International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

B-Spline Curve Fitting with Intelligent Water Drops (IWD)

Kübra Uyar*[1], Erkan Ülker[2],Ahmet Arslan[3]

Accepted 3rd September 2016

Abstract:The use of B-spline curves has spreaded too many fields such as computer aided design (CAD), data visualization, surface modelling, signal processing and statistics. The flexible and powerful mathematical properties of B-spline are the cause of being one of the most preferred curve in literature. They can represent a large variety of shapes efficiently. The curve behind of the model can be obtained by doing approximation of control points, approximation of knot points or parameterization. It is obvious that the selection of knot points in B-spline curve approximation has an important and considerable effect on the behaviour of final approximation. In addition to this, an unreasonable knot vector may introduce unpredictable and unacceptable shape. Recently, in literature, there has been a considerable attention on the algorithms inspired from natural processes or events to solve optimization problems such as simulated annealing, ant colony optimization, particle swarm optimization, artificial bee colony optimization, and genetic algorithms. This paper implements and analyses a solution to approximate B-spline curves using Intelligent Water Drops (IWD) algorithm. This algorithm is a swarm based optimization algorithm inspired from the processes that happen in the natural river systems. The algorithm is based on the actions and reactions that take place between water drops in the river and the changes that happen in the environment. Some basic properties of natural water drops are adopted in the algorithm here to solve B-spline curve fitting problem. Optimal knots are selected through IWD algorithm. The proposed algorithm convergences optimal solutions and finds good and promising results.

Keywords:*Intelligent water drops, natural water drops, evolutionary algorithms, B-spline curves, knot points, optimization, reverse engineering.*

1. Introduction

B-spline curves are usually used in computer aided design (CAD), data visualization, surface modeling and many other fields. B-spline curve data fitting is a challenging problem encountered in reverse engineering. However, B-spline curves are the most preferred approximation curve because they are very flexible and have powerful mathematical characteristics. Because of this feature, they can offer a large variety of shapes efficiently. In literature, many approaches and methods have been developed for B-spline curve approximation. Tirandaz et al.[1] studied on curve matching and character recognition using B-spline curves. In this work, dominant points on the borders of the object were calculated by Local Curvature Maximum (LCM) and control points were calculated by least square method. Consequently, similar characters were determined by utilizing data set of sample characters. Valenzuela and Pasadas[2] used simulated annealing method for cubic spline approximation for knot adjustment. Discrete curvature of data points was smoothed by low pass filters. Artificial immune system was used for optimization of the knots. Gálvez and Iglesias[3]calculated positions of the optimal knots using particle swarm optimization algorithm which was one of the most important metaheuristic approach in literature. They highlighted parameters of the algorithm which has an significiant effect on the performance of the curve fitting problem. Finally, Gálvez et al.[4]studied elistist clonal selection algorithm to select optimal free knots. In their study, they focused on artificial immune system for the problem of knot adjustment. They mentioned that adjustment of computation time and parameters were basic limitation of the method but it was a matter of tuning parameters based on metaheuristic technical problems and they emphasized that this problem was inevitable. On the other hand, standart problems solved by IWD algorithm which is well known in literature are such as vehicle routing problem, travelling salesman problem and robot routing problem. For eaxmple, Duan et. al[5] proposed an improved IWD optimization algorithm for solving the air robot path planning problems which is a complicated global optimum problem in various environment. A multi-objective IWD algorithm [6] was carried out to minimise cost of goods sold and time to market in logistic networks. Design and configuration of supply chain and logistics network were determined by proposed algorithm. This study contributed to enhance the current knowledge of expert and intelligent system with IWD algorithm. Selvarani and Sadhasivam[7] tried to find a solution to the problem of optimizing task scheduling in grid environment. In their study IWD algorithm was adopted to improve the performance of task scheduling in grid environment. They compared the performance of IWD and Ant Colony Optimization (ACO) and it was proved that task scheduling using IWD gives better results than ACO. Differently, Dadaneh et.al[8] applied IWD algorithm to tackle the graph coloring problem.Niu et. al [9] carried out the problem of multi-objective job shop scheduling by customizing intelligent water drop algorithm. Agarwal et. al[10] proposed an optimised code coverage which is widely used testing paradigm algorithm with the help of the IWD.

[1] *Selçuk University, Konya– 42250, Turkey*
[2] *Selçuk University, Konya – 42250, Turkey*
[3] *Konya Food and Agriculture University, Konya – 42080, Turkey*
** Corresponding Author:Email: kubrauyar@selcuk.edu.tr*
Note: This paper has been presented at the 3rd International Conference on Advanced Technology & Sciences (ICAT'16) held in Konya (Turkey), September 01-03, 2016.

Their approach uses dynamic parameters for finding all the optimal paths using basic properties of natural water drops and the algorithm guarantees complete code coverage by generating automated test squences.
This study focuses on the swarm based IWD algorithm which is inspired from natural colonizing behaviour of water drops in nature. Optimal knot points for drawing the curve are determined by applying proposed algorithm.
The rest of this paper is organized as follows: In this section, some problem solving approaches are briefly reported with their basic properties. Then, some fundamental concepts about B-spline curves are given in Section 2. In Section 3 and 4, we introduce swarm based IWD algorithm to solve B-spline curve fitting problem. In section 5, the efficiency of the proposed algorithm is illustrated by comparing some other studies. Finally, we summarize and conclude this study in Section 6.

2. B-Spline Curves

B-spline curves can be determined for a collection of $n+1$ control points. The first and the last control points intersect with curve as seen Figure 1.

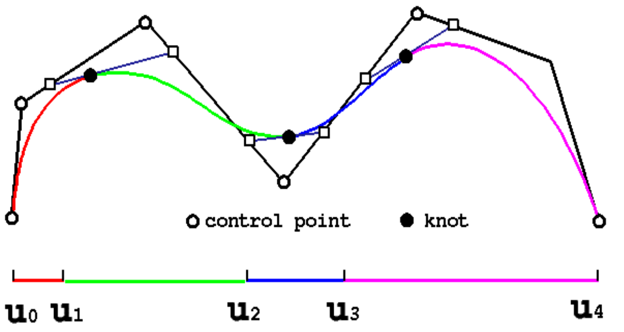


Figure 1. B-spline curve model

The degree of the curve is d and must be satisfy the equation $1 \leq d \leq n$. B-spline curves use blending functions which have local domain areas to overcome restrictions and disadvantages. These functions are equal to zero outside of ownparts. The mathematical definition of the B-spline curve is formulated as follows:

$$P = \sum_{i=0}^{n} (p_i N_{i,d}) \quad (1)$$

where p is the control points, d is the curve degree, $n+1$ is the number of control points and $N_{i,d}$ is the blending function which is calculated as follows:

$$N_{i,1}(u) = \begin{cases} 1, & t_i \leq u \leq t_{i+1} \\ 0, & otherwise \end{cases} \quad (2)$$

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

where t is the elements of knot vector. The main framework of the B-Spline curve drawing is demonstrated as follows[11]:

1. When a point cloud $(F_i \ , i = 0, \ldots, M\_u)$ is given, some points $Q_i, i = 0, \ldots, m \ and \ m \leq M\_u)$ are selected and Centripetal knots are calculated with these points. Here u shows each Centripetal knot:

$$\overline{u_0} = 0 \qquad \overline{u_m} = 1 \quad (4)$$

$$\overline{u_i} = \overline{u_{i-1}} + \frac{\sqrt{|Q_i - Q_{i-1}|}}{\sum_{j=0}^{m} \sqrt{|Q_j - Q_{j-1}|}} \quad (5)$$

$$Q_i - Q_{i-1}| = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (6)$$

2. Approximate B-Spline knots can be found by using calculated Centripetal knots. At this step, the following equation is used:

$$U = [0,0, \ldots, 0, u_{d+1}, \ldots, u_m, 1,1, \ldots, 1] \quad (7)$$

$$u_{j+d} = \frac{1}{d} \sum_{i=j}^{j+d-1} \overline{u_i}, \quad j = 1, \ldots, m-d \quad (8)$$

3. Control points can be obtained by using the equation $Q = PxR$ which is the B-spline curve formula. In this equivalance, in order to obtain P(control points) matrix, at first, R matrix should be obtained by using blending functions. Matrix operations are applied over the equation $Q = PxR$ to left Palone. Accordingly, matrix P is calculated using the equation $P = QxR^{-1}$.
4. As a result, B-spline curve can be drawn.

$$S(u) = \sum_{i=0}^{n} (P_i N_{i,d}(u)) \quad (9)$$

5. The error value is calculated by using data estimated and data obtained . This approach is called Euclidean error sum which is expressed as follows:

$$Error = \sum_{i=1}^{M\_u} |S_i - F_i|^2 \quad (10)$$

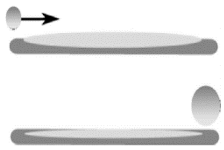
3. Intelligent Water Drops

IWD is a swarm based numerical optimization algorithm inspired from natural colonizing behaviour of natural water drops in river. This algorithm contains a few essential elements of natural water drops and actions and reactions that occur between river's bed and the water drops that flow within. This IWD has two important attributes:

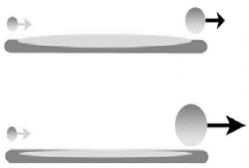
a) The amount of soil it carries now, $soil^{IWD}$
b) The velocity that it is moving now, vel^{IWD}

The values of the both properties may change as the IWD flows in its environment. This environment depends on the problem at hand.
The water drop has three basic behaviours during its movement in the river bed. These behaviours can be summarized as follows:

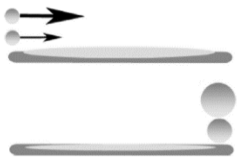
- The water drop that moves from the location i to $j$carries some amount of soil depending on its environment. Therefore, the amount of soil between location i to j diminishes as seen below.



- The water drop which moves from the path that includes less soil on its bed gain more velocity.



- The water drop which have more velocity carrries more soil.



The goal of the problem is to find the best path from the source to the destination.One of the most important mechanism that exists in the behavior of an IWD is that it prefers the path with low soils on its bed. In this point, the probability of the next path to choose inversely proportional to the soils of the available paths. As a result; the lower the soil of the path, the more chance it has for being selected by the IWD. In the following, we will discuss the algorithm rules in details.

Path selecting rule: For each IWD, the probability $(p(i,j;IWD))$ of the choosing next location is given by Equations11, 12 and 13 respectively[12]:

$$p(i,j;IWD) = \frac{f(soil(i,j))}{\sum_{k\notin \mathbf{vc(IWD)}} f(soil(i,k))} \qquad \textbf{(11)}$$

$$f\big(soil(i,j)\big) = \frac{1}{\varepsilon_s + g\big(soil(i,j)\big)} \qquad \textbf{(12)}$$

$$g\big(soil(i,j)\big) = \begin{cases} soil(i,j) & if\ min_l\big(soil(i,l)\big) \geq 0 \\ soil(i,j) \ - min_l\big(soil(i,l) & else \end{cases} \qquad \textbf{(13)}$$

where the set $vc(IWD)$ represents the locations that the IWD should not visit to keep satisfied the constraints of the problem, ε_s is a small positive number to prevent possible zero division, the function min(.) returns the minimum value of its arguments.

Velocity updating rule: For each IWD that moves from current location i to next location j, the velocity of IWD is updated as follows[12]:

$$vel^{IWD}(t+1) = vel^{IWD}(t) + \frac{a_v}{b_v + c_v.soil(i,j)} \qquad \textbf{(14)}$$

where $vel^{(IWD)}(t+1)$ shows the updated velocity of an IWD at the location j, $soil(i,j)$ is the soil on the path joining the current location (i) and the next location (j), a_v, b_v and c_v are constant velocity parameters.

Local soil updating rule: For each IWDs moving, the amount of soil ($soil(i,j)$) and the soil that each IWD carries ($soil^{IWD}$) are updated with the Equations 15, 16, 17 and 18 respectively[12]:

$$soil(i,j) = (1-\rho).soil(i,j) - \rho.\Delta soil(i,j) \qquad \textbf{(15)}$$

$$soil^{IWD} = soil^{IWD} + \Delta soil(i,j \qquad \textbf{(16)}$$

$$\Delta soil(i,j) = \frac{a_s}{b_s + c_s.time(i,j;vel^{IWD})} \qquad \textbf{(17)}$$

$$time(i,j;vel^{IWD}) = \frac{\|c(i)-c(j)\|}{\max(\varepsilon_v, vel^{IWD})} \qquad \textbf{(18)}$$

where $\Delta soil(i,j)$ is the soil which vel^{IWD} removes from the path between i and j and ρ is a small positive number generated from the interval [0,1]. In addition $a_s, b_s$and c_s are constant soil parameters and ε_v is a small positive number to prevent possible zero division (According to original algorithm for finding the best tour we calculate the path with minimum length, however sometimes depending on the problem different measures can be used. For this study, we use error value between original curve and fitted curve instead of Euclidean error.)

Global soil updating rule: At the end of the each iteration, the amount of soil on the edges of the best tour (solution) (T_B) is updated via Equation 19[12]:

$$soil(i,j) = (1-\rho).soil(i,j) + \rho.\frac{2.soil^{IWD}}{N_c(N_c-1)} \quad \forall\,(i,j) \in T_B \qquad \textbf{(19)}$$

where N_c is the number of city that IWD visits on its path. According to obtained best solutions, total best solution is updated.
Taking into account four key rules described above, the steps of implementing standard IWD algorithm can be summarized as follows[12]:

1. The initialization of static parameter values.
   - **a.** Representation of the problem in a graph format.
   - **b.** Setting values for static parameters.
2. The initialization of dynamic parameter values.
3. Distribution of IWDs on the problem's graph.
4. Updating the list of visited cities for each IWD.
5. Following of steps a-d for partial solutions:
   - **a.** Selecting the next city for each IWDs.
   - **b.** Velocity updating of each the IWD which moves from location i to j.
   - **c.** Calculation of soil.

d. Updating of soil.
6. Finding the best of the solutions obtained by IWDs in the respective iteration.
7. Updating of soil amount of the edges which is on the best solution.
8. Updating of best solution.
9. Increasing the number of iteration.
10. Returning the best solution when the termination criterian is satisfied.

There are some improved versions of IWD method in literature. These adaptive IWD, improved IWD, modified IWD and neural IWD are examples.

4. B-Spline Curve Fitting with IWD

This paper focuses on the problem of selecting optimal knots for the best B-spline curve fitting. Data points have been accepted as a cities and all water drops start from the same point. When a water drop reaches predetermined point, it is accepted that this water drop completed its tour. At the end of the algorithm, the cities on the best path are accepted optimal points to fit data.
In this study, some modifications have been made on the original IWD method. The selection of some unnecessary points has been prevented by adding the concept of the radius of curvature. This parameter was determined by semi-autonomously. Furthermore,neighborhood-based local search increases the probability of selecting better ways. Another concept that has been added to the steps of the algorithm is the diversity rate used in local search step. In this way, paths that water drops may follow are gone up. Experimental studies demonstrate that diversity rate prevents hanging out the local minimum. In the following section, some experimental studies were represented to demonstrate the effectiveness of the proposed algorithm.

5. Results and Discussion

In this section, numerous numerical simulations are performed to demonstrate the effectiveness of the proposed optimization algorithm. The scope of B-spline curve fitting, IWD algorithm is coded. On the knot prediction of curve in question, mentioned reverse engineering approaches and optimization process steps have been combined. Some parameters and values used in the problem are given in the following table:

Table 1. Algorithm parameters (*:all values in the range used in simulations)

Parameter	Value
Number of knot	> 6
Curve degree	3
Number of iteration	< 30
Number of IWD	[3,10]*
Tolerance error	0,0001
The number of neighbors	[1,10]*
Diversity rate	[10,100]*

Static parameter values used in the proposed algorithm for the solution of B-spline curve fitting problem are listed as follows:

Table 2. Static parameters of the proposed algorithm

Parameter	a_s	b_s	c_s	a_v	b_v
Value	1000	0.01	1	1000	0.01

Parameter	c_v	$inSoil$	$inVel$	ρ	ε_s	ε_v
Value	1	1000	100	[0.7-0.9]*	0.01	0.0001

All experiments in this paper have been performed on a 2.4 GHz. Intel Core i7 processor with 12 GB. Of RAM. The source code has been implemented by the authors in the programming language of the popular scientific program Matlab, version R2015a. The reason for selection of Matlab is that it is very suitable tool for this task. Because it is fast, and it provides reliable, well tested routines for efficient matrix manipulations.
For this study we compute two fitness functions: Akaike Information Criterian (AIC) and Bayesian Information Criterian (BIC). AIC and BIC which are used to measure performance are defined as follows:

$$AIC = NLog_e SE + 2(2Nod + m) \quad \textbf{(20)}$$

$$BIC = NLog_e SE + Log_e(N)(2Nod + m) \quad \textbf{(21)}$$

$$SE = \sum_{k=1}^{K^*} (y_k - F(\vec{x_k}))^2 \quad \textbf{(22)}$$

where N is the number of data used in the approximation by cubic spline, Nod is the number of interior knots used for construction of B-spline, m is the order of the spline to be fitted for the given data, and SE is sum of squared error.
The reason why we used AIC and BIC as a fitness function is that they do not use subjective parameters such as error limits and smoothing functions. Therefore, these functions provide simple and straightforward procedure for identifying the best result[3]. According to these functions, lower values denote better fitness values.
Proposed method is compared with the Artifical Immune System (AIS)[13] and Pareto Envelope-based Selection Algorithm (PESA)[14]. In this section, performance of the methods was compared by calculating AIC and BIC. Our algorithm for B-spline curve fitting has been applied to the test function which is shown Figure 2.

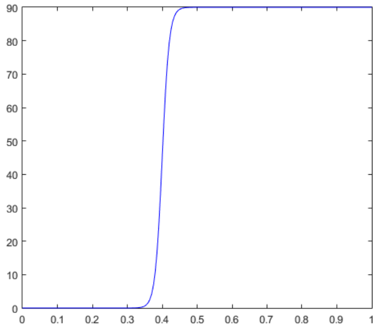


Figure 2. Graphical representation of the test function

Table 3 shows corresponding mathematical definition and associated domain of the test function. The function is evaluated at uniformly distributed values of t in its domain to generate a collection of 201 data points on the interval [0,1].

Table 3. Test function used in this paper: mathematical definition[3]

Equation	Domain
$f_1(t) = \frac{90}{1+e^{-100(t-0.4)}}$	$t\epsilon[0,1]$

Table 4 summarizes AIC and BIC values of the approximationdepending on interior knot number. According to our simulation results, optimal knot number to obtain best approximation is determined as 14. Table 5demonstrates our simulation results of B-spline curve fitting with other two approaches. Here, best results are highlighted in bold. Figure 4 demonstrates the approximated results of the function that we tested for the best solutions visually.

Our comperative results reported in Table 5 confirm that our method outperforms in terms of the fitting errors for the knot selection problem. The number of iterations used to obtain best fitting is nearly a hundred times lower than other methods. By using the AIC and BIC error functions we also provide efficient procedure to determine the optimal number of interior knots.

Table 4. Error functions (AIC, BIC) depending on interior knots.

Number of interior knots	AIC	BIC
$k = 6$	1016.874	1023.481
$k = 7$	1042.236	1055.449
$k = 9$	1031.405	1057.831
$k = 10$	991.9302	1024.963
$k = 11$	867.9182	907.5579
$\boldsymbol{k = 14}$	**704.2195**	**763.679**
$k = 17$	731.878	811.1573

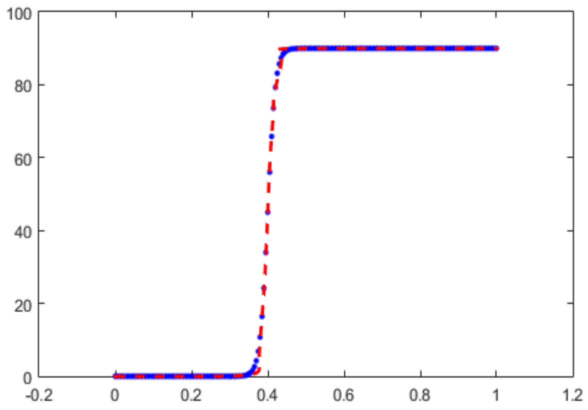


Figure 3. Approximated curve obtained by using 14 interior knots

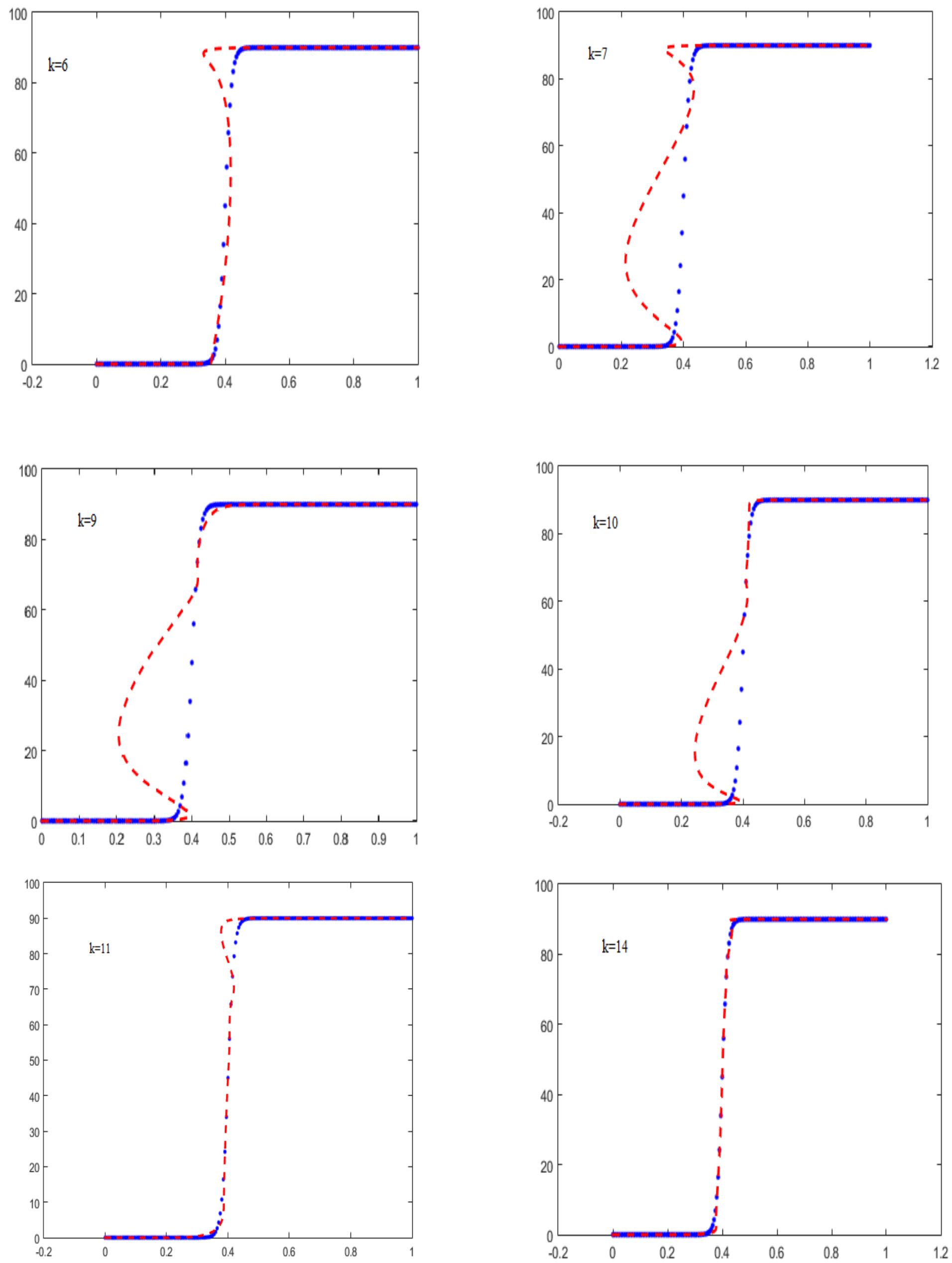


Figure 4. Simulation results depending on the interior knot numbers (Blue dots represents true function and red dashed lines represent approximated curve.)

Table 5. Comparison of methods. Best results highlighted in bold[4].

Methodand reference	Behaviour of the method	Number of iterations	AIC	BIC	Number of interior knots
AIS[13]	Discrete	500	1873	1924	15
PESA[14]	Discrete	500	>3000	>3000	>50
Proposed method (IWD)	**Discrete**	**5**	**704.2195**	**763.679**	**14**

6. Conclusion

We proposed the use of the IWD algorithm for solving curve fitting problem. The points have been accepted as cities and a number of IWDs were placed on the cities randomly. At the end of the algorithm, the best path followed by IWD was determined. In this study, the points on the path were accepted the best points that represent the curve optimally.

Metaheuristic approaches frequently used in optimization problems are very strong but the convergence slows down when the generations approach to optimal solutions. Because of this, the cost of the problem rises up. Proposed method solves this problem in the local search step of the algorithm. In this step, diversity rate is used to prevent hanging out the local minimum. As a result, the proposed algorithm converges fast to optimal solutions and finds good, efficient and satisfactory results.

Acknowledgements

This study was funded by OYP with Project 2015-OYP-075. Kübra UYAR received financial support from TUBITAK BIDEB 2210A: National Graduate Funding Program. In addition, this paper work was summerized from master's thesis study.

This study was presented at the 3rd International Conference on Advanced Technology & Sciences (ICAT'16) which was organized by AT Science Academic Publishing Group, Selcuk University together with other collaborating universities in Konya, Turkey on 1-3 September, 2016. This work was also accepted among the "selected papers".

References

[1] Tirandaz, H., A. Nasrabadi, and J. Haddadnia, *Curve Matching and Character Recognition by Using B-Spline Curves.* IACSIT International Journal of Engineering and Technology, 2011. **3**(2).

[2] Valenzuela, O. and M. Pasadas. *Using Simulated Annealing for knot placement for cubic spline approximation*. in *International Conference on Mathematical Models for Engineering Science - Proceedings*. 2010.

[3] 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.

[4] 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.

[5] Duan, H., S. Liu, and X. Lei. *Air robot path planning based on intelligent water drops optimization*. in *Proceedings of the International Joint Conference on Neural Networks*. 2008.

[6] Moncayo-Martínez, L.A. and E. Mastrocinque, *A multi-objective intelligent water drop algorithm to minimise cost Of goods sold and time to market in logistics networks.* Expert Systems with Applications, 2016. **64**: p. 455-466.

[7] Selvarani, S. and G. Sadhasivam, *An intelligent water drop algorithm for optimizing task scheduling in grid environment.* International Arab Journal of Information Technology, 2016. **13**(6).

[8] Dadaneh, B.Z., H.Y. Markid, and A. Zakerolhosseini, *Graph coloring using intelligent water drops algorithm.* Iranian Conference on Electrical Engineering, 2015: p. 595-600.

[9] Niu, S.H., S.K. Ong, and A.Y.C. Nee, *An improved intelligent water drops algorithm for solving multi-objective job shop scheduling.* Engineering Applications of Artificial Intelligence, 2013. **26**(10): p. 2431-2442.

[10] Agarwal, K., M. Goyal, and P.R. Srivastava, *Code coverage using intelligent water drop (IWD).* International Journal of Bio-Inspired Computation, 2012. **4**(6): p. 392-402.

[11] Ulker, E., *Surface Modeling Using Artificial Intelligence Techniques.* Selcuk University– Electronic Electrical Engineering Department, 2007: p. 142.

[12] Shah-Hosseini, H., *Problem solving by intelligent water drops.* 2007 Ieee Congress on Evolutionary Computation, Vols 1-10, Proceedings, 2007: p. 3226-3231.

[13] 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.

[14] Ulker, E., *B-Spline curve approximation using Pareto envelope-based selection algorithm-PESA.* Int. J. Comput. Commun. Eng., 2013. **2**(1): p. 60-63.