Soft Computing
https://doi.org/10.1007/s00500-020-05114-0

METHODOLOGIES AND APPLICATION

Curve approximation by adaptive neighborhood simulated annealing and piecewise Bézier curves

E. K. Ueda[1] · A. K. Sato[1] · T. C. Martins[1] · R. Y. Takimoto[1] · R. S. U. Rosso Jr[2] · M. S. G. Tsuzuki[1]

Abstract
The curve approximation problem is widely researched in CAD/CAM and geometric modelling. The problem consists in determining an approximating curve from a given sequence of points. The usual approach is the minimization of the discrepancy between the approximating curve and the given sequence of points. However, the minimization of just the discrepancy leads to the overfitting problem, in which the solution is not unique. A new approach is proposed to overcome this problem, in which the length of the approximating curve is used as a regularization increasing the algorithm stability. Another new proposal is the discrepancy determination, in which a method that has the best ratio between accuracy and processing time is proposed. A new simulated annealing (SA) approach is used to minimize the problem, in which the next candidate is determined by a probability distribution controlled by the crystallization factor. The crystallization factor is low for higher temperatures ensuring the exploration of the domain. The crystallization factor is high for lower temperatures, corresponding the refinement phase of the SA. The approximating curve is represented as a piecewise cubic Bézier curve, which is a sequence of several connected cubic Bézier curves. The piecewise Bézier curve supports a new proposed data structure that improves the proposed algorithm. A comparison is also made between the used single-objective SA and the AMOSA multi-objective SA. The results showed that the proposed single-objective SA finds a solution which is not dominated by the Pareto front determined by AMOSA. The results also showed that the regularization stabilized the algorithm, in which the increase in parameters does not lead to the overfitting problem. The proposed algorithm can process even complex curves with self-intersections and higher curvature.

Keywords Piecewise Bézier curve · Curve approximation · Adaptive neighborhood simulated annealing · Multi-objective simulated annealing

1 Introduction

The curve approximation problem is an area of research in geometric modeling (Piegl and Tiller 1997; Prautzsch et al. 2002) and CAD/CAM (Patrikalakis and Maekawa 2009; Maekawa et al. 2007). The curve fitting algorithm is used in the reverse engineering (Várady et al. 1997; Sobrinho et al. 2009), path planning (Lin et al. 2019; Tavares et al. 2011) and calligraphy digitization (Itoh and Ohno 1993; Yang et al. 2001; Dong et al. 2008). In the literature, the number of parameters is selected by the user, and the required number of parameters is still an open issue in the curve approximation problem. If the number of parameters is low, the approximating curve will not be good. Conversely, if the number of parameters is too high, there is the overfitting problem. Thus, this work proposes a method that overcomes the overfitting problem even if the number of parameter is higher than necessary.

Another problem of the curve approximation is the determination of the discrepancy between a point and the approximating curve. There are several approaches for the discrepancy determination in the literature; some of them focus on precision, and others focus on speed. Pandunata and Shamsuddin (2010) and Hasegawa et al. (2013) used a predetermined parametrization, which is fast, but it can converge to a wrong value. Maekawa et al. (2007) and Gofuku

Communicated by V. Loia.

✉ M. S. G. Tsuzuki
mtsuzuki@usp.br

[1] Computational Geometry Laboratory, Escola Politécnica da Universidade de São Paulo , São Paulo, Brazil

[2] Department of Computer Science, Universidade do Estado de Santa Catarina, Florianópolis, Brazil

Published online: 09 July 2020

et al. (2009) used an iterative method which is accurate; however, for higher accuracy; the processing time is longer. This work proposes a method to determine the discrepancy between a point and the approximating curve in which the accuracy is good, but the processing time is still low. Another advantage of the proposed method is determining the discrepancy in the case of more complex curves, i.e., curves with self-intersection and high curvature. The iterative methods existing in the literature can diverge in these cases; on the other hand, the proposed method does not have any restriction.

1.1 Literature review

For a given sequence of points, interpolation and approximation curves can be determined. An interpolation goes through all the given points, and an approximation curve approaches the given points. The exact definition of how near the given points are to an approximation curve depends on the chosen algorithm. Several interpolation curves were proposed in the literature (Piegl and Tiller 1997; Prautzsch et al. 2002), such as the piecewise Hermite curve. Maekawa et al. (2007) developed a geometric algorithm to determine a B-spline curve that interpolates a sequence of points. Gofuku et al. (2009) improved the geometric algorithm by including additional curve information, such as tangent and normal vectors. This geometric algorithm has limitations; it is valid only for plane curves, as for 3D curves, the normal vector is not unique. Hence, Okaniwa et al. (2012) again improved the geometric algorithm to include the curvature vector; this way, the B-spline interpolation curve can be applied to a 3D sequence of points.

Interpolation curves have a drawback; it is necessary to include at least one control point for each point in the sequence. If a sequence has a huge number of points, too many control points are thus included. A curve with many control points is not editable. Conversely, approximation curves require a small number of control points and are editable. Piegl and Tiller (1997) describes an algorithm that determines an approximation Bézier curve based on the linear least squares. Other researchers use metaheuristics. Pandunata and Shamsuddin (2010) and Hasegawa et al. (2013) used the differential evolution (DE) method to determine the curve that approximates a sequence of points. These three works Piegl and Tiller (1997), Pandunata and Shamsuddin (2010) and Hasegawa et al. (2013) approximate the sequence of points with just one Bézier curve. Bézier curves are known not to have local controllability, and once a control point is modified, the entire curve suffers modification. Sobrinho et al. (2009) used the simulated annealing (SA) to determine a B-spline curve that approximates a sequence of points. B-spline curves are known to have local controllability; once a control point is modified, only part of the B-spline curve is affected. Adi et al. (2010) used the particle swarm optimization (PSO) to determine an approximation NURBS curve to a sequence of points. A NURBS curve has similar properties when compared to B-splines, but a larger number of parameters.

Three additional approaches do not use metaheuristics to directly determine the control points to approximate the sequence of points. In the first approach, the relevance of each sampled point is evaluated and the control points are determined by solving a linear system (Loucera et al. 2014; Zhao et al. 2013). In the second approach, the spline representation with free knots is considered and the spline curve knots are determined (Jupp 1978; Schwetlick and Schütze 1995). This approach has some advantages; it can be used when the sequence of points to be approximated is discontinuous and it has turning points. The last approach is a hybridization of the previous two; the knot vector and the curve parametrization or the curve control points are determined (Gálvez and Iglesias 2013; Gálvez et al. 2015).

This work proposes a method of curve approximation by using piecewise Bézier curve segments to approximate a sequence of points. An adaptive neighborhood SA (ANSA) is used with a new cost function, which is the result of the discrepancy between the approximating curve and the sequence of points summed with and the approximating curve length. The length is a regularization function that increases the stability of the algorithm. ANSA and a multi-objective SA are compared.

This work is structured as follows: In Sect. 2, the curve approximation problem is defined, along with the curve representation and discrepancy determination. The optimization algorithms (ANSA and multi-objective) are described in Sect. 3. Section 4 gives an overview of the proposed algorithm. The proposed curve parametrization is given in Sect. 5, and the improved algorithm to calculate the discrepancy between the sequence points and the approximating curve is described in Sect. 6. The improvements are incorporated in the proposed algorithm and detailed in Sect. 7. Section 8 exhibits experimental results, and the conclusions are drawn in Sect. 9.

2 Problem description

The bi-dimensional approximating curve problem consists in generating a curve that minimizes its discrepancy with a sequence of points. This is a broad definition, as discrepancy is defined according to each application. Moreover, the solution also depends on the adopted curve representation. In this work, the piecewise Bézier curve is manipulated by modifying its control points. Next, the piecewise Bézier curve is discretized and transformed into a new sequence of points. This new sequence of points is compared with the given

sequence of points, and their discrepancy is calculated. Some aspects of the problem are discussed in this section so as to define the approximating curve problem as an optimization problem.

2.1 Bézier curve and the piecewise Bézier curve

A Bézier curve is given by

$$\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{p}_i B_{i,n}(u), \quad u \in [0, 1] \tag{1}$$

where $\mathbf{p}_i$ are the control points, u is the parameter, the total number of control points is $n + 1$, and $B_{i,n}(u)$ are the Bernstein polynomial basis functions defined as

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad i = 0, \ldots, n \tag{2}$$

where u, i and n are the input parameters and $\binom{n}{i}$ is the binomial function, given by

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad \binom{0}{0} \equiv 1. \tag{3}$$

Note that by modifying any of the control points, the entire Bézier curve is modified. However, using Bézier curves is preferred instead of B-splines and NURBS curves because of their simplicity. One solution is using piecewise Bézier curves. Piecewise Bézier curves, as shown in the following sections, support a data structure that accelerates the determination of the curve discrepancy. The piecewise Bézier curves consist of a sequence of Bézier curves in which its last control point connects to the following Bézier curve by being its first control point. Additionally, the control points adjacent to a connecting control point have a relation defined by

$$\mathbf{p}_{3i+4} = \mathbf{p}_{3i+3} - (\mathbf{p}_{3i+2} - \mathbf{p}_{3i+3}) \cdot \beta_{i+1}, \quad i = 0, \ldots, j-1. \tag{4}$$

where j is the number of Bézier curves and β_{i+1} is the continuity factor. If $\beta_{i+1} = 1$ for $0 \leq i \leq j-1$, the piecewise Bézier curves are connected with G^1 continuity. This expression ensures that two curves sequentially connected have the tangent vectors at the connecting point with the same direction, but the intensities can be different. An example is shown in Fig. 1, which is a composition of two cubic Bézier curves.

2.2 Discrepancy definition

In approximating curve algorithms, the objective is to reduce the discrepancy between a sequence of the points and the approximating curve, which is also denoted as the approximation error. It is usually assumed as the squared sum of the distance from each point in the sequence to the approximating curve. When using Bézier curves, this distance is defined as the Euclidean distance from the point to its closest point lying on the approximating Bézier curve, given by

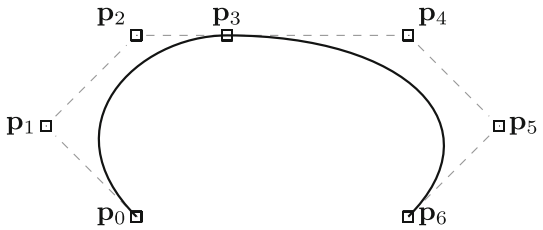


Fig. 1 Piecewise cubic Bézier curve with 2 Bézier curves, in which $\mathbf{p}_3$ is a connecting control point. Notice that control points $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ are colinear

$$||\mathbf{d}_k - \mathbf{P}(u_k)||, \tag{5}$$

where d_k is a point from the sequence and u_k is the parameter corresponding to the nearest point on the approximating curve.

2.3 Search for the closest point for Bézier curves

The determination of u_k is not straightforward. There are three main approaches on how to determine the nearest point on the Bézier curve $\mathbf{P}(u_k)$: chord length parametrization, point-curve projection and curve discretization. We here use the curve discretization approach.

Piegl and Tiller (1997), Hasegawa et al. (2013) and Pandunata and Shamsuddin (2010) calculated u_k using chord length parametrization, which approximates u_k as a fraction between the sum of chords up to the respective given point $\mathbf{d}_k$ and the sum of all chords, as shown in

$$u_k = \frac{\sum_{i=1}^{k} ||\mathbf{d}_i - \mathbf{d}_{i-1}||}{\sum_{i=1}^{m} ||\mathbf{d}_i - \mathbf{d}_{i-1}||}, \quad 1 < k < m. \tag{6}$$

Figure 2 shows an example in which this method does not determine the correct nearest point on the curve. For a given point $\mathbf{d}_3$ from the sequence, the point on the curve should be $\mathbf{d}_3$ itself; instead, this method determines $\mathbf{P}(u_3)$.

Maekawa et al. (2007) determined the closest point using a point-curve approach and a Newton–Raphson algorithm that searches for a parameter u_k where

$$(\mathbf{d}_k - \mathbf{P}(u_k)) \cdot \dot{\mathbf{P}}(u_k) = 0. \tag{7}$$

This represents that the vector between $\mathbf{P}(u_k)$ and $\mathbf{d}_k$ is orthogonal to $\dot{\mathbf{P}}(u_k)$, and $\dot{\mathbf{P}}(u_k)$ represents the curve tangent at u_k, as shown in Fig. 3.

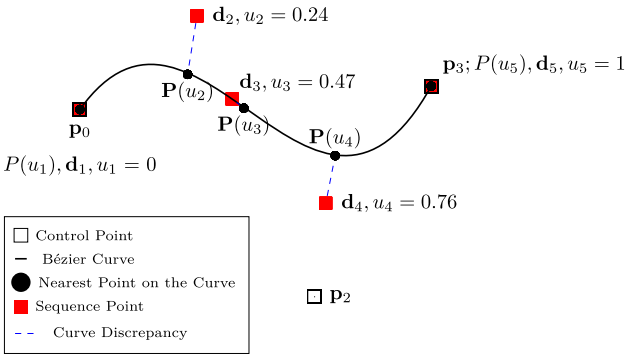


Fig. 2 Nearest point determination using chord length parametrization. For the point on curve $\mathbf{d}_3$, the nearest point on the curve is wrongly calculated as $\mathbf{P}(u_3)$

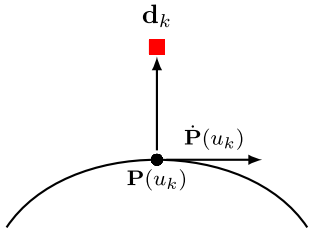


Fig. 3 Determination of u_k that defines the nearest point $\mathbf{P}(u_k)$ on the curve to $\mathbf{d}_k$ using the point-curve approach. The vector connecting $\mathbf{P}(u_k)$ to d_k is orthogonal to tangent $\dot{\mathbf{P}}(u_k)$ at $\mathbf{P}(u_k)$

The latter approach is the curve discretization in which the curve is represented by its set of sampled points. Then, the point on the curve nearest to $\mathbf{d}_k$ is determined by comparing $\mathbf{d}_k$ with the set of sampled points. Tavares et al. (2011) utilized this method; however, according to Hasegawa et al. (2013), the number of points discretizing the curve influences the result quality and if a higher number of points is used, the running time increases.

This work uses the latter approach, with additional processing to enhance the calculation and to avoid the problems pointed out by Hasegawa et al. (2013). The algorithm complexity is reduced, and the quality is preserved.

2.4 Approximation Bézier curve problem definition

Given a sequence $\{\mathbf{d}\}$ of m points, the discrepancy between $\{\mathbf{d}\}$ and a Bézier curve defined by control points $\{\mathbf{p}_i\}$ is given by

$$f(\{\mathbf{p}_i\}, \{\mathbf{d}\}, \{u_k\}) = \sum_{k=1}^{m} ||\mathbf{d}_k - \mathbf{P}(u_k)||^2, \tag{8}$$

where, for a given point $\mathbf{d}_k \in \{\mathbf{d}\}$, the nearest point on the Bézier curve is $\mathbf{P}(u_k)$. The Bézier curve is constructed such that $\mathbf{d}_1 = \mathbf{P}(u_1 = 0)$ and $\mathbf{d}_m = \mathbf{P}(u_m = 1)$. Therefore, the problem could be defined as the minimization of function f.

However, as there are usually multiple solutions for Eq. (8), other factors can also be considered. Here, the total curve length is also minimized, along with discrepancy f, as it usually leads to more practical solutions.

Considering the approach in which the Bézier curve $P(u_k)$ defined by control points $\{\mathbf{p}_i\}$ is discretized by a set of sampled points $\{\mathbf{q}\}$ with l points. In this situation, the discrepancy between $\{\mathbf{q}\}$ and $\{\mathbf{d}\}$ is given by

$$f(\{\mathbf{q}\}, \{\mathbf{d}\}) = \sum_{k=1}^{m} ||\mathbf{d}_k - \mathbf{q}_j||^2, \tag{9}$$

where $\mathbf{q}_j$ is the closest point to $\mathbf{d}_k$ considering the sample sequence of points $\{\mathbf{q}\}$. As pointed out by Hasegawa et al. (2013), the number of points discretizing the curve influences the result quality and if a higher number of points is used, the running time increases. A data structure is here proposed, whereby a reasonable number of discretizing points is enough to avoid the problems pointed out by Hasegawa et al. (2013). Additionally, self-intersecting curves and curves with higher curvature can be correctly processed.

3 Simulated annealing optimization

As mentioned in the previous section, the approximating curve can be obtained through an optimization approach. When achieving exact solutions through mathematical approaches is unfeasible, as is the case of most NP-hard problems, a popular solution is to use a metaheuristics. Most metaheuristics are inspired by natural phenomena and are applicable to a wide range of cases. Some examples of metaheuristic are the genetic algorithm (Abualigah and Hanandeh 2015; Song and Chen 2018), the particle swarm optimization (Abualigah and Khader 2017) and the krill herd algorithm (Abualigah et al. 2018a, b; Abualigah 2019). See Abualigah (2019) for a more comprehensive overview of metaheuristics.

For the approximation curve problem, the application of metaheuristics achieves good solution in a reasonable amount of time. In the proposed solution, the SA metaheuristic is adopted to approximate a piecewise Bézier curve to the given sequence of points. As the objective is to minimize the discrepancy as well as the length, two approaches are considered. The first uses ANSA and penalizes the cost with the curve length. The second adopts a multi-objective SA, which generates a family of solutions. Both, ANSA and multi-objective SA, are introduced in general terms, and their application to the approximating curve problem is presented in the following sections.

3.1 Adaptive neighborhood SA

SA is a probabilistic metaheuristic used to solve a variety of combinatorial and continuous optimization problems. SA is based on the Metropolis algorithm (Metropolis et al. 1953), which simulates the atom configuration at a certain temperature. Kirkpatrick et al. (1983) proposed SA to solve combinatorial optimization problems by adopting a cooling scheme combined with the Metropolis algorithm. Theoretical studies have shown that the global optimum of a combinatorial optimization problem can be reached with probability one (Aarts and Korst 1989; Geman and Geman 1984; Lundy and Mees 1986). SA deals with all types of variables: combinatorial, fixed precision, cyclic and float precision. At higher temperatures, the SA performs the exploration of the domain space (global search). At lower temperatures, the SA performs the refinement stage (exploitation, local search).

During the execution of the SA, candidates are generated by applying random modification to the current solution and then, the objective function is reevaluated. If the new objective function is lower, the next candidate is accepted and the algorithm continues with the next iteration. However, if the objective function value increases, an additional evaluation is performed. In this case, considering the current temperature T, the next candidate is accepted with probability

$$P(\Delta E) = e^{-\frac{\Delta E}{T}} \tag{10}$$

where ΔE is the objective function variation. Different strategies exist for modifying continuous variables. Bohachevsky et al. (1986) proposed that, to obtain the next candidate $\mathbf{x}_{\text{nex}}$, one must generate a random direction unity vector $\boldsymbol{u}$, multiply it by a step size Δr and add this to the current solution $\mathbf{x}_{\text{cur}}$. Thus,

$$\boldsymbol{x}_{\text{nex}} = \boldsymbol{x}_{\text{cur}} + \Delta r \cdot \boldsymbol{u}, \tag{11}$$

where the step size Δr is unique in all directions, and $\boldsymbol{u}$ is a randomly generated vector from a uniform distribution. An anisotropic approach was proposed by Corana et al. (1987), in which the number of accepted candidates is kept as constant as possible by varying the step size in each direction according to the current temperature. This is achieved by modifying just one variable at a time. The next candidate is determined by

$$\boldsymbol{x}_{\text{nex}} = \boldsymbol{x}_{\text{cur}} + v \cdot \Delta r_i \cdot \boldsymbol{e_i} \tag{12}$$

where i is the index of the modified variable, Δr_i is its step size in direction $\boldsymbol{e_i}$, a unitary vector that represents each variable, and v is a uniform random number in $[-1, 1]$. Then, for higher temperatures, Δr_i is larger and exploration is allowed; for lower temperatures, Δr_i is smaller and SA focuses on refining the solution.

3.1.1 Adaptive neighborhood

The strategy used by Corana et al. (1987) may not be the best to escape from local optima, as the maximum step size decreases (Martins et al. 2012). The ANSA (Martins et al. 2012) avoids this drawback. As rejected candidates do not contribute to the progress of the SA, the probability distribution associated with each variable is adjusted to increase the number of accepted solutions. At each iteration, only one variable is modified and the next candidate is generated by

$$\boldsymbol{x}_{\text{nex}} = \boldsymbol{x}_{\text{cur}} + \frac{1}{c_i} \sum_{1}^{c_i} \text{random}(-1, +1) \cdot \boldsymbol{e_i} \cdot \Delta r_i \tag{13}$$

where c_i is the crystallization factor for the continuous variable with index i. The modification applied follows a Bates distribution centered on zero with amplitude 1.

Throughout the optimization process, the crystallization factor c_i of each variable is adjusted by the feedback that can be positive or negative. When a candidate is rejected, the standard deviation of the modified variable probability distribution is reduced, resulting in smaller modifications with higher probability. If a candidate is rejected, it is necessary to perform more refinement in the modified dimension and the crystallization factor is increased. If the new candidate is accepted, the standard deviation of the modified variable probability distribution is increased so as to increase the probability of larger modifications. If a candidate is accepted, it is necessary to perform more exploration in the modified dimension and the crystallization factor is decreased. Figure 4 shows the crystallization factor feedback. This behavior can be explained by observing that the crystallization factor c_i controls the standard deviation associated with the Bates distribution (see Fig. 5).

The ANSA had already been used to solve several problems in the literature, Martins and Tsuzuki (2008, 2010) used this algorithm in the cutting and packing problem; and Martins et al. (2016, 2019) used it in the electrical impedance tomography (EIT) inverse problem.

3.1.2 Annealing schedule

The annealing schedule is the function that defines how the temperature is cooled during the SA execution. It has a great influence on the final result. The logarithmic schedule leads to the slowest convergence; however, the convergence of the SA was only proved for the logarithmic cooling schedule (Cohn and Fielding 1999).

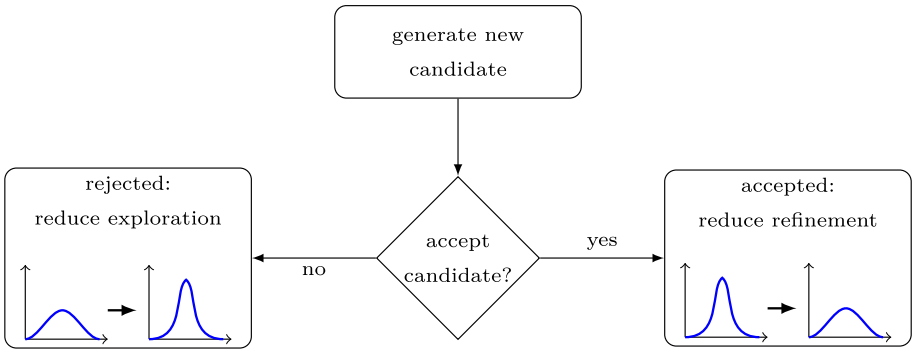


Fig. 4 Crystallization factor feedback scheme. If a candidate is accepted, the probability distribution is widened to improve the exploration. Otherwise, the distribution is narrowed and the search improves refinement

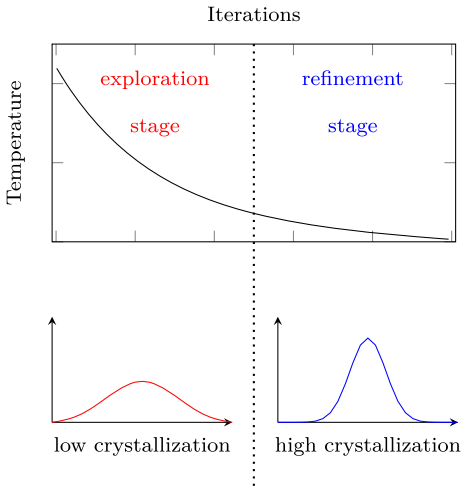


Fig. 5 Modification amplitude for different phases of the SA. The exploration phase has a wider probability distribution and, in the refinement phase, the probability distribution is narrowed

The geometric cooling schedule is the most commonly used, and it is given by

$$T_{k+1} = \alpha * T_k, \tag{14}$$

where $0 < \alpha < 1$ is the cooling factor and its value is kept constant. Adaptive cooling is a variation of geometric cooling and tries to determine the value of α by analyzing the standard deviation of the objective function values at each temperature. The adaptive cooling schedule is given by

$$T_{k+1} = T_k * e^{-\frac{\beta \cdot T_k}{\sigma(T_k)}}, \tag{15}$$

where β is a tunable factor and

$$\alpha = e^{-\frac{\beta \cdot T_k}{\sigma(T_k)}}. \tag{16}$$

Standard deviation $\sigma(T_k)$ is calculated from the probability distribution of the objective function, and it is given by

$$\text{Prob}_i(T_k) = \frac{e^{\frac{-C_i}{k_B T_k}}}{\sum_j e^{\frac{-C_j}{k_B T_k}}}. \tag{17}$$

The mean cost in a specific temperature is given by

$$\langle C(T_k) \rangle = \sum C_i \text{Prob}_i(T_k). \tag{18}$$

The quadratic mean cost is defined as

$$\langle C^2(T_k) \rangle = \sum C_i^2 \text{Prob}_i(T_k). \tag{19}$$

The variance is expressed as

$$\sigma^2(T_k) = \langle C^2(T_k) \rangle - \langle C(T_k) \rangle^2. \tag{20}$$

3.2 Multi-objective SA

Multi-objective optimization algorithms do not determine only one optimum global solution as the single-objective optimization. In the multi-objective optimization, there is not a single global optimum and there is a set of solutions that are equally important. This set of solutions is named Pareto curve or Pareto front, and each solution is a global optimum.

A multi-objective optimization algorithm was proposed by Bandyopadhyay et al. (2008), named AMOSA (archive multi-objective simulated annealing). This algorithm can accept a candidate in several ways; and the next candidate is compared with the current solution and with the solutions present in the archive. In some situations, the next candidate replaces the current solution and can be included in the archive. Additionally, some solutions from the archive can be removed. It uses clustering algorithms to keep the size of the Pareto front constant and small.

The AMOSA is initialized by creating an archive. Next, a random solution of the archive is selected as the initial current solution. There are two loops: The external loop uses the temperature as a stop criterion. The inner loop uses a maximum number of iterations. This algorithm uses a function named quantity of dominance, which evaluates the relation between

the current solution and the next candidate. The quantity of dominance is given by

$$\Delta \text{dom}_{a,b} = \prod_{i=1}^{M} \left(\frac{|f_i(a) - f_i(b)|}{R_i} \right), \tag{21}$$

where a is a solution and b is a candidate, or vice versa, M is the number of objective functions, with $f_i(a) \neq f_i(b)$, R_i is the range of the i-th objective function.

Considering that a is the current solution and b is the next candidate, there are three possible combinations:

- a dominates b—all the objective functions of a are smaller than the objective functions of b. Figure 6 shows two cases in which the current solution dominates the next candidate. In this case, the next candidate can be accepted with the probability

$$\text{prob} = \frac{1}{1 + \exp(\Delta \text{dom}_{\text{avg}} * T)} \tag{22}$$

where

$$\Delta \text{dom}_{\text{avg}} = \frac{\left(\sum_{i=1}^{k} \Delta \text{dom}_{i,\text{nex}-\text{can}} \right) + \Delta \text{dom}_{\text{cur}-\text{sol},\text{nex}-\text{can}}}{k+1}. \tag{23}$$

- There is no dominance between a and b—there are objective functions of a that are smaller than the objective functions of b; however, there are objective functions of b that are smaller than the objective functions of a. Figure 7 shows three situations in which there is no dominance between the current solution and the next candidate, as follows:

  (a) The next candidate is dominated by k solutions of the archive, as shown in Fig. 7a. In this case, the next candidate is adopted as current solution with the probability

$$\text{prob} = \frac{1}{1 + \exp(\Delta \text{dom}_{\text{avg}} * T)} \tag{24}$$

  where

$$\Delta \text{dom}_{\text{avg}} = \frac{\left(\sum_{i=1}^{k} \Delta \text{dom}_{i,nex-can} \right)}{k}. \tag{25}$$

  (b) There is no dominance between the next candidate and the solutions of the archive, as shown in Fig. 7b. In this case, the next candidate is adopted as the current solution and added to the archive.

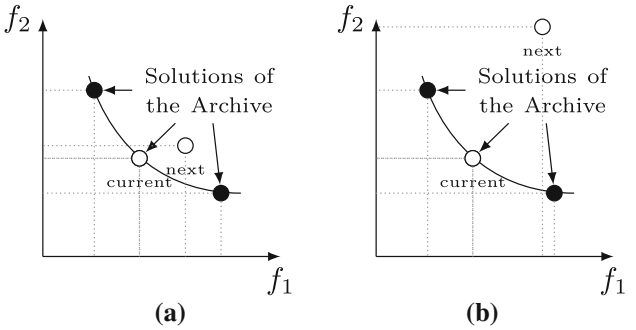


Fig. 6 The next candidate is dominated by the current solution. **a** There is no dominance between the next candidate and the solutions of the archive; **b** some solutions of the archive dominate the next candidate

  (c) The next candidate dominates k solutions of the archive, as shown in Fig. 7c. In this case, the next candidate is adopted as the current solution and added to the archive, and the k solutions of the archive dominated by next candidate are removed from the archive.

- b dominates a—all the objective functions of b are smaller than the objective function of a, as shown in Fig. 8. In this case, there are three situations:

  (a) The next candidate is dominated by k solutions of the archive, as shown in Fig. 8a. This case can only occur if the current solution is not from the archive. In this case, the minimum quantity of dominance between the next candidate and k solutions of the archive is calculated. This minimum quantity of dominance is named $\Delta \text{dom}_{\text{min}}$, and the corresponding solution of $\Delta \text{dom}_{\text{min}}$ is adopted as the current solution with the probability

$$\text{prob} = \frac{1}{1 + \exp(-\Delta \text{dom}_{\text{min}})}. \tag{26}$$

  If the next candidate has the minimum quantity of dominance, it is adopted as the current solution.

  (b) There is no dominance between the next candidate and the solutions of the archive, as shown in Fig. 8b. In this case, the next candidate is adopted as the current solution and added to the archive. If the current solution belongs to the archive, it is removed from the archive.

  (c) The next candidate dominates k solutions of the archive, as shown in Fig. 8c. In this case, the next candidate is adopted as the current solution and added to the archive, and the k solutions of the archive, dominated by next candidate, are removed from the archive.

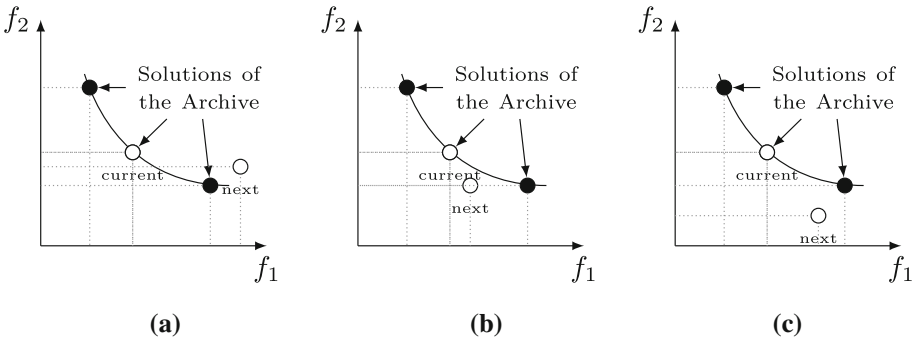


Fig. 7 There is no dominance between the next candidate and the current solution. **a** Some solutions of the archive dominate the next candidate; **b** there is no dominance between the next candidate and the solutions of the archive; **c** the next candidate dominates the k ($k \geq 1$) solutions of the archive

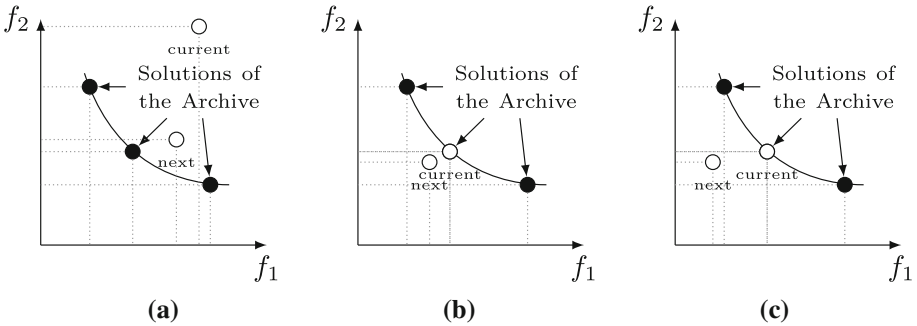


Fig. 8 The next candidate dominates the current solution. **a** Some solutions of the archive dominate the next candidate; **b** there is no dominance between the next candidate and the solutions of the archive, except the current solution; **c** the next candidate dominates some solutions of the archive, except the current solution

4 Proposed solution overview

The proposed methodology for the piecewise Bézier approximating curve is based on the SA metaheuristic. It is an iterative method in which the curve is modified and evaluated at each iteration. Then, the SA determines whether the next candidate is adopted or discarded. A worse next candidate can be accepted to escape local minimum. After the convergence condition is reached, the algorithm stops and the current solution is the final result. The flowchart from Fig. 9 shows the steps for the proposed iterative method.

As discussed in Sect. 2.4, one of the challenges for Bézier approximating curves is the determination of the discrepancy, i.e., the distance from the input sequence of points to the approximating curve. Here, the discretization of the approximating curve is performed to allow a fast discrepancy determination. The discrepancy along with the curve length contributes to the cost function, which is minimized by the SA. As there are two competing parcels to the cost, two approaches are proposed herein: ANSA and AMOSA. In the latter case, the output is a family of solutions which forms the Pareto front. All the other steps are similar for both propositions, with occasional minor modifications.

5 Solution representation for the SA

At each iteration of the SA, the first step consists in modifying a single parameter from the current solution (see Fig. 9). This requires adopting of a specific representation of the piecewise Bézier curve to allow for an efficient control by the SA. Desirably, the parameters should uniquely and fully describe the curve and there should not exist a combination of parameters that generates an unfeasible solution.

As the number of segments and the order is predefined, the number of control points is known. Thus, the proposed representation allows the SA to modify the coordinates of the control points (except the connecting points), while preserving the continuity of the first derivative. The sequence of points is used to define the position of the initial, final and connecting points, whereas the initial and final points are fixed, and the connecting point is controlled by the SA, as a connecting point can be any point from the sequence of points. Due to their different treatment by the SA, the parameters can be classified into continuous and discrete.

5.1 Continuous parameters

The continuous parameters adjusted by SA are the coordinates of the control points, with the exception of the endpoints of each segment and the second control point of all the segments after the second segment. The second control points for all the Bézier curves (except for the first Bézier curve) are controlled by a single continuity factor to enforce the weak-C^1 continuity. The continuity factor corresponds to the β_i value in Eq. (4), which determines the length of the segment connecting $\mathbf{p}_{3i+2}$ and $\mathbf{p}_{3i+3}$, where i is the index of the segment.

Figure 10 shows an example of a piecewise Bézier curve with three segments. In this example, the continuous param-

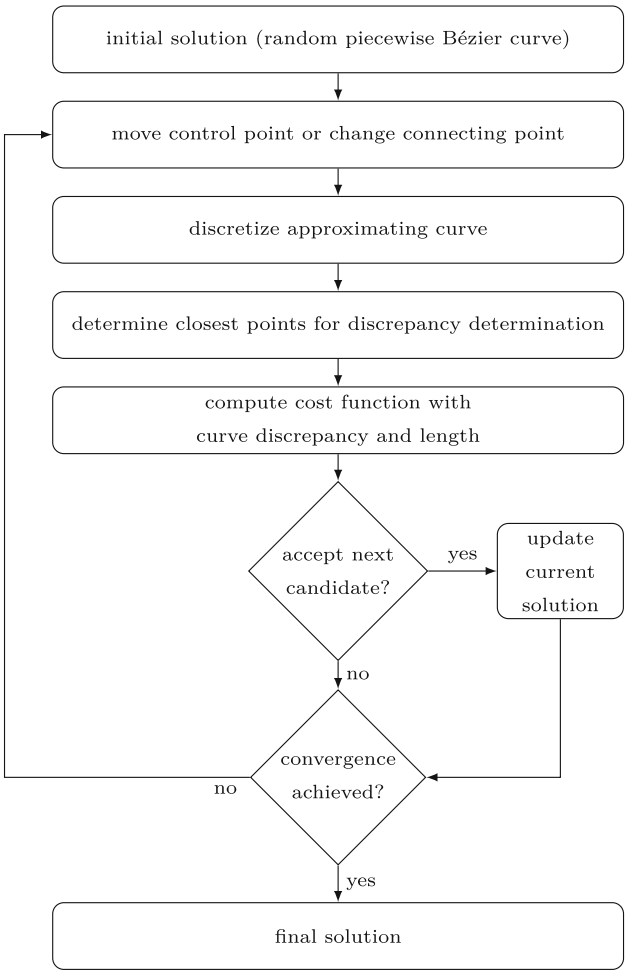


Fig. 9 Proposed methodology exclusively modifies control points and the connecting point. The approximating curve is discretized and its discrepancy related to the given sequence of points is calculated

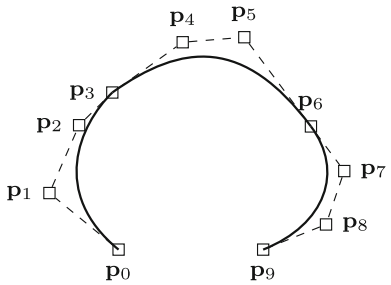


Fig. 10 Example of a piecewise cubic Bézier curve, with three segments. $\mathbf{p}_3$ and $\mathbf{p}_6$ are the connecting points, which are associated with integer parameters. Control points $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_6$ and $\mathbf{p}_8$ have their coordinates as continuous parameters. Control points $\mathbf{p}_5$ and $\mathbf{p}_7$ are indirectly controlled by the SA using the collinearity constraint

eters to be adjusted by SA are the coordinates of control points $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_5$ and $\mathbf{p}_8$ and the two continuity factors used to determine $\mathbf{p}_4$ and $\mathbf{p}_7$.

5.2 Discrete parameters

One of the challenges when adopting piecewise Bézier curve for approximation is the positioning of the connecting points, i.e., the control points shared among two segments. In the proposed parametric representation, the connecting points are assigned positions that coincide with the points belonging to the sequence of points. Thus, the connecting point can be described by the index of the selected point inside the sequence of points.

In the example of Fig. 10, points $\mathbf{p}_3$ and $\mathbf{p}_6$ represent the connection of two segments. Thus, two discrete parameters are needed to represent this curve.

6 Curve discretization and discrepancy determination

After the modification of a single parameter, the cost function must be updated to evaluate the next candidate. The cost is composed of two factors, the curve discrepancy and length. As both are complex to calculate, two steps are introduced, the curve discretization and the discrepancy computation.

6.1 Curve discretization

In order to reduce the computational cost of the objective function determination, the curve is discretized, i.e., converted to a sequence of line segments. The total number of segments (or points) is a parameter that controls the discretization refinement—the larger the number of segments, the more precise is the approximation. However, a very refined discretization incurs high computational costs, invalidating the advantages of the discretization.

Algorithm 1 describes the discretization process. The points are defined using Eq. (1), by adopting equally spaced parameters u. Each segment of the piecewise Bézier curve is divided into Δ sub-segments and are stored in the sequence of sampled points $\{q\}$.

This step is useful to determine both the curve length and the curve discrepancy. The total curve length is obtained by adding up the individual segments length. For determining the discrepancy, a two-step process is proposed to reduce approximation errors. For a given sequence point, its contribution to the discrepancy is determined by finding the closest sampled curve point and finding the distance by analyzing its neighbors sampled points. This procedure is explained as follows.

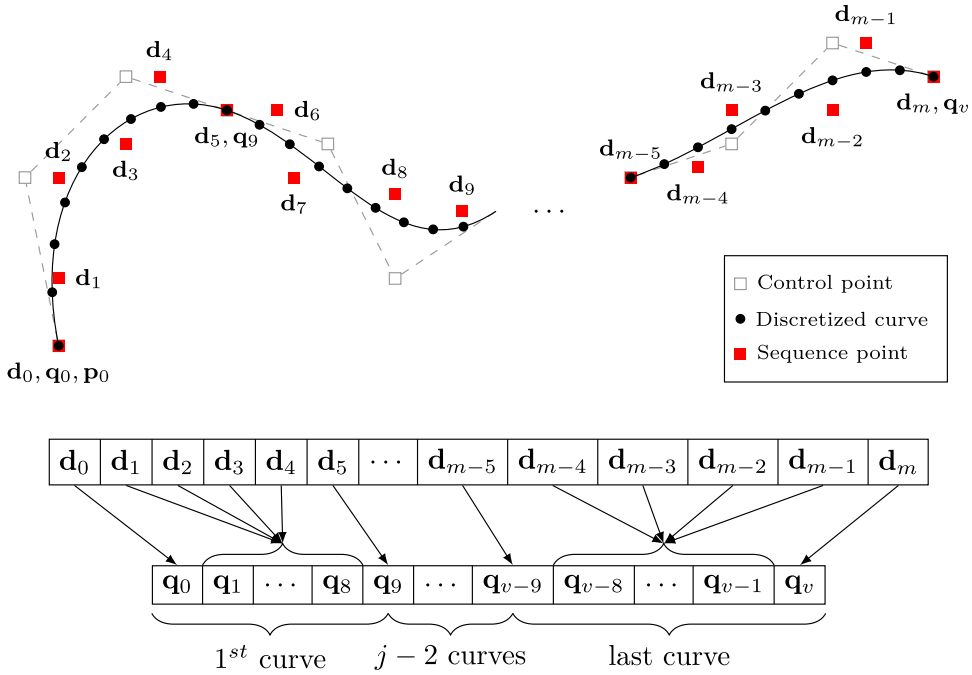


Fig. 11 Local search for the closest point. The given sequence of points {d} is shown, together with the approximating curve and the sequence of sampled points $\{q\}$ representing the approximating curve discretization. Each Bézier curve is associated with a subset of $\{d\}$ and a subset of $\{q\}$. Only corresponding subsets are compared in the search for the closest point

Algorithm 1 Piecewise Bézier curve discretization

```
1: procedure DiscretizeCurve(x)
2:     {p} ← <Extract controls points from x >
3:     l ← 0
4:     while l < j do                          ▷ j is the number of curves
5:         u ← 0.0
6:         while u < 1.0 do
7:             u ← u + 1.0/Δ                   ▷ Δ is the refinement parameter
8:             q_cur ← Σ_{i=3l}^{3l+3} p_i B_{i,3}(u)
9:             <Add q_cur to {q}>
10:        end while
11:        l ← l + 1
12:    end while
13:    return {q}
14: end procedure
```

6.2 Search for the closest point

The total discrepancy is the sum of all individual sequence point discrepancy, i.e., its distance to the approximating piecewise Bézier curve. As explained before, the discrepancy is approximated by adopting a discretization of the curve. The first step to determine the discrepancy contribution of a sequence point is to find the closest point in the sampled point list $\{q\}$. A full search in all the sampled points may be performed; however, a more efficient method is proposed.

As connecting points are always sequence points, each sequence point can be associated with a curve segment. Likewise, the points from $\{q\}$ can be assigned to each segment they were generated from. Therefore, the search for each sequence point can be limited by only investigating the corresponding subset of $\{q\}$. Figure 11 shows an example of the search procedure. For example, the search for the closest point from d_2 is limited to $q_1, \ldots, q_8$, which are associated with the first curve, as is d_2.

A further advantage is that the distance is locally determined, avoiding evaluation with sampled points which are further ahead in the sequence. This advantage is observed in examples in which the curve overlaps itself, as shown in Fig. 12. There are two curves: one represented by a dashed black line and the other by a blue line. On both examples, the curve length is close. If a global search considers all the points of $\{d\}$ and all the points of $\{q\}$, the discrepancy in both cases is similar. However, if $\{d\}$ and $\{q\}$ are divided into subsets and only corresponding subsets are compared, the discrepancy for both cases is very different. The dashed curve intersects itself, and the blue curve has no intersection; this is an important distinction.

6.3 Discrepancy determination

After determining the closest point for a point from $\{d\}$, the discretization approach approximates the discrepancy by the distance between the two points. However, if the curve is sampled with few points, this approximation is significantly imprecise. Therefore an enhancement is proposed here. Using the sequence point d_i, its closest point q_k and its two closest neighbors q_{k-1} and q_{k+1}, two triangles are defined, as shown in Fig. 13a. The height of the smallest triangle with acute angle α is taken as the distance from the curve, which is denoted δ. For the height determination, the segments defined by $\mathbf{q}_{k-1}\mathbf{q}_k$ and $\mathbf{q}_k\mathbf{q}_{k+1}$ are adopted as the base of the triangles. Thus, two situations can be observed: with both acute angles (see Fig.13a), and with one acute and

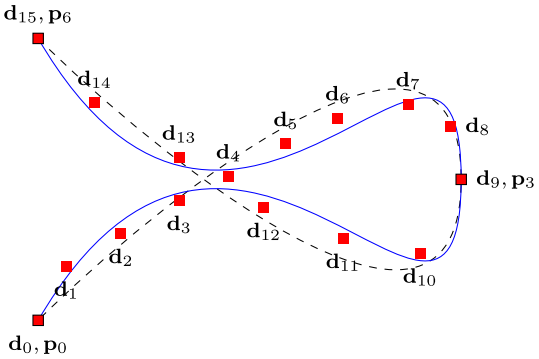


Fig. 12 Two-segments example. There is an intersection on the dashed curve, whereas no intersection for the blue curve. Considering the given sequence of points represented by the red squares, the correct approximating curve has self-intersection. The proposed data structure can differentiate between the two cases, and distinct discrepancy results are produced for each approximating curve. The curve with self-intersection has smaller discrepancy when compared with the curve with no self-intersection (color figure online)

one obtuse angle, as shown in Fig.13b. In the former case, the areas must be evaluated to determine δ, whereas, in the latter case, the height of the triangle with acute angle α is defined as the discrepancy. The third situation is observed when both angles α_1 and α_2 are obtuse, and the discrepancy is the distance between the sequence point and its closest sampled curve point. Figure 13c shows an instance with such conditions.

Algorithm 2 details the curve discrepancy determination. Initially, for each sequence point, the closest point is determined in the sampled sequence of points. Then, the area of each triangle is determined using the Heron formula given by

$$A = \sqrt{(s(s-a)(s-b)(s-c)}, \tag{27}$$

where a, b and c are the edges of the triangle and $s = (a + b + c)/2$ is the semi-perimeter. Also the cosine of angles α_1 and α_2 is computed to determine if they are acute or obtuse. The conditions are then verified and either the height of one of the triangles is accepted as the discrepancy (see Fig. 13a, b), or the distance between the two points is accepted as the discrepancy (see Fig. 13c). Then, the total discrepancy is obtained by summing all the individual discrepancies.

The proposed approach can be understood as the approximation of the distance between a point from $\{d\}$ and the discretized approximating curve represented by the sampled sequence of points $\{q\}$. This is a valid approach as the height of a triangle is orthogonal to the triangle base, which is a segment of $\{q\}$. When one of the angles is obtuse, the projection of one of the sides on the base lies outside the sub-segment and, therefore, cannot be adopted. Therefore, when both angles are obtuse, and the minimum distance is given by the distance between the two points.

Algorithm 2 Discrepancy determination

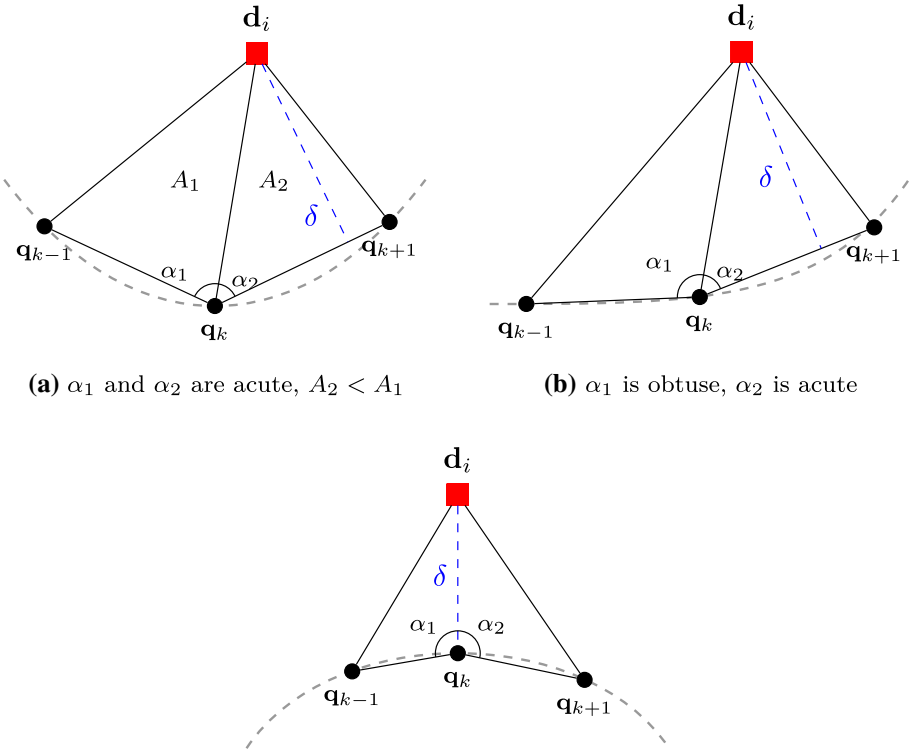
procedure DETERMINEDISCREPANCY($\{q\}$, $\{d\}$)
  for all $d \in \{d\}$ **do**
    $\{i_d\} \leftarrow$ <curve segment point indexes for d > ▷ see Fig. 12
    $min \leftarrow +\infty$
    for all $i_d \in \{i_d\}$ **do** ▷ local search for closest point index
      if $||d - q_{i_d}|| < min$ **then**
        $min \leftarrow ||d - q_{i_d}||$
        $k \leftarrow i_d$
      end if
    end for
    $s_1 \leftarrow \frac{||d-q_{k-1}||+||d-q_k||+||q_k-q_{k-1}||}{2}$
    $A_1 \leftarrow \sqrt{s_1(s_1 - ||d - q_{k-1}||)(s_1 - ||d - q_k||)(s_1 - ||q_k - q_{k-1}||)}$ ▷ 1^{st} triangle area
    $c_{\alpha_1} \leftarrow (d - q_k) \cdot (q_{k-1} - q_k)$ ▷ cosine of α_1 (see Fig. 13)
    $s_2 \leftarrow \frac{||d-q_k||+||q_{k+1}-q_k||+||d-q_{k+1}||}{2}$
    $A_2 \leftarrow \sqrt{s_2(s_2 - ||d - q_k||)(s_2 - ||q_{k+1} - q_k||)(s_2 - ||d - q_{k+1}||)}$ ▷ 2^{nd} triangle area
    $c_{\alpha_2} \leftarrow (d - q_k) \cdot (q_{k+1} - q_k)$ ▷ cosine of α_2 (see Fig. 13)
    if $c_{\alpha_1} < 0$ and $c_{\alpha_2} < 0$ **then** ▷ both angles are obtuse
      $\delta \leftarrow ||d_k - q_k||$
    else if $A_1 > A_2$ and $c_{\alpha_1} < 0$ **then**
      $\delta \leftarrow \frac{2A_1}{||q_k - q_{k-1}||}$ ▷ height of triangle 1
    else
      $\delta \leftarrow \frac{2A_2}{||q_{k+1} - q_k||}$ ▷ height of triangle 2
    end if
    <Add δ to $\{\delta\}$ >
  end for
  return $\{\delta\}$
end procedure

7 SA-based piecewise Bézier approximation curve

The SA algorithm to solve the piecewise Bézier approximation curve, outlined in Sect. 4 and in Fig. 9, is detailed in this section. The main loop of the SA consists of basically three steps: modification of one parameter of the solution (as required by the crystallization heuristic), computation of the new cost function, and the decision of accepting or rejecting the next candidate. These steps are then adopted for the two proposed algorithms: ANSA- and AMOSA-based approximation curve solutions.

7.1 Solution parameter update

The update of parameters is different for continuous parameters and for discrete parameters. For each continuous parameter, ANSA uses a crystallization factor that reduces the probability of large control point displacements with the decrease in temperature (see Sect. 3.1.1). In the case of AMOSA, as there is no adaptive neighborhood, the modification process is akin to adopting a fixed crystallization factor of value 1.



(a) α_1 and α_2 are acute, $A_2 < A_1$

(b) α_1 is obtuse, α_2 is acute

(c) α_1 and α_2 are obtuse

Fig. 13 For a given point $\boldsymbol{d}_i$, the nearest point on the curve is determined. **a** Two acute triangles are determined using the previous $\mathbf{q}_{k-1}$ and next $\mathbf{q}_{k+1}$ points of the sampled points. **b** Solution whereby one triangle is acute and the other is obtuse. **c** solution whereby the two triangles are obtuse

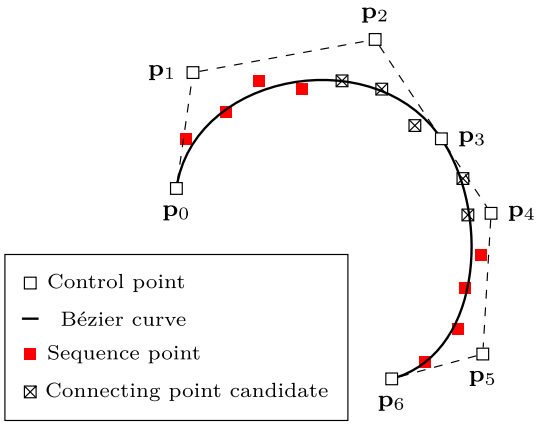


Fig. 14 Position limit of new candidates for connecting points. The crossed out squares are the possible positions at which connecting point can be modified

The discrete parameter in this problem is very particular, as small updates on the connecting points lead to large changes in the curve. Thus, one simple method is created to update discrete parameters: The point index is either incremented by 1 or decremented by 1, with equal probabilities. Moreover, to prevent the creation of very short segments, a minimum distance to the next connecting point is established. This restriction is shown in Fig. 14, where connecting point p_3 is shown at its initial position, which can be modified to any position represented by the crossed out squares.

7.2 Objective function

Finally, after determining the individual discrepancies and the curve length, the cost function can be determined (see the flowchart in Fig. 9). However, larger curves often allow for better approximation, and thus, the minimization conflicts with the main objective of discrepancy reduction. This problem is approached herein through two proposals. One consists in attributing weights to each objective (curve length and curve discrepancy), and the other adopts a multi-objective optimization approach.

The single-objective function contemplates both costs: the discrepancy from Eq. (8) and the curve length. Both are obtained by discretizing the curve, generating a sequence of points $\{q\}$. Assuming δ_k as the discrepancy for a sequence point d_k, the objective function, which simultaneously minimizes discrepancy and length, is given by

$$f(\{\mathbf{q}\},\{\delta\}) = W \cdot \sum_{k=1}^{m} \delta_k + (1-W) \cdot \sum_{k=1}^{l} ||\mathbf{q}_k - \mathbf{q}_{k-1}|| \tag{28}$$

where W is a weight that combines the influence of each objective function in the combined objective function and l is the number of points in which the curve is discretized.

For AMOSA, the objective functions to be minimized are

$$f_1(\{\delta\}) = \sum_{k=1}^{m} \delta_k \tag{29}$$

$$f_2(\{\mathbf{q}\}) = \sum_{k=1}^{l} ||\mathbf{q}_k - \mathbf{q}_{k-1}||. \tag{30}$$

7.3 SA optimization

Using the parameter update and cost function described in this section, two SA optimization approaches are proposed for the approximation curve. The first is ANSA, which adopts the single-objective function from Eq. (28). Algorithm 3 details this procedure. At each iteration, a single parameter is chosen randomly and is modified as required by the crystallization heuristic. Continuous parameters are updated using the adaptive neighborhood approach described in Sect. 3.1.1. If it is a connecting point discrete parameter, it is either increased or decreased, with equal probabilities. The objective function is then evaluated and the crystallization factor, which influences the continuous parameter updates, is refreshed. When the local stop condition is satisfied, the temperature is updated, impacting the domain exploration for the next inner loop. This process is repeated until convergence is achieved.

Algorithm 3 Adaptive neighborhood SA

$\boldsymbol{x}_{cur} \leftarrow$ <Initial random solution>
while <Global stop condition not satisfied> **do**
  $T \leftarrow T * \alpha$
  while <Local stop condition not satisfied> **do**
    $i \leftarrow rnd(0,1) * n$
    if <Parameter i is continuous> **then**
      $\boldsymbol{x}' \leftarrow \boldsymbol{x} + \boldsymbol{e_i} \cdot \Delta r_i \cdot \frac{1}{c_i} \sum_1^{c_i} rnd(-1,+1)$ ▷ move control point
    else
      if $rnd(0,1) < 0.5$ **then** $\boldsymbol{x}' \leftarrow \boldsymbol{x} + \boldsymbol{e_i}$ ▷ move connecting point forward
      else $\boldsymbol{x}' \leftarrow \boldsymbol{x} - \boldsymbol{e_i}$ ▷ move connecting point backward
      end if
    end if
    $\{q\}' \leftarrow$ DiscretizeCurve($\boldsymbol{x}'$) ▷ see Algorithm 1
    $\{\delta\}' \leftarrow$ DetermineDiscrepancy($\{q\}', \{d\}$) ▷ see Algorithm 2
    $\Delta E = f(\{q\}', \{\delta\}') - f(\{q\}, \{\delta\})$
    if $\Delta E < 0$ **then**
      $\boldsymbol{x} \leftarrow \boldsymbol{x}'$; $\{q\} \leftarrow \{q\}'$; $\{\delta\} \leftarrow \{\delta\}'$ ▷ update solution
      $c_i \leftarrow$ <positive feedback>
    else
      if $random(0,1) < e^{-\Delta E/k_b T_k}$ **then**
        $\boldsymbol{x} \leftarrow \boldsymbol{x}'$; $\{q\} \leftarrow \{q\}'$; $\{\delta\} \leftarrow \{\delta\}'$ ▷ update solution
        $c_i \leftarrow$ <positive feedback>
      else $c_i \leftarrow$ <negative feedback>
      end if
    end if
  end while
end while

The second proposal is to adopt two separate cost functions (Eqs. (29), (30)) simultaneously. The AMOSA was chosen to determine a set of solutions in the Pareto front, as described by Algorithm 4. The main difference from the ANSA algorithm is the management of the solution archive. The cost function evaluation must happen for the next candidate. Using the dominance concept described in Sect. 3.2, the current solution may be updated, and the next candidate added to the archive or some solutions may be removed from the archive. The other change from the ANSA algorithm is the update of continuous variables, which does not account for crystallization, and consequently, there is not feedback for continuous variables.

Algorithm 4 AMOSA

$x \leftarrow$ <Random solution from Archive>
while <Global stop condition not satisfied> **do**
  $T \leftarrow T * \alpha$
  while <Local stop condition not satisfied> **do**
    $x' \leftarrow <x$ with random modification (no crystallization)> ▷ see lines xxx from eq. ??
    if $< x$ dominates $x' >$ **then** ▷ case 1
      $\Delta dom_{avg} \leftarrow \frac{(\sum_{i=1}^{k} \Delta dom_{i,x'}) + \Delta dom_{x,x'}}{k+1}$ ▷ k is the size of the Archive
      if $rnd(0,1) < \frac{1}{1+exp(\Delta dom_{avg} * T)}$ **then** $x \leftarrow x'$
      end if
    else if $< x$ and x' are non-dominating to each other> **then** ▷ case 2
      if $< x'$ is dominated by $k (k \geq 1)$ solutions of the Archive) **then** ▷ case 2a
        $\Delta dom_{avg} \leftarrow \frac{(\sum_{i=1}^{k}) \Delta dom_{i,x'}}{k}$
        if $rnd(0,1) < \frac{1}{1+exp(\Delta dom_{avg} * T)}$ **then** $x \leftarrow x'$
        end if
      else if $< x'$ do not dominate any solution of the Archive> **then** ▷ case 2b
        $x \leftarrow x'$; <Add x' to the Archive>
      else if $< x'$ k, $(k \geq 1)$ solutions of the Archive> **then** ▷ case 2c
        $x \leftarrow x'$; <Add x' to the Archive>
        <Remove from the Archive k solutions dominated by x' >
      end if
    else if $< x'$ dominates $x >$ **then** ▷ case 3
      if $< x'$ is dominates? by $k (k \geq 1)$ solutions of the Archive> **then** ▷ case 3a
        $\Delta dom_{min} \leftarrow$ <Min. dominance difference between x' and the k solutions>
        if $(random(0,1) < \frac{1}{1+exp(-\Delta dom_{min})})$ **then**
          <Update with solution of the Archive corresponding to Δdom_{min} >
        else$< x \leftarrow x' >$
        end if
      else if $< x'$ do not dominate any solution of the Archive> **then** ▷ case 3b
        $x \leftarrow x'$; <Add x' to the Archive>
      else if $< x'$ dominates k, $(k \geq)$ solutions of the Archive> **then** ▷ case 3c
        $x \leftarrow x'$; <Add x' to the Archive>
        <Remove from the Archive k solutions dominated by x' >
      end if
    end if
  end while
end while

8 Results

The two SA-based approximation curve algorithms were implemented for computation experiments. For these tests, a series of parameters were defined. For the discretization step, δ was set to 100, creating 100 sampled points per segment. In the case of ANSA, the global stop condition was set to

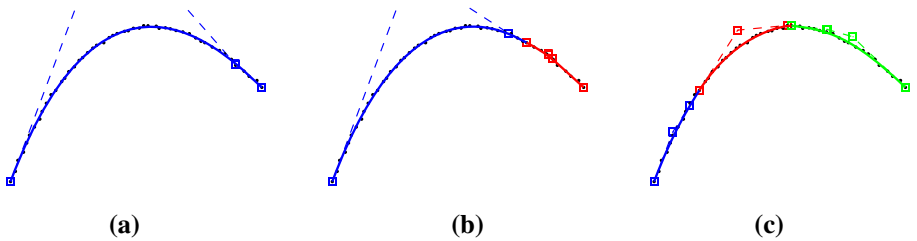


Fig. 15 Parabola example with: **a** 1 curve segment; **b** 2 curve segments; **c** 3 curve segments

Table 1 Parabola average results regarding 100 tests using Eq. (28) as the cost function

W	1 Segment			2 Segments			3 Segments		
	Cost	Disc	Len	Cost	Disc	Len	Cost	Disc	Len
0.10	50.20	2.48	55.50	50.22	2.43	55.53	50.22	2.28	47.46
0.20	44.79	1.06	55.72	44.82	1.15	55.74	44.81	1.08	55.74
0.30	39.30	0.83	55.79	39.35	0.97	55.81	39.34	0.90	55.82
0.40	33.80	0.76	55.83	33.85	0.87	55.84	33.84	0.81	55.86
0.50	28.29	0.72	55.86	28.35	0.82	55.88	28.33	0.79	55.88
0.60	22.78	0.71	55.87	**22.86**	**0.83**	**55.90**	**22.82**	**0.76**	**55.91**
0.70	**17.26**	**0.71**	**55.88**	17.38	0.86	55.94	17.33	0.79	55.92
0.80	11.74	0.71	55.90	11.83	0.79	55.95	11.81	0.78	55.97
0.90	6.23	0.71	55.90	6.31	0.78	56.07	6.30	0.75	56.27
0.95	3.47	0.71	55.91	3.50	0.71	56.52	3.57	0.72	57.74
1.00	0.71	0.71	56.12	0.63	0.63	125.20	0.62	0.62	365.15

Disc and *Len* are the discrepancy and length of the curve, respectively

a minimum temperature of 0.0001 or the acceptance of 1% of the solutions or less. The local temperature loop was limited to 500 accepted solutions or 1000 iterations. The initial temperature was set to 100, and the cooling factor α was chosen using the adaptive cooling schedule from Eq. (16) with β equal to 1 and α at least 0.8. The negative feedback consisted of adding one to the crystallization factor; however, this increase is limited by 20 to avoid numerical errors. Considering the positive feedback, the crystallization factor was divided by five. If, due to a parameter update, the distance between two connecting points (or endpoints) was less than 4, the modification was ignored. The local and global stop conditions for the AMOSA algorithm were: a limit of 500 iteration and a minimum temperature of 0.05, respectively. The initial temperature was 500 and α equal to 0.99. There was no limit size to the archive.

The algorithm was tested with three different examples, and each example was created with a B-spline curve with the desired shape. First, the simplest one is a parabola, the second is a curve with one inflection, and the third example is a curve with self-intersection. The examples are created with the desired shape and sampled with 50 points. A noise of at most 3% was added to each sampled points, i.e., a random noise at maximum 3% of the bounding box was added to coordinates x and y of each point. Each example is tested with several values of W, and several numbers of segments.

8.1 Parabola

In this example, the sequence generated is approximated with 1, 2 and 3 cubic Bézier curve segments, as shown in Fig. 15a–c, respectively. Each test is repeated 100 times, and the results are shown in Table 1, which displays the average values for the discrepancy and length, as well as the cost.

The average curve discrepancy is observed to be higher for $W = 0.10$ and $W = 0.20$, although it remains almost constant for all other values of W. The curve length is significantly higher with $W = 1.00$, except when a single segment is used, where it is only slightly higher. This is expected, as W controls the weight of each part of the cost function and, as W is decreased, the curve discrepancy tends to increase and the curve length to decrease.

Figure 16 shows the comparison between AMOSA (blue points) and two cases of ANSA (red and green points) for the curves with 1, 2 and 3 segments. For each case, the red points correspond to the results of 100 tests with $W = 1.00$ and the green points are the results of 100 tests with an intermediary value of W, which corresponds to a low curve length and discrepancy standard deviations. These plots show that the exclusive minimization of the curve discrepancy is not satisfactory, once it creates several curves with low variation of the curve discrepancy and high variation of the curve length. The use of a weighed sum of the curve discrepancy and curve length minimized this effect. The ANSA found some

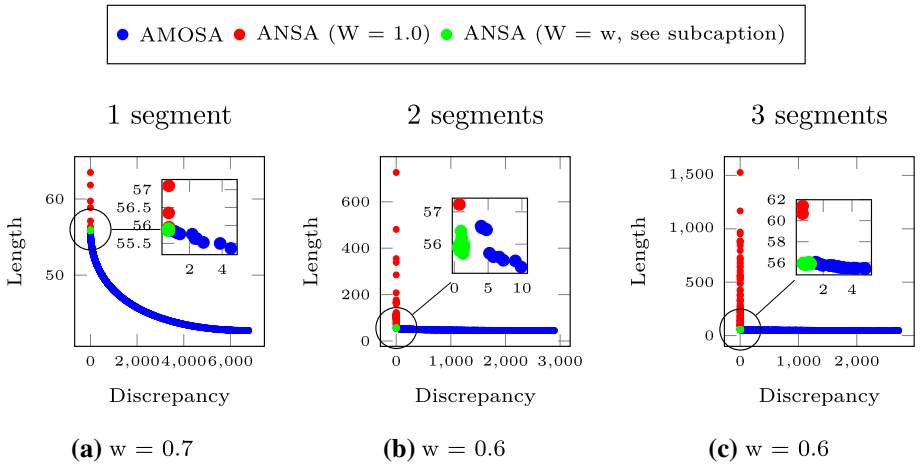


Fig. 16 Comparison between AMOSA and two cases of ANSA in the parabola example. For 2 and 3 curve segments, ANSA determined solutions that dominates the ones obtained by AMOSA

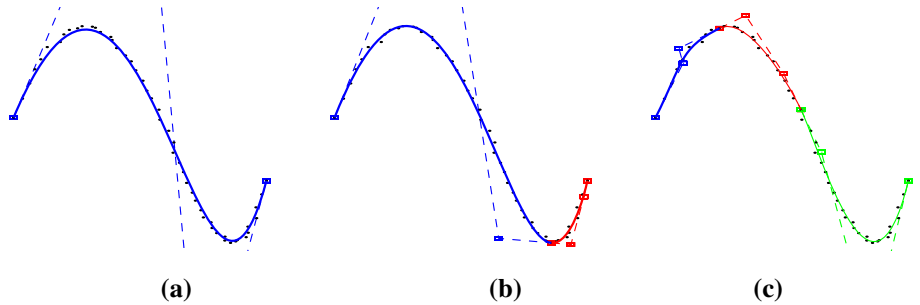


Fig. 17 Inflection example with: **a** 1 curve segment; **b** 2 curve segments; **c** 3 curve segments

Table 2 Inflection curve average results regarding 100 tests using Eq. (28) as the cost function

W	1 Segment			2 Segments			3 Segments		
	Cost	Disc	Len	Cost	Disc	Len	Cost	Disc	Len
0.10	111.46	29.31	120.59	107.40	47.76	114.03	104.27	44.74	110.88
0.20	100.94	12.40	123.07	100.49	12.86	122.40	99.68	10.90	121.88
0.30	89.78	11.33	123.40	89.28	7.69	124.25	89.08	7.05	124.24
0.40	78.55	11.07	123.54	77.76	6.93	124.99	77.50	5.75	125.33
0.50	67.30	10.98	123.62	65.68	5.73	125.63	65.54	5.20	125.89
0.60	56.02	10.88	123.73	53.73	5.64	125.86	**53.39**	**4.87**	**126.16**
0.70	44.73	10.86	123.76	**41.35**	**4.94**	**126.31**	41.40	4.90	126.59
0.80	**33.44**	**10.85**	**123.78**	29.32	5.01	126.57	29.10	4.64	126.92
0.90	22.15	10.85	123.81	16.95	4.71	127.14	17.04	4.68	128.25
0.95	16.50	10.85	123.81	10.72	4.59	127.27	10.76	4.30	133.48
1.00	10.85	10.85	123.82	4.30	4.30	288.66	4.85	4.85	634.76

Disc and *Len* are the discrepancy and length of the curve, respectively

solutions that dominated the solutions obtained by AMOSA, showing that ANSA found better solutions than AMOSA. This was expected, once the method to create new candidates in ANSA is better than the one in the AMOSA. The crystallization heuristic performs a balance between exploration and refinement considering the current temperature. The AMOSA has an exploratory behavior.

8.2 Inflection curve

In this example, the inflection curve sequence is approximated with 1, 2 and 3 cubic Bézier curve segments, as shown in Fig. 17a–c, respectively. The average results from 100 executions are shown in Table 2.

As with the parabola case, the average curve discrepancy is verified to be very high with $W = 0.10$, significantly lower with $W = 0.20$ and, for all the other values of W, its value steadily decreases. Unlikely the parabola example, however, this example presents a slightly lower curve length with $W =$

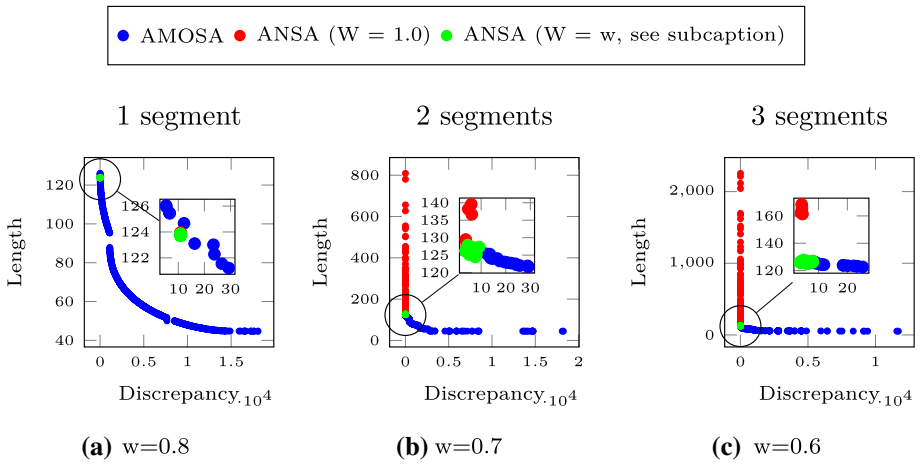


Fig. 18 Comparison between AMOSA and two cases of ANSA in the inflection example. For 2 and 3 curve segments, ANSA determined solutions that dominates the ones obtained by AMOSA

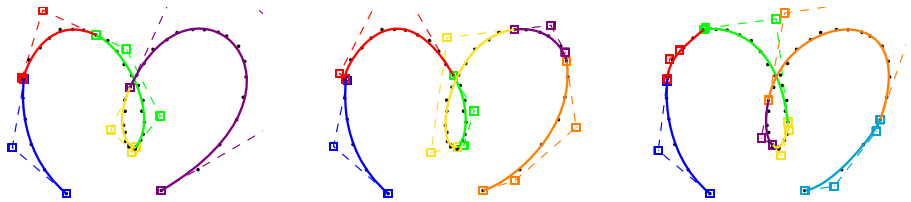


Fig. 19 Self-intersection example with: **a** 5 curve segments; **b** 6 curve segments; **c** 7 curve segments

Table 3 Self-intersection curve average results regarding 100 tests using Eq. (28) as the cost function

W	5 Segments			6 Segments			7 Segments		
	Cost	Disc	Len	Cost	Disc	Len	Cost	Disc	Len
0.10	80.24	43.07	84.37	80.95	37.04	85.83	81.07	31.66	86.56
0.20	77.83	27.39	90.44	77.92	22.88	91.68	79.69	18.08	95.09
0.30	74.61	16.15	99.66	76.10	10.53	104.20	77.05	7.08	107.04
0.40	67.18	6.37	107.73	67.53	4.20	109.76	67.75	2.79	111.06
0.50	57.26	3.14	111.39	57.14	2.06	112.21	56.97	1.53	112.41
0.60	46.60	2.65	112.53	46.03	1.67	112.58	45.88	1.21	112.90
0.70	**35.12**	**1.69**	**113.12**	**34.88**	**1.32**	**113.18**	**34.66**	**1.00**	**113.21**
0.80	24.30	1.98	113.56	23.71	1.26	113.50	23.43	0.91	113.52
0.90	12.78	1.50	114.30	12.50	1.21	114.11	12.28	0.97	114.04
0.95	7.68	1.96	116.32	7.29	1.53	116.89	6.85	1.11	115.97
1.00	4.19	4.19	526.23	3.74	3.74	916.83	3.03	3.03	1,164.58

Disc and *Len* are the discrepancy and length of the curve, respectively

0.10, while slowly increasing with the increasing values of W. When using more than 1 segment, the curve length greatly increases for W equal to 1, showing again that minimizing just the curve discrepancy turns the algorithm unstable. In this example, the algorithm prioritized the minimization of the curve length in favor of the curve discrepancy for the lowest value of W.

Figure 18 shows the comparison of results between the AMOSA and ANSA variants. In the example with 1 curve segment, the ANSA results are close to some AMOSA results. However, there are differences between the ANSA and AMOSA results in the approximations with 2 and 3 curve segments. The results have the same behavior of the parabola examples; the ANSA method for $W = 1.0$ is unstable, and ANSA determines better results than AMOSA.

8.3 Self-intersection curve

Figure 19 shows the self-intersection curve sequence approximated with 5, 6 and 7 segments. The results from 1000 repetitions are displayed in Table 3.

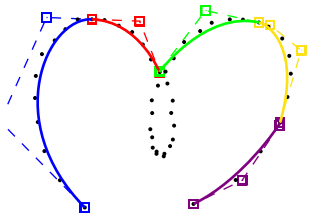


Fig. 20 The tradeoff between the minimization of the curve length or the curve discrepancy results in that several points are not considered in the approximation

Differently from the two previous examples, any change in W significantly affects the average values of the curve discrepancy and length. In the tests with $W = 0.10$, $W = 0.20$ and $W = 0.30$, the average values of the curve length are lower and the curve discrepancy is higher, showing that the algorithm minimizes the curve length instead of the curve discrepancy. This is verified in the result from Fig. 20, in which the approximation curve does not approximate all the points. With the increase in W, the average value of the curve discrepancy decreases. However, differently from the other two examples, with $W = 1.00$, the average curve discrepancy and length greatly increase.

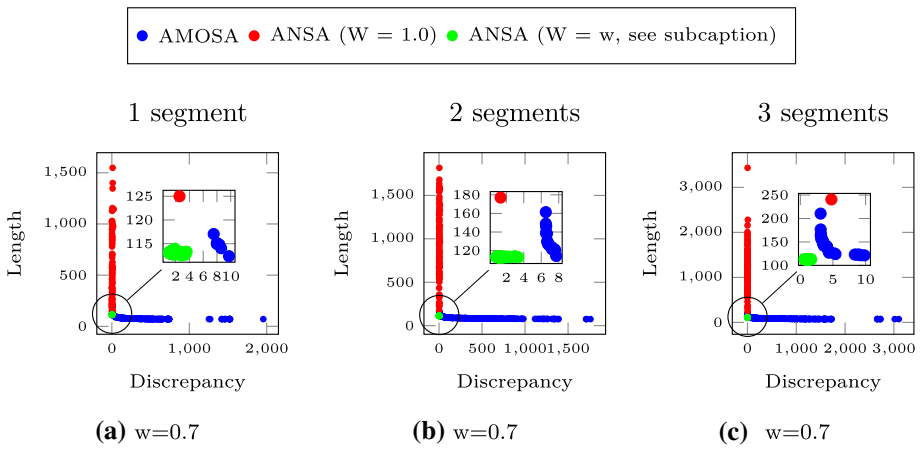
Figure 21 compares the archive solutions from AMOSA with the results from multiple tests of the ANSA algorithm. As in the previous examples, the approximation using ANSA yielded better results than AMOSA. However, the use of $W = 1.00$ turns the algorithm so unstable that the average value of the curve discrepancy is far higher than the case with $W = 0.70$ and the curve length largely increases. The comparison between the examples with $W = 1.00$, at which only the curve discrepancy is minimized, shows the higher the number of curve segments, the more unstable the method. This is expected, once the higher the number of curve segments, the higher the number of control points; and longer curves can be created with more control points.

9 Conclusion

This work used a SA with adaptive neighborhood to solve the curve fitting problem, whereby each continuous parameter is evaluated at each iteration increasing the number of accepted solutions. The use of segments of cubic Bézier curve in this problem takes advantage of the local influence of each control point, in which each of the latter modifies at most 2 curve segments. The use of the curve length as a regularization prevents the determination of long curves making the algorithm stable, and the tuning of the weight of this regularization varies for each example. However, the more complex the example, i.e., the example with a high number of curve segments, the more important this tune is, as observed in the three examples. In the parabola example, there is no best value for W. There is a wide range for W in which the algorithm is stable, as can be observed with the low values of $Std\ Disc$, $Std\ Len$ and $Std\ Cost$. In the inflection curve example, there is no best value for W yet; the range for W in which the algorithm is stable is a little narrow, but still wide. However, in the self-intersection curve example, there is a better value for W. In the comparison between the adaptive neighborhood SA and AMOSA, the adaptive neighborhood SA was observed to obtain better approximation curves than the AMOSA, due to the refinement in the parameters drawn in ANSA, which does not occurs for AMOSA.

As future work, it an automatic method to determine the value of W is proposed as well as the number of segments of cubic Bézier necessary to determine a smooth curve that approximates a sequence of points.

Acknowledgements Edson K. Ueda is supported by CNPq (Grant 140.518/2015-0). André K. Sato is supported by CAPES/PNPD and FUSP/Petrobras. Rogério Y. Takimoto is supported by CNPq (Grant 401.794/2014-8) and FUSP/Shell. Thiago C. Martins is partially supported by CNPq (Grant 306.415/2012-7). Marcos S. G. Tsuzuki is partially supported by CNPq (Grant 311.195/2019-9). The research was supported by CAPES (Grant 817757/38860).



(a) w=0.7 **(b)** w=0.7 **(c)** w=0.7

Fig. 21 Comparison between AMOSA and two cases of ANSA in the self-intersection example. The ANSA determined solutions that dominates the ones obtained by AMOSA in all cases

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

Aarts E, Korst J (1989) Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing. Wiley, New York

Abualigah LMQ (2019) Feature selection and enhanced krill herd algorithm for text document clustering, vol 816. Springer, Berlin

Abualigah L, Hanandeh E (2015) Applying genetic algorithms to information retrieval using vector space model. Int J Comput Sci Eng Appl 5:19–28

Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J Supercomput 73(11):4773–4795

Abualigah LM, Khader AT, Hanandeh ES (2018) Hybrid clustering analysis using improved krill herd algorithm. Appl Intell 48(11):4047–4071

Abualigah L, Khader AT, Hanandeh E (2018) A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis. Eng Appl Artif Intell 73:111–125

Adi D, Shamsuddin S, Hashim SZM (2010) NURBS curve approximation using particle swarm optimization. In: 2010 seventh international conference on computer graphics, imaging and visualization (CGIV), pp 73–79

Bandyopadhyay S, Saha S, Maulik U, Deb K (2008) A simulated annealing-based multiobjective optimization algorithm: AMOSA. IEEE Trans Evol Comput 12(3):269–283

Bohachevsky IO, Johnson ME, Stein ML (1986) Generalized simulated annealing for function optimization. Technometrics 28(3):209–217

Cohn H, Fielding M (1999) Simulated annealing: searching for an optimal temperature schedule. SIAM J Optim 9:779–802

Corana A, Marchesi M, Martini C, Ridella S (1987) Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. ACM Trans Math Softw 13:262–280

Dong J, Xu M, Zhang X, Gao Y, Pan Y (2008) The creation process of Chinese calligraphy and emulation of imagery thinking. IEEE Intell Syst 23(6):56–62

Gálvez A, Iglesias A (2013) A new iterative mutually coupled hybrid GA–PSO approach for curve fitting in manufacturing. Appl Soft Comput 13:1491–1504

Gálvez A, Iglesias A, Avila A, Otero C, Arias R, Manchado C (2015) Elitist clonal selection algorithm for optimal choice of free knots in B-spline data fitting. Appl Soft Comput 26:90–106

Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Mach Intell 6(6):721–741

Gofuku S, Tamura S, Maekawa T (2009) Point-tangent/point-normal B-spline curve interpolation by geometric algorithms. Comput Aided Des 41:412–422

Hasegawa AY, Rosso Jr RSU, Tsuzuki MSG (2013) Bézier curve fitting with a parallel differential evolution algorithm. In: 11th IFAC workshop on intelligent manufacturing systems, pp 233–238

Itoh K, Ohno Y (1993) A curve fitting algorithm for character fonts. Electron Publ 6:195–205

Jupp DLB (1978) Approximation to data by splines with free knots. SIAM J Numer Anal 15(2):328–343

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

Lin F, Shen L-Y, Yuan C-M, Mi Z (2019) Certified space curve fitting and trajectory planning for CNC machining with cubic B-splines. Comput Aided Des 106:13–29

Loucera C, Galvez A, Iglesias A (2014) Simulated annealing algorithm for Bezier curve approximation. In: International conference on cyberworlds, pp 182–189

Lundy M, Mees A (1986) Convergence of an annealing algorithm. Math Program 34(1):111–124

Maekawa T, Matsumoto Y, Namik K (2007) Interpolation by geometric algorithm. Comput Aided Des 39:313–323

Martins TC, Tsuzuki MSG (2008) Rotational placement of irregular polygons over containers with fixed dimensions using simulated annealing and no-fit polygons. J Braz Soc Mech Sci Eng 30(3):205–212

Martins TC, Tsuzuki MSG (2010) Placement over containers with fixed dimensions solved with adaptive neighborhood simulated annealing. Bull Pol Acad Sci Tech Sci 57(3):273–280

Martins TC, Sato AK, Tsuzuki MSG (2012) Adaptive neighborhood heuristics for simulated annealing over continuous variables. In: Tsuzuki MSG (ed) Simulated annealing–advances, applications and hybridization. Intech, pp 3–20

Martins TC, Tsuzuki MSG, de Camargo EDLB, Lima RG, de Moura FS, Amato MBP (2016) Interval simulated annealing applied to electrical impedance tomography image reconstruction with fast objective function evaluation. Comput Math Appl 72(5):1230–1243

Martins TC, Sato AK, Moura FS, Camargo EDLB, Silva OL, Santos TBR, Zhao Z, Möeller K, Amato MBP, Mueller JL, Lima RG, Tsuzuki MSG (2019) A review of electrical impedance tomography in lung applications: theory and algorithms for absolute images. Annu Rev Control 48:442–471

Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21:1087

Okaniwa S, Nasri A, Lin H, Abbas A, Kineri Y, Maekawa T (2012) Uniform B-spline curve interpolation with prescribed tangent and curvature vectors. IEEE Trans Vis Comput Graph 18(9):1474–1487

Pandunata P, Shamsuddin S (2010) Differential evolution optimization for Bezier curve fitting. In: Seventh international conference on computer graphics, imaging and visualization (CGIV), pp 68–72

Patrikalakis NM, Maekawa T (2009) Shape interrogation for computer aided design and manufacturing, 1st edn. Springer, Berlin

Piegl L, Tiller W (1997) The NURBS book, 2nd edn. Springer, Berlin

Prautzsch H, Boehm W, Paluszny M (2002) Bezier and B-spline techniques. Springer, Berlin

Schwetlick H, Schütze T (1995) Least squares approximation by splines with free knots. BIT Numer Math 35(3):361–384

Sobrinho E, Sanomya R, Ueda R, Tiba H, Tsuzuki MSG, Adamowski JC, Silva ECN, Carbonari RC, Buiochi F (2009) Development of a methodology for evaluation of a structural damage in turbine blades from hydropower generators. In: Proceedings of the 20th international congress of mechanical engineering

Song M, Chen D (2018) An improved knowledge-informed NSGA-II for multi-objective land allocation (MOLA). Geo-spat Inf Sci 21(4):273–287

Tavares RS, Martins TC, Tsuzuki MSG (2011) Simulated annealing with adaptive neighborhood: a case study in off-line robot path planning. Expert Syst Appl 38(4):2951–2965

Várady T, Martin RR, Cox J (1997) Reverse engineering of geometric models—an introduction. Comput Aided Des 29(4):255–268 reverse Engineering of Geometric Models

Yang H-M, Lu J-J, Lee H-J (2001) A Bezier curve-based approach to shape description for Chinese calligraphy characters. In: Proceedings of sixth international conference on document analysis and recognition, pp 276–280

Zhao L, Jiang J, Song C, Bao L, Gao J (2013) Parameter optimization for Bezier curve fitting based on genetic algorithm. In: Advances in swarm intelligence, vol 7928 of Lecture notes in computer science. Springer, Berlin, pp 451–458