Using Simulated Annealing for knot placement for cubic spline approximation

Olga Valenzuela
University of Granada, Spain
Department of Applied Mathematics
olgavc@ugr.es

Miguel Pasadas
University of Granada, Spain
Department of Applied Mathematics
mpasadas@ugr.es

Abstract: In this paper, a new methodology is developed for knots placement for cubic spline approximation, using Simulated Annealing. It is not necessary to convert the problem into a discrete combinatorial optimization problem, as presented in other paradigms inspired in genetic algorithm or artificial immune systems, and therefore, removing the constrain of combinatorial optimization problem in the proposed methodology, because the real value of the location of the knots is directly optimized. The accuracy, computationally efficient and robustness of the algorithm presented is compared by different experimental result, with other approaches presented in the bibliography.


1 Introduction

Function approximation is a fundamental problem in almost every scientific field. Given a set of multiple input single-output (MISO) data, with input variables \( X = [x_0, x_1, \ldots, x_n] \) and output variable \( Y = y \), the main objective of function approximation is obtaining a model to approximate the dependent variable \( Y \), given the input variable \( X \), being \( X \) and \( Y \) real numbers.

In the literature, there exist different methodologies for approximating or fitting curves to data. One of frequently approach is based on polynomial interpolation theorem of Lagrange [1], [2]. Other important approach, presented in [3], [4] based on Hermite polynomial. These interpolation methodologies are not suitable when the number of data is large. In fact, for problems with a large data base to be interpolated, the large size of the approximation space becomes the resolution of the interpolation problem to be complex. In general, traditional methods based on polynomial interpolation, have the disadvantage that, when the number of data is large, the interpolant polynomial usually present significant fluctuations in the intervals where no data are available. The classic solution to this problem is considered as interpolating, spline functions, that is actually used in a large variety of applications. Spline interpolation is a form of interpolation where the interpolant is a special type of piecewise polynomial called a spline. Spline interpolation is preferred over polynomial interpolation because the interpolation error can be made small even when using low degree polynomials for the spline. Thus, spline interpolation avoids the problem of Runge’s phenomenon which occurs when using high degree polynomials. The main advantages of spline interpolation are its stability and calculation simplicity. In principle, there are few drawbacks of using spline functions for interpolation, except for the objective to achieve a higher regular order that would require to increase the spline function order.

It is well-known that the placement of the knots in spline approximation has an important and considerable effect on the behavior of the final approximation [5]. However, as pointed out in [6], although spline for approximation is well understood, the knot placement problem has not been dealt with adequately. In theory, knot placement is a multivariate and multimodal nonlinear optimization problem. Especially for problems when dense and noisy data points should be approximated. For this kind of problems, smoothing and resampling are usually employed to pre-process the data in order to make easy the placement of knot and improve the performance of the approximation [7], [8]. But the data smoothing and resampling operations highly depend on the intervention of the designers. In [6] a new knot placement algorithm has been presented for B-spline curve approximation to dense and noisy data points. In the proposed methodology, the discrete curvature of the data points is smoothed using lowpass digital filter to expose the curvature characteristics of the underlying curve of the data. Then knots are automatically placed to make the curve, which passes the corresponding points, be of locally small deflection.

In [9], the authors presented a Bayesian model for
automatically determining knot placement in spline modeling. The random variables of the proposed methodology are the number of knots and their placement, which were estimated via Reversible Jump Markov Chain Monte Carlo sampler. This technique has the ability to maximize the joint posterior distribution of the number of knots and their locations, without becoming stranded on local maxima. The authors presented results which verify the effectiveness of the proposed technique, in accurately fitting a non-uniform, cubic spline to data, whilst maintaining a relatively small number of knots.

In order to obtain a good B-spline curve model from large data, in [10] the knots are respected as variables, and its placement is optimized by a methodology inspired in artificial immune system. A curve is then modeled as a continuous, nonlinear and multivariate optimization problem with many local optima. For this reason it is very difficult to reach a global optimum. In the proposed methodology, the author convert the original problem into a discrete combinatorial optimization problem like in other approaches inspired in Genetic Algorithm in [11]. The authors presented a new method that solves the converted problem by artificial immune systems, using the candidates of the locations of knots as antibodies. The proposed method determines the appropriate location of knots automatically and simultaneously. Furthermore, the presented methodology did not need any subjective parameter or good population of initial location of knots for a good iterative search.

The main drawback of the method previously presented is its computationally inefficient, due the great complexity of the algorithm to be resolved. In this paper, a fast, computationally efficient and robust method is presented for placement of the knots, based on Simulated Annealing.

The rest of this paper is organized as follows. After this Introduction, Section 2 briefly presents some basic concepts about the optimization method used in this paper for knot placement of cubic spline approximation: the Simulated Annealing. In Section 3, the proposed methodology for approximation with cubic splines, using Simulated Annealing as optimization paradigm for placement the knots, is presented in detail. Several experiments are carried out in Section 4 to analyze the behavior of the approximation of the proposed methodology. Comparison with others approaches are also presented. The conclusions are discussed in Section 5.

2 Simulated Annealing

A combinatorial optimization problem is a minimization or maximization problem which involves finding the optimal, or globally optimal configuration of discrete variables, with respect to some function of the variables. Many combinatorial optimization problems are very difficult and are NP-hard, requiring the characterization of the search space and the cost/evaluation or objective function. The knot placement for cubic spline interpolation can be considered as an example of combinatorial optimization problem, where the values of the position of the knots have a great influence on the behavior of the cubic spline interpolation.

The search space $\mathbb{S}$ of potential solutions is a finite or countably infinite set (in our particular case, the placement of the different knots selected for interpolation), and the objective function $f : \mathbb{S} \rightarrow \mathbb{R}$ obtain, for each multidimensional point of the search space (possible solutions of the problem), a fitness measure of the accuracy of the solution. For the minimization situation, the desired optimal solution $x_{optimal} \in \mathbb{S}$, fulfill the following requirement: $f(x_{optimal}) \leq f(x)$, for all $x \in \mathbb{S}$. Simulated annealing can be used as a powerful tool for the minimization problem then can be declared as: $\minimize_{x \in \mathbb{S}} f(x)$.

Simulated annealing is a single-objective optimization technique, that comes from an analogy between combinatorial optimization and the physical process of annealing.

The physical process of annealing is inspired in the cooling of a metal sufficiently slowly so that it adopts a low-energy, crystalline state. When the temperature of the metal is high, the particles within the metal are able to move around, changing the structure of the metal, without restraint.

As the temperature is lowered, or cooled slowly so that thermal equilibrium is achieved at each temperature, the particles are limited in the movements they can make as many movements have a high energy cost and are increasingly limited to only those configurations with lower energy than the previous state. Thermal equilibrium can be characterized by the Boltzmann distribution:

$$P_T\{X = x\} = \frac{e^{-E_x/K_B T}}{\sum_{\text{all states } i} e^{-E_i/K_B T}} \tag{1}$$

where $X$ is a random variable indicating the current state, $E_x$ is the energy of state $x$, $K_B$ is Boltzmann’s constant, and $T$ is temperature.

The configuration is typically a solution to the optimization, and at each iteration of the algorithm this
solution is agitated/modified in some manner to produce a novel solution. The quality of a solution is said to be the energy of the state, analogous with real-world annealing. The quality of both solutions is evaluated, using the objective function $f$, and a new state is selected from the two solutions. If the new solution is better than the previous solution (in minimization problem, $f(x_t) < f(x_{t-1})$), the new solution is selected as the state. Where the new solution is worst the existing solution, it may be accepted with a probability dependent upon both the current computational temperature and the magnitude of the difference in quality.

This simple algorithm is based on Monte Carlo techniques which was proposed by Metropolis et. al. [18] in 1953. Mathematically, the transition from state $x_{t-1}$ to state $x_t$ is then accepted with probability:

$$P_{\text{accept}}(x_{t-1}, x_t) = \begin{cases} 1 & \text{if } E_{x_{t-1}} - E_{x_t} > 0 \\ e^{-(E_{x_{t-1}} - E_{x_t})/k_B T} & \text{if } E_{x_{t-1}} - E_{x_t} \leq 0 \end{cases}$$

If accepted, $x_t$ becomes the current state and the procedure is repeated. This acceptance rule is known as the Metropolis criterion. For a particular combinatorial optimization problem let the solution $x$ correspond to the current state of the solid, the objective function correspond to the energy of the current state, and the factor $T$ can be considered as the temperature of the solid. The simulated annealing algorithm evolution is the iterative evaluation of the Metropolis algorithm for decreasing values of the artificial temperature factor $T$.

The statement that the acceptance probability is equal to 1 when the transition from state $x_t$ to state $x_{t-1}$, minimizes the corresponding associated energies, indicates that transitions which improve the cost or objective function are always accepted. On the other hand, unlike “greedy” algorithms which tolerate only decreases in objective functions, simulated annealing may allow cost increases, enabling the algorithm to escape from local minima.

### 3 Proposed Methodology

In this section, the knots placement algorithm is presented in detail. First, briefly, and introduction to Cubic spline spaces of class $C^2$ is presented, in order to define the notation required for optimization the knots placement of cubic spline.

#### 3.1 Cubic spline spaces of class $C^2$

A spline is a function, typically constructed using low order polynomial functions, joined at breakpoints with certain smoothness conditions. The breakpoints are defined in this context as knots. If $n$ is the order of the spline, in order to ensure the smoothness of the interpolation, typically $(n-2)$ continuity conditions should be fulfilled.

The order $n$ refers to the number of coefficients in the polynomial pieces ($n = 2$ therefore corresponds with linear splines, whereas cubic splines are of the order $n = 4$). The order also determines the smoothness of the resulting function approximation as splines fulfilling $(n-2)$ continuous derivatives. We start with a partition or knot sequence of $[a, b]$ in $m$ subintervals, i.e., an increasing sequence that is uniform or not uniform $\Delta_m = \{a = t_0 < t_1 < \ldots < t_m = b\}$, and we define the cubic spline of class $C^2$ on the partition $\Delta_m$ as every function

$$s : [a, b] \rightarrow \mathbb{R}$$

such that

i) $s \in C^2[a, b]$

ii) $s \mid_{[t_i, t_{i+1}]} \in P_3[t_i, t_{i+1}], \; i = 0, \ldots, m - 1,$

where $P_3[t_i, t_{i+1}]$ is the space of all the restrictions of the polynomial functions of a degree less than or equal to three in the interval $[t_i, t_{i+1}]$.

Given $t_{-3}, t_{-2}, t_{-1}, t_{m+1}, t_{m+2}, t_{m+3} \in \mathbb{R}$ such that,

$$t_{-3} \leq t_{-2} \leq t_{-1} \leq t_0 < \cdots < t_m \leq t_{m+1} \leq t_{m+2} \leq t_{m+3},$$

we define for each $t \in [a, b]$

$$B^k_i(t) = \begin{cases} 1, & t_{i-3} \leq x < t_{i-2}, \\ 0, & \text{otherwise} \end{cases}$$

$i = 0, \ldots, m + 5 - k$ and, $B^k_i(t), \; k = 1, 2, 3$, is defined from the recursive relation:

$$B^k_i(t) = \frac{t - t_{i-3}}{t_{i+k-3} - t_{i-3}} B^{k-1}_i(t) + \frac{t_{i+k-2} - t}{t_{i+k-2} - t_{i-2}} B^{k-1}_{i+1}(t), \; i = 0, \ldots, m + 5 - k.$$

These functions verify the following properties:

i) They are positive in the interior of their support,

$$B^k_i(t) \geq 0, \; \forall t \in [a, b].$$
ii) They form a partition of unity
\[ \sum_{i=0}^{m+5-k} B_i^k(t) = 1, \forall t \in [a, b]. \]

iii) \( \{B_0^k, \ldots, B_{m+5-k}^k\} \) are linearly independent for all \( k = 0, 1, 2, 3. \)

Meanwhile, if \( S_3(\Delta_m) \) represents the set of cubic spline functions of degree less than or equal to three and the class \( C^2 \), then \( \dim S_3(\Delta_m) = m + 3 \) and \( \{B_0^3, \ldots, B_{m+2}^3\} \) would be a basis of \( S_3(\Delta_m) \), called B-spline basis functions of the third order.

Some examples of spline basis functions for different orders \( n = 2, n = 3, \) and \( n = 4 \) are presented in Figure 1.

![Figure 1: B-splines of orders n=2, 3, and 4 using ten nodes in [0,1].](image)

3.2 Simulated Annealing as optimization paradigm for placement the knots in Cubic spline

The purpose of this subsection is to clarify and determine the different set of data and functional block within the proposed methodology for using SA as an optimization strategy for the determination of the knot placement for cubic spline approximation. First, it is necessary to define several set of data to evaluate the accuracy of the system:

i) Let \( n_{nod} \) the number of knots to build the Basis of Spline (BS) functions. These knots are denoted by \( X^{BS} = [x_1^{BS}, \ldots, x_{n_{nod}}^{BS}] \) and its corresponding output value, i.e., \( TDS = [X^{BS}, Y^{BS}] \).

ii) Let \( TDS \) be a test data set \( X^{Test} = [x_1^{Test}, \ldots, x_{n_{test}}^{Test}] \) to verify the ability of the spline interpolation method and its corresponding output, i.e., \( TDS = [X^{Test}, Y^{Test}] \).

iii) Finally, let \( Y^{Test} = [y_1^{Test}, \ldots, y_{n_{test}}^{Test}] \) be the output (real numbers) obtained by the spline function on \( X^{Test} \).

The use of a SA requires the determination of three fundamental issues: solution representation, determination of parameters/factors of the algorithm, and the evaluation function. The rest of this section describes each of these issues for the problem of knot placement in cubic spline.

1. Solution Representation. For any SA, a solution representation is needed to describe each individual of interest to be evaluated. The representation scheme determines how the problem is structured in the SA and also determines the behavior of the SA. Each solution is made up of a sequence of variables (in the field of evolutionary computation, each of these variables represent genes) from a certain alphabet. In the bibliography, there exist mainly two different codifications or solution representation: a) binary codification, b) real codification. In our example, the alphabet consist of an array of floating point numbers (real codification), with represent the location of each knot. Figure 2 shows the codification of the knots for an example, function \( f_1 \) used in section Simulation Result. The start point represent the location of the knots for the set Basis of Spline (BS). The dashed line is the function to be approximated, and the dotted line the approximation obtained with the cubic spline.

2. Selection Function. In SA, there are some important parameters that should be defined before running the algorithm. The most important factor are [12]:

- **Initial temperature**: The temperature is the control parameter in simulated annealing that is decreased gradually as the algorithm proceeds. It determines the probability of accepting a worse solution at any step and is used to limit the extent of the search in a given dimension. In the simulation carried out in this paper, the initial temperature has a value for each of the variables to be optimized of 100 (it used to be a default value in software package [12])

- **Annealing Schedule**: The annealing schedule is the rate by which the temperature is
decreased as the algorithm proceeds. The slower the rate of decrease, the better the chances are of finding an optimal solution, but the longer the run time. There exist different options to modify the temperature, in each iteration. For example, it is possible to use a exponential temperature update, which decreases as $0.95^{\text{iteration}}$ the temperature value. Also it is possible to use a logarithmic temperature update, which decreases as $1/\log(\text{iteration})$ the temperature value, or finally a linear temperature update, where the temperature decreases as $1/\text{iteration}$. In the simulation carried out in this paper, these different schedules will be experimentally analyzed.

- **Reannealing**: Annealing is the technique of closely controlling the temperature when cooling a material to ensure that it is brought to an optimal state. Reannealing raises the temperature after a certain number of new points have been accepted, and starts the search again at the higher temperature. Reannealing avoids getting caught at local minima. In the simulation carried out in this paper, this factor has a value of 20.

3. **Fitness function**. Evaluation functions of many forms can be used in a SA, subject to the minimal requirement that the function can map the different solution obtained, into a partially ordered set. As stated, the evaluation function is independent of the SA. In the simulation performed in this paper, the main goal is to minimize the mean squared error (MSE) between the real output of the test data set, and the output obtained by the cubic spline, for a defined number of knots.

It is important to note, analyzing 2, that each variable to be optimized by the SA, is directly the location or placement of a knot. Therefore, it is not necessary to convert the problem into a discrete combinatorial optimization problem, as presented in other paradigms inspired in genetic algorithm [11] or artificial immune systems [10]. In fact, in these methodologies, where different potential solution represent a population which evolves in each iteration, where for a large set of potential candidates to be selected knots, this algorithm selects a subset of them (and therefore coding is performed using integer or binary numbers, which determine whether a node has been selected or not). However, the proposed methodology inspired in SA, for a given number of knots to be used to build the Basis of Spline (BS) functions, determining the optimal positions/location of the knots, without the need to establish an initial set of nodes or select a subset of them. Starting from a initial solution (random or homogenous placement of the knots), the proposed methodology optimize the location of the knots in order to minimize the mean square error obtained in the approximation with a test data set.

4 **Simulations results**

To analyze the behavior of the approximation of the proposed methodology, performed by optimization of the knots placement of cubic spline functions by simulated annealing, various simulations have been carried out, in which various important factors can be modified:

- The number of knots to build the Basis of Spline (BS) functions, that will be denoted as n nod.
- The initial point or solution to carried out the SA algorithm.
- The annealing schedule. In this section, three different options will be experimentally analyzed: exponential temperature update, logarithmic temperature update and linear temperature update.

In order to perform the approximation using the proposed methodology, a function defined in [10] will be used. The function is defined by the following equation:

$$F_j = \frac{90}{1 + e^{-100(x_j - 0.4)}}$$
for $j=1,2,...,N$, where any error value is not used, and the values of variable $x_j$ is 0.0, 0.01,..., 1.0, being, therefore the number of them equal to 101. The interval of the fitting was set to $[a, b] = [0, 1]$.

The number of iteration is fixed to 50 for all the simulations carried out in this section. In Table 1, a summary of simulation for the Function $F_j$, for several values of parameter $Nnod$ and different values of the factor within Simulated Annealing procedure is presented, where, for each configuration a total of 100 simulations are carried out in order to obtain statistical parameter such as the mean and standard deviation. Analyzing this information graphically, in Figure 3, the evolution of the mean of MSE for different values of $Nnod$ and Annealing Schedule, showing that when $Nnod$ increases, the average value of MSE decreases. It can also be appreciated that using the linear temperature update (where the temperature decreases as 1/iteration) produces slightly better result than the other methods. In Figure 4, the behaviour of the SA for the mean temperature (it is important to note, that each variable has his own temperature), and the evolution of the best fitness, is presented, for an example of $F_j$ with $Nnod = 15$ and Annealing Schedule being Exponential. Although it may get better results by increasing the number of iterations of the algorithm, for reasons of computational time complexity, have been set as 50 the maximum number.

![Figure 3](image)

**Figure 3:** Evolution of the mean of MSE for different values of $Nnod$ and Annealing Schedule

### 5 Conclusion

In this paper, a new methodology for knot placement of cubic spline is presented, analyzing the performance of the system, when different factors or parameters within the methodology are modified. The motivation for using an optimization tool such as Simulated Annealing for placement the knot is both: a)
It is well-known that the placement of the knots in spline approximation has an important and considerable effect on the behavior of the final approximation, but the optimal placement or location is not known a priori; b) In theory, knot placement is a multivariate and multimodal nonlinear optimization problem, with can has many local minimum, therefore, using a tool such as SA, it is possible to avoid local minimum in order to obtain the global one. It is also important to note that due the codification used in this paper, real value of the location of the knots is directly optimized, and therefore it is not necessary to convert the problem into a discrete combinatorial optimization problem. Several simulation results have been carried out, showing that increasing the number of knots in the definition of the BS, increases the accuracy of the approximation. It can also be appreciated that using the linear temperature update (where the temperature decreases as 1/iteration) produces slightly better result than the other methods, although it will require a greater number of simulations and a deeper analysis, to generalize this statement.

References:


