Comput Syst Sci & Eng (2020) 6: 431–440

International Journal of
Computer Systems
Science & Engineering

B-Spline Curve Approximation by Utilizing Big Bang-Big Crunch Method

Özkan İnik[1,*], Erkan Ülker[2], İsmail Koç[2]

[1] *Department of Computer Engineering, Gaziosmanpaşa University, Tokat, Turkey*
[2] *Department of Computer Engineering, Konya Technical University, Konya, Turkey*

The location of knot points and estimation of the number of knots are undoubtedly known as one of the most difficult problems in B-Spline curve approximation. In the literature, different researchers have been seen to use more than one optimization algorithm in order to solve this problem. In this paper, Big Bang-Big Crunch method (BB-BC) which is one of the evolutionary based optimization algorithms was introduced and then the approximation of B-Spline curve knots was conducted by this method. The technique of reverse engineering was implemented for the curve knot approximation. The detection of knot locations and the number of knots were randomly selected in the curve approximation which was performed by using BB-BC method. The experimental results were carried out by utilizing seven different test functions for the curve approximation. The performance of BB-BC algorithm was examined on these functions and their results were compared with the earlier studies performed by the researchers. In comparison with the other studies, it was observed that though the number of the knot in BB-BC algorithm was high, this algorithm approximated the B-Spline curves at the rate of minor error.

Keywords: B-Spline curve fitting; Big Bang–Big Crunch; Knot placement; Reverse engineering

1. INTRODUCTION

Curve fitting problem for data points is the primary of the problems in many application areas. For instance, for representation of objects in the real-world on digital media, these objects are firstly required to be scanned. Then, curve and surface modeling, which will represent the objects by the acquired data points should be obtained. Reverse engineering [1] is utilized in order to obtain curve and surface models by data points. Curve fitting technique is known to be the main of several innovations in industrially car bonnet designs, ship hull designs, medical sector in addition to developments in computer modeling and design [2]. In addition to these, curve fitting method takes an important place within research issues in geometrical modeling [3], computer-aided design (CAD), computer-aided modeling (CAM) [4, 5] and computer-aided manufacturing areas.

*Corresponding author: ozkan.inik@gop.edu.tr

Mathematical functions are necessary to be used for reproducing the original aspect of the object by utilizing data points acquired from real objects. Different functions can be utilized for this process. Especially, as long as complexity level of shape increases, it is necessary to use different mathematical functions. Functions with free-form piecewise polynomial functions such as Bezier, B-Spline and Non-Uniform Rational B-Spline (NURBS) [6–13] are used for the complex shapes. The most commonly used functions within these functions are B-Splines. In B-Spline curves, the whole curve is not influenced when any parameter of the curve is changed. So, parameter changes have only local effects on the B-Spline curve. Calculating B-Splines causes more effort than Bezier basis functions and evaluating NURBS should cause even more effort because they consist of a fraction with B-Splines in the numerator and denominator. In contrast to B-Splines, NURBS allow to represent a perfect circle so they are more powerful in geometrical terms. In contrast to Bezier curves, B-Spline curves are continuous for a higher derivative.

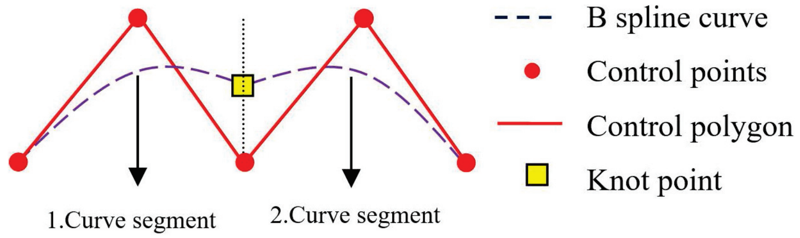

Figure 1 B-Spline curve and control polygon.

The most important issue for B-Splines is the knot vector. Particularly, knot selection remarkably affects the shape of curve [14, 15]. In order to obtain a good approximation, a suitable determination of B-Spline parameters is essential [12]. This problem is NP-Hard problem. Especially given the point cloud is large, Utilization of other alternative optimization techniques instead of mathematical techniques has given more successful results for B-Spline knot vector approximation. Optimization methods available in the field of artificial intelligence become a part of an activity in solving maximization or minimization problems. The most optimal solutions for B-Spline knot vector have been seen to be obtained by several artificial intelligence techniques [16, 17]. Big Bang-Big Crunch (BB-BC) which is one of the evolutionary based optimization algorithms was firstly discussed for B-Spline knot approximation in this paper. The most important reason for choosing the BB-BC method is to obtain the best value with less iteration.

For B-Spline curve fitting problem, different artificial intelligence techniques have been utilized in the literature. Yoshimoto et al. [18] used genetic algorithm (GA) for the operation of automatic knot placement in data fitting problem. Gálvez and Iglesias [10] applied the GA paradigm iteratively to fit a given cloud of data points by using strictly polynomial B-spline surfaces. Sarfraz and Raza [19] also incorporated a corner detection algorithm to detect significant points which are necessary to capture a pleasant looking spline fitting for shapes such as fonts. Kumar et al. [20] presented an approach based on GA for the parameter optimization in Non-Uniform B-Spline curve fitting. Pittman [21] presented an adaptive modeling technique referred to as adaptive genetic splines which combines the optimization power of a GA with the flexibility of polynomial splines. Sarfraz and Raza presented a method to transform the original problem into a discrete combinatorial optimization problem and solved it by a GA. They also incorporated a comer detection algorithm to detect significant points (comer points), which are necessary to capture a pleasant looking spline fitting for 2D and 3D data [22]. Ulker and Arslan [16] performed B-Spline knot placement by using Artificial Immune System. Gálvez et al. [23] introduced an adapted elitist clonal selection algorithm for automatic knot adjustment of B-spline curves. Gálvez and Iglesias presented a new method to overcome difficulties of reconstruction of freeform objects. Some these difficulties are correct determination of the length of knot vectors and the calculation cost and slow in the concavities or holes that contain numerous data points. The method applies the particle swarm optimization (PSO) paradigm to compute an appropriate location of knots automatically [24].

The rest of the paper is organized as follows. In Section 2 describes previous work related to this study and this section gives information about B-Spline curves. BB-BC algorithm [25] which is one of the evolutionary based optimization algorithms is explained in Section 3. How a B-Spline curve approximation is carried out is gradually described in Section 4. The test results obtained by BB-BC method on B-Spline curve fitting and comparisons with other studies are given in Section 5. The final section concludes the article.

2. B-SPLINE CURVES

Although occurrence of B-Splines dates back to 1947, when they were firstly proposed by De Boor [26] , they gained industrial popularity [16]. B-Spline curves are developed of Bezier curves. However, B-Spline curves do not always consist of one-piece curve like Bezier curve. B-Spline curve is composed of combination of at least one or more polynomial segments. In case B-Spline curve is composed of only one segment, this curve is Bezier curve as well.

B-Spline curves and surfaces are identified by vertices named as control points. Although the curves and surfaces obtained by using these points do not come around to the control points, the form of the curve or surface completely shapes according to positions of these points. The polygon which these control points generate is named as the control polygon. These points enable the curve to track the shape of the polygon by acting like a magnet and ultimately, the characteristic and smooth curve located within the boundaries of the control polygon is acquired [27]. B-Spline curves own an effective geometric feature that a local effect occurs by changing only one part of the curve when one of the points moves. However, in the Bezier curves, the whole curve is affected from the first point to the end when only one of the points in data set moves. A typical B-Spline curve and control polygon are presented in Figure 1.

The definition of B-Spline curves are as follow:

$$C(t) = \sum_{i=0}^{n} P_i N_{i,k}(t) \tag{1}$$

P_i is one of n+1 control point. t is knot vector. The ith B-Spline basis function $N_{i,k}(t)$ of order k (or degree d $= k - 1$) can be defined recursive relations given as follow [28, 29].

$$N_{i,1}(t) = \begin{cases} 1 & t_i \leqslant t < t_{i+1} \\ 0 & otherwise \end{cases} \tag{2}$$

$$N_{i,k}(t) = \frac{(t - t_i)}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{(t_{i+k} - t)}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t) \tag{3}$$

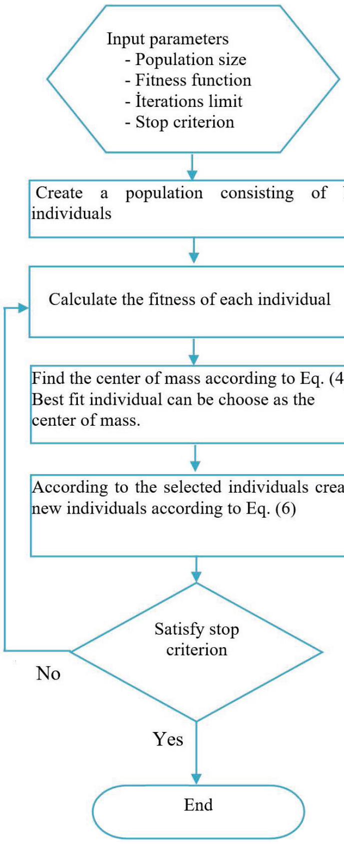

Figure 2 The flowchart of BB-BC algorithm.

3. BIG BANG-BIG CRUNCH(BB-BC) METHOD

BB-BC algorithm was firstly presented in 2006 as an evolution-based optimization algorithm [25]. The most important feature of the algorithm is that it has high convergence speed in addition to low computational time. For instance, while evolution-based algorithms used in solution of an optimization problem present the best optimal solution in the result of too much iteration, BB-BC algorithm generally attains solutions very close to the optimal solution of this problem on far less iterations. The operating logic of this evolutionary method is indicated as transformation of ideal solution into an irregular status which comprises of new solution set [25]. The flowchart of BB-BC algorithm is given in Figure 2.

BB-BC algorithm consists of two phases: Big Bang and Big Crunch.

First Phase: This stage is named as *Big Bang* phase. At this phase, initial population is randomly generated similar to genetic algorithm. The candidate solutions which are randomly generated are spread into search space in a uniform way.

Second Phase: At the phase called Big Crunch, the center of gravity of the population and the most optimal individual of the population are calculated. The point representing the center of gravity of mass is indicated as X^c and it is calculated according to the following formula.

$$X^c = \frac{\sum_{i=1}^{\mathrm{H}} \frac{1}{f^i} \mathrm{X}^i}{\sum_{i=1}^{\mathrm{H}} \frac{1}{f^i}} \tag{4}$$

where x^i is a point generated in h-dimensional search space and f^i is a fitness function of this point value. H is a population size in Big Bang phase as well After the second stage completes, i^{th} new individuals (X_i^{new}) are once again calculated for Big Bang stage according to the formula below.

$$X_i^{new} = X^c + \frac{lr}{s} \qquad i = 1, 2, 3, \ldots\ldots H \tag{5}$$

where r is a randomly generated value between 0–1. s is iteration number. l upper bounds of parameters. l possible value for each candidate that creates a solution. If a binary problem is solved, the possible values are 0 or 1. In this case, l becomes 1. If the problem is a numerical problem, the solution interval depends on the problem. Accelerate convergence with r close 1. Convergence is slow if the number of iterations increases. In Table 1, the steps of BB-BC method are given in summary.

4. B-SPLINE CURVE APPROXIMATION BY BB-BC METHOD

Today, computer modeling, computer aided design and computer analysis are seen to take part on the basis of many technological developments. Mathematical optimization methods are seen not to be sufficient in finding solution for obtaining objects in the real world especially in modeling and design. There may be many parameters in designing the object desired to be obtained. So, artificial intelligence techniques have been developed for solving these kind of problems. In this paper, approximation of B-spline curves by BB-BC are implemented. The problem aimed

Table 1 The steps of BB-BC method

	Description
Preparation Stage	Specify the population size, the number of iterations and the fitness function to start the algorithm.
Step 1:	Create a population of random consists of N individuals in the search field.
Step 2:	Calculate all candidate solutions (individuals) using their fitness function.
Step 3:	Determine the center of gravity or the most optimal individual as Big Bang point by the help of Eq. (4).
Step 4:	Generate a new population around the center of gravity or the most optimal individual By Eq. (5).
Step 5:	Go to Step 2 until the stopping criterion (number of iteration or error value) is reached.

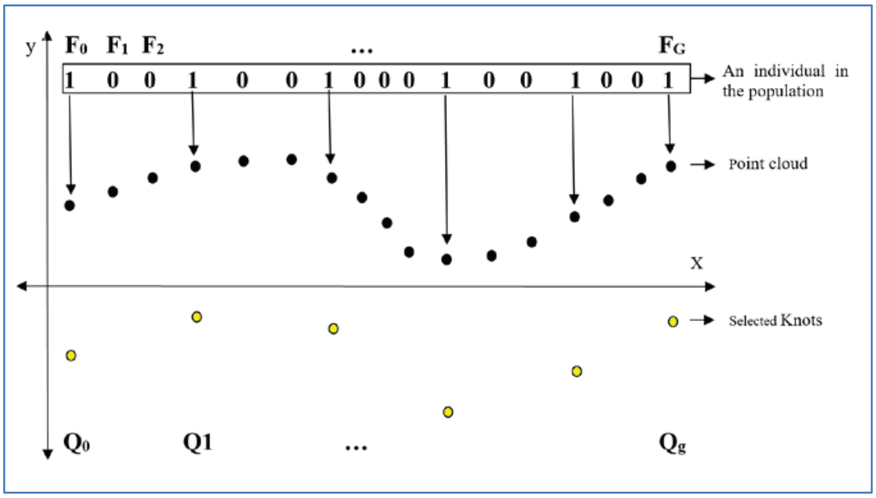

Figure 3 Selection of knots with BB-BC

at being solved is to obtain B-Spline curves to ideally represent a point cloud. Control points, knot vector and parameterization of B-Spline are known to be required for obtain a B-Spline curve. B-spline curve fitting is based on reverse engineering. For a better understanding of the proposed method, the B-spline curve fitting main frame is shown as follows:

1. $F_i, (i = 0, 1, 2, \ldots G)$ is given as a point cloud. Some of these points $(Q_i, (i = 0, 1, 2, \ldots g)\ g < G)$ are selected by BB-BC as knots. As an example in Figure 3 Fconsists of 17 points($G = 16$). Six of these points(Q) are selected for knots by $BB - BC(g = 5)$ in initial. Each individual consists of 0–1. Each bit represents a point in F. The Size of the individual in BB-BC algorithm is equal to the point number of point cloud The points denoted by 1 are designated as knot points. Thus, every iteration the BB-BC algorithm selects different points as knot points.

2. The knot vector is calculated after the knot points are determined. A variety of methods are available for calculation of knot vector. These methods are Uniform, chord and Centripetal methods [30]. According to these methods, suitable parameterization for the curve can be calculated. In this study, calculation of knot vector was conducted by means of Centripetal method. Centripetal knot are calculated as follows.

$$\beta_0 = 0, \qquad \beta_g = 1 \tag{6}$$

$$\beta_i = i - 1 + \frac{\sqrt{|Q_i - Q_{i-1}|}}{\sum_{j=0}^{g} \sqrt{|Q_j - Q_{j-1}|}} \tag{7}$$

$$|Q_i - Q_{i-1}| = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{8}$$

Where β shows Centripetal knot, x_i and y_i are the coordinate value of i^{th} point.

3. After calculated Centripetal knots, estimated B-Spline knots are calculated by the following equation.

$$u_i = (u_{,...,} u_d, u_{d+1, ...} u_g, u_{g+1....} u_{g+d})$$

$$u_{j+d} = \frac{1}{d} \sum_{i=j}^{j+d-1} u_i\, j = 1, \ldots g - d \tag{9}$$

Where d shows B-Spline curve degree. For non-periodic B-Spline curve a knot vector with multiple knots at the beginning and end. Usually the number of identical knots depends on the function degree, thus the first $d + 1$ point forming the knot vector consist of 0 and the last $d + 1$ point consist of 1.

4. $P = Q \times R$ can be accepted as matrix representation for defination of B-Spline curve. R is a matrix produced according to the B-spline blending functions (Eq. 2–3). P represents control points matrix and is calculated as $P = Q \times R^{-1}$.

5. After obtaining P matrix, B-Spline curve approximation can be obtained by the following defination.

$$S(t) = \sum_{i=0}^{n} P_i N_{i,k}(t) \tag{10}$$

Table 2 The performance of BB-BC algorithm in titanium heat data.

	De Boor and Rice (1968)	Jupp's Algorithm (1978)	Yuan Yuan et al. (2013)	Proposed Algorithm
Number of Knot	5	5	6	7
RMSE	0.01305	0.01227	0.01174	0.0302

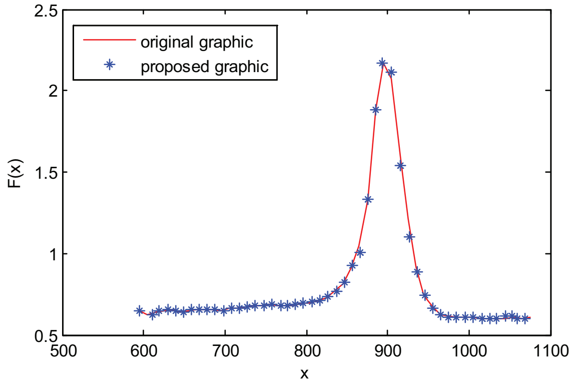

Figure 4 The graph of titanium heat data obtained by BB-BC algorithm.

6. The total error value is calculated between estimated point and real point for all point in F as follows.

$$\text{Error} = \sqrt{\sum_{i=0}^{G}(F_i - S_i)^2} \quad (11)$$

The aim in our proposed method is to obtain the lowest Error by selecting knot points with BB-BC. There exist different error functions in the literature. In this paper, the best ideal solution have been tried to find by means of Mean Squared Error (MSE) and Residual Mean Squared Error (RMSE) error calculation method. MSE is calculated by the following formula.

$$MSE = \frac{1}{G}Error \quad (12)$$

$$RMSE = \frac{1}{G-2}Error^2 \quad (13)$$

In addition to the MSE and RMSE, Akaike Information Criterion (AIC) [31, 32] and Bayesian Information Criterion (BIC) [33] can be calculated. The calculation of these models is calculated according to the following equation.

$$AIC = G * Ln\,(MSE) + 2\,(2 * g + d) \quad (14)$$

$$BIC = G * Ln\,(MSE) + Ln\,(G) * 2 * (2 * g + d) \quad (15)$$

5. EXPERIMENTAL STUDIES

In this study, many commonly used functions in the literature were used. Seven of these functions are presented in this paper. Especially the reason for choosing these functions is that they are frequently used in the curve fitting problems. The size of the initial population of the method used for all functions is between 50 and 150, and the number of iterations is between 10 and 25. This interval was found as the number of interferences in which the lowest error was obtained in the experiments performed. B-Spline degree is given as three (cubic) unless there is any special condition. The test functions used in the study and experimental results are presented below. All the experimental results have been obtained by utilizing *MATLAB R2014a* software package.

5.1 Test Functions 1

The first function is the test function known as titanium heat data in the literature. This test function was used by De Boor and Rice [34]. For knot selection algorithms and was used by Jupp [35] in estimating the optimal internal knots. In addition to, this function was used by Yuan at al. [36] for adaptive knots placement for B-Spline curve fitting. The titanium heat data function consists of 49 points. In Figure 4, original curve points and predicted curve points via BB-BC method are given. In curve estimation, the knots and error have been stored. The results obtained by the proposed algorithm were compared with the result found by De Boor and Rice [34], Jupp [35] and Yuan Yuan et al. [36] are presented in Table 2.

Because only interior knots are given in the studies in the literature, the selected first and last knots were ignored in Table 2 in the process of calculating the number of knots. When examined Table 2, the obtained number of knots is close to the other studies although the error value appears to be a little high compared to other studies.

5.2 Test Function 2

This test function is used by Yuan et al. [36] and Schwetlick and Schütze [37]. Noise was added to the point cloud in their work. The equation of the function is as follow:

$$F\,(x) = \frac{10x}{\left(1 + 100x^2\right)}, \quad x \in [-2, +2] \quad (16)$$

Table 3 The performance of BB-BC algorithm in Function 2.

	Schwetlick and Schutze (1995)	Yuan Yuan et al. (2013)	Proposed Algorithm
Number of Knot	Not reported	6	42
MSE	0.0739568	0.067471	0.012038

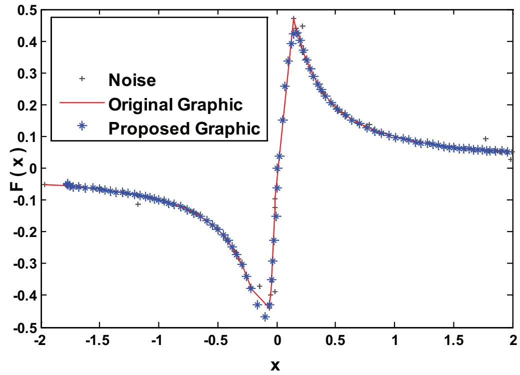

Figure 5 The graph of Function 2 obtained by BB-BC algorithm.

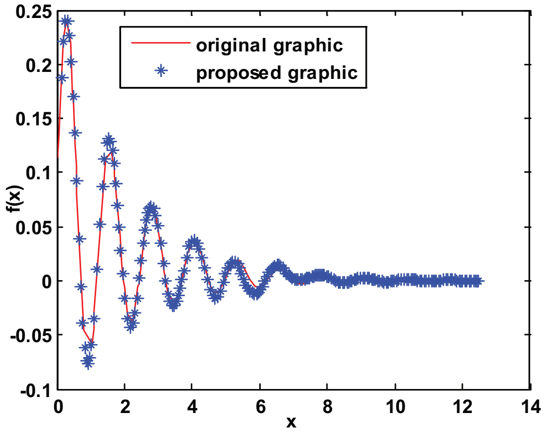

Figure 6 The graph of Function 3 obtained by BB-BC algorithm.

Table 4 The performance of BB-BC algorithm in Function 3.

Valenzuela at all. (2013)					Proposed Algorithm				
Knot	Min MSE	Max MSE	Mean MSE	STD MSA	Knot	Min MSE	Max MSE	Mean MSE	STD MSA
10	0.00241	0.0915	0.0208	0.0395	115	0.00823	0.0110	0.0096	0.00117

Schwetlick and Schutze have studied to solve knot fitting problem by using Gauss-Newton with this function. Besides, Yuan at al. have utilized this function for adaptive B-Spline knot selection. Because the point number is taken as 90 in these studies, the point number is taken as 90 for the proposed algorithm as well. 10% noise value in the range of -0.05 and 0.05 is randomly inserted into the point set obtained by the function. The original curve points and curve points approximated by the BB-BC method are given in Figure 5. The *MSE* error and knot points, which are calculated for the curve, are stored. The results of Schwetlick and Schutze and Yuan et al. and the results obtained by the proposed algorithm are presented in Table 3.

Because the obtained knot numbers have not been reported in the other studies, knot numbers are not presented in the table.

When looked at the error in Table 3, the error acquired by the proposed method is observed to be lower than those of the other studies.

5.3 Test Function 3

Improved Clustering Algorithm was used for this test function in the B-Spline knot fitting problem [38]. The equation of this function is as follow:

$$F(x) = 0.2e^{-0.5x} \sin 5x + 4 \quad x \in 0, 4\pi \qquad (17)$$

Initially, 200 points are used for properly obtaining the original function. The most suitable knot number and the lowest MSE error are searched by running ten times for the detection of knot number and knot location owing to the BB-BC method. The original and obtained curves are presented in Figure 6. The knot numbers, *Min*, *Max*, *MSE* values and standard deviation are given in Table 4 by calculating these values in the experimental study.

When viewed the *Min* and *Max* error in Table 4, it is observed that while the *Max MSE-Min MSE* values have been obtained as

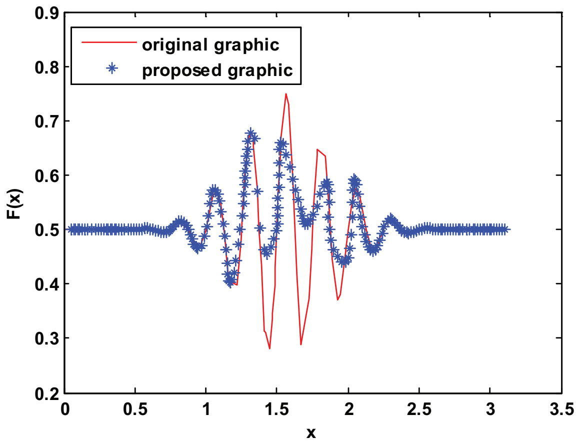

Figure 7 The graph of Function 4 obtained by BB-BC algorithm.

Table 5 The performance of BB-BC algorithm in Function 4.

O. Valenzuela at all. (2013)					Proposed Algorithm				
Knot	Min MSE	Max MSE	Mean MSE	STD MSA	Knot	Min MSE	Max MSE	Mean MSE	STD MSA
10	0.0094	0.2170	0.0649	0.0867	117	0.0105405	0.0157915	0.013098	0.00193

Table 6 The performance of BB-BC algorithm for Function 5.

	Yoshimoto et al. (2003)	Zhao et al (2011)	Akemi Gálvez, Andrés Iglesias (2011)	Proposed Algorithm
Number of iterations	200–300	200	10	10
BIC	1150–1170	Not applicable	1012	2259
Computation time	Tens of seconds	Not reported	0.1–1 s	77s
MSE	Not reported	Not reported	Not reported	1.9

0.08909 in the studies of Valenzuela et al. (2013), this values has been found as 0.00277 in the proposed algorithm. When looking at the standard deviation in Table 4, while the standard deviation is 0.00117 for the proposed approach, Valenzuela at al. (2013) obtained this value as 0.0395. The proposed algorithm has been observed to consistently converge the optimal solution for this difficult function as well.

5.4 Test Function 4

Equation of another test function used in studies of Valenzuela et al. [38] is given as follow:

$$F(x) = 0.5 + 0.5e^{-5(x-\frac{\pi}{2})^2}\sin(4\pi x)\cos(4\pi x) \quad x \in 0, \pi \quad (18)$$

As in Test Function 3, the number of knot for this test function is also selected as 200. The graph of the test function is presented in Figure 7. The knot vector is randomly selected by the program for the curve predicted through the BB-BC approach. *MSE* and standard deviation values obtained in the consequence of experimental studies and these values reported in the literature are presented in Table 5. The proposed algorithm is seen converge to optimal solution for this hard problem. When examining the mean error values obtained from the results, it is seen that while the proposed algorithm has been found as 0.013098 for the mean error value, Valenzuela et al. [38] found as 0.0649 in their study.

5.5 Test Function 5

This problem is used in solution of different problems by Yoshimoto et al. [39], Zhao et al. [40] and Akemi Gálvez, Andrés Iglesias [24]. The definition of the function is given as follow:

$$F(x) = \frac{100}{e^{|10x-5|}} + \frac{(10x-5)^5}{500} \qquad x \in 0, 1 \qquad (19)$$

The number of point is taken as 201 for this problem. The graph acquired by the BB-BC is presented in Figure 8. The number of iteration is taken as 10 in the experimental studies. Although MSE values are not report in other studies, MSE values were calculated in this study. The BIC results and the calculation time are based on the comparison. In Table 6, the performance of the developed algorithm is summarized.

When analyzed Table 6, the results acquired by Gálvez and Iglesias could not be achieved for this problem. However, the approximation curve is obtained quite close to the original curve in terms of figure. The MSE value is gained as 1.9.

5.6 Other Test Functions

These functions are used by Li et al. [41], Yoshimoto et al. [18], Sarfraz and Raza [19] Yoshimoto et al. [39] , Zhao et al. [40] and Gálvez, Iglesias [24]. The equations of the function are presented as follow:

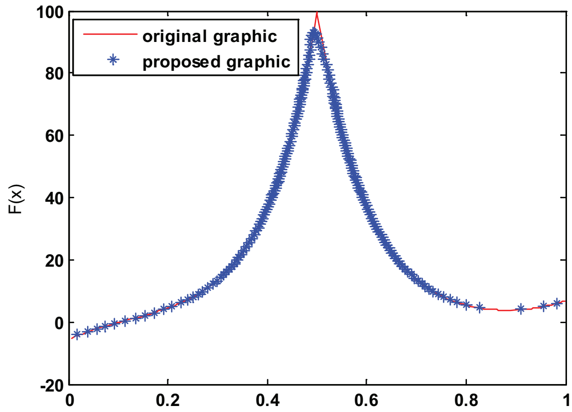

Figure 8 The graph of Function 5 by BB-BC algorithm.

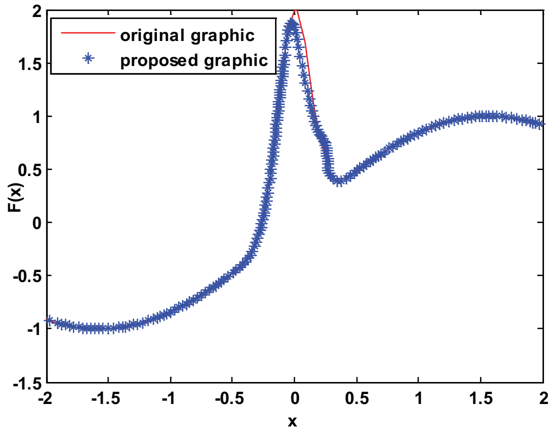

Figure 9 The graph of Function 6 obtained by the BB-BC algorithm.

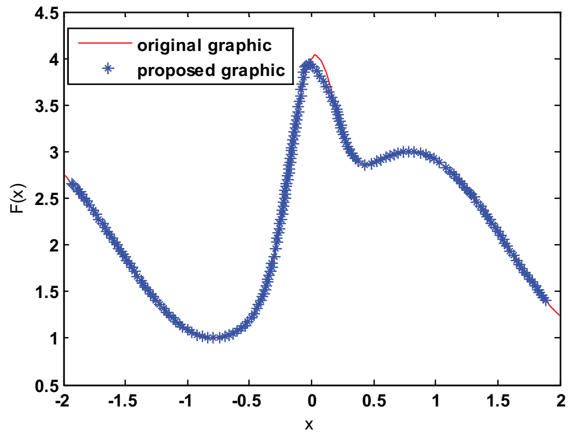

Figure 10 The graph of Function 7 obtained by the BB-BC algorithm.

$$F(x) = \sin(x) + 2e^{-30x^2} \quad x \in -2, 2 \tag{20}$$

$$F(x) = \sin(2x) + 2e^{-16x^2} + 2 \quad x \in -2, 2 \tag{21}$$

The points number are 201 for these functions and the curves are given in Figure 9 and 10 respectively. In the experimental study, the iteration numbers, the *BIC* values, the computational times and *MSE* errors are calculated. The performance of the proposed algorithm and the results of the other approaches are presented in Table 7 and 8, respectively.

When examining Table 7 and 8, it is seen that though the *BIC* values is high, the computational time is reasonable. In addition, it is observed that the proposed approach has been reached to the optimal solutions in short periods such as 10 iterations in terms of especially iteration number. However, when viewed at Figure 9 and 10, the obtained curve has been seen to a large extent to seem to the original curve except for sharp winding.

6. CONCLUSION

In this paper, within the scope of the B-Spline curve approximation, utilization of Big Bang-Big Crunch algorithm (BB-BC)

Table 7 The performance of the BB-BC algorithm for Function 6.

	Yoshimoto et al. [18]	Sarfraz and Raza [19]	Yoshimoto et al. [39]	Zhao et al. [40]	Akemi Gálvez and Andrés Iglesias [24]	Proposed Algorithm
Number of iterations	200	120	200–300	200	10	10
BIC	−46	Not reported	−193	Not applicable	−279	1684
Computation time	5–15s	Tens of seconds–minutes	Tens of seconds	Not reported	0.1–1s	98s
MSE	Not reported	Not reported	Not reported	Not reported	Not reported	0.0411

Table 8 The performance of the BB-BC algorithm for Function 7.

	Yoshimoto et al. [18]	Sarfraz and Raza [19]	Yoshimoto et al. [39]	Ulker and Arslan [16]	Zhao et al. [40]	Gálvez and Iglesias [24]	Proposed Algorithm
Number of iterations	200	120	200–300	500	200	10	10
BIC	134	Not reported	49	362	Not applicable	−63	1543
Computation time	5–15s	Tens of seconds–minutes	Tens of seconds	Tens of seconds–minutes	Not reported	0.1–1s	388s
MSE	Not reported	Not reported	Not reported	Not reported	Not reported	Not reported	0.0265

which is one of the optimization algorithms has been given together with its results. While reverse engineering has been used in curve approximation, *Centripetal* technique has been used for knot approximation. The BB-BC algorithm has stepped in the process of knot approximation. Finally, the following results have been obtained in the B-Spline curve approximation:

1. By means of the developed method, the knot points have been dynamically determined in the B-Spline curve approximation.

2. It is observed that because the number of knot point has been assigned by the program, this number has been generally at high rate. For the flexibility of the program, the number of nodes is not limited. Some problems may have the best results if the number of nodes is high. On the contrary, there may be better results when the number of nodes is small. As the best results were shared, the number of knots has been high. The number of nodes can be lower in the second best solution and third best solution etc.

3. The studies over seven different sample curves from the test functions which are frequently used in the literature have been carried out, and the reasonable curves have been attained from all the functions.

4. For the B-Spline curve estimation, the convergence speed of the BB-BC algorithm is high and curve has been obtained with much shorter iterations.

REFERENCES

1. Varady, T. and RR.Martin, *Reverse engineering*, J.H. G. Farin, M. Kim(Eds.), Editor. 2002, Handbook of Computer Aided Geometric Design, Elsevier Science.
2. Pottmann, H., et al., *Industrial geometry: recent advances and applications in CAD*. Computer-Aided Design, 2005. **37**(7): p. 751–766.
3. Hoschek. J, a.D.L., *Fundamentals of computer aided geometric design*. 1993.
4. Barnhill, R., *Geometric processing for design and manufacturing*, in *SIAM*. 1992: Philadelphia:.
5. Patrikalakis, N., Maekawa T. Shape, *Shape interrogation for computer aided design and manufacturing.* Heidelberg: Springer Verlag, 2002.
6. De Boor, C., *A Practical Guide to Splines.* Springer Verlag, 2001.
7. Ma, W.Y. and J.P. Kruth, *Parameterization of Randomly Measured Points for Least-Squares Fitting of B-Spline Curves and Surfaces.* Computer-Aided Design, 1995. **27**(9): p. 663–675.
8. Ma, W. and J.P. Kruth, *NURBS curve and surface fitting for reverse engineering.* International Journal of Advanced Manufacturing Technology, 1998. **14**(12): p. 918–927.
9. Echevarria, G., A. Iglesias, and A. Galvez, *Extending neural networks for B-spline surface reconstruction.* Computational Science-Iccs 2002, Pt Ii, Proceedings, 2002. **2330**: p. 305–314.
10. Galvez, A. and A. Iglesias, *Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points.* Information Sciences, 2012. **192**: p. 174–192.
11. Varady, T., R.R. Martin, and J. Cox, *Reverse engineering of geometric models - An introduction.* Computer-Aided Design, 1997. **29**(4): p. 255–268.
12. Piegl, L.A.a.W.T., *The Nurbs Book*. 1997, Heidelberg, Germany Springer Verlag.
13. Galvez, A., A. Iglesias, and J. Puig-Pey, *Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction.* Information Sciences, 2012. **182**(1): p. 56–76.
14. Goldenthal, R. and M. Bercovier, *Spline curve approximation and design by optimal control over the knots.* Computing, 2004. **72**(1–2): p. 53–64.
15. Farin, G., *Curves and Surfaces for CAGD*, in *5th ed*, M. Kaufmann, Editor. 2002.: SanFrancisco.

16. Ulker, E. and A. Arslan, *Automatic knot adjustment using an artificial immune system for B-spline curve approximation.* Information Sciences, 2009. **179**(10): p. 1483–1494.
17. Ulker, E., *B-Spline curve approximation using Pareto envelope based selection algorithm-PESA*, in *Int. J. Comput. Commun. Eng.* 2013, 2. p. 60–63.
18. Yoshimoto, F., M. Moriyama, and T. Harada, *Automatic knot placement by a genetic algorithm for data fitting with a spline.* Shape Modeling International '99 - International Conference on Shape Modeling and Applications, Proceedings, 1999: p. 162–169.
19. Sarfraz, M., and S.A. Raza. , *Capturing outline of fonts using genetic algorithms and splines*, in *Fifth International Conference on Information Visualization IV*. 2001, IEEE Computer Society Press. p. 738–743.
20. Kumar, G.S., P.K. Kalra, and S.G. Dhande, *Parameter optimization for B-spline curve fitting using genetic algorithms.* Cec: 2003 Congress on Evolutionary Computation, Vols 1–4, Proceedings, 2003: p. 1871–1878.
21. Pittman, J., *Adaptive splines and genetic algorithms.* Journal of Computational and Graphical Statistics, 2002. **11**(3): p. 615–638.
22. Sarfraz, M. and S.A. Raza, *Visualization of data using genetic algorithm.* Soft Computing and Industry, 2002: p. 535–544.
23. Galvez, A., et al., *Elitist clonal selection algorithm for optimal choice of free knots in B-spline data fitting.* Applied Soft Computing, 2015. **26**: p. 90–106.
24. Galvez, A. and A. Iglesias, *Efficient particle swarm optimization approach for data fitting with free knot B-splines.* Computer-Aided Design, 2011. **43**(12): p. 1683–1692.
25. Erol, O.K. and I. Eksin, *A new optimization method: Big Bang Big Crunch.* Advances in Engineering Software, 2006. **37**(2): p. 106–111.
26. De Boor, C., *A practical guide to splines.* springer, 1978.
27. Sarıöz, E., *Course Notes of Computer Aided Ship Design and Production*. 2005: Istanbul Technical University.
28. Cox, M.G., *The numerical evaluation of B-splines.* Journal of Mathematic Application, 1972. **10**: p. 134–149.
29. De Boor, C., *On calculating with B-splines.* Journal of Aproximate Theory, 1972. **10**: p. 50–62.
30. Lee, E.T.Y., *Choosing Nodes in Parametric Curve Interpolation.* Computer-Aided Design, 1989. **21**(6): p. 363–370.
31. Akaike, H. *Information theory and an extension of the maximum likelihood principle*. in *Second international symposium on information theory*. 1973. Budapest: Akademiai Kiado.
32. Akaike, H., *A new look at the statistical model identification.* , in *IEEE Transactions on Automatic Control* 1974. p. 19(6):716–23.
33. Schwarz, G., *Estimating the dimension of a model.* Annals of Statistics, 1978. **6**(2): p. 461–464.
34. De Boor, C.R., J. R., *Least Squares Cubic Spline Approximation Variable Knots.* Computer Science Technical Reports, Purdue University, 1968.
35. Jupp, D.L.B., *Approximation to Data by Splines with Free Knots.* SIAM Journal on Numerical Analysis, 1978: p. 328–343
36. Yuan, Y., N. Chen, and S.Y. Zhou, *Adaptive B-spline knot selection using multi-resolution basis set.* Iie Transactions, 2013. **45**(12): p. 1263–1277.
37. Schwetlick, H. and T. Schutze, *Least-Squares Approximation by Splines with Free Knots.* Bit, 1995. **35**(3): p. 361–384.
38. Valenzuela, O., et al., *Automatic Knot Adjustment For B-Spline Smoothing Approximation Using Improved Clustering Algorithm.* 2013 Ieee International Conference on Fuzzy Systems (Fuzz - Ieee 2013), 2013.
39. Yoshimoto, F., T. Harada, and Y. Yoshimoto, *Data fitting with a spline using a real-coded genetic algorithm.* Computer-Aided Design, 2003. **35**(8): p. 751–760.
40. Zhao, X.Y., et al., *Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation.* Computer-Aided Design, 2011. **43**(6): p. 598–604.
41. Li, W.S., et al., *Adaptive knot placement in B-spline curve approximation.* Computer-Aided Design, 2005. **37**(8): p. 791–797.