Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 1364–1373

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

Memetic Simulated Annealing for Data Approximation with Local-Support Curves

Carlos Loucera[1], Andrés Iglesias[2,3], and Akemi Gálvez[2,3]

[1] Department of Communications Engineering, University of Cantabria, Santander, Spain
[2] Department of Information Science, Toho University, Funabashi, Japan
[3] Department of Applied Maths & Comp. Sciences, University of Cantabria, Santander, Spain
{loucerac,iglesias,galveza}@unican.es

Abstract

This paper introduces a new memetic optimization algorithm called MeSA (*Me*metic *S*imulated *A*nnealing) to address the data fitting problem with local-support free-form curves. The proposed method hybridizes simulated annealing with the COBYLA local search optimization method. This approach is further combined with the centripetal parameterization and the Bayesian information criterion to compute all free variables of the curve reconstruction problem with B-splines. The performance of our approach is evaluated by its application to four different shapes with local deformations and different degrees of noise and density of data points. The MeSA method has also been compared to the non-memetic version of SA. Our results show that MeSA is able to reconstruct the underlying shape of data even in the presence of noise and low density point clouds. It also outperforms SA for all the examples in this paper.

Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

Keywords: Simulated Annealing, Memetic, Data Fitting, Spline, COBYLA

1 Introduction

This paper deals with the problem of data approximation with free-form parametric functions. Examples of this problem range from digital reconstruction of the boundary of a physical model with free-form surfaces [23] to the use of geometric modeling tools for computer-aided design and manufacturing [6, 25]. A way to address this problem is to use local-support splines, which are piecewise parametric functions defined on intervals with the end points called knots. Unfortunately, the high dependence of nonlinear interactions among the numerous parameters makes spline fitting a very complicated problem. Furthermore, although the quality of the reconstruction increases if the knots are treated as free variables, so does the problem difficulty [5, 8]. In fact, classical mathematical optimization tools fail to solve the whole problem at once, so they usually pre-compute a data parameterization; then, a closely interlinked knot vector is found through classical optimization methods. A lot of variations to this approach have been proposed: from threshold-driven knot insertion/removal [18], to exploit knowledge

1877-0509
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.048

about the data and the inclusion of subjective parameters. Although a variety of shapes can be reconstructed with these methods, the general case is beyond their limits as the fitness landscape is full of valleys and local minima, were classical methods tend to get stuck [3].

Recent advances in nature-inspired evolutionary algorithms have made it possible to perform data fitting successfully in some particular cases. References [11, 14, 15, 21] report data fitting with global-support curves by using differential evolution, artificial immune systems, bat algorithm, and simulated annealing, respectively. Other examples with local-support curves can be found in [7, 30, 31]. In this context, this work introduces a new memetic optimization algorithm called MeSA (*Me*metic *S*imulated *A*nnealing) to address the data fitting problem with local-support free-form curves. Our approach is based on the hybridization of simulated annealing with the COBYLA local search optimization method. This approach is further combined with the centripetal parameterization and the Bayesian information criterion to compute all free variables of the curve reconstruction problem with B-splines.

This paper is organized as follows: Section 2 provides the basic mathematical background about the problem. The fundamentals about simulated annealing are described in Section 3, while Section 4 describes our new memetic approach in detail. Section 5 outlines the methodology used in this paper for data fitting with B-splines. Finally, we present four representative examples an a comparative analysis with the non-memetic simulated annealing method in Section 6. The paper closes with the main conclusions and some future work in the field.

2 Mathematical Background

In this section we introduce the main mathematical concepts about the parametric B-spline curves along with the problem we want to solve. From now on, vectors are denoted in bold.

2.1 Parametric B-spline Curves

Mathematically, a *parametric B-spline curve* $\mathbf{C}(t) \subset \mathbb{R}^d$ of order p is a piecewise function expressed as:

$$\mathbf{C}(t) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,p}(t) \tag{1}$$

where $t \in [\alpha, \beta]$ represents the *curve parameter*, $\{\mathbf{P}_i\}_i$ are the *control points* of the curve, and $\{N_{i,p}(t)\}_i$ are the so-called *B-spline basis functions* of order p defined on a knot vector $\mathcal{U} = \{u_0 = \alpha, u_1, u_2, \ldots, u_{n+p} = \beta\}$, comprised of non-decreasing real numbers u_i called *knots*. The B-spline basis functions $N_{j,p}(t)$ can be computed through the Cox de-Boor recursive formula (see [2] for details):

$$N_{j,p}(t) = \frac{t - u_j}{u_{j+p-1} - u_j} N_{j,p-1}(t) + \frac{u_{j+p} - t}{u_{j+p} - u_{j+1}} N_{j+1,p-1}(t) \tag{2}$$

for $p > 1$, while for $p = 1$ we have:

$$N_{j,1}(t) = \begin{cases} 1 & \text{if} \quad u_j \leq t < u_{j+1} \\ 0 & \text{otherwise} \end{cases} \qquad (j = 0, \ldots, n + p - 1) \tag{3}$$

In this paper we consider the case of B-spline curves clamped at the end points, based of repeating the end knots as many times as the order, i.e., $u_0 = \cdots = u_{p-1} = \alpha$, and $u_{n+1} = \cdots = u_{n+p} = \beta$. In this case, the curve interpolates the end control points, a valuable property in several real-world applications. Without loss of generality, we can assume that $[\alpha, \beta] = [0, 1]$.

2.2 Data Fitting

Let $\{\mathbf{Q}_k\}_{k=1,\dots,M}$ be a set of points in $\mathbb{R}^d$. Our goal consists of finding a B-spline curve $\mathbf{C}(t)$ approximating the given data with high fidelity while trying to keep the model complexity as low as possible. Because the curve is parametric, our method must perform data parametrization, i.e., finding the $\{t_k\}_k$ associated with the original data. Then, we have to compute the control points $\{\mathbf{P}_i\}_i$ as well as the knots $\{u_j\}_j$ and, finally, deal with the model complexity: how to minimize the number of free parameters of the system.

Due to the constraints imposed on the boundary knots, we can assume that $\mathbf{C}(t_1) = \mathbf{Q}_1$ and $\mathbf{C}(t_M) = \mathbf{Q}_M$. As a result, the equation to minimize in a least-squares sense is given by:

$$E = \sum_{k=2}^{M-1} \left|\left| \mathbf{Q}_k - \sum_{i=0}^{n} \mathbf{P}_i N_{i,p}(t_k) \right|\right|_2^2 \tag{4}$$

where $||.||_2$ indicates the Euclidean norm. Note that when the order of the curve, the data parameterization, and the knot vector are known, Eq. (4) becomes a simple linear system. However, in many real-world problems, such values cannot be obtained directly from the data; instead, they have to be fully computed. In such a case, the least-squares minimization problem (4) becomes highly nonlinear, continuous, and multivariate. In addition, the computation of the knot vector has been proved to be a non-convex and multi-modal optimization problem [2, 20]. To overcome such difficulties we propose an optimization schema that deals with each sub-problem sequentially: firstly, data parameterization; then, knot vector and control points computation; and, finally, model complexity.

3 Simulated Annealing

The *simulated annealing* (SA) is a thermodynamics-inspired metaheuristic algorithm originally proposed by Kirkpatrick *et al.* in 1983 to solve large-scale combinatorial optimization problems [19]. The algorithm was inspired by the annealing process of a physical system, a technique where a material (typically a metal) is subjected to a process of heating and then controlled slow cooling in order to improve the material inner structure and hence, to increase its toughness. The optimal configuration for the system is reached at the state of minimal energy.

In its most basic form, the SA algorithm identifies the state (through its energy) of a physical system with the function to be minimized. In resemblance to the annealing physical process, the method tries to achieve the optimum (state of minimal energy) from an arbitrary initial solution (initial state). At the beginning, the system is exposed to very high temperatures so that the particles can move freely. The temperature is then slowly decreased in order to let the particles adopt a more stable configuration. At the final stages there is almost no particle movement, as the system is near the state of minimal energy in which the particles find the most stable configuration.

The original SA algorithm is based on the *Metropolis-Hastings algorithm* [22] to generate sample states of a thermodynamic system. Given an initial (usually random) *state* in the solution domain, the algorithm iteratively perturbs it. If a better solution is found, the change is always accepted; otherwise, it is accepted only with a certain probability. This probability is higher at the beginning that at the end. In this way, this idea of slow cooling translates into that of a slow decrease of the probability of accepting such worse solutions. As a result, the system evolves from a free exploration of the search space at the beginning to a stochastic hill-climbing at latter stages.

3.1 The Algorithm

The SA algorithm is designed to minimize a real-valued fitness function (usually called the *system energy*) $f : \mathcal{D} \subseteq \mathbb{R}^d \longrightarrow \mathbb{R}$, within a problem domain $\mathcal{D}$, assumed to be continuous in this paper. Each point $\mathbf{x} \in \mathcal{D}$ is a *state* of the physical system. Given an initial (usually random) state $\mathbf{x}_0$, the algorithm performs an iterative process; at each iteration step, a new state $\mathbf{x}_{new}$ is generated from the current one, $\mathbf{x}_{old}$, through a *neighborhood function*, denoted by $\mathfrak{N} : \mathcal{D} \longrightarrow \mathcal{D}$, i.e., $\mathbf{x}_{new} = \mathfrak{N}(\mathbf{x}_{old})$. Let now $f_{old} \equiv f(\mathbf{x}_{old}), f_{new} \equiv f(\mathbf{x}_{new})$ be their associated energies, respectively. The algorithm probabilistically decides between moving the system to the new state $\mathbf{x}_{new}$ or staying in the current state $\mathbf{x}_{old}$. This new state is chosen with a probability function $\mathfrak{P} : \mathcal{D} \times \mathcal{D} \longrightarrow [0, 1]$, called the *acceptance function*, which depends on two factors: (1) the difference $\triangle = f_{old} - f_{new}$ of the energy values; and (2) a global parameter called *temperature*, denoted by T, which varies according to a strictly decreasing function $\mathfrak{T} : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ called the *cooling function*.

In addition, two more conditions are required. The first one is that $\mathfrak{P} > 0$ if $\triangle < 0$, meaning that the system may move to the new state even if it is worse than the current one. This condition is imposed with the goal to prevent *stagnation* (when the system gets trapped in the neighborhood of local optima, leading to premature convergence). The second one is that the lower the temperature, the easier to reject a worse solution. In fact, in the particular case $T = 0$, the procedure will only allow downhill moves, meaning that the algorithm reduces to a greedy search algorithm. The interested reader is referred to [16] for further details about the algorithm and the corresponding pseudocode.

4 MeSA: Memetic Simulated Annealing

Memetic algorithms were originally introduced as an enhancement for genetic-driven metaheuristics by introducing the idea of individual learners potentially able to refine some members of a population [9], thus mimicking more closely the domain-specific processes of the universal Darwinian theory (incorporate knowledge of the problem). Nowadays, the memetic optimization approach has been applied with varying grades of success to a wide range of metaheuristic optimization techniques beyond the genetic algorithms paradigm. Some illustrative examples of memetic approaches can be found, for instance, in [7, 12, 13, 15, 24].

In this paper we propose a memetic variant of the simulated annealing algorithm that performs solution refinement by means of the *constrained optimization by linear approximations* (COBYLA) local search procedure. The proposed optimization framework is described in Algorithm 1. Our memetic approach can be divided into three closely intertwined phases: information gathering, cooling, and local learning. About the *information gathering phase*, a remarkable feature of recent developments for memetic optimization is the inclusion of problem knowledge during the generation of the initial population [10]. Our MeSA approach learns the temperature parameter by initially exploring the fitness landscape via the `LearnParameters` process. The method generates a random population within the search space and sets the initial temperature to 0.8 times the worst energy transition.

During the *cooling phase*, the general SA schema is applied to compute the knots. It consists of two nested loops. The main or outer loop, labeled in the literature as the annealing or cooling loop, controls the temperature update process and the stop criterion. In our implementation, the stop criterion consists of running the outer loop for a predefined number of iterations N_{outer}. Let k be the outer iteration index from now on. The method keeps track of two control parameters, the acceptance (T^{ac}) and generation (T^{gen}) temperatures, which are initialized

Algorithm 1: MeSA: Memetic Simulated Annealing (by linear approximations)

Input: An intial guess $\mathbf{x}_0$, the function to optimize f, lower and upper bounds $\mathbf{l}, \mathbf{u}$
Output: The final solution $\mathbf{x}$
$T_0^{ac} \leftarrow \texttt{LearnParameters}(f, \mathbf{l}, \mathbf{u})$
$\mathbf{x} \leftarrow \mathbf{x}_0$ and $f_{\mathbf{x}} \leftarrow f(\mathbf{x})$
$T^{ac} \leftarrow T_0^{ac}$
$T^{gen} \leftarrow T_0^{gen}$
while *The System is not Frozen* **do**
  while *Thermal Equilibrium is not Reached* **do**
    $\mathbf{x}_{new} \leftarrow \mathfrak{N}(\mathbf{x})$ and $f_{new} \leftarrow f(\mathbf{x}_{new})$
    if $\mathfrak{A}(f_{new}, f_{\mathbf{x}}, T^{ac})$ **then**
      $\mathbf{x} \leftarrow \mathbf{x}_{new}$ and $f_{\mathbf{x}} \leftarrow f_{new}$
    end
  end
  $\mathbf{x} \leftarrow \texttt{LocalLearning}(\mathbf{x}, \mathbf{l}, \mathbf{u})$ and $f_{\mathbf{x}} \leftarrow f(\mathbf{x})$
  $T^{ac} \leftarrow \mathfrak{T}^{ac}(T_0^{ac})$
  $T^{gen} \leftarrow \mathfrak{T}^{gen}(T_0^{gen})$
end
return x

during the information gathering phase and subsequently updated during the cooling phase as: $\mathfrak{T}^{ac} : T_{k+1}^{ac} \leftarrow \frac{T_0^{ac}}{k}$ and $\mathfrak{T}^{gen} : T_{k+1}^{gen} \leftarrow \left(\frac{k_{outer}}{N_{outer}}\right)^{-1}$, respectively. The inner loop mimics the achievement of thermal equilibrium system state at a given temperature. Similarly to the outer loop, our implementation runs the inner loop N_{inner} iterations. During the inner loop, the new candidate solutions, $\mathfrak{N} : \mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}$, are generated according to the inverse μ-law function [27] given by:

$$\Delta\mathbf{x} = g_\mu^{-1}(\mathbf{y}) \odot (\mathbf{u} - \mathbf{l}) \quad \text{with} \quad g_\mu^{-1}(\mathbf{y}) = \frac{(1+\mu)^{|\mathbf{y}|} - 1}{\mu} \odot \text{sign}(\mathbf{y})$$

where the $\odot$ symbol represents the element wise vector multiplication, $\mu = 10^{100T^{gen}}$ and $\mathbf{y} \in U^d([-1, 1])$. See [29] for a more detailed discussion on this search procedure. Then, the law governing the probability of accepting a given transition follows the modified Metropolis criterion [22]: $\mathfrak{A} \leftarrow \min\left\{1, \left(1 + \exp\left(\frac{\Delta f}{T_k^{ac}}\right)\right)^{-1}\right\}$.

At the end of each inner loop , i.e. when the thermal equilibrium for a given temperature has been reached, the algorithm performs a local search procedure with the last accepted point as an initial guess. This is called the *local learning phase*. This local search is performed via the COBYLA algorithm with a budget of N_{local} function evaluations.

5 The Proposed Method

As described in Section 2.2, our problem consists of finding the best B-spline fitting curve to a given set of (possibly noisy) data points while keeping the complexity of the model as low as possible. To do so, we need to compute a suitable parameterization of the data, the optimal number of knots along with their location, and the B-spline control net. We solve

Shape	Parameters	σ	Method	E	BIC	$NMSE_x$	$NMSE_y$
Elephant	208	35	MeSA	0.0029422	-64.777	0.99988	0.99974
			SA	0.0081525	147.21	0.99911	0.99972
Camel	204	50	MeSA	0.0012596	9.968	0.99993	0.99992
			SA	0.0027904	172.24	0.99984	0.99983
Beetle	208	60	MeSA	0.0008605	79.845	0.99992	0.99991
			SA	0.0059459	481.88	0.99946	0.99935
Bell	752	40	MeSA	0.058327	1187.7	0.99991	0.99993
			SA	0.0700	1325.25	0.99981	0.99991

Table 1: Numerical fitting errors for MeSA and SA on the four examples in our benchmark.

all those problems by combining four different techniques: the centripetal method for data parameterization, our memetic simulated annealing (MeSA) method – based on hybridizing simulated annealing with the COBYLA local optimization method – along with least-squares minimization for the determination of the knot vector and the control points respectively, and finally, BIC (Bayesian information criterion) for model selection.

The centripetal parameterization is one of the most popular data parametrization methods [4], as it takes into account both the distribution of data and sharp turns. It is given by:

$$t_1 = 0 \qquad \text{and} \qquad t_k = \frac{\sum_{j=2}^{k} |\mathbf{Q}_j - \mathbf{Q}_{j-1}|^{\frac{1}{2}}}{\sum_{j=2}^{M} |\mathbf{Q}_j - \mathbf{Q}_{j-1}|^{\frac{1}{2}}} \qquad \text{for } k = 2, \ldots, M. \tag{5}$$

In this paper, we consider cubic B-spline curves (although our method is actually independent on the order of the fitting curve). Now, assuming a given parameterization and the number of knots, the only unknowns in (4) are the control points and the knot vector. To compute the knots, we apply the MeSA algorithm, where each state is given by the following representation scheme: $\mathbf{x} \in (0,1)^{\sigma}$, where σ refers to the number of free (i.e., not-clamped) knots. The elements in $\mathbf{x}$ are then sorted to conform to the ordered structure of the knot vector. Finally, our fitness function is taken as $f \equiv E$. Regarding the parameter tuning, it is as follows: $N_{outer} = 500, N_{inner} = 50, N_{local} = 200, \sigma \in \{1, \ldots, 70\}$. We use the same parameter setup for the non-memetic SA approach, used here for comparative purposes. However, we increase the number of iterations as $N_{outer} = 1000, N_{inner} = 100$ for the non-memetic version to allow this simpler (and arguably slower) version to reach convergence. Once the knots are obtained, the control points can be computed by linear least-squares minimization of the functional E, leading to an overdetermined linear system that can be solved by standard numerical methods.

The functional E does not contain any information about the model complexity, so the model might (potentially) be affected by overfitting. This is a common problem when approximating data with B-splines, as E decreases as the number of knots increases. To overcome this limitation, we compute the modified BIC (Bayesian information criterion) cost [26], given by:

$$BIC = M \log(E) + \zeta \log(M) \tag{6}$$

where ζ represents the total number of free parameters of the problem and log(.) represents the natural logarithm function. As a general rule, the model with the lower BIC is preferred.

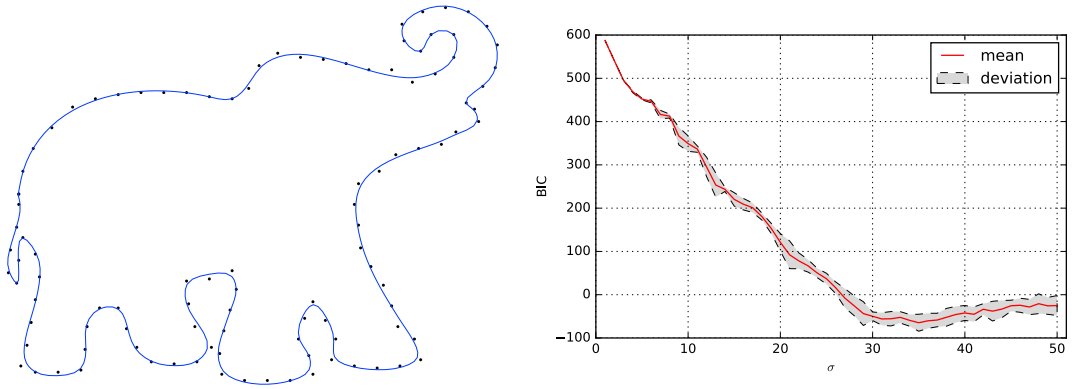

Figure 1: Reconstruction of the elephant shape: (l) best fitting curve ($\sigma = 35$); (r) BIC vs. σ.

To measure the goodness of the fit, we also compute the *normalized mean square error* NMSE (7) for each spatial component. The NMSE is given by:

$$NMSE_i = 1 - \left\| \frac{\mathbf{Q}_i - \mathbf{Q}_i^*}{\mathbf{Q}_i - \text{mean}(\mathbf{Q}_i^*)} \right\|^2 \tag{7}$$

where $\mathbf{Q}_i^*$ represents the reconstructed point associated with $\mathbf{Q}_i$, and i represents the spatial component (x and y in our examples). Note that the NMSE values vary between $-\infty$ and 1; the closer to one, the better the fit.

6 Experimental Results

To assess our MeSA method, we consider four datasets of synthetic shapes (elephant, camel, beetle, and bell) from [1, 28], depicted as black dots on the left of Figs. 1–4. The datasets consist of 104, 102, 104, and 376 points, respectively (only 124 are drawn for the bell example for better visualization). The figures also show the best fitting curve (blue solid line) obtained with the MeSA method according to the BIC. To this aim, Figs. 1–4 (right) show the evolution of the BIC value against σ. A total of 26 independent runs are executed for each σ value. The three best and worst runs are then removed to provide statistical evidence for the results and assert the experiment reproducibility. The mean BIC value is displayed as a red solid line, while the minimum and maximum values are represented by the lower and upper dashed lines. The deviation area is filled up with a gray tone for better visualization. From the figures and the numerical results, the best value for σ is determined and used for the fitting curves in Figs. 1–4 (left). Note that all shapes present difficult geometric features, such as strong changes of slope and curvature. Still, our method is able to reconstruct the general shape of the data with good accuracy, as confirmed visually in those figures. Table 1 summarizes our numerical results. The following data are reported (in columns): the shape, number of free parameters, best value for σ, the method used, and the E, BIC, and NMSE (for x and y) fitting errors. For further assessment, the results for the MeSA method are compared with those of a standard implementation of the (non-memetic) SA. As shown in Table 1, our method outperforms SA for all shapes in our benchmark and others not included here because of limitations of space.

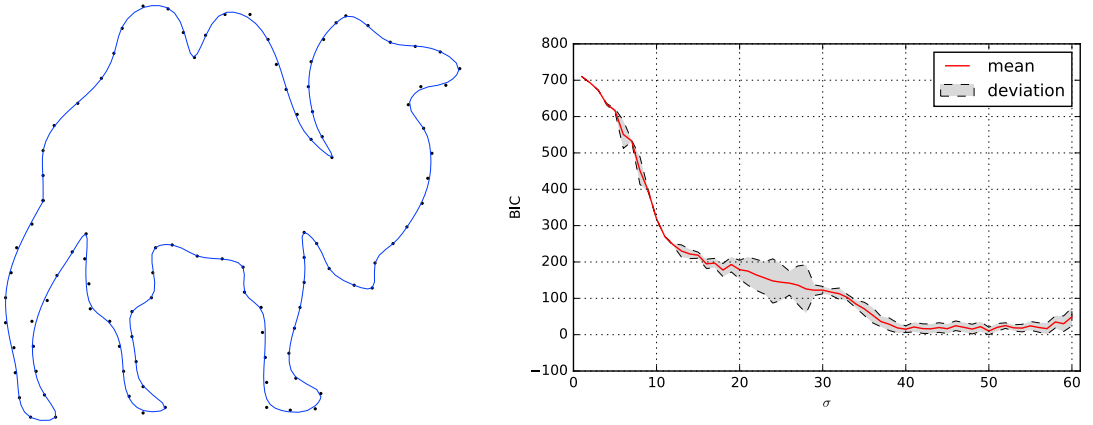

Figure 2: Reconstruction of the camel shape: (l) best fitting curve ($\sigma = 50$); (r) BIC vs. σ.

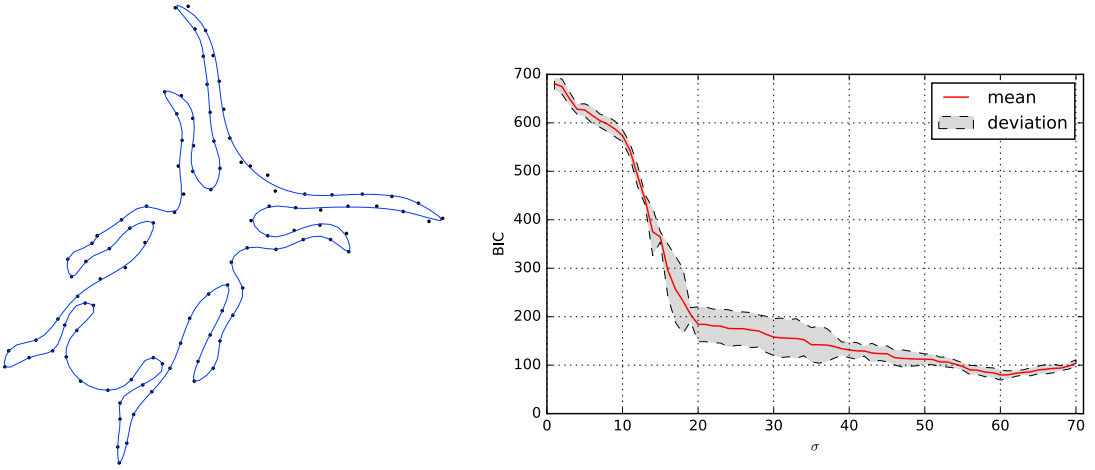

Figure 3: Reconstruction of the beetle shape: (l) best fitting curve ($\sigma = 60$); (r) BIC vs. σ.

7 Conclusions and Future Work

In this paper, we present a new memetic optimization algorithm called MeSA that hybridizes simulated annealing with the COBYLA local optimization method. This approach is further combined with the centripetal parameterization and the Bayesian information criterion to solve the curve reconstruction problem to a given set of data points. Our curve fitting model is based on local support splines to take advantage of their ubiquitousness in the CAD/CAM field. We have illustrated the excellent performance of our algorithm in representing a variety of shapes with local deformations and different degrees of noise and density of data points. The MeSA method has also been compared to the non-memetic version of SA. Our results show that MeSA outperforms SA for all the examples in this paper.

Regarding the implementation issues, all computations have been carried out on an `Intel i7-7600` quad-core processor with 16GB of RAM. The source code has been implemented by the authors in the native programming language of MATLAB, v.2014b. We also used the COBYLA implementation in the `NLopt` library [17]. All the simulations took less than

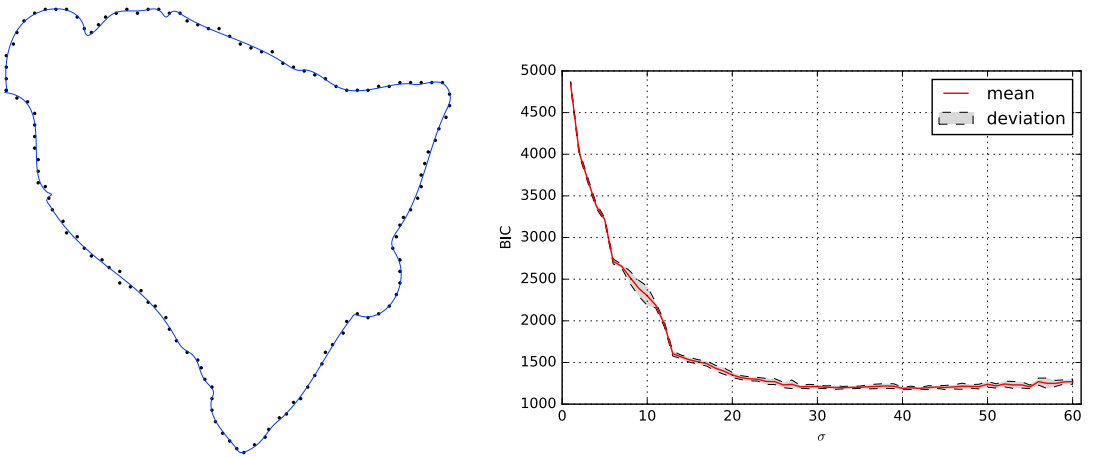

Figure 4: Reconstruction of the bell shape: (l) best fitting curve ($\sigma = 40$); (r) BIC vs. σ.

30 seconds. Future work includes the comparison with other optimizations schemes reported in the literature, as well as the extension of the proposed methodology to surface fitting.

Acknowledgments

This work has been supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under grants TEC2013-47141-C4-R (RACHEL) and #TIN2012-30768 (Computer Science National Program) and Toho University (Funabashi, Japan).

References

[1] Axel Carlier, Kathryn Leonard, Stefanie Hahmann, Geraldine Morin, and Misha Collins. The 2D shape structure dataset: A user annotated open access database. *Computers & Graphics*, 58:23–30, 2016.

[2] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag GmbH, 2001.

[3] Paul Dierckx. *Curve and surface fitting with splines*. Oxford University Press, 1995.

[4] Gerald Farin. *Curves and Surfaces for CAGD*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2002.

[5] Akemi Gálvez and Andrés Iglesias. Efficient particle swarm optimization approach for data fitting with free knot B-splines. *Computer-Aided Design*, 43(12):1683–1692, 2011.

[6] Akemi Gálvez and Andrés Iglesias. A new iterative mutually coupled hybrid GA-PSO approach for curve fitting in manufacturing. *Applied Soft Computing*, 13(3):1491–1504, 2013.

[7] Akemi Gálvez and Andrés Iglesias. New memetic self-adaptive firefly algorithm for continuous optimisation. *International Journal of Bio-Inspired Computation*, 8(5):300–317, 2016.

[8] Akemi Gálvez and Andrés Iglesias. Particle-based meta-model for continuous breakpoint optimization in smooth local-support curve fitting. *Applied Mathematics and Computation*, 275:195–212, 2016.

[9] Stephen Harris and Emmanuel Ifeachor. Automatic design of frequency sampling filters by hybrid genetic algorithm techniques. *IEEE Transactions on Signal Processing*, 46(12):3304–3314, 1998.

[10] William Hart, Natalio Krasnogor, and James Smith. *Recent advances in memetic algorithms*, volume 166. Springer Science & Business Media, 2004.

[11] Allan Hasegawa, Roberto Rosso, and Marcos Sales-Guerra. Bézier curve fitting with a parallel differential evolution algorithm. *IFAC Proceedings Volumes*, 46(7):233–238, 2013.

[12] Andrés Iglesias and Akemi Gálvez. Memetic firefly algorithm for data fitting with rational curves. *IEEE Congress on Evolutionary Computation, CEC'2015*, pages 507–514, 2015.

[13] Andrés Iglesias and Akemi Gálvez. Memetic electromagnetism algorithm for surface reconstruction with rational bivariate Bernstein basis functions. *Natural Computing* (*in press*) doi:10.1007/s11047-016-9562-5.

[14] Andrés Iglesias, Akemi Gálvez, and Andreina Avila. Discrete bézier curve fitting with artificial immune systems. In *Intelligent Computer Graphics 2012*, pages 59–75. Springer, 2013.

[15] Andrés Iglesias, Akemi Gálvez, and Marta Collantes. Four adaptive memetic bat algorithm schemes for bézier curve parameterization. *Transactions on Computational Science XXVIII*, pages 127–145. Springer, 2016.

[16] Andrés Iglesias, Akemi Gálvez, and Carlos Loucera. Two simulated annealing optimization schemas for rational Bézier curve fitting in the presence of noise. *Mathematical Problems in Engineering*, Volume 2016, Article ID 8241275, 17 pages, 2016.

[17] Steven Johnson. The NLopt nonlinear-optimization package. *http://ab-initio.mit.edu/nlopt*.

[18] David Jupp. Approximation to data by splines with free knots. *SIAM Journal on Numerical Analysis*, 15(2):328–343, apr 1978.

[19] Scott Kirkpatrick, Daniel Gelatt, and Mario Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[20] Pascal Laurent-Gengoux and Mounib Mekhilef. Optimization of a NURBS representation. *Comput. Des.*, 25(11):699–710, 1993.

[21] Carlos Loucera, Akemi Gálvez, and Andrés Iglesias. Simulated annealing algorithm for Bézier curve approximation. In *Int. Conf. on Cyberworlds, CW'2014*, pages 182–189. IEEE, 2014.

[22] Nicholas Metropolis, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087, 1953.

[23] Michael Milroy, Colin Bradley, Geoffrey Vickers, and DJ Weir. G1 continuity of B-spline surface patches in reverse engineering. *Computer-Aided Design*, 27(6):471–478, 1995.

[24] Yannis Petalas, Konstantinos Parsopoulos, and Michael Vrahatis. Memetic particle swarm optimization. *Annals of Operations Research*, 156(1):99–127, aug 2007.

[25] H. Pottmann, S. Leopoldseder, M. Hofer, T. Steiner, and W. Wang. Industrial geometry: recent advances and applications in CAD. *Computer-Aided Design*, 37(7):751–766, jun 2005.

[26] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[27] Bernard Sklar. *Digital communications*, volume 2. Prentice Hall NJ, 2001.

[28] Ninad Thakoor, Jean Gao, and Sungyong Jung. Hidden Markov model-based weighted likelihood discriminant for 2-D shape classification. *IEEE Transactions on Image Processing*, 16(11):2707–2719, 2007.

[29] Won Yang, Wenwu Cao, Tae-Sang Chung, and John Morris. *Applied numerical methods using MATLAB*. John Wiley & Sons, 2005.

[30] Fujiichi Yoshimoto, Toshinobu Harada, and Yoshihide Yoshimoto. Data fitting with a spline using a real-coded genetic algorithm. *Computer-Aided Design*, 35(8):751–760, 2003.

[31] Yuan Yuan, Nan Chen, and Shiyu Zhou. Adaptive b-spline knot selection using multi-resolution basis set. *IIE Transactions*, 45(12):1263–1277, 2013.