Chaotic Fruit Fly Optimization Algorithm

Xiujuan Lei[1,2], Mingyu Du[1], Jin Xu[2], and Ying Tan[2]

[1] School of Computer Science, Shaanxi Normal University, Xi'an, 710062, China
xjlei168@163.com
[2] School of Electronics Engineering and Computer Science, Peking University, Beijing,100871, China (Visiting Scholar)

Abstract. Fruit fly optimization algorithm (FOA) was a novel swarm intelligent algorithm inspired by the food finding behavior of fruit flies. Due to the deficiency of trapping into the local optimum of FOA, a new fruit fly optimization integrated with chaos operation (named CFOA) was proposed in this paper, in which logistic chaos mapping was introduced into the movement of the fruit flies, the optimum was generated by both the best fruit fly and the best fruit fly in chaos. Experiments on single-mode and multi-mode functions show CFOA not only outperforms the basic FOA and other swarm intelligence optimization algorithms in both precision and efficiency, but also has the superb searching ability.

Keywords: Fruit fly optimization, logistic chaos, function optimization.

1 Introduction

Bio-inspired algorithms provide a new perspective for solving complex problems by mimicking the biological behaviors and nature phenomenon, with the characteristics of high robust, low complexities, excellent efficiency and superb performance, and also overcoming the weakness in searching and calculation for finite solutions and high complexity in traditional algorithms. As a significant branch of bio-heuristic research, swarm intelligence is inspired by the behavior of birds, fish, ants and bee colonies and so on in order to search global optima. Besides the characteristics of the meta-heuristic algorithms, swarm intelligent algorithms have the advantages of easy operating and having good parallel architecture. In recent years, novel swarm intelligent optimization algorithms spring up continually and have driven many researches. For example, particle swarm optimization algorithm (PSO) [1], proposed in 1995, imitated the behavior of birds; Bacterial foraging optimization algorithm (BFO) [2], introduced in 2002, simulated the foraging of bacteria; Glowworm swarm optimization algorithm (GSO) [3], developed in 2009, inspired by the glowworms for searching the light. Artificial bee colony algorithm (ABC) has two different mechanisms consisting of foraging behavior [4] and propagating behavior [5]. Swarm intelligent algorithms have been applied in many fields such as function optimization [6, 7], traveling salesman problem [8], path planning [9], image segmentation[10], spam detection [11], data clustering [12], and functional modules detection in protein-protein interaction network [13, 14] *etc*..

Y. Tan et al. (Eds.): ICSI 2014, Part I, LNCS 8794, pp. 74–85, 2014.

Fruit fly optimization algorithm (FOA) [15] is a novel swarm intelligent algorithm proposed by Pan in 2011, mimicking the foraging behavior of fruit flies for searching global optimum. With the outstanding olfactory, fruit flies can perceive the smell in the air even the food source beyond 40 meters and fly toward it. Then, after it gets close to the food location, it can also use its sensitive vision to find food and the company's flocking location, and also fly towards that direction.

FOA has been applied in many field such as neural network parameters optimization [16], [17], financial distress [18], PID controller [19], scheduling [20], and knapsack [21] and so on. Because FOA is a novel algorithm, its application in scientific fields is not very extensive, what's more, its weakness avoid it using for many fields. In order to overcome the weakness, we adopt chaos to the basic FOA.

Chaos [22] is a stochastic phenomenon created in nonlinear and ensured system with the characteristics of randomness, regularity, ergodicity, and sensitive to the initial values, which makes it applied to many scientific research field such as image processing [23], signal processing [24], electric power system [25], optical assessment [26], and neural networks [27] and so on. Due to the features of chaos are corresponding to the features in swarm intelligence, chaos is combined with swarm intelligent algorithms for optimization problems to strengthen the performance of the swarm intelligent algorithms, such as PSO [28, 29], ABC [30], and bat algorithm [31] and so on.

In this paper, logistic chaos operation is introduced into FOA in the movement of the fruit flies, in which the optimum was generated by both the best fruit fly and the best fruit fly in chaos. Besides, the calculation of high dimension distance was adopted to basic FOA to overcome the drawback that it is only used for one dimensional problems.

The rest of this paper is organized as follows. Section 2 introduces the basic concepts and principles including the chaotic mapping, distance metric, and basic FOA. Section 3 provides the chaotic fruit fly optimization algorithm. Results from experiments are described in Section 4. Finally, in section 5, conclusions about the paper and future research are shown.

2 Basic Concepts and Principles

2.1 The Chaotic Mapping

Some statistic distribution is used for enhancing the randomness of algorithms, such as uniform and Gaussian distribution. With the randomness properties, chaos is a superb choice to generate random data. Because of the chaotic characteristics of ergodicity and mixing of chaos, algorithms can potentially carry out iterative search steps at higher speeds than standard stochastic search with standard probability distributions [32]. As a typical chaotic system, logistic mapping is the most representative chaotic mapping with simple operation and well dynamic randomness introduced by May [33] in 1976. Logistic mapping is defined as:

$$z(t+1) = \mu z(t)(1 - z(t)) \quad z \in (0,1) \quad 0 < \mu \le 4 \ . \tag{1}$$

in which c is a control parameter and determines whether chaotic variable z stabilizes at a constant value and t denotes the iteration number. Variable z cannot be assigned to 0, 0.25, 0.75, 0.5 and 1. When $\mu = 4$, the sequence of the logistic mapping is chaotic. In later experiments, $\mu = 4$ is adopted.

2.2 Basic Fruit Fly Optimization Algorithm

Fruit fly optimization algorithm is a novel swarm intelligent optimization algorithm with the property of simple operation. Figure 1 shows the fruit fly and group iterative food searching process of fruit fly [15].

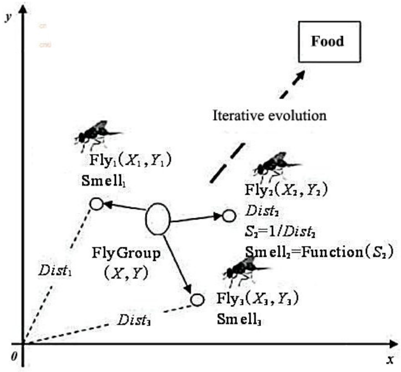

Fig. 1. Illustration of the group iterative food searching of fruit fly

According to the basic FOA [15], several steps are involved as below:

Step 1. Randomly initialize fruit fly swarm location which is shown in Fig.1. The initial location is marked as (*InitX_axis, InitY_axis*).

Step 2. Give the random direction and distance for the search of food using osphresis by an individual fruit fly. New location can be calculated using:

$$\begin{aligned} x(t+1) &= x(t) + randomvalue_x \\ y(t+1) &= y(t) + randomvalue_y \end{aligned} . \tag{2}$$

where *randomvalue* is the movment value in each coordinate. As shown in Fig.1, Fly group move to the new locations like Fly1, Fly2, Fly3, the new locations compose the new fly group and new locations take place of the former fly group locations for calculation.

Step 3. Due to the food location cannot be known, the distance to the origin is thus estimated first, marked as *Dist* calculated by:

$$Dist_i = \sqrt{x_i^2 + y_i^2} . \tag{3}$$

The smell concentration judgment value (S) is calculated, and this value is the reciprocal of *Dist*.

Step 4. Substitute smell concentration judgment value (S) into smell concentration judgment function (or called Fitness function) so as to find the smell concentration ($Smell_i$) of the individual location of the fruit fly.

$$Smell_i = Function(S_i) \ . \tag{4}$$

Step 5. Find out the fruit fly with minimal smell concentration (finding the maximal value marked as [*bestSmell bestIndex*]) among the fruit fly swarm.

Step 6. Keep the best smell concentration value (marked as *Smellbest*) and x, y coordinates, and at this moment, the fruit fly swarm will use vision to fly towards that location.

Step 7. Enter iterative optimization to repeat the implementation of *Steps* 2-5, then judge if the smell concentration is superior to the previous iterative smell concentration, if so, implement *Step* 6.

3 Chaotic Fruit Fly Optimization Algorithm

3.1 Principle of Chaotic Fruit Fly Optimization Algorithm

As we know above, only one variable is referred in the basic FOA, we tried to seek for the algorithm for multiple variables, so comes to chance the distance metrics in tradition. In general, basic FOA is a powerful algorithm in swarm intelligent algorithms with the features of simple calculation and high efficiency.

As a consequence of basic FOA, several local optima are achieved instead of global optima. Aimed at this deficiency, chaotic mapping is adopted to improve the performance of basic FOA escaping from the local optima in this paper. The modified FOA proposed in this paper is marked as chaotic fruit fly optimization algorithm (CFOA, for short).

3.2 Distance Metric

Distance is the metric for two variables in similarity, the larger distance, the more difference is. Several distance metrics [34] are used frequently such as Euclidean distance, Mahattan distance, Minkowski distance and Mahalanobis distance. Euclidean distance is taken advantage to calculate the distance resulting in variable one dimension, while experiments show Mahalanobis distance performs well in high dimension as a consequence of vector variable. As can be seen above, high dimension problems are not involved in the basic FOA. It becomes obviously that Euclidean distance is not appropriate for the high dimension problems; meanwhile the complexity for calculating is a very time-consuming process. Hence, distance metric is redesigned to make the algorithm apply to problems in high dimension and reduce the computational complexity. Due to the unknown location of the food source, we assume that it locates in zero in coordinates, then absolute distance is adopted in each dimension for lessening the calculation complexity and insuring the vector result required. That is to say, smell concentration judgment value (S) is a multidimensional variable for high dimension problems in CFOA instead of single dimension smell concentration judgment value in FOA.

3.3 New Location Update

The new location of the fruit fly group is combined the best location in the basic movement (x_b) with the best chaotic location (x_c) in logistic mapping. The new location is defined as:

$$x(t+1) = x(t) + ax_b(t) + (1-a)x_c(t)r \quad . \tag{5}$$

where t stands for the iteration, r is a random number, a denotes the balance parameter ranging from 0 to 1. If $a = 1$, new location depends on the movement of the fruit group independently; if $a = 0$, new location only depends on the chaotic mapping. In order to acquire the outperformance, random number r is introduced to avoid the absoluteness and increase the possibility of seeking for the global optimum.

3.4 CFOA Process

CFOA includes several steps as below:

Step 1. Initialization. Initialize the locations of the first fruit fly group, where uniform distribution is used for experiments to generate the random locations between the *max* and *min* values in the real models. The maximum iteration t_{max}, group size n, problem dimension d, and the bound values should be given at the beginning.

Step 2. Fly group movement. According to the new location calculation method, use Eq. (5) to get the new location. Let the best location in the basic movement be equal to the best chaotic location ($x_b = x_c$) in the initial stage of the algorithm.

Step 3. Calculation for smell concentration. As discussed above, absolute distance is introduced to calculate the smell concentration judgment value (S). After that, perform the 4th step of basic FOA in section 2.3, using Eq. (4) to achieve the value of smell concentration. Smell is the objective function value as well.

Step 4. Frist selection. Find out the best location (x_b) in fruit group with minimal smell concentration, mark the value of smell ($smell_1$) as the same operation of 5th step of basic FOA in section 2.2

Step 5. Chaotic operation. Let the whole fruit group in logistic mapping. On account of data in logistic mapping ranges from 0 to 1, variables in fruit group should be standardized in order to match the variable z in logistic mapping. Assume variables of fruit group (x) ranging from the low bound (low) to up bound (up), standardized variable (z') defines as:

$$z' = \frac{x - low}{up - low} \qquad x \in [low, up] \ . \tag{6}$$

z' is matching the variable z in logistic mapping, operate the Eq. (1) to transform $z'(n)$ to $z'(n+1)$, where n denoting the iteration n in searching space.

After the chaotic operation, the variable $z'(n+1)$ ranges from 0 to 1, therefore, inverse substitution should be taken to transform $z'(n+1)$ in logistic mapping to data in fruit group. Corresponding to Eq. (6), the substitution is presented below:

$$x' = z'(up - low) + low \quad z' \in (0,1) \ . \tag{7}$$

Step 6. Best selection. Following the chaotic operation, Find out the best x' in fruit group with the minimal smell concentration, mark the best chaotic location (x_c) and value of smell ($smell_2$) similar to the 3th step above. Compare the value of best smell in the basic movement and chaotic movement, mark the smaller as the best smell (*bestsmell*).

Step 7. Enter iterative optimization to judge whether the iteration achieve n_{max} or not, if archives, end up the optimization get rid of loops and output global optima. Otherwise, go to the *Step* 2.

4 Evaluation and Analysis of Experimental Results

Algorithms are tested in Matlab 7.13 and experiments are executed on Pentium dual-core processor 3.10 GHz PC with 4G RAM. 6 Benchmark functions are experimented to testifying the CFOA algorithm compared with PSO [1], BFO [2], GSO [3] and the basic FOA algorithms.

4.1 Benchmark Functions

In the experiments, benchmark functions are used to demonstrate the performance of the algorithm shown in Table 1. Among which, the former 3 functions (*f*1-*f*3) are unimodal and the others (*f*4-*f*6) are multimodal functions.

Table 1. Benchmark functions.

Id	Name	Equation	Domain
*f*1	Sphere	$\sum_{i=1}^{d} x_i^2$	±5.12
*f*2	Tablet	$10^6 x_1^2 + \sum_{i=2}^{d} x_i^2$	±100
*f*3	Quadric	$\sum_{i=1}^{d}(\sum_{j=1}^{i} x_j)^2$	±100
*f*4	Rastrigin	$\sum_{i=1}^{d}(x_i^2 - 10\cos(2\pi x_i) + 10)$	±5.12
*f*5	Ackley	$20 + e - 20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)}$	±32
*f*6	Schaffer	$\frac{\sin^2\sqrt{x_1^2 + x_2^2}}{(1 + 0.001(x_1^2 + x_2^2))^2} + 0.5$	±100

4.2 Parameters Setting

It can be seen from swarm intelligent algorithms such as PSO and GSO, the group size assignment is 50 in general. Here, the group size ranging from 10 to 60 are tested in search of the most suitable value, taking Sphere function in 30 dimensions as an example, which is shown in Table 2. “Convergence” is defined as the iteration where the value of objective function becomes changeless. “Time” denotes the running time.

Table 2. Influences of group size in CFOA

Size	Convergence	Time(s)
10	31	0.0921
20	20	0.1909
30	20	0.3081
40	21	0.3890
50	19	0.3720
60	18	0.4827

From Table 2, we can see that running time increase along with group size. In pace with group size augments, convergent iteration varies slightly. When group size ranged from 10 to 20, running time double increased, but from 20 to 30, running time changes little than double. As noted above, group size should be assigned to 20.

Maximal iteration is set to 1000 in comparison between CFOA and FOA algorithm due to the curves converge approximately about 600 iteration; when comparing with other algorithms in Section 4.4, the maximal iteration is set to 200; other parameters are set in Table 3.

Table 3. Parameters of algorithms

Algorithm	Parameters
PSO	$n = 50, w = 0.8, c_1 = 2, c_2 = 2$
BFO	$n = 20, n_c = 10, n_s = 5, n_r = 2, c_r = 0.025$
GSO	$n = 20, \rho = 0.4, \gamma = 0.6, l_i(0) = 4, n_t = 4, r_d = 50, r_s = 50$

4.3 Comparison between CFOA and FOA

By virtue of FOA without the ability to handle high dimensional problems, the distance metric mentioned in section 3.2 is adopted for high dimensional problems here. We compare the best, mean and worst values of the 150 times running for the benchmark functions in 30 dimensions, along with the convergent searching curves gained by FOA and CFOA in Table 4 and Fig. 2, respectively.

Table 4. Comparison of performances between CFOA and FOA

Function	FOA			CFOA		
	Best	Mean	Worst	Best	Mean	Worst
Sphere	1.0588e-04	1.1214e-04	1.1370e-04	0	0	0
Tablet	2.4690	2.7316	2.9027	0	0	0
Quadric	0.0016	0.0017	0.0018	0	0	0
Rastrigin	0.0211	0.2606	0.0219	0	0	0
Ackley	0.0077	0.0080	0.0082	-8.8818e-16	-8.8818e-16	-8.8818e-16
Schaffer	1.0700e-04	1.1320e-04	1.1535e-04	0	0	0

As can be seen from Table 4, CFOA outperforms the basic FOA in aspects of best, mean and worst values, the results win a perfect victory to the problems, reaching the precise value in real problems model mostly, expect for the Ackley problem. On the other hand, FOA is a stable algorithm which can be referred from the little deviation among the best, mean and worst values in Table 4. However, CFOA is more stable than FOA from the data in Table 4.

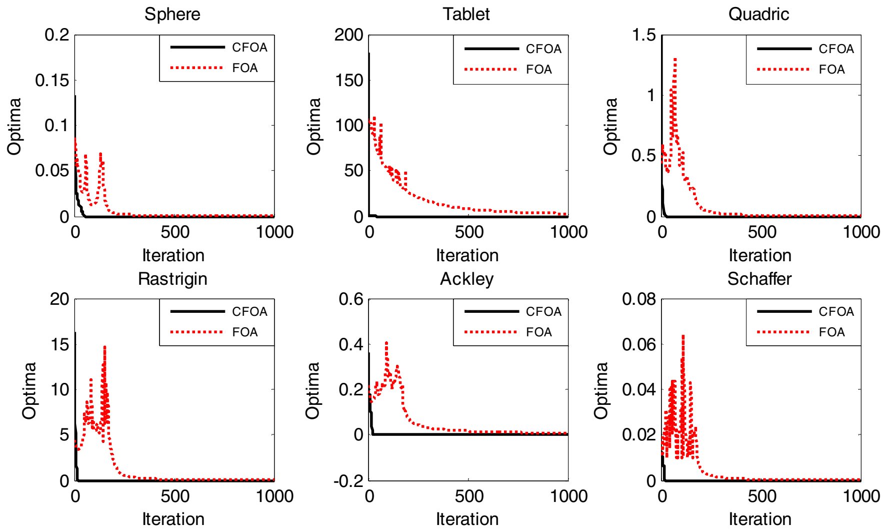

Fig. 2. Searching curves between FOA and CFOA

Basic FOA is not convergent commendably in the problem of Tablet, which can be brought out from Fig. 2. We can see the curves of basic FOA changes sharply and go oscillating in the former iterations, most get smooth in about 300 iterations, and even some diverges in the last such as Tablet. However CFOA converges faster evidently and its curves are smoother than FOA.

4.4 Comparison among CFOA and other Swarm Intelligent Algorithms

As noted above, some typical swarm intelligent optimization algorithms were emerged, in which PSO a well-known algorithm is applied in various fields, after that came BFO, then GSO in recent years. CFOA with PSO, BFO, and GSO are compared to reveal its excellent performance. Maximal iteration is assigned to 200 and d = 30.In addition, ABC cited in [35] is used for comparison. The best, mean and worst values are shown in Table 5. Searching curves of CFOA, PSO, BFO, and GSO are compared show in Fig. 3.

Table 5. Comparison of the best value among CFOA and other algorithms

Function	Algorithm	Best value	Mean value	Worst value	Time(s)
Sphere	PSO	0.7082	4.2435	16.3361	0.1536
	ABC	6.9216e-06	9.71e-06	1.7306e-05	-
	BFO	0.0569	27.9043	34.4376	2.2938
	GSO	15.1463	24.1407	37.9462	0.8207
	CFOA	0	0	0	0.1584
Tablet	PSO	9.2533	47.5864	151.2483	0.1633
	ABC	-	-	-	-
	BFO	0.8019	27.8247	54.2484	2.8789
	GSO	21.5472	26.2276	84.2679	0.7496
	CFOA	0	0	0	0.1858
Quadric	PSO	0.1471	6.8473	18.2545	0.2270
	ABC	-	-	-	-
	BFO	1.4393e-07	35.8423	62.8472	5.4017
	GSO	55.2578	89.1475	107.3562	0.8838
	CFOA	0	0	0	0.2755
Rastrigin	PSO	20.1543	70.5196	135.2792	0.1681
	ABC	9.6741e-04	0.0024	0.0054	-
	BFO	207.5546	222.6783	236.8989	2.7732
	GSO	203.8972	227.2889	278.0422	0.8361
	CFOA	0	0	0	0.1700
Ackley	PSO	0.0924	2.3001	4.1087	0.2427
	ABC	-	-	-	-
	BFO	0.0940	1.4919	5.3649	2.8435
	GSO	4.3467	4.6689	5.5228	0.8678
	CFOA	-8.8818e-16	-8.8818e-16	-8.8818e-16	0.2138
Schaffer	PSO	0.0372	0.0622	0.0782	0.2109
	ABC	0.8701	1.0657	1.2542	-
	BFO	0.0165	0.0372	0.0412	2.4955
	GSO	0.0372	0.0372	0.0372	0.9135
	CFOA	0	0	0	0.1952

From Table 5 we can see that CFOA reaches the best value of problems mostly to be the best in actual, values among the best, mean and worst are equivalent. Other algorithms cannot gain the accurate value in actual and even traps into local optima. Furthermore, although PSO runs faster than the other algorithms, CFOA has the advantage in running time, better than PSO. Values obtained by CFOA, which exceeded far from BFO and GSO, better than PSO as well. As a consequence, CFOA is a superb algorithm with outstanding robustness and wonderful accuracy for the functions above.

We tested PSO, BFO and GSO along with CFOA of performance in convergence and searching abilities. Searching curves of algorithms show in Fig.3, we can see that the convergent iteration begin to converge and the values changing in the iteration period. CFOA reaches the smallest values in Fig.3, what's more, the CFOA curves

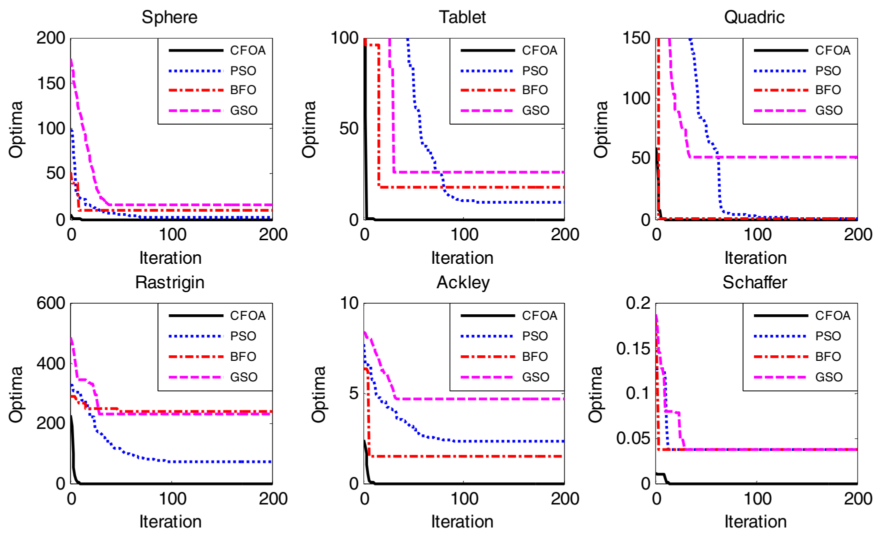

Fig. 3. Searching curves of CFOA and other swarm intelligent algorithms

converges better than the other three algorithms, especially for the functions such as Tablet, Quadric, Rastrigin and Ackley. In addition, the approximately equivalent values are apparently for the Schaffer function.

5 Conclusion and Discussion

Aiming at the deficiencies of trapping into local optimum, converging slowly as well as not suitable for high dimension problems in the basic fruit fly optimization algorithm, chaotic fruit fly optimization algorithm is presented in this paper. In the first place, we modified the distance metric to suit the high dimension problems, absolute distance is adopted here in each dimension to transform the distance into a vector. Secondly, we introduced logistic mapping, the famous typical chaotic mapping to the new algorithm to expand the searching space. Last but not least, new location update is designed to improve the optimum gained by the group. As superior results gained above, CFOA algorithm performs outstanding in both optima searching and running time, not only outperforms the basic FOA, but also other swarm intelligent algorithms. We are intending to apply it to other fields for scientific research in the near future to testify whether it works well.

Acknowledgement. This paper is supported by the National Natural Science Foundation of China (61100164, 61173190), Scientific Research Start-up Foundation for Returned Scholars, Ministry of Education of China ([2012]1707) and the Fundamental Research Funds for the Central Universities, Shaanxi Normal University (GK201402035, GK201302025).

References

1. Kennedy, J., Eberhard, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, USA, pp. 1942–1948 (1995)
2. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine 22, 52–67 (2002)
3. Krishnanand, K.N., Ghose, D.: Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In: IEEE Swarm Intelligence Symposium, Pasadena, California, USA, pp. 84–91 (2005)
4. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department 129(2), 2865–2874 (2005)
5. Hadded, O.B., Afshar, A., Marino, M.A.: Honey-bees mating optimization (HBMO) algorithm. Earth and Environmental Science 20(5), 661–680 (2006)
6. Sun, S.Y., Li, J.W.: A two-swarm cooperative particle swarms optimization. Swarm and Evolutionary Computation 15, 1–18 (2014)
7. Chatzis, S.P., Koukas, S.: Numerical optimization using synergetic swarms of foraging bacterial populations. Expert Systems with Applications 38(12), 15332–15343 (2011)
8. Mavrovouniotis, M., Yang, S.X.: Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors. Applied Soft Computing 13(10), 4023–4037 (2013)
9. Ma, Q.Z., Lei, X.J., Zhang, Q.: Mobile Robot Path Planning with Complex Constraints Based on the Second-order Oscillating Particle Swarm Optimization Algorithm. In: 2009 World Congress on Computer Science and Information Engineering, Los Angeles, USA, vol. 5, pp. 244–248 (2009)
10. Lei, X.J., Fu, A.L.: Two-Dimensional Maximum Entropy Image Segmentation Method Based on Quantum-behaved Particle Swarm Optimization Algorithm. In: Proceedings of the 4rd International Conference on Natural Computation, Jinan, China, vol. 3, pp. 692–696 (2008)
11. Tan, Y.: Particle Swarm Optimizer Algorithms Inspired by Immunity-Clonal Mechanism and Their Application to Spam Detection. International Journal of Swarm Intelligence Research 1(1), 64–86 (2010)
12. Kuo, R.J., Syu, Y.J., Chen, Z.Y., Tien, F.C.: Integration of particle swarm optimization and genetic algorithm for dynamic clustering Original Research Article. Information Sciences 195, 124–140 (2012)
13. Lei, X.J., Tian, J.F., Ge, L., Zhang, A.D.: Clustering and Overlapping Modules Detection in PPI Network Based on IBFO. Proteomics 13(2), 278–290 (2013)
14. Lei, X.J., Wu, S., Ge, L., Zhang, A.D.: The Clustering Model and Algorithm of PPI Network Based on Propagating Mechanism of Artificial Bee Colony. Information Sciences 247, 21–39 (2013)
15. Pan, W.T.: A new evolutionary computation approach: Fruit fly optimization algorithm. In: 2011 Conference of Digital Technology and Innovation Management, Taipei (2011)
16. Li, H.Z., Guo, S., Li, C.J., et al.: A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. Knowledge-Based Systems 37, 378–387 (2013)
17. Lin, S.M.: Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network. Neural Comput. Appl. 22(3-4), 783–791 (2013)

18. Pan, W.T.: A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. Knowledge-Based Systems 26, 69–74 (2012)
19. Wang, S., Yan, B.: Fruit fly optimization algorithm based fractional order fuzzy-PID controller for electronic throttle. Nonlinear Dynamics 73(1-2), 611–619 (2013)
20. Zheng, X.L., Wang, L., Wang, S.Y.: A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem. Knowledge-Based Systems 57, 95–103 (2014)
21. Wang, L., Zheng, X.L., Wang, S.Y.: A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem. Knowledge-Based System 48, 17–23 (2013)
22. Pecora, L., Carroll, T.L.: Synchronization in chaotic system. Phy. Rev. Lett. 64(8), 821–824 (1990)
23. Zhou, Y.C., Bao, L., Chen, C.L.P.: A new 1D chaotic system for image encryption. Signal Processing 97, 172–182 (2014)
24. Kassem, A., Hassan, H.A.H., Harkouss, Y., et al.: Efficient neural chaotic generator for image encryption. Digital Signal Processing 25, 266–274 (2014)
25. Ugur, M., Cekli, S., Uzunoglu, C.P.: Amplitude and frequency detection of power system signals with chaotic distortions using independent component analysis. Electric Power Systems Research 108, 43–49 (2014)
26. Petrauskiene, V., Survila, A., Fedaravicius, A., et al.: Dynamic visual cryptography for optical assessment of chaotic oscillations. Optics & Laser Technology 57, 129–135 (2014)
27. Yang, G., Yi, J.Y.: Delayed chaotic neural network with annealing controlling for maximum clique problem. Neurocomputing 127(15), 114–123 (2014)
28. Wang, J.Z., Zhu, S.L., Zhao, W.G., et al.: Optimal parameters estimation and input subset for grey model based on chaotic particle swarm optimization algorithm. Expert Systems with Applications 38(7), 8151–8158 (2011)
29. Lei, X.-J., Sun, J.-J., Ma, Q.-Z.: Multiple Sequence Alignment Based on Chaotic PSO. In: Cai, Z., Li, Z., Kang, Z., Liu, Y. (eds.) ISICA 2009. CCIS, vol. 51, pp. 351–360. Springer, Heidelberg (2009)
30. Gao, W.F., Liu, S.Y., Jiang, F.: An improved artificial bee colony algorithm for directing orbits of chaotic systems. Applied Mathematics and Computation 218, 3868–3879 (2011)
31. Gandomi, A.H., Yang, X.S.: Chaotic bat algorithm. Journal of Computational Science 5(2), 224–232 (2014)
32. Coelho, L., Mariani, V.C.: Use of chaotic sequences in biologically inspired algorithm for engineering design optimization. Expert Systems with Applications 34, 1905–1913 (2008)
33. May, R.: Simple mathematical models with very complicated dynamics. Nature 261, 459–467 (1976)
34. Zerzucha, P., Walczak, B.: Concept of (dis)similarity in data analysis. TrAC Trends in Analytical Chemistry (38), 116–128 (2012)
35. Lei, X.J., Huang, X., Zhang, A.D.: Improved Artificial Bee Colony Algorithm and Its Application in Data Clustering. In: The IEEE Fifth International Conference on Bio-Inspired Computing, Theories and Applications (BIC-TA 2010), Changsha, China, pp. 514–521 (2010)