Versatility of Simulated Annealing with Crystallization Heuristic: Its Application to a Great Assortment of Problems

Tiago G. Goto, Hossein R. Najafabadi, Guilherme C. Duran, Edson K. Ueda, André K. Sato, Thiago C. Martins, Rogério Y. Takimoto, Hossein Gohari, Ahmad Barari and Marcos S.G. Tsuzuki

Abstract

This chapter is related to several aspects of optimization problems in engineering. Engineers usually mathematically model a problem and create a function that must be minimized, like cost, required time, wasted material, etc. Eventually, the function must be maximized. This function has different names in the literature: objective function, cost function, etc. We will refer to it in the chapter as objective function. There is a wide range of possibilities for the problems and they can be classified in different ways. At first, the values of the parameters can be continuous, discrete (integers), cyclic (angles), intervals, and combinatorial. The result of the objective function can be continuous, discrete (integers) or intervals. One very difficult class of problems have continuous parameters and discrete objective function, this type of objective function has very weak sensibility. This chapter shows the versatility of the simulated annealing showing that it can have different possibilities of parameters and objective functions.

Keywords: Simulated Annealing, Optimization Problems, Cutting Packing, Electrical Impedance Tomography, Topological Optimization, Curve Interpolation

1. Introduction

Optimization problems are common in engineering, and the problem must be optimized regarding one function (usually called as cost function). The cost function has several parameters used to model the optimization problem. Several methods have been proposed in the literature to optimize functions, and the method is selected according the cost function and parameters characteristics.

Deterministic methods usually have an initial point, called as seed. Using the gradient of the cost function, the optimization method starts on the initial point and converges to the optimal parameter value. The parameters are usually continuous [1].
However, engineering problems have problems which cannot be solved by this approach.

Some cost functions have several minimum, and the convergence strongly depends on the seed (or initial point). To overcome this issue, several meta-heuristics were proposed, like Genetic Algorithms (GA) [2, 3], Particle Swarm Optimization (PSO) [4], Simulated Annealing (SA) [5], and others. Optimization methods based on meta-heuristics do not need the gradient information. For some problems, it is very hard to determine the gradient. This is particularly true for problems with discrete cost function. The existing meta-heuristics, usually, are focused on some specific type of parameters. For example, PSO can be applied to engineering problems with continuous parameters. GA can be applied to engineering problems with integer or fixed precision parameters.

SA was first proposed to solve combinatorial problems [5]. Later on, Corana et al. [6] extended the SA to incorporate continuous parameters. It was shown that the proposal made by Corana et al. [6] only converged to the global optimum in specific problems. Ingber [7] improved the SA to overcome this issue, however he used the cost function gradient for some calculations. Martins and Tsuzuki [8] proposed the SA with crystallization heuristic which does not use any gradient information.

This chapter shows the versatility of the SA with crystallization heuristic. Initially, the SA with crystallization heuristic is explained in Section 2. It was used to solve different problems related to Cutting and Packing, particularly the case with discrete cost function and cyclic continuous parameters [8–10] will be explained in Section 3. A third application is Topological Optimization (TO), it has a large number of parameters, like the EIT problem. This application is explained in Section 4. The curve interpolation problem, explained in Section 5, has two different types of parameters: continuous and integer. It is relevant to point that the SA with crystallization heuristic has been applied with success to interval based objective function evaluation [11–13]. The conclusions are in Section 6.

2. Simulated annealing with crystallization heuristic

The SA is an optimization algorithm that uses a random local search and is able to find the global optimum solution. It was proposed by [5, 14], and it was inspired by the metal annealing process, which starts with a high temperature and reduces the temperature in small increments until reaching the minimum temperature. In each step, the new solution is accepted if it improves the current solution. Otherwise, it is only accepted if a probability factor $P(T)$ is more than a random number [10, 14, 15]. The $P(T)$ can be determined by

$$P(T) = \exp \left( \frac{-\Delta E}{T} \right)$$

where $T$ is the current temperature, and $\Delta E$ is the energy state variation which is the difference between the objective function $F(x)$ of the new candidate $a$ and the current solution $b$, as presented in $\Delta E = F(a) - F(b)$.

Algorithm 1 describes a possible implementation of a conventional SA. It initializes with a random solution and an initial temperature, which is an important parameter. There are two loops. The external loop controls the temperature. The temperature decreases according to a geometric cooling schedule with factor $\alpha$, and it stops when the frozen state is reached. The cooling schedule, in this case the value of $\alpha$, must be carefully chosen, as the convergence to the optimal distribution.
strongly depends on it. In the majority of the applications explained here, we used $\alpha = 0.98$. The internal loop, performs the random local search, until the thermal equilibrium is reached. The random search consists of modifying a single parameter and a new candidate $x^*$ is obtained. The new objective function value is evaluated and compared with the cost of the current solution. Then, it is either accepted or rejected according to (1). The convergence conditions are usually controlled by algorithm parameters. They influence the quality of the final result as well as the speed of the algorithm.

**Algorithm 1**: Conventional SA

1: $i \leftarrow 0$
2: $x \leftarrow \text{random initial solution}$
3: $T_0 \leftarrow \text{initial temperature}$
4: while $\text{Global condition not satisfied}$ do
5:     while $\text{Local condition not satisfied}$ do
6:         $x^* \leftarrow \text{modify a parameter}$
7:         $\Delta E = F(x^*) - F(x)$
8:         if $\Delta E < 0$ or random$(0, 1) < e^{-\Delta E/T_i}$ then
9:             $x \leftarrow x^*$
10:     end if
11: end while
12: $i \leftarrow i + 1$
13: $T_i \leftarrow T_{i-1} \times \alpha$
14: end while

The random search in the SA algorithm can use different strategies to improve finding a new solution. One of them is the SA with crystallization heuristic proposed by [9, 16]. The SA has two phases when the search is performed: exploration and refinement. **Figure 1** shows the two phases and their connection with the crystallization factor. The exploration of the domain space, usually happens in higher temperatures. The refinement of the solution happens in lower temperatures. There is no clear interface between the two phases.

For problems with combinatorial and integer parameters, the generation of the new candidate can be easily performed. In the case of combinatorial parameters, the algorithm exchanges the position of two parameters. In the case of integer parameters, the algorithm increments, decrements or keeps the current value. There are

![Figure 1](image-url)

*The crystallization heuristic represented by SA controlling each parameter crystallization factor.*
possibilities, considering that integer parameters have lower and upper bounds, it is possible to generate a new value performing a random selection in the range.

The control of continuous parameters is challenging. The crystallization heuristic proposed by [9, 16] considers a fixed maximum step, and the SA controls the probability density. Figure 1 shows that wider probability density allows larger jumps with higher probability. While, thinner probability density allows smaller jumps with higher probability. However, a larger jump still possible to happen with very small probability. This feature allows the SA to escape from local minima.

Each continuous parameter has a crystallization factor, which is represented by \( c_j \). Considering the crystallization heuristic, the new candidate is determined by summing \( c_j \) times a random number, this procedure determines a Bates distribution. It is represented by

\[
x^* = x + \frac{1}{c_j} \sum_{k=1}^{c_j} \text{random}(-1/2, 1/2) \cdot \Delta r_j \cdot e_j.
\]

where \( x \) is the current solution, \( x^* \) is the new candidate, they are represented by vectors. \( \Delta r_j \) is the fixed step size associated with continuous parameter \( j \). \( e_j \) represents the selected parameter, \( e_j \) is a vector with all elements equal to zero except one, the position associated with the \( j \)-th parameter. A common value assigned to \( \Delta r_j \) depends on the search interval, as \( \Delta r_j = (\max_j - \min_j)/4 \). This value will provide enough exploration for the algorithm.

It remains to explain the procedure in which the SA controls the crystallization factor for each parameter. The SA modifies only one parameter at a time, and according the decision of accepting or rejecting the new candidate, an action is performed. The procedure is represented in Figure 2. If the new candidate is rejected, it is assumed that the SA is performing exploration and to increase accepted solutions the crystallization factor associated with this parameter is increased. The increase in the crystallization factor will reduce the probability of larger jumps for this parameter. On the other hand, if the new candidate is accepted, it is assumed that the SA is performing refinement and to increase exploration the crystallization factor associated with this parameter is reduced. The reduction of the crystallization factor will enlarge the probability of larger jumps for this parameter.
Algorithm 2: SA with Crystallization Heuristic.

1: \(i \leftarrow 0; \ c \leftarrow (1, 1, \ldots, 1)\)
2: \(x \leftarrow \text{< random initial solution >}\)
3: \(T_0 \leftarrow \text{< initial temperature >}\)
4: \(\Delta r \leftarrow \text{< range of the parameters >}\)
5: \(\text{while } < \text{ Global condition not satisfied } > \text{ do}\)
6: \(\text{ while } < \text{ Local condition not satisfied } > \text{ do}\)
7: \(j \leftarrow \text{< select parameter to modify >}\)
8: \(x^* \leftarrow x + \frac{1}{c_j} \sum_{i=1}^{c_j} \text{random} \left( -\frac{1}{2}, \frac{1}{2} \right) \cdot \Delta r_j \cdot e_j\)
9: \(\Delta E = F(x^*) - F(x)\)
10: \(\text{if } \Delta E < 0 \text{ then}\)
11: \(x \leftarrow x^*\)
12: \(\text{if } c_j > 1 \text{ then}\)
13: \(c_j \leftarrow c_j - 1\)
14: \(\triangleright \text{ Positive Feedback}\)
15: \(\text{else}\)
16: \(\text{if } \text{random}(0, 1) < e^{-\Delta E/T_i} \text{ then}\)
17: \(x \leftarrow x^*\)
18: \(\text{if } c_j > 1 \text{ then}\)
19: \(c_j \leftarrow c_j - 1\)
20: \(\triangleright \text{ Positive Feedback}\)
21: \(\text{else}\)
22: \(c_j \leftarrow c_j + 1\)
23: \(\triangleright \text{ Negative Feedback}\)
24: \(\text{end if}\)
25: \(\text{end if}\)
26: \(i \leftarrow i + 1\)
27: \(T_i \leftarrow T_{i-1} \star \alpha\)
28: \(\text{end while}\)

Algorithm 2 describes the implementation of SA with crystallization heuristic. In this algorithm, the crystallization heuristic adjusts the modification to increase the acceptance of new solutions. The crystallization factor \(c\) is initialized with 1 for all continuous parameters, and the fixed step size \(\Delta r\) is defined as 25\% of the search range. For example, if the search happens between \(-100\) and \(+100\), in this case \(\Delta r_i = 50\).

3. Application in cutting and packing

The field of operations research concentrates many real world applications of optimization techniques in engineering, as it essentially aims to increase the efficiency of industry operations [8, 9, 16–19]. Interest in this area has grown recently due to advances in optimization algorithms and the importance of the waste and pollution reduction, which is usually an effect of an improved solution.

Among operations research subjects, cutting and packing (C&P) problems can be highlighted due to its importance and singular combination of geometry and optimization. These characterises are even more prominent in irregular packing problems, which involves simple polygonal items. Essentially, C&P problems consists of assigning a set of small items to a set of set of large containers, maintaining some geometric restrictions, while minimizing an objective function.
In this section, a SA based solution for the irregular packing problem is proposed. It adopts a discrete objective function, as it is related to the number of items in the container, while using continuous parameter for items rotations.

3.1 Irregular bin packing problem with free rotations

In the 2D irregular single bin packing problem, given a collection of items, one must place a subset of these inside a rectangular container with the aim of minimizing the unused space inside the bin. Each item is represented by a simple polygon and may be rotated by any angle. The main geometric restrictions dictate that no two items may overlap and there should be no protrusion from the container.

Consider a set of items $P = \{P_1, P_2, \ldots, P_n\}$ and a rectangular container $C$. The layout can be represented by the translation vector $\{t_1, t_2, \ldots, t_n\}$, a rotation vector $\{r_1, r_2, \ldots, r_i\}$ and an assignment set $T$, which contains the subset items placed inside the container. The irregular bin packing problem can be described as the minimization of the difference between the area of the container $A(C)$ and the assigned set of items $A(T)$, as

$$\text{minimize } A(C) - A(T)$$

subject to

$$i(P_i(r_i) \oplus t_i) \cap (P_j(r_j) \oplus t_j) = \emptyset \text{ for } i, j \in T \text{ and } i < j$$

$$(P_i(r_i) \oplus t_i) \subseteq C \text{ for } 1 \leq i \leq T$$

$$0 \leq r_i < 2\pi \text{ for } 1 \leq i \in T$$

$$t_i \in \mathbb{R}^2 \text{ for } 1 \leq i \in T$$

(3)

where $P(r)$ represents an item rotate by $r$, operator $i(P)$ expresses only the interior of $P$ and $P \oplus t$ indicates a translation $t$ applied to $P$.

3.2 Solution using SA

One of the main challenges in irregular packing is the complexity of managing the geometric constraint, which results in fewer proposed solutions in the literature [20]. In order to optimize the layout without compromising the geometric feasibility, two main strategies are often employed. The first is to define a constructive heuristic, which places one item at a time, usually at the bottom-left position. In this approach, the optimization algorithm only controls the placement order; a popular solution is to employ genetic algorithms [2, 21]. The alternative is to allow the items to move freely inside the container while applying penalization on the item overlaps [22–24].

In [9], however, a different approach was adopted: a SA based algorithm was employed to directly control the placement order, as well as the position and rotation of each item. Items were placed sequentially, using the parameters given by the SA, until no more items fitted the container. Then, the objective function, which was the unoccupied area of the container, which is directly related to the subset of placed items, was evaluated.

The main difficulty was the definition of the item position parameter, which should always correspond to a valid placement, i.e., without overlap or protrusion from the container. It was given by the collision free region (CFR), a polygon describing the allowed placement region, as shown in Figure 3. The CFR can be obtained using modified Boolean operations on polygons, described in [25]. A continuous parameter evaluated to a position along the perimeter of the CFR, and the closest vertex was chosen as the placement position of the item. The rotation
was defined by a second controlled variable, and it had to be applied prior to the determination of the CFR.

Therefore, the solution optimization basically consisted of a SA controlling the placement and rotation parameter of each item in the layout. At each iteration, one parameter for a single item was changed, or two items were swapped in the placement order. Then, the cost was evaluated and the new solution was accepted according to (1). The crystallization factor described in Section 2 was applied to the rotation parameters.

3.3 Results and discussion

Six broken glass puzzle instances were created to evaluate the performance of the algorithm. These instances have a known optimal solution, which are displayed in Figure 4. Therefore, the effectiveness of the algorithm can be measured by its success rate, which relates to the number of executions converging to the optimal solution.

The tests were executed on a Phenom 9550 2.21GHz and the convergence condition for the SA was that (1) the cost variation for the final temperature was zero, and (2) the final solution has the lowest cost found. The initial temperature was adjusted by admitting initial 50% acceptance and a geometric cooling schedule was adopted with $\alpha = 0.98$ (as shown in Algorithm 1).

Table 1 shows the results data obtained for 30 executions of each instance. Results show that only very simple problems were solved optimally in every execution. This indicates that the SA algorithm should be complemented with heuristics approaches to increase the convergence rate. Nevertheless, given the complexity of the problem with free rotations, the fact that it found the optimal solutions for all instances is important, as some could not be achieved by enforcing simple heuristics such as bottom-left or larger first.

One important characteristic of the irregular bin packing problem with free rotations is that the objective function is discrete and some of the parameters are continuous. This is illustrated in Figure 5, which shows discrete cost values for different values for the leftmost item rotation. The SA solution was not affected by

![Figure 3](image-url)  
*Example of collision free region for item P.*

![Figure 4](image-url)  
*Instances for the irregular bin packing problem.*
this issue, as the results show, it can handle both continuous and discrete parameters and objective function. Moreover, the continuous rotation convergence for each item was improved by adopting the crystallization factor.

4. Application in TO

TO is a mathematical approach to determine distribution of material in a design domain such that the performance becomes maximized. The definition of performance could be different in each application. The physical problem and desired application determines the objective function and the constraints. Depending on the problem, different methods have been developed in the literature [26, 27].

For the TO problems with well defined objective functions and constraints, the gradient based algorithms have been widely used [28–30]. The sensitivity information converges the final results to the optimized topology. This method shows fast convergence and low computational costs. The gradient based TO methods use Optimality Criteria (OC), Method of Moving Asymptotes (MMA), Sequential Linear Programming (SLP), etc. to optimize the objective functions [31, 32]. In the cases that the objective function or its derivatives are not mathematically modeled or hard to calculate, using the non-gradient based algorithms are more advantageous [33, 34]. In the non-gradient based TO methods, GA and SA are the two most popular algorithms of optimization [3, 35–37]. Even these methods have high computational costs, but can optimize the topology without need to calculation of the derivatives and sensitivity information.

Using the SA method in the non-gradient based TO is beneficial because of reaching global minimum as well as provide the information of convergence.

| Instance | $N_{\text{items}}$ | $N_{\text{conv}}$ | $T_{\text{conv}}$ (s) | $P_{\text{conv}}$ |
|----------|-------------------|------------------|---------------------|----------------|
| Tangram  | 7                 | 130,169          | 722.83              | 58.8%          |
| Hole     | 7                 | 12,847           | 28.89               | 93.5%          |
| Simple   | 4                 | 24,604           | 49.58               | 100.0%         |
| Concave  | 5                 | 17,912           | 43.96               | 86.7%          |
| LF Fails | 5                 | 4,426            | 10.61               | 63.6%          |
| BL Fails | 5                 | 117              | 0.16                | 100.0%         |

Table 1. The results for the SA based irregular bin packing solution. $N_{\text{items}}$: total number of items. $N_{\text{conv}}$: average number of iterations. $T_{\text{conv}}$: average time until convergence (in seconds). $P_{\text{conv}}$: success rate.

Figure 5. Discrete cost behaviour with single continuous parameter variation.
The information of convergence such as the number of accepted and rejected solutions could be used in the evaluation of TO results. The available SA for TO uses random search to generate new solutions, while by using SA with crystallization heuristic [10], the search for new accepted solutions has been improved. After finishing the optimization process, a density filter can reduce gray area and discontinued regions.

4.1 SA for non-gradient based TO

The structural TO to minimize compliance in beams is a classic problem. This problem has been solved with gradient based methods in the literature [38] and the results have been used to verify TO with crystallization heuristic SA (as described in Section 2). The problem of minimizing compliance can be represented as the problem of minimizing strain energy, modeled as

$$\text{minimize } S = U^T K U, \text{ subjected to }: KU = F$$

(4)

where $F$, $U$, and $K$ are, respectively, external force, elastic deformation, and stiffness. The constraint is volume fraction that represents the final optimized topology should have less or equal volume of the design domain. By discretizing the design domain to $N$ square elements, the total strain energy can be calculated as

$$S = \sum_{e=1}^{N} (x_e p) u_e^T k_e u_e$$

(5)

where $x_e$ is the density of each element varying from a minimum value (to avoid singularity in matrix calculation) to 1, $p$ is the penalization parameter, $u_e$ is elastic deformation for element $e$ and $k_e$ is the stiffness for element $e$. The penalization factor $p$ penalizes the intermediate gray area in the Solid Isotropic Material with Penalization (SIMP) method to reduce gray area. In this case, the TO has continuous parameters and it is solved using SA with crystallization heuristic, as described in Section 2.

A new heuristic is included, after reaching the thermal equilibrium, the domain is regularized by filtering. The new density of each element after filtering gets some effects from the adjacent elements, as

$$x_{\text{filter}} = \frac{\sum_{e=1}^{N} w_e x_e}{\sum_{e=1}^{N} w_e}$$

(6)

where $w_e$ is the weighting function which is the filter radius minus distance of each adjacent element. It should be noted that the weighting function is zero outside of the filter radius and the density changes just inside the filter domain. The design domain and loading conditions for cantilever and half-MBB beam problems are shown in Figure 6. A comparison of obtained compliance from proposed method and some results from the literature is shown in Table 2.

As shown in Table 2, the results from this non-gradient based TO method are very close to the gradient based results. The main advantage of the proposed method is that there is no need to calculate derivatives of the objective function.

4.2 SA for multi-objective TO

The TO objective function can be complex and combine two or more objective functions. In such situations, the solution is not necessarily unique and comes with a
set of optimum solutions called Pareto Front. The Pareto Front curve shows the solutions where none of the objective functions can be improved without degrading the other objective functions. The Pareto Front curve can be used for trade-off the suitable solution within this set instead of considering the full range of every parameter. The traditional TO usually optimizes one objective function while considering the other objective functions fixed or as a constraint [39]. The SA showed also this ability to incorporate multiple objective functions, like the one presented by the CoAnnealing [40] and AMOSA [41].

| Volume fraction | Cantilever in [38] | Cantilever in SA | Half-MBB in [38] | Half-MBB in SA |
|-----------------|--------------------|-----------------|------------------|----------------|
| 0.5             | 75.05              | 79.87           | 82.37            | 89.65          |
| 0.6             | 62.89              | 65.70           | 69.19            | 73.36          |
| 0.7             | 55.27              | 57.10           | 60.61            | 63.39          |
| 0.8             | 49.89              | 51.29           | 54.85            | 56.63          |
| 0.9             | 46.29              | 47.36           | 50.57            | 51.79          |

*Table 2. The results for compliance of cantilever and half-MBB beam for different volume fractions.*

![Figure 6](image1.png)
*Figure 6. The design domain and loading for cantilever beam (left) and half-MBB beam (right).*

![Figure 7](image2.png)
*Figure 7. The Pareto Front curve for minimization of compliance and weight in cantilever problem.*
The CoAnnealing was used to solve TO problems and the Pareto Front was obtained; particularly, compliance and volume fraction as considered as cost functions in a cantilever beam, as shown in Figure 7. The points on the curve showed a good agreement with the results for that volume fraction and compliance [42]. The results shows that Coannealing can be used for the multiple objective TO problems.

5. Application to curve interpolation

The curve fitting is a CAD (Computer Aided Design) study area. This problem consists in determining a curve from a set of points, as shown in Figure 8. Two different types of curves can be determined, an interpolating curve that passes through the set of points and an approximating curve that passes near to the points. The set of points is shown in blue points, the interpolating curve in red line and the approximating curve the dashed line.

In this section, a SA approach is proposed to determine an approximation curve from a sequence of points, in which the control points are the continuous parameters and index corresponding points of the sequence are discrete parameters. All these parameters are adjusted by the SA. This study was started by Ueda et al. [43]. The piece-wise Bézier curve is used to overcome some difficulties, as explained in the following section.

5.1 Piece-wise Bézier curve

There are several curves that can be used in the curve fitting problem, each one of them has features that is a drawback. The Bézier curve’s control points modify all the curve, and regions that the curve is already fitted. On the other hand, the control points of the B-spline have a local influence, i.e., just a region of the curve is modified. One feature of the Bézier curve that aids in the fitting problem is that the curve always interpolates the first and last control point, while in the B-spline these points usually are not interpolated. It is possible to interpolates these points in the B-spline. However, a higher number of parameter is needed to achieve such feature, compared to the Bézier curve.

A piece-wise Bézier curve overcome the problem of the global influence of the control point. This curve is a sequence of cubic Bézier curves as shown in Figure 9, in which the last control point of curve \( p_3 \) is the first control point of the following curve. The determination of the second control point of the second curve \( p_4 \) is given by

\[
p_4 = p_3 - \beta \cdot (p_2 - p_3),
\]

where \( \beta \) is a parameter that controls the degree of curvature at the point of continuity. This parameter can be adjusted to achieve the desired smoothness in the curve.

Figure 8.
Approximating curve (dashed) versus the interpolating curve (red).
with \( \beta \) being a proportional factor that ensures the weak-G1 continuity between the curves, i.e., the tangent vector of the end of the first curve has the same direction but not necessary the same intensity of the tangent of the start of the second curve. Ueda et al. [44] proposed an algorithm for automatic evaluate the number of piece-wise Bézier curves necessary to interpolate a sequence of points.

### 5.2 SA for the approximation curve determination

Using a piece-wise Bézier curve has a advantage in two aspects. First, the curve is divided in several pieces, in which each curve segment approximates only part of the sequence. Second, it is known that a Bézier curve always interpolates its end points. The used cost function is

\[
F(\{p_i\}) = W \cdot \sum_{k=1}^{m-1} ||d_k - P_v(d_k)||^2 + (1 - W) \cdot L(P(u)).
\]

with \( p_i \) are the control points defining the curve, \( d \) is the sequence of points, \( P_v(d_k) \) is the projection of point \( d_k \) in the curve \( P \), \( L(P(u)) \) is the length of the curve, \( P \) is a piece-wise Bézier curve and \( W \) is a weight factor. This object function consists of two parts. \( \sum_{k=1}^{m-1} ||d_k - P_v(d_k)||^2 \) is the first part and it represents the discrepancy between the piece-wise curve and the sequence of points. The first and last points in the sequence are not considered as they already are on the determined piece-wise Bézier curves. For every point in the sequence \( d_k \) it is determined its projection in the piece-wise Bézier curve \( P_v(d_k) \) [45]. \( L(P(u)) \) is the second part, representing a regularization. It is used to avoid the over-fitting problem, the algorithm will search for short and smooth curves (and avoid long and sinuous curves).

SA needs to adjust the parameters defining a piece-wise Bézier curve that minimizes (8). Each piece-wise Bézier curve has two internal control points (as shown in Figure 9). The coordinates for each of these control points are continuous parameters and they are controlled using the crystallization heuristic. The connection point between two piece-wise Bézier curves (as point \( p_3 \) shown in Figure 9) is represented as an integer parameter. It is an index representing a point from the given sequence.

Consider the example from Figure 10, the continuous parameters are the coordinate of control points \( p_1, p_2 \) and \( p_5 \). Control point \( p_4 \) is determined by modifying \( \beta \) from (7). Control points \( p_0 \) and \( p_6 \) are the first and last point of the sequence of points; and control point \( p_3 \) is a connecting point. The definition of this point is the discrete parameter adjusted by the SA.

The continuous parameters are controlled by the crystallization heuristic. The discrete parameters have specific operator, a random value is picked from a
uniform distribution between $[0, 1]$, if this value is greater than 0.5 the discrete parameter is increased by 1, otherwise it is decreased by 1.

5.3 Results and discussion

A curve is defined and sampled, and a random noise is added to each sampled point creating an artificial example to be tested by the proposed algorithm. Two example were tested, shown in Figure 11. Figure 11(a) is sampled with 101 points and Figure 11(b) is sampled with 300 points. The initial temperature is set as 100 and for each temperature there are 1000 iteration or 500 accepted solutions, and an adaptive cooling schedule [45] was adopted with a variable $\alpha$. A minimum of 5 accepted solutions is used as stop criteria.

Table 3 show the result of these two test examples. The objective function is higher in the second example, once $N_{pts}$ is higher as well the curve is longer. $N_{iter}$ is close in both example, there are less than 7% of difference between both tests.

The proposed interpolation algorithm was applied in different applications showing its versatility and robustness. Ueda et al. [46] proposed an algorithm to determine the curves which interpolates the boundary of a point cloud in space.
Ueda et al. [47] proposed an algorithm to determine the defect zone from a 3D scanned turbine blade. Ueda et al. [48] also proposed an algorithm to determine open boundaries in point clouds with symmetry (like injuries in a skull).

6. Conclusions

This chapter describes the SA with crystallization heuristic and three different applications: cutting and packing, topology optimization and curve interpolation. The cutting and packing problem determines the layout for a set of items to be placed inside a container with fixed dimensions. The cost function is the unused area inside the container. The items can be translated and rotated in the process of determining the layout. This cost function is discrete and the parameters are continuous. As there is no need of any gradient information, it can optimize discrete cost functions.

The topology optimization problem has continuous parameters. The number of parameters is much larger when compared with the cutting and packing problem. Usually, the cost function is one of the two possibilities: volume fraction and compliance. As there are two possibilities for cost function, it is considered the application of the CoAnnealing, which is a multi-objective version of the SA with crystallization heuristic. The last application is the curve interpolation. It has two different types of parameters: continuous and integer. The objective function is composed of two parts: the error between the the given points and the interpolating curve, and the length of the curve. The length of the curve is a regularization to force the curve to be shorter and smoother. The curve interpolation algorithm was used in different applications.

These examples show that the SA with crystallization heuristic is very generic, versatile and easy to implement.

Acknowledgements

G. C. Duran is supported by CNPq (process 140.299/2020-3). H. R. Najafabadi is supported by FAPESP (grant 2019/03453-2). M. S. G Tsuzuki is partially supported by CNPq (process 311.195/2019-9). A. K. Sato is supported by FUSP/Petrobras.

Nomenclature

| Example | \( N_{pts} \) | \( N_{segm} \) | \( N_{iter} \) | Cost |
|---------|--------------|--------------|--------------|------|
| 1       | 101          | 4            | 508,404      | 10.37|
| 2       | 300          | 5            | 473,680      | 134.97|

Table 3. The results for the SA based curve interpolation solution. \( N_{pts} \): total number of points, \( N_{segm} \): number of curve segments, \( N_{iter} \): number of iterations. Cost: objective function.
\( \Delta E \) Energy Variation
\( \Delta r_j \) Step Variation for Variable \( j \)
\( e_j \) Position Vector for Variable \( j \) (all elements are zero, except one)
\( F_e \) External Force
\( F(x) \) Objective Function
\( k_e \) Stiffness for Element \( e \)
\( i(P_k) \) Interior of the Item
\( L(P(u)) \) Length of Curve \( P(u) \)
\( K \) Stiffness
\( N \) Number of Square Elements
\( p \) Penalization Parameter
\( P_k \) Control Point
\( \mathcal{P} \) Set of Items
\( P_k \) Item
\( P_k(r) \) Item Rotated by \( r \)
\( P_k \oplus t \) Item Translated by \( t \)
\( P(u) \) Piece-wise Bézier Curve
\( P_v(\mathbf{d}_k) \) Projection of Point \( \mathbf{d}_k \) in Curve \( P(u) \)
\( \mathbb{R} \) Space of Continuous Numbers
\( r \) Rotation
\( S \) Strain Energy
\( t \) Translation
\( T \) Assigned Set of Items
\( T_i \) Temperature at iteration \( i \)
\( U \) Elastic Deformation
\( u_e \) Elastic Deformation for Element \( e \)
\( x \) Current Solution
\( x^* \) Solution Candidate
\( x_e \) Density of Element \( e \)
\( w_e \) Weighting Function
\( W \) Weight Used in the Approximation Curve

Abbreviations

AMOSA Archived Multiobjective Simulated Annealing
CAD Computer-Aided Design
CFR Collision Free Region
C&P Cutting & Packing
EIT Electrical Impedance Tomography
FEM Finite Element Method
GA Genetic Algorithm
MAP Maximum a Posteriori
MMA Method of Moving Asymptotes
OC Optimality Criteria
PDF Probability Density Function
PSO Particle Swarm Optimization
RANSAC Random Sample Consensus
SA Simulated Annealing
SIMP Solid Isotropic Material with Penalization
SLP Sequential Linear Programming
TO Topology Optimization
Author details

Tiago G. Goto¹,²†, Hossein R. Najafabadi²†, Guilherme C. Duran²†, Edson K. Ueda²†, André K. Sato²†, Thiago C. Martins²†, Rogério Y. Takimoto²†, Hossein Gohari³†, Ahmad Barari³† and Marcos S.G. Tsuzuki²*†

¹ Universidade Federal de Rondonópolis, Rondonópolis, Brazil

² Computational Geometry Laboratory, Escola Politécnica da USP, São Paulo, Brazil

³ Ontario Tech University, Oshawa, Canada

*Address all correspondence to: mtsuzuki@usp.br

† These authors contributed equally.
References

[1] J. V. Burke and M. C. Ferris, “A Gauss-Newton method for convex composite optimization,” Math Program, vol. 71, no. 2, pp. 179–194, 1995.

[2] P. R. Pinheiro, B. Amaro Júnior, and R. D. Saraiva, “A random-key genetic algorithm for solving the nesting problem,” Int J Comp Integ M, vol. 29, no. 11, pp. 1159–1165, 2016.

[3] S. Wang, K. Tai, and M. Y. Wang, “An enhanced genetic algorithm for structural topology optimization,” Int J Numer Meth Eng, vol. 65, no. 1, pp. 18–44, 2006.

[4] E. G. Castro and M. S. G. Tsuzuki, “Swarm intelligence applied in synthesis of hunting strategies in a three-dimensional environment,” Expert Syst Appl, vol. 34, no. 3, pp. 1995–2003, 2008.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science, vol. 220, no. 4598, pp. 671–680, 1983.

[6] 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, vol. 13, pp. 262–280, 1987.

[7] L. Ingber, “Very fast simulated re-annealing,” Mathematical and Computer Modelling, vol. 12, no. 8, pp. 967–973, 1989.

[8] T. C. Martins and M. S. G. Tsuzuki, “Simulated annealing applied to the rotational polygon packing,” IFAC Proc Vol (IFAC-PapersOnline), vol. 12, no. PART 1, 2006.

[9] 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 Braz Soc Mech Sci, vol. 30, no. 3, pp. 205–212, 2008.

[10] T. C. Martins, A. K. Sato, and M. S. G. Tsuzuki, “Adaptive neighborhood heuristics for simulated annealing over continuous variables,” in Simulated Annealing - Advances, Applications and Hybridizations (M. S. G. Tsuzuki, ed.), pp. 3–20, INTECHOpen, 2012.

[11] T. C. Martins, E. D. L. B. Camargo, R. G. Lima, M. B. P. Amato, and M. S. G. Tsuzuki, “Electrical impedance tomography reconstruction through simulated annealing with incomplete evaluation of the objective function,” in 33rd IEEE EMBC, pp. 7033–7036, 2011.

[12] T. C. Martins and M. S. G. Tsuzuki, “Electrical impedance tomography reconstruction through simulated annealing with total least square error as objective function,” in 34th IEEE EMBC, pp. 1518–1521, 2012.

[13] R. S. Tavares, T. C. Martins, and M. S. G. Tsuzuki, “Electrical impedance tomography reconstruction through simulated annealing using a new outside-in heuristic and GPU parallelization,” J Phys: Conf Series, vol. 407, p. 012015, dec 2012.

[14] V. Černý, “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm,” J Optimiz Theory App, vol. 45, no. 1, pp. 41–51, 1985.

[15] E. H. L. Aarts and P. J. M. van Laarhoven, “Statistical cooling: a general approach to combinatorial optimization problems,” Philips J Res, vol. 40, no. 4, pp. 193–226, 1985.

[16] T. C. Martins and M. S. G. Tsuzuki, “Placement over containers with fixed dimensions solved with adaptive neighborhood simulated annealing,” B
Pol Acad Sci-Tech, vol. 57, pp. 273–280, 2009.

[17] A. K. Sato, T. C. Martins, and M. S. G. Tsuzuki, “Rotational placement using simulated annealing and collision free region,” IFAC Proceedings Volumes, vol. 43, no. 4, pp. 234–239, 2010.

[18] A. K. Sato, T. C. Martins, and M. S. G. Tsuzuki, “An algorithm for the strip packing problem using collision free region and exact fitting placement,” CAD, vol. 44, no. 8, pp. 766–777, 2012.

[19] A. K. Sato, T. de Castro Martins, and M. S. G. Tsuzuki, “A pairwise exact placement algorithm for the irregular nesting problem,” Int J Comp Integ M, vol. 29, no. 11, pp. 1177–1189, 2016.

[20] J. A. Bennell and J. F. Oliveira, “The geometry of nesting problems: A tutorial,” Eur J Oper Res, vol. 184, no. 2, pp. 397 – 415, 2008.

[21] L. R. Mundim, M. Andretta, and T. A. de Queiroz, “A biased random key genetic algorithm for open dimension nesting problems using no-fit raster,” Expert Syst Appl, vol. 81, pp. 358–371, 2017.

[22] J. Egeblad, B. K. Nielsen, and A. Odgaard, “Fast neighborhood search for two- and three-dimensional nesting problems,” Eur J Oper Res, vol. 183, pp. 1249–1266, 2007.

[23] A. Elkeran, “A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering,” Eur J Oper Res, vol. 231, no. 3, pp. 757–769, 2013.

[24] A. K. Sato, T. C. Martins, A. M. Gomes, and M. S. G. Tsuzuki, “Raster penetration map applied to the irregular packing problem,” Eur J Oper Res, vol. 279, no. 2, pp. 657–671, 2019.

[25] A. K. Sato, T. C. Martins, and M. S. G. Tsuzuki, “Collision free region determination by modified polygonal boolean operations,” CAD, vol. 45, no. 7, pp. 1029–1041, 2013.

[26] M. P. Bendsoe and O. Sigmund, Topology optimization: theory, methods, and applications. Springer Science & Business Media, 2013.

[27] T. R. Marchesi, R. D. Lahuerta, E. C. N. Silva, M. S. G. Tsuzuki, T. C. Martins, A. Barari, and I. Wood, “Topologically optimized diesel engine support manufactured with additive manufacturing,” IFAC-PapersOnLine, vol. 28, no. 3, pp. 2333–2338, 2015.

[28] O. Sigmund and K. Maute, “Topology optimization approaches,” Struct Multidiscip O, vol. 48, no. 6, pp. 1031–1055, 2013.

[29] G. I. Rozvany, “A critical review of established methods of structural topology optimization,” Struct Multidiscip O, vol. 37, no. 3, pp. 217–237, 2009.

[30] A. Shukla and A. Misra, “Review of optimality criterion approach scope, limitation and development in topology optimization,” Int J Adv Eng Techn, vol. 6, no. 4, p. 1886, 2013.

[31] S. Rojas-Labanda and M. Stolpe, “Benchmarking optimization solvers for structural topology optimization,” Struct Multidiscip O, vol. 52, no. 3, pp. 527–547, 2015.

[32] F. A. Gomes and T. A. Senne, “An slp algorithm and its application to topology optimization,” Comput Appl Math, vol. 30, no. 1, pp. 53–89, 2011.

[33] O. Sigmund, “On the usefulness of non-gradient approaches in topology optimization,” Struct Multidiscip O, vol. 43, no. 5, pp. 589–596, 2011.

[34] D. Guirguis, W. W. Melek, and M. F. Aly, “High-resolution non-gradient topology optimization,” J Comput Phys, vol. 372, pp. 107–125, 2018.
[35] N. Garcia-Lopez, M. Sanchez-Silva, A. Medaglia, and A. Chateauneuf, “A hybrid topology optimization methodology combining simulated annealing and simp,” *Comput Struct*, vol. 89, no. 15-16, pp. 1512–1522, 2011.

[36] G. Y. Cui, K. Tai, and B. P. Wang, “Topology optimization for maximum natural frequency using simulated annealing and morphological representation,” *AIAA J*, vol. 40, no. 3, pp. 586–589, 2002.

[37] M. Ohsaki, “Genetic algorithm for topology optimization of trusses,” *Comput Struct*, vol. 57, no. 2, pp. 219–225, 1995.

[38] O. Sigmund, “A 99 line topology optimization code written in matlab,” *Struct Multidiscip O*, vol. 21, no. 2, pp. 120–127, 2001.

[39] K. Suresh, “A 199-line matlab code for pareto-optimal tracing in topology optimization,” *Struct Multidiscip O*, vol. 42, no. 5, pp. 665–679, 2010.

[40] T. C. Martins and M. S. G. Tsuzuki, “EIT image regularization by a new multi-objective simulated annealing algorithm,” in *IEEE EMBC 2015*, pp. 4069–4072, IEEE, 2015.

[41] T. C. Martins, A. V. Fernandes, and M. S. G. Tsuzuki, “Image reconstruction by electrical impedance tomography using multi-objective simulated annealing,” in *IEEE ISBI 2014*, pp. 185–188, jul 2014.

[42] H. R. Najafabadi, T. G. Goto, T. C. Martins, A. Barari, and M. S. G. Tsuzuki, “Multi-objective topology optimization using simulated annealing method,” in *ICGG 2020*, pp. 343–353, Springer.

[43] E. K. Ueda, M. S. G. Tsuzuki, R. Y. Takimoto, A. K. Sato, T. C. Martins, P. E. Miyagi, and R. S. U. Rosso Jr, “Piecewise Bézier curve fitting by multiobjective simulated annealing,” *IFAC-PapersOnLine*, vol. 49, pp. 49–54, jan 2016.

[44] E. K. Ueda, T. C. Martins, and M. S. G. Tsuzuki, “Planar curve fitting by simulated annealing with feature points determination,” *IFAC-PapersOnLine*, vol. 51, pp. 290–295, jan 2018.

[45] E. K. Ueda, A. K. Sato, T. C. Martins, R. Y. Takimoto, R. S. U. Rosso Jr, and M. S. G. Tsuzuki, “Curve approximation by adaptive neighborhood simulated annealing and piecewise Bézier curves,” *Soft Comput*, vol. 24, p. 18821–18839, 2020.

[46] E. K. Ueda, M. S. Tsuzuki, and A. Barari, “Piecewise Bézier curve fitting of a point cloud boundary by simulated annealing,” in *INDUSCON 2018*, pp. 1335–1340, jan 2019.

[47] E. K. Ueda, A. Barari, A. K. Sato, and M. S. G. Tsuzuki, “Detection of defected zone using 3D scanning data to repair worn turbine blades,” in *21st IFAC World Congress*, pp. 10666–10670, 2020.

[48] E. K. Ueda, A. Barari, and M. S. G. Tsuzuki, “Determination of open boundaries in point clouds with symmetry,” in *ICGG 2020*, pp. 332–342, Springer.