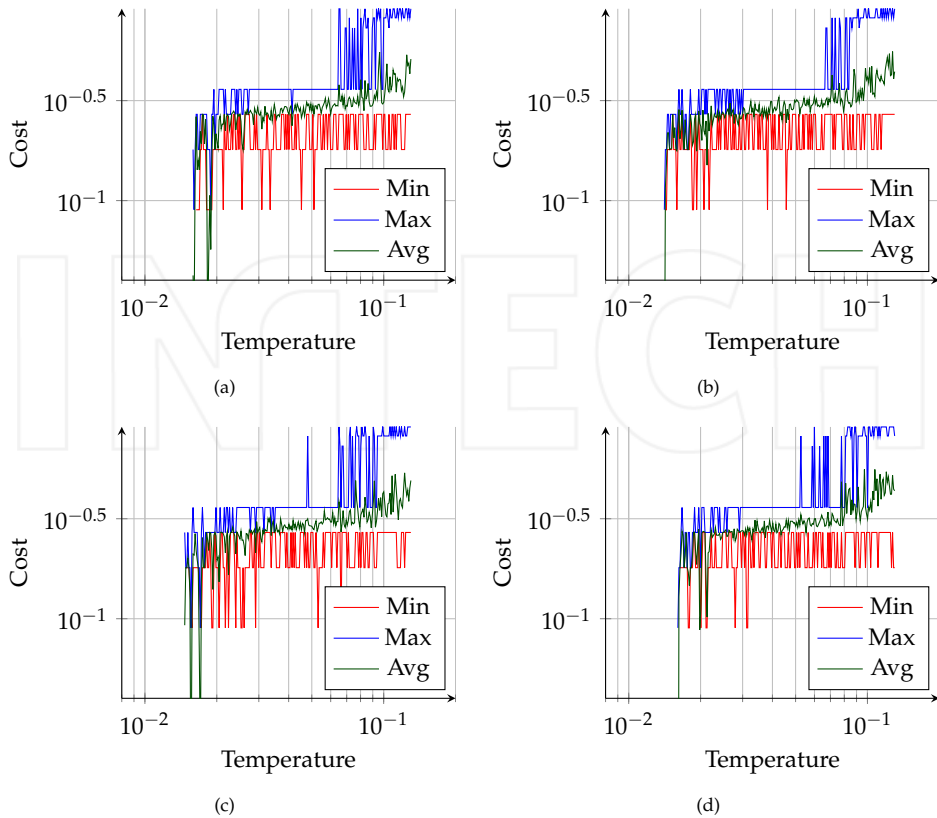


Figure 5. Minimum, maximum and average costs for the LF Fails with different feedbacks. (a) Feedback A. (b) Feedback B. (c) Feedback C. (d) Feedback D.

Problem	Feedback Method	N_{conv}	N_{min}	T_{conv}	P_{conv}
LF fails	A	228935	188814	17.48	1.00
	B	235986	197038	17.78	0.99
	C	235595	195377	17.64	1.00
	D	235481	194394	17.67	1.00
Tangram	A	303517	255611	64.33	0.56
	B	315019	268996	69.07	0.65
	C	319440	270484	69.27	0.62
	D	317403	267057	71.09	0.61

Table 2. Statistics for the LF fails and Tangram puzzles. The columns respectively represent the adopted problem instance, the feedback method, number of iterations to converge, number of iterations to reach the minimum, time in seconds to converge, and the percentage of runs that converged to the global optimum.

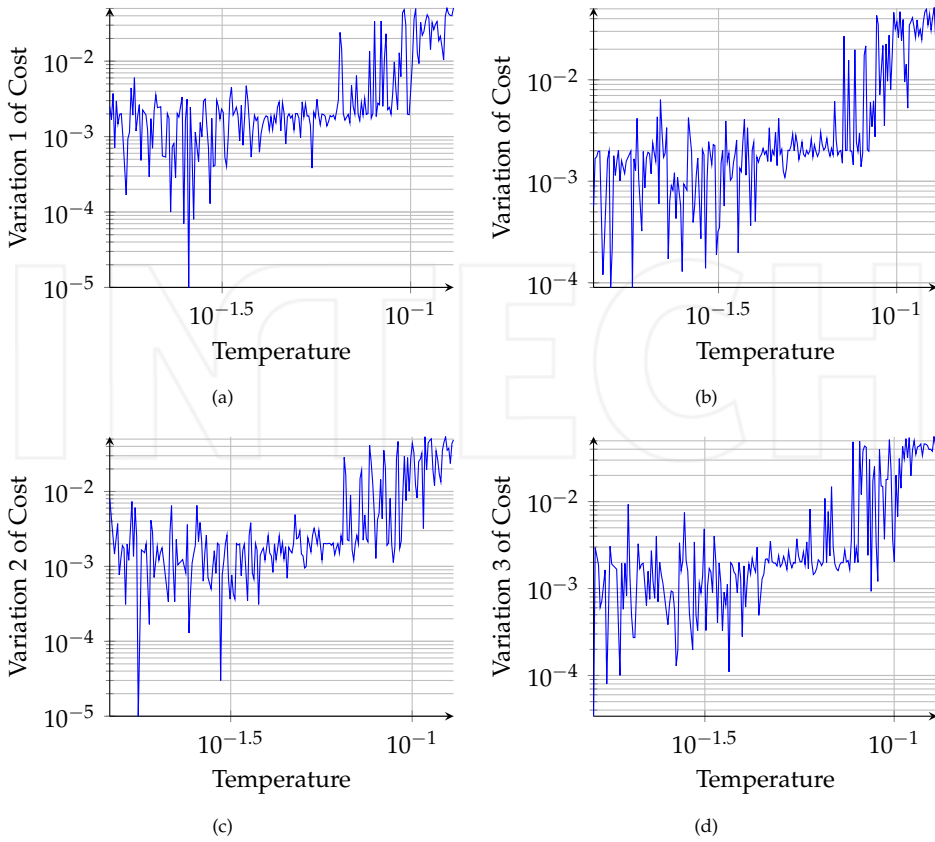


Figure 6. Variation of cost for the LF Fails with different feedbacks. (a) Feedback A. (b) Feedback B. (c) Feedback C. (d) Feedback D.

decreases, the crystallization factor is expected to increase, which is confirmed by the graphics in Fig 7. Positive feedback A is very stable, showing that it is less exploratory. Because of the small number of items, it was not necessary to use the scale factor. Fig. 8 shows the specific heat for each case considered. The specific heat is calculated as [14]

$$C_H(T) = \frac{\sigma^2(T)}{T^2 k_B^2} \quad (2)$$

where T is temperature, $\sigma^2(T)$ is the variation of cost, k_B is a constant. A phase transition occurs at a temperature at which specific heat is maximum, and this triggers the change in state ordering. In several processes, it represents the transition from the exploratory to the refining phase. However, in this specific case, this transition is not observable.

For the Tangram problem, the minimum, maximum and average costs explored by the algorithm in one execution are shown in Fig. 9. The increase in allowable cost function values can be observed. In each of these executions the global minimum was reached only at low

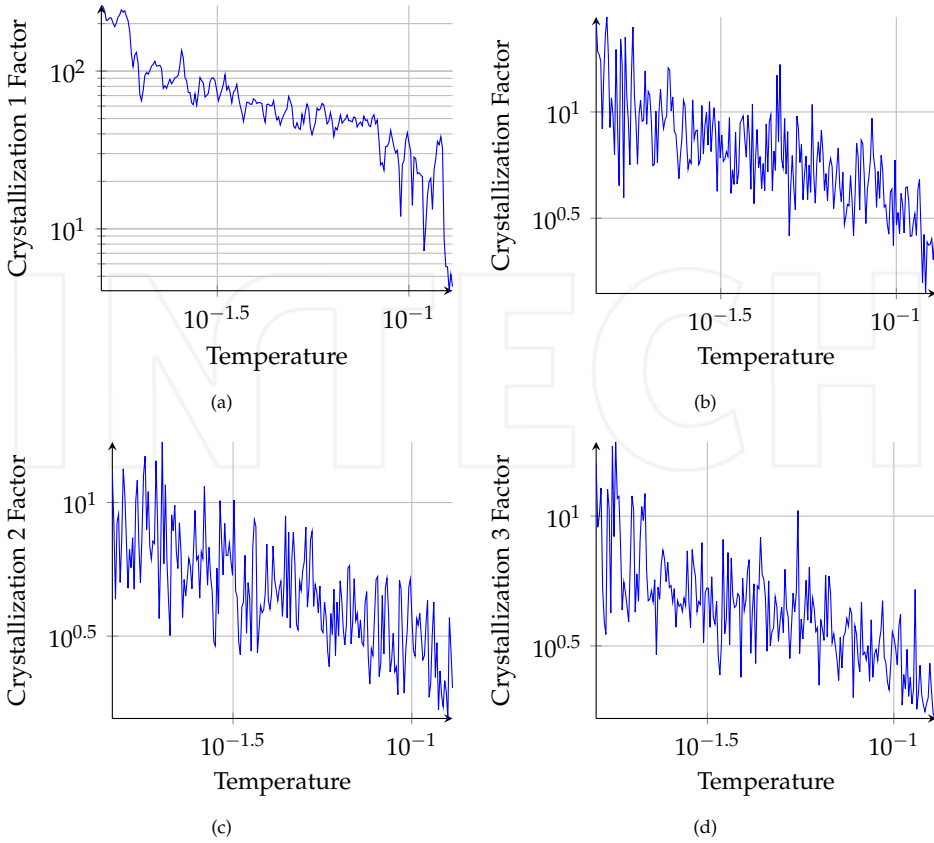


Figure 7. Crystallization factor for the largest item of the LF Fails problem, with different feedbacks. (a) Feedback A. (b) Feedback B. (c) Feedback C. (d) Feedback D.

temperatures. Independently of the used positive feedback, the proposed algorithm reached local minima at lower temperatures, but successfully escaped from them. Cost variance is displayed in Fig. 10. The rotation crystallization factor evolution is shown in Fig. 11, for one of the large triangles. It is possible to observe, when adopting feedback strategies A and C, that there are two distinguishable levels. The final level, at lower temperatures, is very high, indicating that the rotation parameter of the item is crystallized. Again, feedback A is more stable when compared to the others, showing that is less exploratory. As the convergence rate is very poor, the scale factor should be used. Fig. 12 shows the specific heats obtained. In the Tangram casem, it seems that a peak is present. However, further investigations need to be done.

6.2. Influence of the binary search

Binary search is used to improve the sensibility of the discrete objective function, aiming to obtain a higher percentage of convergence for puzzle problems. Its application is not necessary in the case of the LF Fails problem, as almost all executions converged. As a

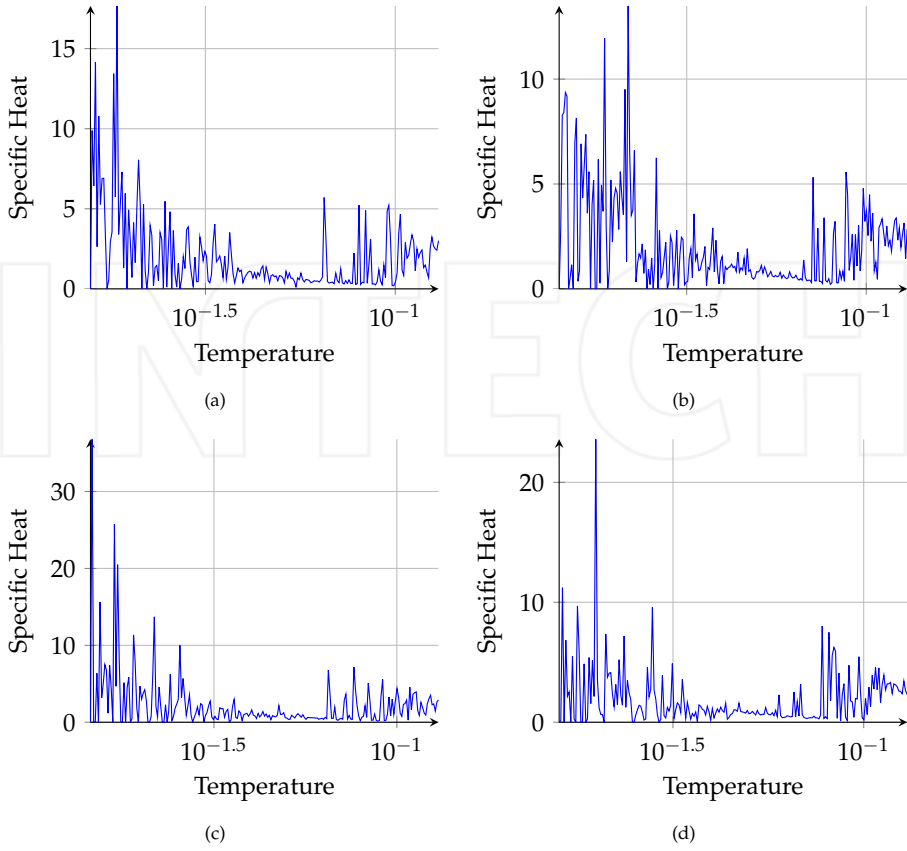


Figure 8. Specific heat for the LF Fails with different feedbacks. (a) Feedback A. (b) Feedback B. (c) Feedback C. (d) Feedback D.

consequence, the binary search was employed only in the Tangram problem. The fixed search depth was set to 1. Table 3 shows the results of the tests. Comparing with the results obtained in Table 2, the convergence rate is observed to be considerably higher when binary search is adopted, reaching 98% in the best case. Drawback is the large number of iterations needed to converge, resulting in longer execution times, approximately 3.5 times higher. As with the previous tests, feedback strategy A obtained less optimum solutions.

The behavior of the optimization process is illustrated through cost function (energy) histograms of the search while the temperature diminishes. For a given temperature, a gray-level histogram of the distribution of the cost function at that temperature is plotted. The resulting graph shows a plot of cost histograms (horizontal bars) and temperature (dots) versus the number of iterations. Darker horizontal bars in the histogram, indicate a higher frequency of occurrence of a particular level of energy at a given temperature. Fig. 13 shows the histogram of the objective function value during the course of the algorithm, without the use of the binary search. Fig. 14 shows the same type of histogram employing the binary search with depth 1. Observing both graphics, one can note the extra levels of energy which

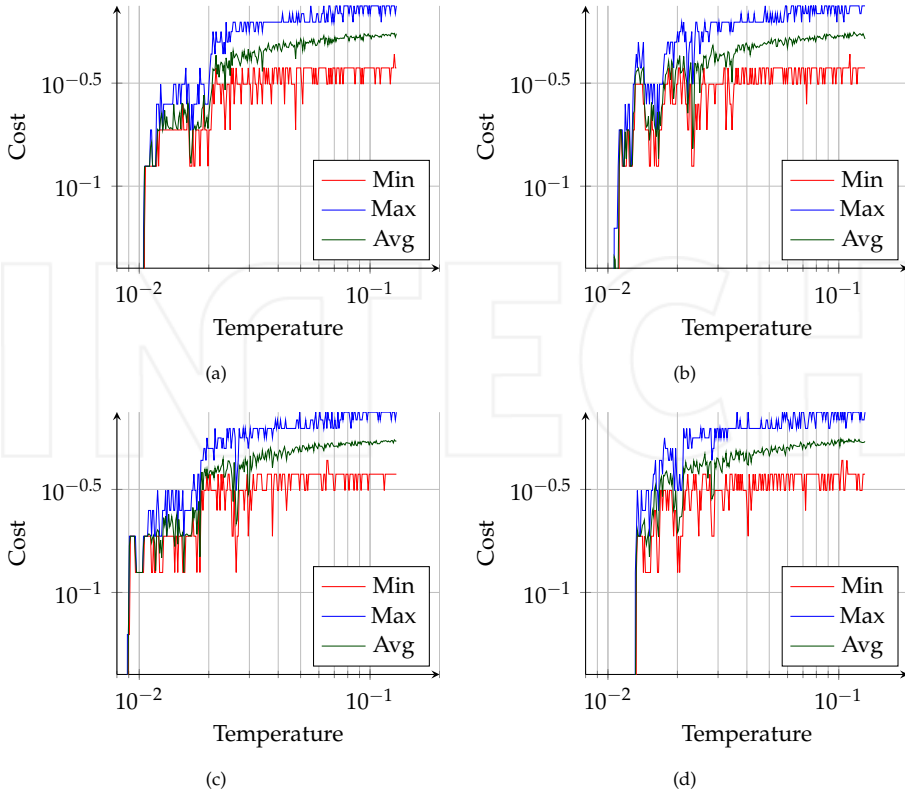


Figure 9. Minimum, maximum and average costs for the Tangram with different feedbacks. (a) Feedback A. (b) Feedback B. (c) Feedback C. (d) Feedback D.

Feedback Method	N_{conv}	N_{min}	T_{conv}	P_{conv}
A	370667	290044	222.54	0.78
B	351141	299052	227.39	0.91
C	343652	327037	228.40	0.98
D	338394	312867	213.91	0.97

Table 3. Statistics for the Tangram puzzles using a binary search with unitary depth. The columns respectively represent the feedback method, number of iterations to converge, number of iterations to reach the minimum, time in seconds to converge, and the percentage of runs that converged to the global optimum.

appears in Fig. 14. Higher values of the search depth were tested, however the convergence rate deteriorates.

From the studied problems, it is possible to observe that positive and negative feedbacks must not be opposites. The negative feedback increases the crystallization factor by a unit and the positive feedback needs to decrease the crystallization factor at a faster speed. If this is not the case, the parameters might get crystallized.

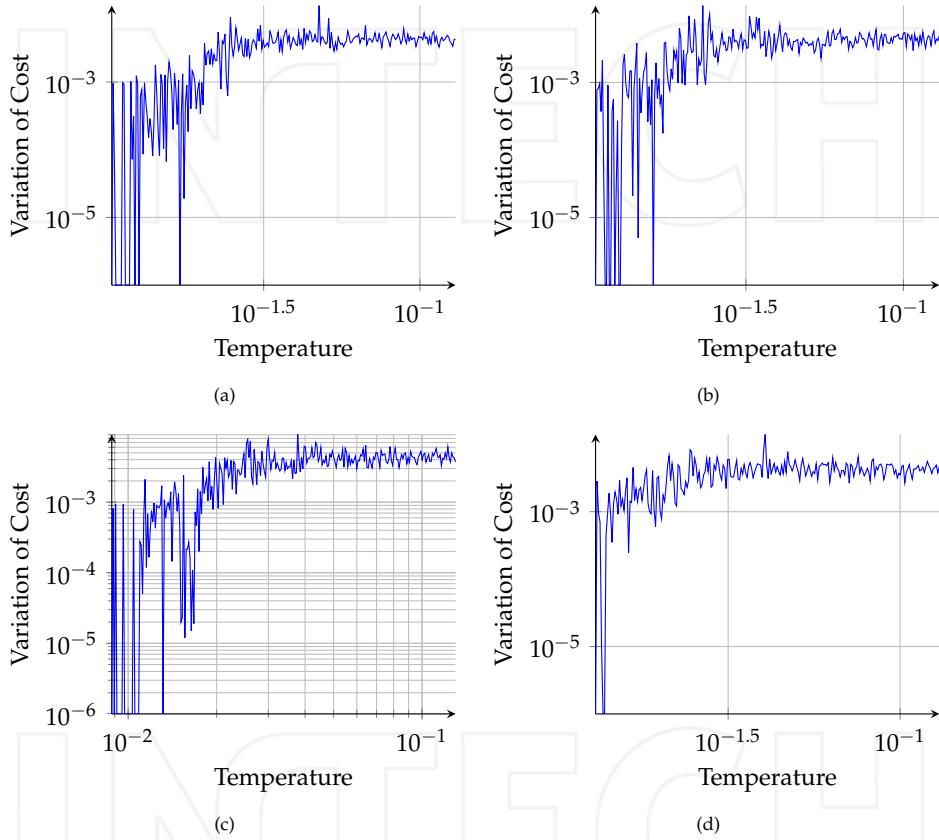


Figure 10. Variation of cost for the Tangram with different feedbacks. (a) Feedback A. (b) Feedback B. (c) Feedback C. (d) Feedback D.

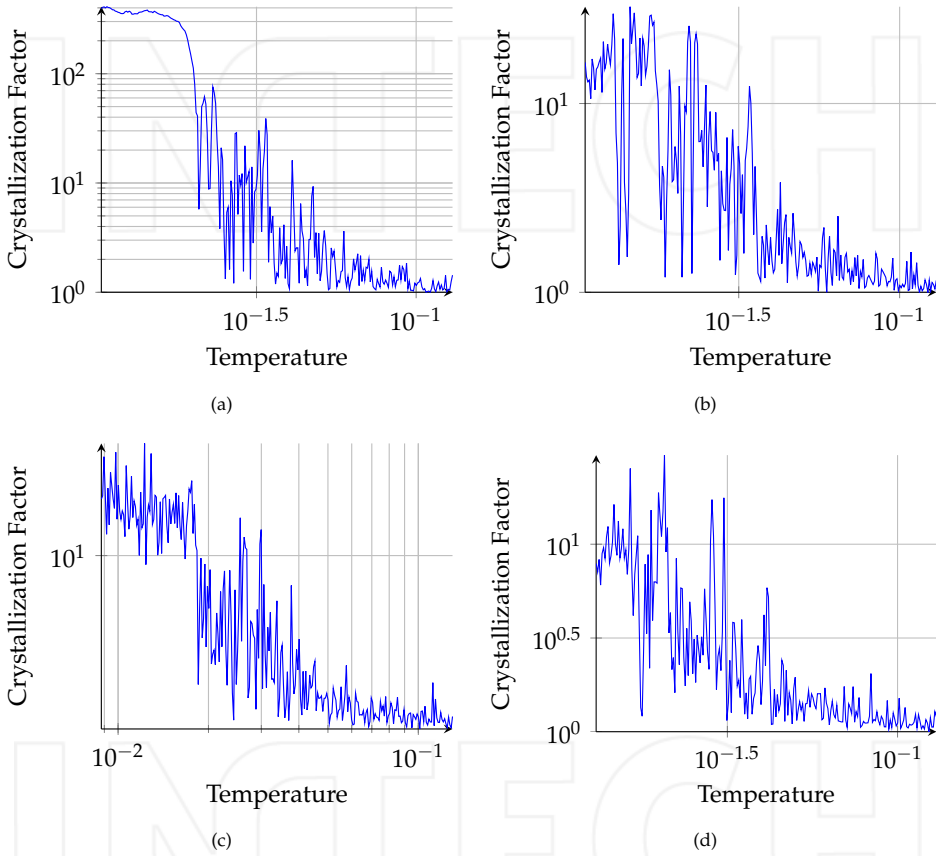


Figure 11. Crystallization factor for one of the large item of the Tangram problem, with different feedbacks. (a) Feedback A. (b) Feedback B. (c) Feedback C. (d) Feedback D.

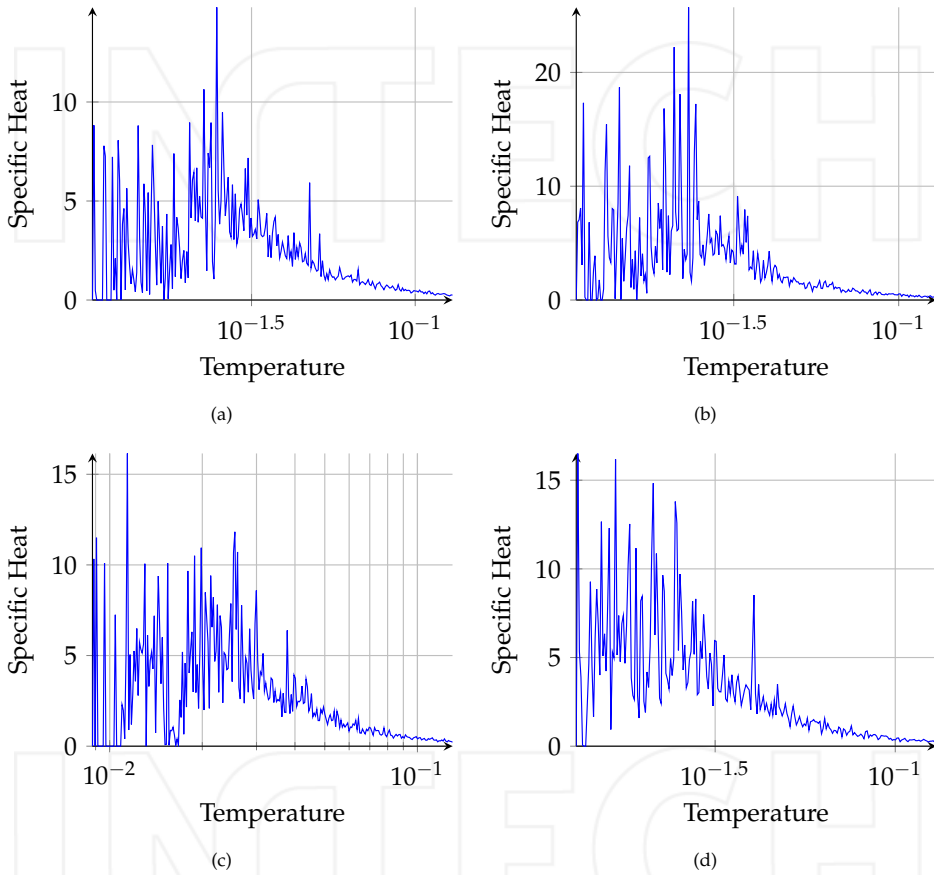


Figure 12. Specific heat for the Tangram with different feedbacks. (a) Feedback A. (b) Feedback B. (c) Feedback C. (d) Feedback D.

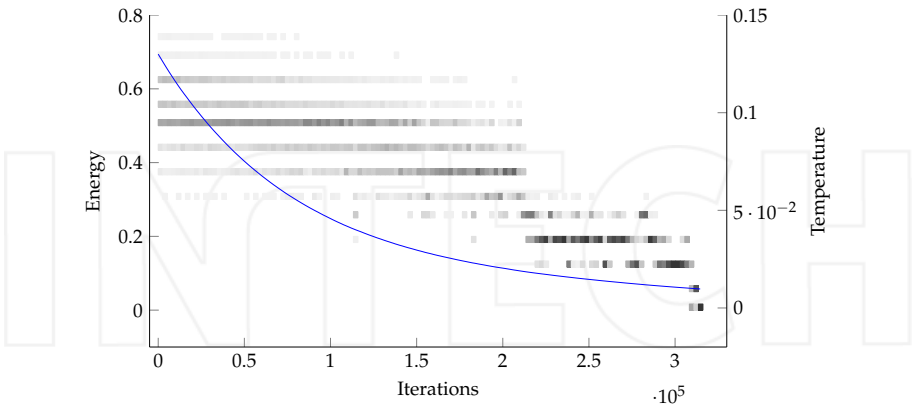


Figure 13. Histogram for the Tangram, not employing binary search.

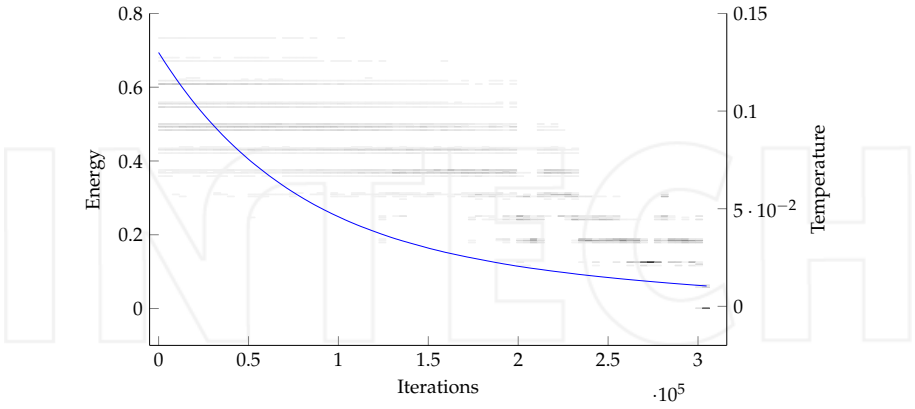


Figure 14. Histogram for the Tangram, employing binary search with fixed depth equal to 1.

7. Conclusion

This work proposed a new simulated quenching algorithm with adaptive neighborhood, in which the sensibility of each continuous parameter is evaluated at each iteration increasing the number of accepted solutions. The proposed simulated quenching was successfully applied to other types of problems: robot path planning [14] and electrical impedance tomography [12]. The placement of an item is controlled by the following simulated quenching parameters: rotation, translation and sequence of placement.

8. Acknowledgements

AK Sato was supported by FAPESP (Grant 2010/19646-0). MSG Tsuzuki was partially supported by the CNPq (Grant 309.570/2010-7). This research was supported by FAPESP (Grants 2008/13127-2 and 2010/18913-4).

Author details

T.C. Martins, A.K.Sato and M.S.G. Tsuzuki
Computational Geometry Laboratory - Escola Politécnica da USP, Brazil

9. References

- [1] Bohachevsky, I. O., Johnson, M. E. & Stein, M. L. [1986]. Generalized simulated annealing for function optimization, *Technometrics* 28(3): pp. 209–217.
- [2] Brooks, D. G. & Verдини, W. A. [1988]. Computational experience with generalized simulated annealing over continuous variables, *Am. J. Math. Manage. Sci.* 8(3–4): 425–449.
- [3] Chen, S. & Luk, B. [1999]. Adaptive simulated annealing for optimization in signal processing applications, *Signal Processing* 79(1): 117 – 128.
- [4] Corana, A., Marchesi, M., Martini, C. & Ridella, S. [1987]. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm, *ACM Trans. Math. Softw.* 13(3): 262–280.
- [5] Fowler, R. J., Paterson, M. & Tanimoto, S. L. [1981]. Optimal packing and covering in the plane are np-complete, *Inf. Process. Lett.* pp. 133–137.
- [6] Heckmann, R. & Lengauer, T. [1995]. A simulated annealing approach to the nesting problem in the textile manufacturing industry, *Annals of Operations Research* 57: 103–133.
- [7] Ingber, L. [1996]. Adaptive simulated annealing (asa): Lessons learned, *Control and Cybernetics* 25: 33–54.
- [8] Johnson, N. [1994]. *Continuous univariate distributions*, Wiley, New York.
- [9] Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. [1983]. Optimization by simulated annealing, *Science* 220: 671–680.
- [10] Martins, T. C. & Tsuzuki, M. S. G. [2009]. Placement over containers with fixed dimensions solved with adaptive neighborhood simulated annealing, *Bulletin of the Polish Academy of Sciences Technical Sciences* 57: 273–280.
- [11] Martins, T. C. & Tsuzuki, M. S. G. [2010]. Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions, *Expert Systems with Applications* 37: 1955–1972.
- [12] Martins, T. C., Camargo, E. D. L. B., Lima, R. G., Amato, M. B. P. & Tsuzuki, M. S. G. [2012]. Image reconstruction using interval simulated annealing in

electrical impedance tomography, *IEEE Transactions on Biomedical Engineering*, URL: <http://dx.doi.org/10.1109/TBME.2012.2188398>.

- [13] Miki, M., Hiroyasu, T. & Ono, K. [2002]. Simulated annealing with advanced adaptive neighborhood, *In Second international workshop on Intelligent systems design and application*, Dynamic Publishers, Inc. ISBN, pp. 113–118.
- [14] Tavares, R. S., Martins, T. C. & Tsuzuki, M. S. G. [2011]. Simulated annealing with adaptive neighborhood: A case study in off-line robot path planning, *Expert Systems with Applications* 38(4): 2951–2965.

INTECH

INTECH