Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 375926, 10 pages
http://dx.doi.org/10.1155/2015/375926

Research Article

Polygonal Approximation Using an Artificial Bee Colony Algorithm

Shu-Chien Huang

Department of Computer Science, National Pingtung University, Pingtung City 90003, Taiwan

Correspondence should be addressed to Shu-Chien Huang; schuang@mail.nptu.edu.tw

Received 8 August 2014; Revised 11 December 2014; Accepted 15 December 2014

Academic Editor: Debasish Roy

Copyright © 2015 Shu-Chien Huang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A polygonal approximation method based on the new artificial bee colony (NABC) algorithm is proposed in this paper. In the present method, a solution is represented by a vector, and the objective function is defined as the integral square error between the given curve and its corresponding polygon. The search process, including the employed bee stage, the onlooker bee stage, and the scout bee stage, has been constructed for this specific problem. Most experiments show that the present method when compared with the DE-based method can obtain superior approximation results with less error norm with respect to the original curves.

1. Introduction

Polygonal approximation is a common representation of digital curves [1]. A polygonal representation of a digital curve can significantly reduce the amount of data needed to be processed while at the same time preserving important information about the curve. It can be applied in shape analysis, pattern classification, image understanding, 3D reconstruction, and CAD applications. In recent decades, a number of methods have been proposed for approximation. In general, they can be categorized into three approaches: (1) given the number of breakpoints, m, find the approximation with m breakpoints such that the approximation error is minimized; (2) given an error ε, find the approximation with the minimum number of breakpoints that is distant from the given curve by no more than ε; and (3) dominant point-detection approaches. In this paper, the present method is categorized into the first approach for polygonal approximation.

Teh and Chin [2] indicated that dominant point detection relies on the determination of a local support region rather than the estimation of discrete curvature. B. K. Ray and K. S. Ray [3] developed a technique to determine the polygon of an object curve by simultaneously maximizing the line length and minimizing the approximation error. Pei and Horng [4] proposed a dynamic programming technique for constructing an approximation using circular arcs. Huang and Sun [5] developed a method based on genetic algorithms for the polygonal approximation of plane curves. Ho and Chen [6] solved the polygonal approximation problem using an efficient evolutionary algorithm with a novel orthogonal array crossover, which is an efficient general-purpose algorithm capable of solving large parameter optimization problems. Yin [7] proposed a polygonal approximation method using an ant colony search algorithm. Sarkar [8] formulated the polygonal approximation task as a minimization problem with ISE as the objective function and then used the differential evolution-based method to solve it. Kolesnikov [9] presented a method based on searching for the shortest path in a feasibility graph for the polygonal approximation of digital curves with a minimum number of approximation segments for a given error bound. Wang et al. [10] proposed an approach, integer particle swarm optimization, for polygonal approximation. Their method found a polygon with the minimum number of vertices whose integral square error is less than a given maximum tolerance.

In recent years, researchers have attempted to apply nature-inspired methods to the problem of pattern recognition and image processing. In this paper, the new artificial bee colony (NABC) algorithm [11] is employed for approximation of digital curves with line segments. The present method can output solutions very close to the optimal one in a relatively short time. The remainder of this paper is organized as

follows. In Section 2, the problem is illustrated. In Section 3, the NABC algorithm is described. In Section 4, the NABC algorithm for polygonal approximation is presented. In Section 5, the experimental studies for the comparison of the present method with the DE-based method are shown. Finally, conclusions are given in Section 6.

2. Problem Statement

Given a curve with n clockwise-ordered points, denoting $C = \{p_0, p_1, p_2, \ldots, p_{n-1}\}$, let $\widehat{p_x p_y}$ and $\overline{p_x p_y}$ represent the arc and the chord starting at point p_x and continuing to point p_y in the clockwise direction along the original curve C. The approximation error of line segment $\overline{p_x p_y}$, denoted as $e_{x,y}$, is defined as

$$e_{x,y} = \sum d^2\left(p_i, \overline{p_x p_y}\right), \quad \forall p_i \in \widehat{p_x p_y}, \tag{1}$$

where $d(p_i, \overline{p_x p_y})$ is the perpendicular distance of arc point p_i to the chord $\overline{p_x p_y}$.

The objective of the polygonal approximation problem is to find a polygon, assuming that the set of its breakpoints is $C^* = \{p_{a(1)}, p_{a(2)}, \ldots, p_{a(m)}\}$, where $0 \le a(1) < a(2) < \cdots < a(m) \le n-1$, such that the integral square error between the polygon and the given curve is minimized. The integral square error (ISE) is defined as follows:

$$\mathrm{ISE} = e_{a(1),a(2)} + e_{a(2),a(3)} + \cdots + e_{a(m-1),a(m)} + e_{a(m),a(1)}. \tag{2}$$

Therefore, there exists $C_{(n,m)}$ $(= n!/[m!(n-m)!])$ combinations to consider when searching for the m breakpoints of the given curve. Generally, this problem is not trivial and always turns into a time-consuming iterative search.

3. The New Artificial Bee Colony (NABC) Algorithm

The artificial bee colony (ABC) algorithm [12–14] was proposed by Karaboga for optimizing numerical problems. Karaboga and Ozturk [15] applied it to training feed-forward neural networks to classify nine data sets. Xu and Duan [16] proposed an artificial bee colony algorithm with edge potential function to accomplish the target recognition task for aircraft. Zhang et al. [17] used the ABC algorithm for clustering. Zhu and Kwong [18] proposed the Gbest-guided ABC (GABC) algorithm, which takes advantage of the information of the global best solution to guide the search for new candidate solutions in order to improve the exploitation. A set of numerical benchmark functions were tested, and the results show that the GABC algorithm can outperform the ABC algorithm in most of their experiments. Pan et al. [19] proposed a discrete artificial bee colony (DABC) algorithm to solve the lot-streaming flow shop scheduling problem. Horng [20] proposed the multilevel thresholding algorithm for image segmentation based on the technology of the artificial bee colony algorithm, while Gao et al. [21] proposed a modified ABC algorithm, which is based on each bee searching only around the best solution of the previous iteration in order to improve the exploitation. Draa and Bouaziz [22] used an artificial bee colony algorithm for image contrast enhancement. In their method, the grey levels of the input image are replaced by a new set of grey levels. Talatahari et al. [23] presented a new optimization method based on the artificial bee colony algorithm for solving parameter identification problems. Sarkar et al. [24] presented a perturbed martingale approach to global optimization, and the results show that their method appears to substantively improve upon the performance of the particle swarm optimization method.

The ABC algorithm simulates the intelligent foraging behavior of honey bee swarms. The colony of artificial bees contains three groups of bees: employed bees, onlooker bees, and scout bees. In the ABC algorithm, the first half of the colony consists of employed bees while the second half is constituted of the onlooker bees. A bee associated with a food resource is called an employed bee, while a bee waiting in the dance area to get information concerning food sources is called an onlooker bee. An employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout. The scout bee carries out random searches to discover new sources. The number of food sources, denoted by SN, is equal to the number of employed bees. The position of a food source represents a possible solution to the optimization problem, while the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. Each solution x_i $(i = 1, 2, \ldots, \mathrm{SN})$ is an m-dimensional vector. Here, m is the number of optimization parameters and is also the number of breakpoints of a polygon in this paper.

In [11], an improved version of ABC, called NABC, was proposed. A set of twelve benchmark functions were used in their experiments. The simulation results showed that the NABC algorithm is significantly better or at least comparable to the original ABC algorithm. In the initialization stage, NABC generates a randomly distributed initial population of SN solutions (food source positions). The fitness of all solutions is evaluated. Then, a solution pool is constructed by storing the best $100p\%$ solutions in the current swarm with $p \in (0, 1]$.

In the employed bee stage, the original ABC algorithm generates a candidate solution v_i in the neighborhood of its present solution x_i as follows:

$$v_{i,j} = x_{i,j} + \phi_{ij}\left(x_{i,j} - x_{k,j}\right), \tag{3}$$

where k is a randomly produced index which is different from i, j is a random dimension index selected from the set $\{1, 2, \ldots, m\}$, and ϕ_{ij} is a random number in the range $[-1, 1]$. In [11], the new ABC/best/1 strategy was proposed in the employed bee stage. The NABC algorithm generates a candidate solution v_i as follows:

$$v_{i,j} = x^p_{\mathrm{best},j} + \phi_{ij}\left(x_{r1,j} - x_{r2,j}\right), \tag{4}$$

where $x^p_{\mathrm{best},j}$ is randomly chosen from the solution pool and x_{r1}, x_{r2} are two randomly selected solutions from the current swarm with $i \ne r1 \ne r2$.

In the onlooker bee stage, an artificial onlooker bee chooses a food source depending on the probability value

```
(1) // Initialization stage
    Generate the initial population x_i, i = 1, ..., SN
    trial_i = 0, i = 1, ..., SN
    set cycle to 1
(2) Evaluate the fitness fit(x_i) of the population
Repeat
(3) Update the solution pool
(4) // Employed bee stage
    For each employed bee
     Produce new solution (food source position) v_i by using (4)
     Calculate the fitness value fit(v_i)
     If fit(v_i) > fit(x_i) then x_i is replaced by v_i and trial_i is set to 0 else trial_i = trial_i + 1
    End
(5) Calculate the probability values prob_i by using (5) for the solutions x_i (i = 1, ..., SN)
(6) // Onlooker bee stage
    For each onlooker bee
     Select a solution x_i depending on prob_i
     Produce new solution (food source position) v_i by using (4)
     Calculate the fitness value fit(v_i)
     If fit(v_i) > fit(x_i) then x_i is replaced by v_i and trial_i is set to 0 else trial_i = trial_i + 1
    End
(7) Memorize the best solution so far
(8) // Scout bee stage
     If there is an abandoned solution (trial_i > limit) for the scout then replace it with
     a new solution which will be randomly produced
     cycle = cycle + 1
Until cycle ≦ MCN
```

Algorithm 1

associated with that food source, prob_i, calculated by the following expression:

$$\text{prob}_i = 0.9 \times \frac{\text{fit}(x_i)}{\max\{\text{fit}(x_1), \text{fit}(x_2), \ldots, \text{fit}(x_{\text{SN}})\}} + 0.1, \quad (5)$$

where $\text{fit}(x_i)$ is the fitness value of the solution x_i. In the scout bee stage, if a solution x_i is abandoned, then the scout bee discovers a new food source to replace x_i.

The difference between the NABC algorithm and the original ABC algorithm is that the NABC algorithm constructs a solution pool and a candidate solution v_i is generated by using (4) instead of (3). In the NABC approach, bees can search the neighborhood of different best solutions. This can help avoid fast attraction and is helpful for accelerating the convergence speed.

The flowchart of the NABC algorithm is given in Figure 1. The pseudocode of the NABC algorithm is shown in Algorithm 1.

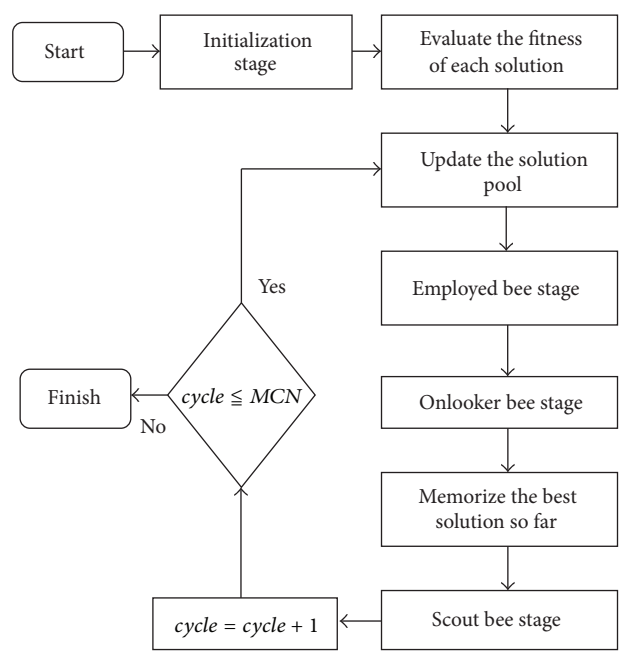
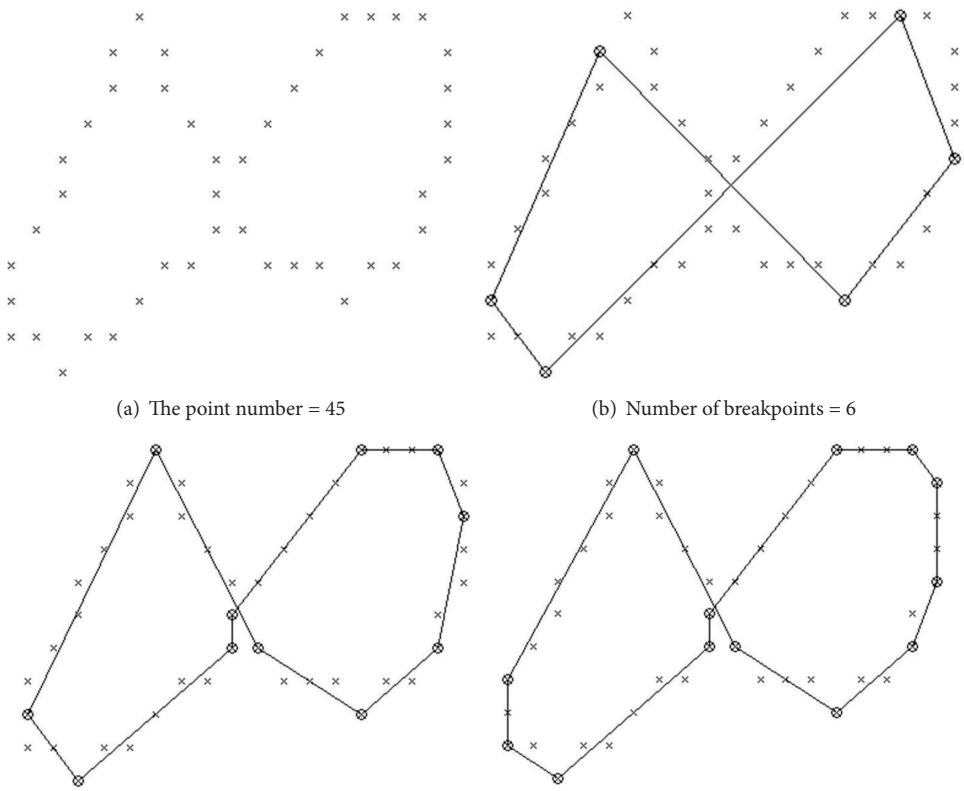


Figure 1: Flow chart of the NABC algorithm.

4. The Proposed NABC Algorithm for Polygonal Approximation

In this section, the representation of solutions and the fitness function are described, and the search process of the NABC algorithm for accomplishing the task of polygonal approximation is also presented.

4.1. Representation of Solutions. In the initialization stage, the SN solution x_i $(i = 1, 2, \ldots, \text{SN})$ is generated, where $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m})$, $x_{i,j}$ is the real number, and the



(a) The point number = 45

(b) Number of breakpoints = 6

(c) Number of breakpoints = 11

(d) Number of breakpoints = 13

Figure 2: The approximated polygons for the figure-of-eight curve obtained by the present method. (a) Original curve and (b)–(d) results using the present method.

$0 \le x_{i,j} < x_{i,j+1} < n$ for all j. Then, the value of *cycle* is set to 0 and the trial number of each solution x_i, trial_i, is equal to 0.

4.2. Fitness Function. The solution $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m})$ corresponds to a polygon and the set of its breakpoints is $\{p_{a(1)}, p_{a(2)}, \ldots, p_{a(m-1)}, p_{a(m)}\}$, where $a(j) = \text{floor}(x_{i,j})$, $j \in \{1, 2, \ldots, m\}$. First, the integral square error (ISE) is computed by using (2). Then, the fitness value can be calculated by the following expression:

$$\text{fit}(x_i) = \frac{1}{(\text{ISE} + 1)}. \tag{6}$$

4.3. The Search Process

4.3.1. The Employed Bee Stage. In this stage, the ith ($i = 1, 2, \ldots, \text{SN}$) employed bee produces a new solution v_i by using (4) and computes the fitness value of the new solution. To ensure that $v_{i,j} < v_{i,j+1}$, the elements of each solution vector are sorted in ascending order of magnitude. If the fitness of the new solution v_i is higher than that of the old solution x_i, the solution x_i is replaced by v_i and trial_i is set to 0; otherwise, the old solution x_i is kept and trial_i is increased by 1.

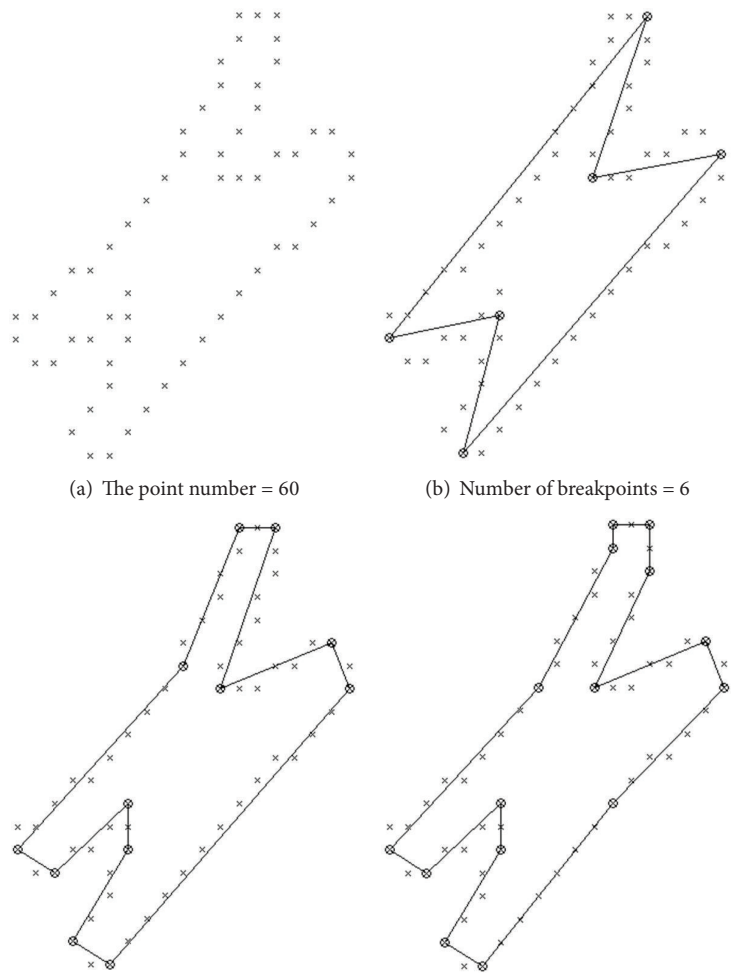
4.3.2. The Onlooker Bee Stage. First, the probability value prob_i is calculated by using (5) for the solution x_i. Each onlooker bee which selects a solution x_i depending on prob_i also produces a new solution v_i by using (4). Then, it applies the greedy selection process between v_i and x_i. As described in Section 4.3.1, if the fitness of the new solution v_i is higher than that of the old solution x_i, the solution x_i is replaced by v_i and trial_i is set to 0; otherwise, the old solution x_i is kept and trial_i is increased by 1.

4.3.3. The Scout Bee Stage. As described above, if the solution x_i is not improved through the employed stage and the onlooker bee stage, the trial_i value of solution x_i will be increased by 1. If the trial_i of solution x_i is more than the parameter "*limit*," the solution x_i is considered to be an abandoned solution; meanwhile, the employed bee will be changed into a scout. The scout randomly produces the new solution to replace x_i. Then, the value of trial_i is set 0, and this scout is changed into an employed bee.

4.3.4. Check the Termination Criterion. If the *cycle* is greater than the maximum cycle number (MCN), then the algorithm is finished and the best solution found so far is output.

5. Experimental Results

The present method is experimentally tested and compared with the DE-based method [8]. The program is coded in C language and run on a PC with a Pentium 4 CPU. The control



(a) The point number = 60 (b) Number of breakpoints = 6

(c) Number of breakpoints = 12 (d) Number of breakpoints = 15

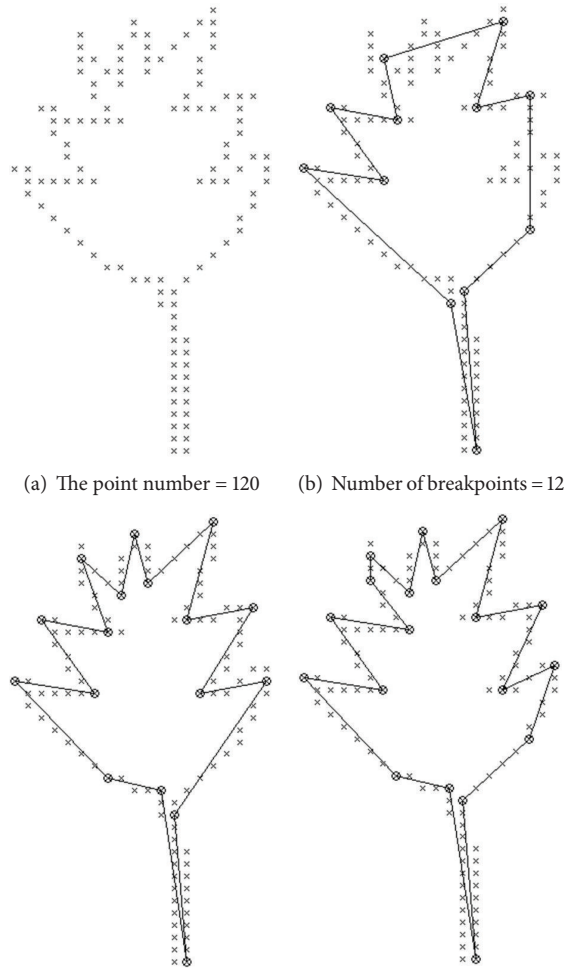
Figure 3: The approximated polygons for the chromosome-shaped curve obtained by the present method. (a) Original curve and (b)–(d) results using the present method.

parameter that is the number of food sources, which is equal to the number of employed bees or onlooker bees (SN), is set to 100, the value of "*limit*" is set to 50, the maximum cycle number (MCN) is set to 600, and the parameter p for constructing a solution pool is set to 0.06 in the experiments.

The DE algorithm is an evolutionary algorithm for solving numerical problems. The NABC algorithm is a novel swarm intelligent algorithm inspired by the foraging behaviors of a honeybee colony. The two methods have attracted the attention of researchers and have been widely used in solving engineering optimization problems. The main difference between the two methods is that they adopt different strategies to produce new solutions. The two methods start with an initial population of SN solutions. In the DE algorithm, new solutions are generated through the operations of mutation, crossover, and selection. In the NABC algorithm, as shown in Figure 1, new solutions are generated after performing a solution pool construction, employed bee stage, onlooker bee stage, and scout bee stage.

In order to verify the performance of the present method, four digital curves, namely, a figure-of-eight curve (see Figure 2(a)), a chromosome-shaped curve (see Figure 3(a)), a leaf-shaped curve (see Figure 4(a)), and a semicircle-shaped curve (see Figure 5(a)), are applied to be tested. The simulation conducted twenty independent runs. The best solutions, average solutions, and the standard deviations (S.D.) of the solutions are reported. The simulation results for the figure-of-eight curve, the chromosome-shaped curve, the leaf-shaped curve, and the semicircle-shaped curve are listed in Tables 1, 2, 3, and 4, respectively. For the average solutions, the present method outperforms the DE-based method. For the best solutions, it is shown that the present method produces better polygonal approximations than the DE-based method in most cases. In addition, the low S.D. reveals that the present method has higher stability than the DE-based method.

In Figures 2–5, Parts (b)–(d) show the best results obtained by the present method with different numbers of breakpoints employed. The computation times for the four digital curves are listed in Table 5. The present method can output solutions immediately because the processing time of all cases is less than 1 second in the experiments.



(a) The point number = 120 (b) Number of breakpoints = 12

(c) Number of breakpoints = 17 (d) Number of breakpoints = 19

Figure 4: The approximated polygons for the leaf-shaped curve obtained by the present method. (a) Original curve and (b)–(d) results using the present method.

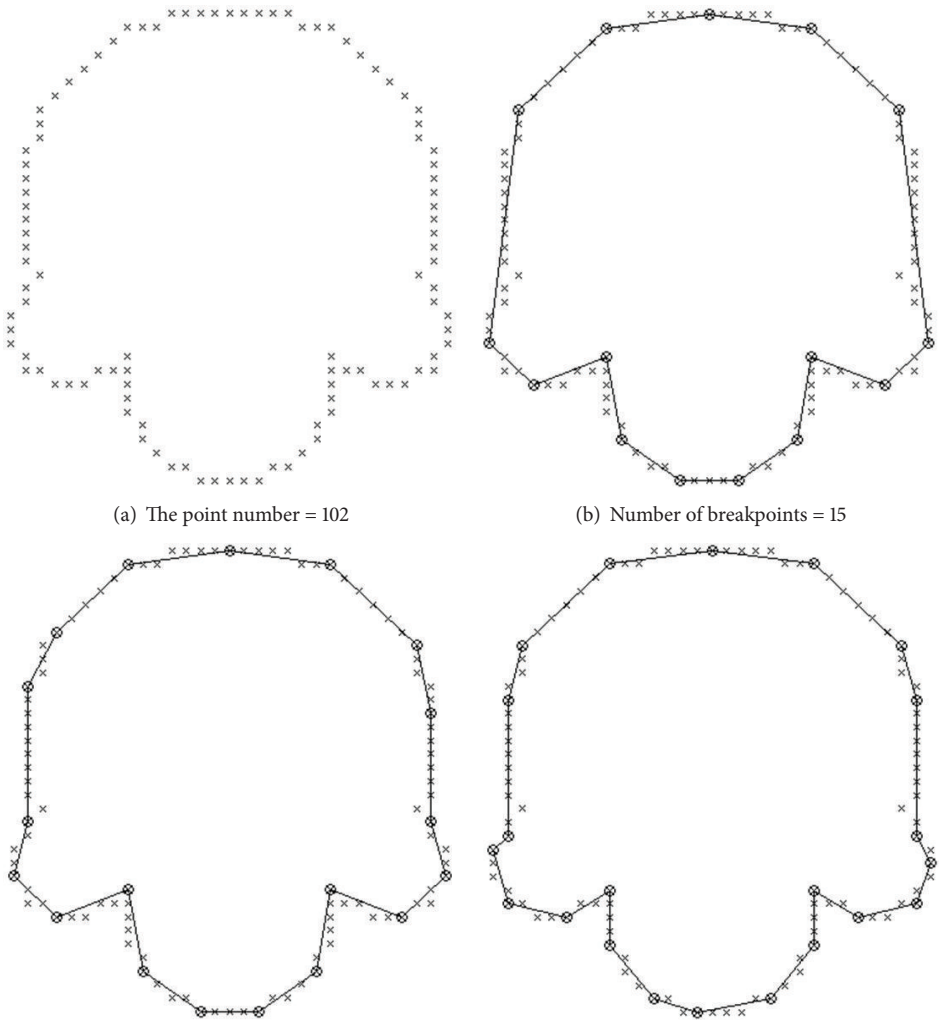
Table 1: Experimental results for the figure-of-eight curve.

Number of breakpoints	DE-based method [8]			Present method		
	ISE (best)	ISE (average)	S.D.	ISE (best)	ISE (average)	S.D.
6	17.494	17.494	0.000	17.494	17.494	0.000
11	2.869	3.235	0.240	2.896	2.896	0.000
13	2.042	2.509	0.277	2.042	2.042	0.000

To demonstrate the applicability of the present method, it is applied to the screwdriver-shaped curve (see Figure 6(a)) and the fish-shaped curve (see Figure 7(a)). The maximum cycle number (MCN) is set to 1,000 for these two relatively larger curves. For the screwdriver-shaped curve, the point number of this curve is 267. Figures 6(b) and 6(c) show the best approximated polygon obtained through the present method with breakpoint numbers of 40 and 45, respectively. The computation times for Figures 6(b) and 6(c) are 2.547 s and 2.656 s, respectively. For the fish-shaped curve, the point number is 537. Figures 7(b) and 7(c) show the best approximated polygon obtained through the present method with breakpoint numbers of 50 and 60, respectively. The computation times for Figures 7(b) and 7(c) are 4.671 s and 4.891 s, respectively. The results of the approximation are tabulated in Table 6. It is seen that the present method can also give good approximation results.

6. Conclusions

A new approach based on the new artificial bee colony (NABC) algorithm has been developed for approximation of digital curves with line segments. Given the number of breakpoints, the algorithm finds an approximation with the



(a) The point number = 102

(b) Number of breakpoints = 15

(c) Number of breakpoints = 19

(d) Number of breakpoints = 22

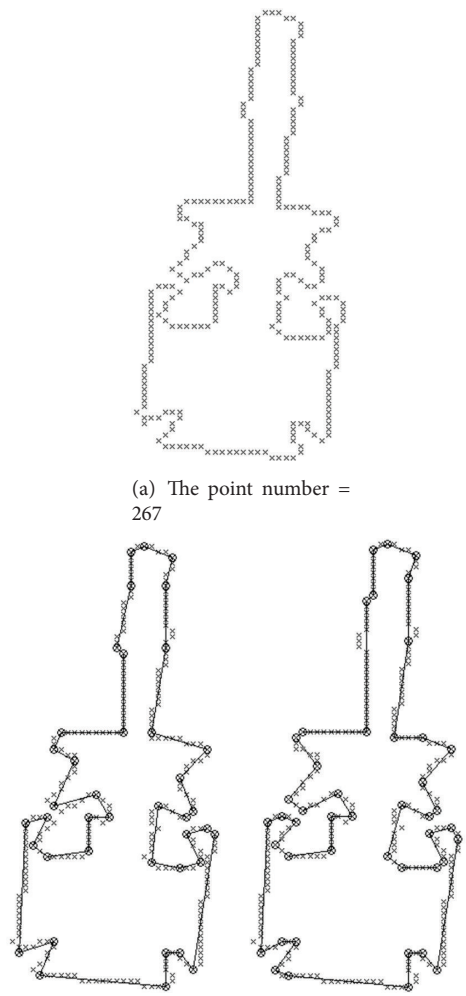
Figure 5: The approximated polygons for the semicircle-shaped curve obtained by the present method. (a) Original curve and (b)–(d) results using the present method.

Table 2: Experimental results for the chromosome-shaped curve.

Number of breakpoints	DE-based method [8]			Present method		
	ISE (best)	ISE (average)	S.D.	ISE (best)	ISE (average)	S.D.
6	23.730	23.730	0.000	23.730	23.730	0.000
12	6.403	6.905	0.342	5.818	5.818	0.000
15	4.232	4.533	0.205	3.796	3.796	0.000

Table 3: Experimental results for the leaf-shaped curve.

Number of breakpoints	DE-based method [8]			Present method		
	ISE (best)	ISE (average)	S.D.	ISE (best)	ISE (average)	S.D.
12	80.834	103.944	11.751	84.519	86.236	1.339
17	42.400	47.354	4.699	21.314	26.219	2.685
19	30.862	36.158	3.281	15.331	15.909	0.336



(a) The point number = 267

(b) Number of breakpoints = 40

(c) Number of breakpoints = 45

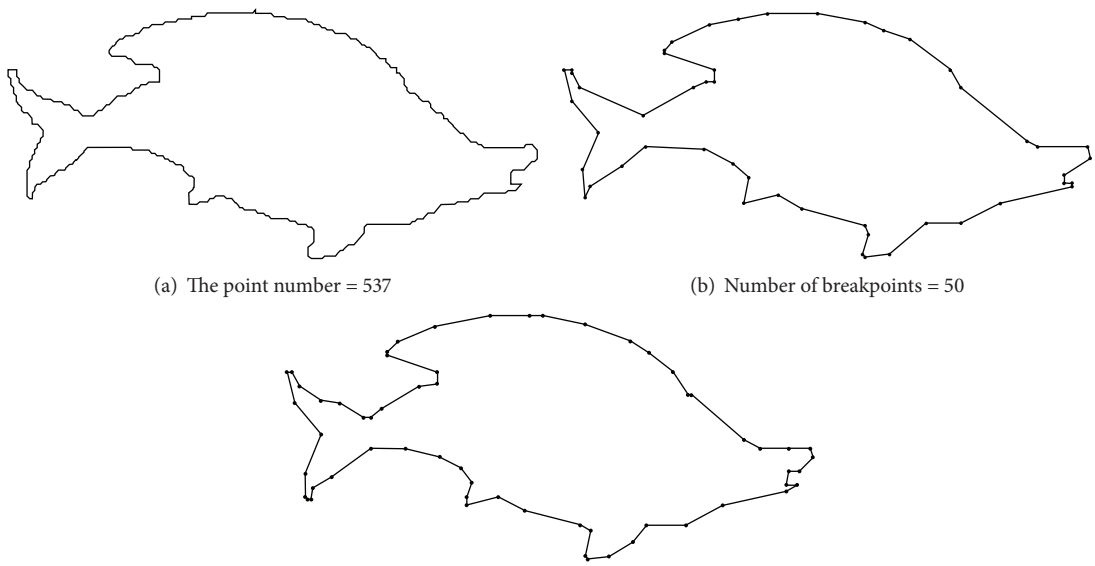
Figure 6: The approximated polygons for the screwdriver-shaped curve obtained by the present method. (a) Original curve and (b)-(c) results using the present method.

Table 4: Experimental results for the semicircle-shaped curve.

Number of breakpoints	DE-based method [8]			Present method		
	ISE (best)	ISE (average)	S.D.	ISE (best)	ISE (average)	S.D.
15	18.531	20.731	0.840	14.399	14.399	0.000
19	12.519	13.377	0.543	10.115	10.439	0.217
22	9.542	10.023	0.594	7.129	7.393	0.096

minimum integral square error. Consequently, the NABC algorithm is a new population-based method and it has the advantages of finding the global or near-global minimum of a multimodal search space, having fast convergence, being simple and flexible, and using few control parameters. The present method simulates the behavior of honey bees foraging to develop the algorithm to select adequate breakpoints for polygonal approximation.

The method has been tested with four different well-known curves. The present method is efficient in dealing with the application because the processing time of all cases is less than 1 second. Compared with the DE-based method, the present method can obtain better approximation results in most cases for both average solutions and best solutions. For the screwdriver-shaped curve and the fish-shaped curve, the present method can also obtain good approximation results. From these experiments, these results have demonstrated the feasibility of the present method for polygonal approximation of digital curves with line segments.



(a) The point number = 537

(b) Number of breakpoints = 50

(c) Number of breakpoints = 60

Figure 7: The approximated polygons for the fish-shaped curve obtained by the present method. (a) Original curve and (b)-(c) results using the present method.

Table 5: Processing time for finding various approximation results.

Approximation	Number of breakpoints	Time (s)
Figure 2(b)	6	0.216
Figure 2(c)	11	0.262
Figure 2(d)	13	0.272
Figure 3(b)	6	0.273
Figure 3(c)	12	0.319
Figure 3(d)	15	0.346
Figure 4(b)	12	0.544
Figure 4(c)	17	0.581
Figure 4(d)	19	0.609
Figure 5(b)	15	0.488
Figure 5(c)	19	0.525
Figure 5(d)	22	0.562

Table 6: Experimental results for the screwdriver-shaped curve and the fish-shaped curve.

Curve	Number of breakpoints	ISE (best)	ISE (average)	S.D.
Screwdriver-shaped curve	40	44.489	47.636	1.562
	45	35.475	38.240	1.786
Fish-shaped curve	50	88.722	93.215	2.657
	60	72.864	76.574	2.323

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] R. C. Gonzalez and R. D. Woods, *Digital Image Processing*, Addison-Wesley, Reading, Mass, USA, 1992.

[2] C.-H. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 859–872, 1989.

[3] B. K. Ray and K. S. Ray, "Determination of optimal polygon from digital curve using L_1 norm," *Pattern Recognition Letters*, vol. 26, no. 4, pp. 505–509, 1993.

[4] S.-C. Pei and J.-H. Horng, "Optimum approximation of digital planar curves using circular arcs," *Pattern Recognition*, vol. 29, no. 3, pp. 383–388, 1996.

[5] S.-C. Huang and Y.-N. Sun, "Polygonal approximation using genetic algorithms," *Pattern Recognition*, vol. 32, no. 8, pp. 1409–1420, 1999.

[6] S.-Y. Ho and Y.-C. Chen, "An efficient evolutionary algorithm for accurate polygonal approximation," *Pattern Recognition*, vol. 34, no. 12, pp. 2305–2317, 2001.

[7] P.-Y. Yin, "Ant colony search algorithms for optimal polygonal approximation of plane curves," *Pattern Recognition*, vol. 36, no. 8, pp. 1783–1797, 2003.

[8] B. Sarkar, "An efficient method for near-optimal polygonal approximation based on differential evolution," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 6, pp. 1267–1281, 2008.

[9] A. Kolesnikov, "ISE-bounded polygonal approximation of digital curves," *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1329–1337, 2012.

[10] B. Wang, D. Brown, X. Zhang, H. Li, Y. Gao, and J. Cao, "Polygonal approximation using integer particle swarm optimization," *Information Sciences*, vol. 278, pp. 311–326, 2014.

[11] Y. Xu, P. Fan, and L. Yuan, "A simple and efficient artificial bee colony algorithm," *Mathematical Problems in Engineering*, vol. 2013, Article ID 526315, 9 pages, 2013.

[12] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[13] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.

[14] D. Karaboga and B. Akay, "A comparative study of artificial Bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[15] D. Karaboga and C. Ozturk, "Neural networks training by artificial bee colony algorithm on pattern classification," *Neural Network World*, vol. 19, no. 3, pp. 279–292, 2009.

[16] C. Xu and H. Duan, "Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1759–1772, 2010.

[17] C. Zhang, D. Ouyang, and J. Ning, "An artificial bee colony approach for clustering," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4761–4767, 2010.

[18] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.

[19] Q.-K. Pan, M. F. Tasgetiren, P. N. Suganthan, and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," *Information Sciences*, vol. 181, no. 12, pp. 2455–2468, 2011.

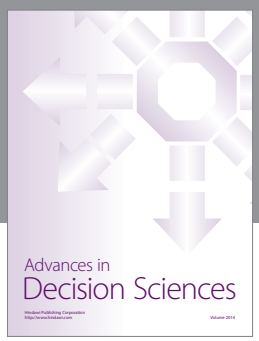
[20] M.-H. Horng, "Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13785–13791, 2011.

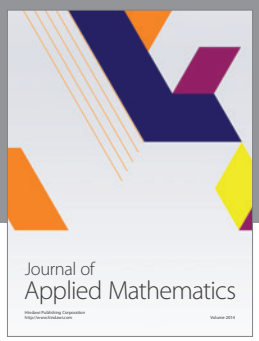
[21] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741–2753, 2012.

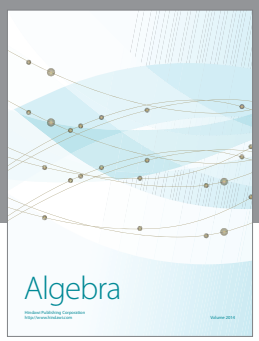
[22] A. Draa and A. Bouaziz, "An artificial bee colony algorithm for image contrast enhancement," *Swarm and Evolutionary Computation*, vol. 16, pp. 69–84, 2014.

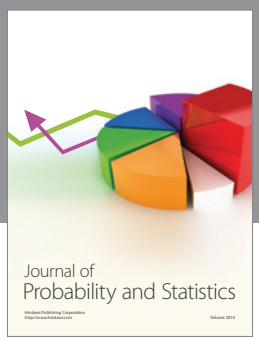
[23] S. Talatahari, H. Mohaggeg, K. Najafi, and A. Manafzadeh, "Solving parameter identification of nonlinear problems by artificial bee colony algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 479197, 6 pages, 2014.

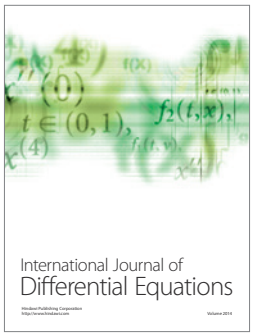
[24] S. Sarkar, D. Roy, and R. M. Vasu, "A perturbed martingale approach to global optimization," *Physics Letters A*, vol. 378, no. 38-39, pp. 2831–2844, 2014.











Hindawi

Submit your manuscripts at
http://www.hindawi.com