

Research Article

Adaptive Grouping Cloud Model Shuffled Frog Leaping Algorithm for Solving Continuous Optimization Problems

Haorui Liu,¹ Fengyan Yi,² and Heli Yang¹

¹*School of Automotive Engineering, Dezhou University, Dezhou 253023, China*

²*Automotive Engineering College, Shandong Jiaotong University, Jinan 250023, China*

Correspondence should be addressed to Fengyan Yi; yi.fengyan@163.com

Received 8 September 2015; Revised 9 November 2015; Accepted 17 November 2015

Academic Editor: Silvia Conforto

Copyright © 2016 Haorui Liu et al. 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.

The shuffled frog leaping algorithm (SFLA) easily falls into local optimum when it solves multi optimum function optimization problem, which impacts the accuracy and convergence speed. Therefore this paper presents grouped SFLA for solving continuous optimization problems combined with the excellent characteristics of cloud model transformation between qualitative and quantitative research. The algorithm divides the definition domain into several groups and gives each group a set of frogs. Frogs of each region search in their memplex, and in the search process the algorithm uses the “elite strategy” to update the location information of existing elite frogs through cloud model algorithm. This method narrows the searching space and it can effectively improve the situation of a local optimum; thus convergence speed and accuracy can be significantly improved. The results of computer simulation confirm this conclusion.

1. Introduction

Shuffled frog leaping algorithm (SFLA) proposed by Eusuff and Lansey [1] has the advantages of being easy to implement and having fast speed and global optimization capability and has been widely used in many areas. Elbeltagi et al. [2] proved that SFLA is superior to genetic algorithms which approximates particle swarm algorithm in success rate and convergence rate when solving some continuous optimization problems. Amiri et al. [3] improved *K*-means clustering method by using SFLA and the result showed that the new method is superior to other clustering methods such as ant colony optimization (ACO) in solution quality and speed. However, SFLA has shortcomings of premature, slow convergence and less precision, so that it is not ideal in solving high-dimensional continuous optimization problem. The main cause of the defect is that, in the later stage of evolution, population diversity declines rapidly, and the method lacks locally refined search capabilities. In order to improve the performance of SFLA optimization algorithms, many scholars [4–36] improved parameter adjustment and memplexes update mode and combined SFLA with other

intelligent algorithms, and so forth, and achieved good optimization results.

The cloud mode [37–40] is proposed to represent the transformation of the uncertainty between the qualitative concept and quantitative value and describes the qualitative knowledge. It is also being widely used in the fields such as the fuzzy evaluation and intelligent control. Characterized by the coexistence of the uncertainty and certainty as well as the stability during the course of knowledge representation, the cloud model has reflected the fundamental principle of the biological evolution in nature.

Some groups of animals in the foraging process have two roles: either looking for prey or sharing current best prey. “Elite frog” finds current best prey, and “variation frog” is involved in foraging and shares “elite frog” results. “Elite frog” may be fooled by illusion which corresponds to local optimal solution in intelligent optimization problems. Variation frog is in charge of discovering prey in the vicinity or a new search track and helps “elite frog” to avoid being confused by some illusion and leaving foraging groups into local optimum. “Elite frog” and “variation frog” guide groups to better search trajectory and find their prey eventually.

In summary, the existing research has made many achievements. However, convergence speed and optimization accuracy need to be improved; particularly when solving multimode function the SFLA easily falls into local optima. This paper presents a continuous optimization algorithm based on grouping SFLA optimization; the goal is to improve the convergence speed and accuracy of the optimization process, especially to improve the optimizing performance for multimode function. Firstly, the cloud model is used to update the existing elite frog. Random uniform design method (RUDM) is used to construct a frog of variation group. Adaptive grouping algorithm is used to determine the content of each group. Optimization is completed through several iterations. A large number of simulation experiments confirm that the algorithm has high speed and optimization precision.

2. Basic Definitions and Proposed Algorithm

2.1. Basic SFLA. Shuffled frog leaping algorithm (SFLA) is a swarm evolution algorithm which imitates frogs exchanging information as the divided memplexes when they are searching for food. The combination of global search and local search in memplex develops the algorithm to approach the optimal solution [26].

In SFLA, memplex is virtually composed of frogs which represent a candidate solution. The population is divided into several memplexes each of which includes a certain amount of frogs. Different memplexes have their own culture as well as the behavior which can be influenced by each other. All memplexes will be mixed up to form a new population after the local search has been implemented for defined times for each of them, which can exchange information globally among all the memplexes [3, 28]. Alternate the global and local search until the information desired convergence is reached which is expressed as certain convergence accuracy or the maximum of iterations. Solution steps of unconstrained function optimization are as follows.

(1) *Initialization.* p candidate solution can be generated in its feasible domain $\Omega \subset \mathbf{R}^D$ for a D -dimensional problem. The p candidate solution can be expressed as initial swarm $S = (X_1, X_2, \dots, X_p)$ in which $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ describes the i th ($1 \leq i \leq p$) candidate solution.

(2) *Memplex Classification.* Partition the population into m memplexes as follows: allocate the frogs to the groups according to the fitness values. The first frog with the highest value moves to the first memplex, the second highest frog moves to the second memplex, and the m th highest frog moves to the last memplex. Then, $m + 1$ th frog moves to the first memplex again. These operations continue until the last frog is allocated to a memplex. Finally, each memplex contains n frogs. Thus, $p = n \times m$.

(3) *Local Search in Memplex.* Let the best frog of a memplex be X_b , let the worst one be X_w , and let the global best one

be X_g . The search in each memplex is to renew X_w of the memplex in the following strategy:

$$D_i = \begin{cases} \min \{ \text{int} [r(X_b - X_w)], D_{\max} \}, & r(X_b - X_w) \geq 0 \\ \max \{ \text{int} [r(X_b - X_w)], -D_{\max} \}, & r(X_b - X_w) \leq 0 \end{cases} \quad (1)$$

$$X_{\text{neww}} = X_w + D_i. \quad (2)$$

In the function above, D_i is renewing value of step size; $\text{int}[x]$ is roundness of x ; r is the random number in the range of $(0, 1)$ whose effective bit depends on problems and simulation environment; D_{\max} is the maximum distance allowing frog moving.

If there exists $X_{\text{neww}} \in \Omega$, after update, then substitute X_w with X_{neww} . Otherwise replace X_b in Function (1) with X_g . The new X'_{neww} is computed by Functions (1) and (2). If $X'_{\text{neww}} \in \Omega$ and $F(X'_{\text{neww}}) < F(X_w)$ exist, then substitute X_w with X'_{neww} ; otherwise a new arbitrary candidate solution is generated to replace X_w . The iteration will not end until the search of the designed search time has been reached.

(4) *Global Information Exchange.* After the local search is completed, all the memplexes are mixed into one swarm which is then operated as the methods shown in steps (2) and (3) until the result meets the termination criterion.

The termination criteria for the algorithm could be satisfied by one of the following three conditions:

- (1) The value of the main objective function F reaches an acceptable and optimum value.
- (2) The number of iterations reaches a predefined value (N); it varies for different number of dimensions in a problem.
- (3) During several consecutive iterations, no progress could be seen in the value of the main objective function F .

The standard SFLA's diagram is shown in Figure 1.

2.2. Cloud Model. Cloud model provides a solution for the problem in combination with both qualitative and quantitative concept, which realizes the conversion between qualitative concept and quantitative concept and shows the fuzziness and randomness of concept expressing. Let U be a universe in accurate value, which corresponds to a qualitative concept A and could be one-, two-, or several-dimensional. There exists a random value satisfying the formula of $y = U_A(x)$ steadily for an arbitrary element in the universe, reflecting the certainty degree of x corresponding to the concept A . The distribution of x in the universe of U is called cloud model or cloud for short. x is denoted as cloud droplet in universe space [37].

Expectation E_x , entropy E_n , and hyperentropy H_e describe the digital characteristics of cloud, and they are

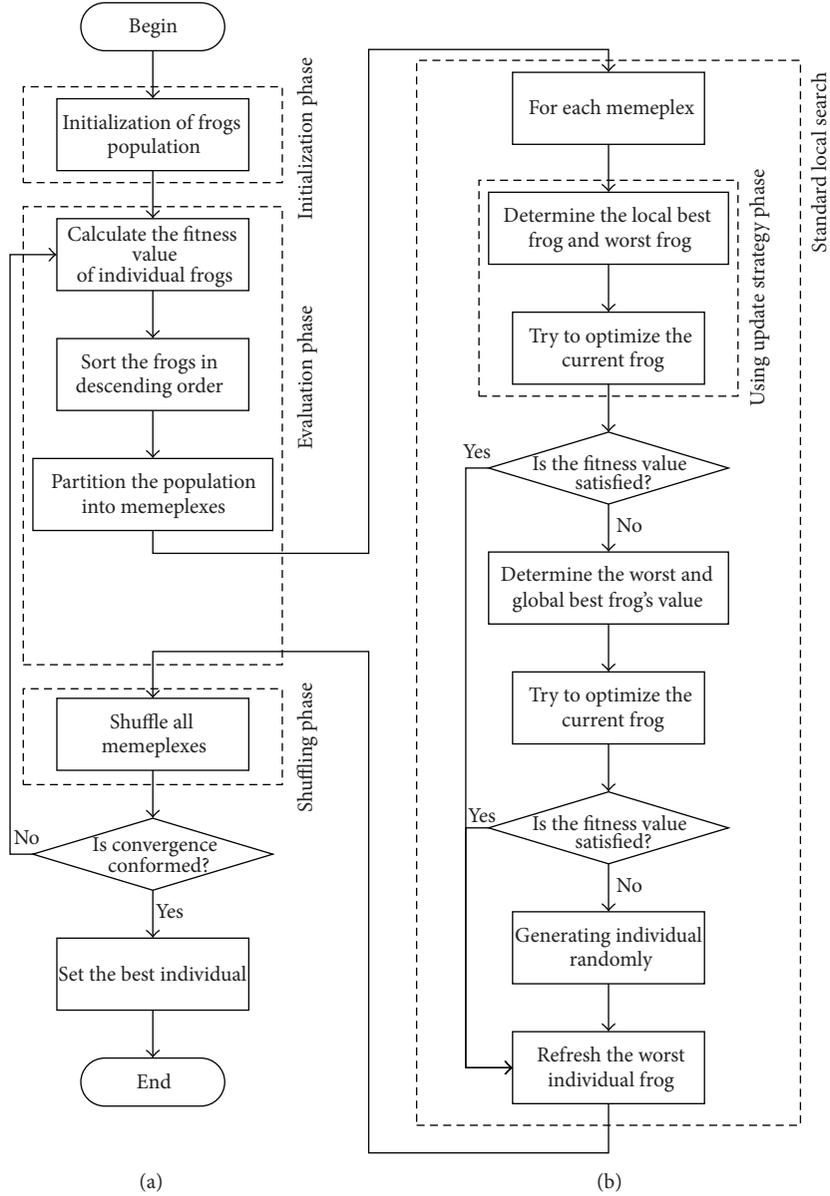


FIGURE 1: (a) The standard SFLA algorithm's main diagram; (b) local search for each memeplex.

illustrated in Figure 2. Expectation E_x is the most typical sample point relating to concept A . Entropy E_n represents the uncertainty of the concept A , revealing both the acceptance range of universe for languages and the randomness of forming cloud droplet. The higher E_n value expresses the wider range of cloud droplet generating and the stronger randomness. Hyperentropy H_e is measurement of E_n . In other words, H_e is the entropy of E_n .

The algorithm of the basic normal cloud generator is operated in following steps.

Firstly, a random number E_n is achieved taking E_n as expectation and H_e as standard deviation.

Secondly, a random number x is achieved taking E_x as expectation and absolute value of E_n as standard deviation. x is the cloud droplet.

Thirdly, y is calculated in the formula as follows:

$$y = e^{-(x-E_x)^2/2(E_n')^2} \quad (3)$$

which is certainty degree of qualitative concept A .

Finally, the first, second, and third step are repeated until cloud droplet number reaches object value.

$3E_n$ rule of basic normal cloud can be described as follows. Among all the cloud droplets that make contribution to qualitative concept A , 99.7% of them are located in the range of $[E_x - 3E_n, E_x + 3E_n]$; thus cloud droplets outside this range only have a tiny possibility of making contribution to qualitative concept A which can be ignored [38–40].

2.3. Adaptive Grouping Cloud Model SFLA. An important reason of the existing algorithms easily falling into local

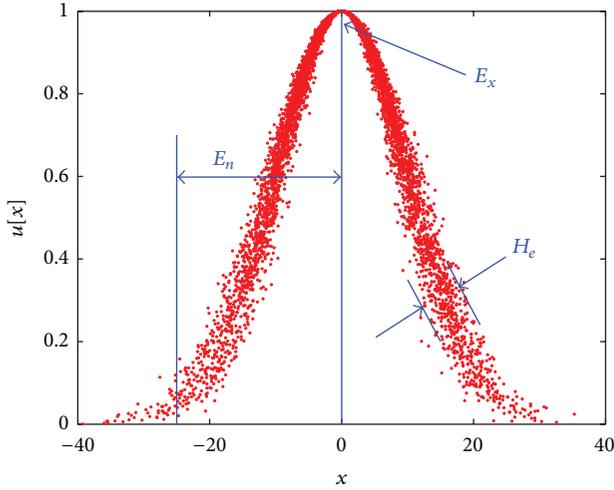


FIGURE 2: Digital features of cloud model.

optimum is as follows: when the optimization process reaches a local bottom or peak, the algorithm lacks mechanisms to jump out of the local optima.

The grouping optimization algorithm makes the other groups work regularly when one group runs into local optimum and that greatly reduces search space so that the optimization process avoids local optima and laid the foundation for the parallel algorithms. The grouping optimization algorithm divides definition region into three groups and then assigns certain memplexes to each group. The adaptive grouping cloud model SFLA's diagram is shown in Figure 3.

2.3.1. Relevant Definition. Group includes some memplexes.

Elite group has bigger fitness and its memplexes may have more excellent next generation by using cloud model.

Normal group uses standard local search.

Variation group guarantees any possibility of evolution, and only a few individuals of that will undergo mutation during evolution.

2.3.2. Elite Group Policy. In the search process, the “elite policy” means that the best individual in each generation is defined as the elite frog and kept to the next cloud model generation and used to update the location best frog and accelerate the convergence rate. When each memplex is updated, the worst individual will be updated, as well as the best individual in memplex. The worst individual in memplex will be updated by traditional frog leaping algorithm and the best individual by the normal cloud model algorithm in Section 2.2. The best individual in memplex will be taken as a normal cloud droplet, and it is used to produce a group of cloud droplets with the same number as the memplex, that is, a group of frogs. In this new memplex, if a frog is better than the best individual within the original memplex, the new best individual will replace the old one; otherwise it remains the same. Specific generation algorithm is as follows: E_x is the center of new individuals generation; the best individual in memplex will be regarded as E_x . Take

modern fitness variance σ^2 as E_n , dynamical changing search scope; take E_n/T as H_e [41], and T can be selected from 3 to 6.

2.3.3. Variation Group Policy. Using random uniform design method [42] construct the initial population. Random uniform design (RUD) is using generally generated vector replacing primitive roots of uniform design generating randomized uniform design points set. The deviation of many randomized uniform design points set is obviously smaller than the original uniform design; thus it has better representativeness. Therefore, in this paper more uniform variant frogs constructed by randomized uniform design method are adopted, replacing the existing best frog.

2.3.4. Adaptive Grouping Policy. Frogs with better adaptive fitness are divided into the elite group, and those with poor adaptive fitness are always divided into variant group. The adaptive fitness grouping method is given as follows.

Step 1. Figure out adaptive value's average fitness N_{avg} of the best individuals in all memplexes.

Step 2. Figure out adaptive value's average fitness N_{l-avg} of the best individuals whose adaptive value is larger than N_{avg} in all memplexes.

Step 3. Figure out adaptive value's average fitness N_{s-avg} of the best individuals whose adaptive value is smaller than N_{avg} in all memplexes.

Step 4. Figure out the number of memplexes whose best individuals' adaptive fitness is larger than N_{l-avg} , N_{elite} , constructing the elite group.

Step 5. Figure out the number of memplexes whose best individuals' adaptive fitness is smaller than N_{s-avg} , $N_{variation}$, constructing the variant group.

Step 6. The rest belong to normal group.

3. Results and Discussion

By taking three-function extreme optimization as example, a contrast is conducted through the particle swarm optimization (PSO), quantum ant colony algorithm (QACA), grouping ant colony algorithm (GACA), cloud model shuffled frog leap algorithm (CM-SFLA), and adaptive grouping chaos cloud model frog leap algorithm (AGCM-SFLA), so as to verify the optimum performance of the algorithm. The parameter setting of frog leap algorithm is as follows: the population size is 300. It is divided into 15 memplexes, each memplex having 20 frogs. The iteration time inside each memplex is 15, and the maximum iterative algebra is 6000. Each function runs for 30 times independently. The error precision is 10^{-10} . Three functions are listed in Table 1 to compare the optimal result, worst result, average result, average time, and variance of the five algorithms: f_1 is a two-dimensional complex function with numerous minimum

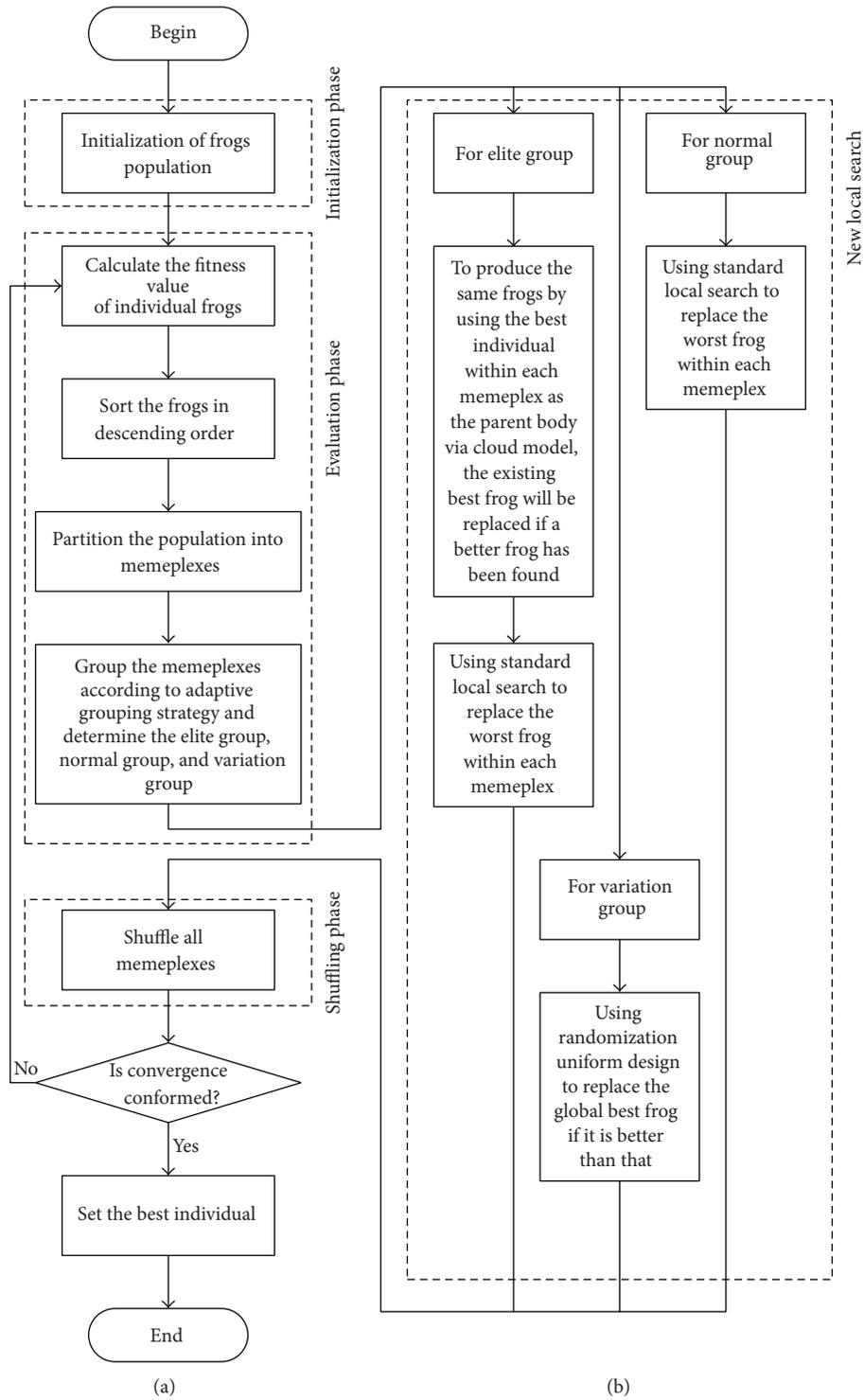


FIGURE 3: (a) The proposed SFLA algorithm’s main diagram; (b) local search for each memeplex.

points. The minimum value is 0. f_2 has six local optimal solutions and two global minimum points. f_3 is a multipeak function. Its global minimum value is 0. The operational simulation result comparison of function $f_1 \sim f_3$ is shown in Tables 2, 3, and 4.

As can be seen from the simulation results, the adaptive grouping cloud model shuffled frog leap algorithm (AGCM-SFLA) proposed by this paper can achieve good solution precision and solution speed. There are two main reasons below for that.

TABLE 1: Tested function.

Number	Function	Interval of x_i
f_1	$f_1 = 0.5 + \frac{\sin^2\left(\sqrt{x_1^2 + x_2^2}\right) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	$-100 \leq x_i \leq 100$
f_2	$f_2 = \left[4 - 2.1x_1^2 + \frac{1}{3}x_1^4\right]x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	$-3 \leq x_1 \leq 3, -2 \leq x_2 \leq 2$
f_3	$f_i = 10 \times n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$	$-5 \leq x_i \leq 5, n = 10$

TABLE 2: Simulation results of function f_1 .

Algorithm	Optimal results	Worst result	Result	Time/s	Variance
PSO	$1.55166e - 12$	0.01067	$10.6975e - 04$	0.286	$7.01e - 06$
QACA	$8.93574e - 12$	$10.27554e - 11$	$6.3734e - 11$	0.199	$4.66e - 21$
GACA	$6.36979e - 12$	$10.73633e - 11$	$4.85551e - 11$	0.187	$4.12e - 21$
CM-SFLA	$4.53099e - 12$	$7.89415e - 11$	$3.9303e - 11$	0.179	$2.64e - 21$
AGCM-SFLA	$5.03124e - 13$	$8.9265e - 11$	$3.04139e - 11$	0.165	$2.11e - 21$

TABLE 3: Simulation results of function f_2 .

Algorithm	Optimal results	Worst result	Result	Time/s	Variance
PSO	-1.135745	-1.135745	-1.135745	1.699	$2.35e - 19$
QACA	-1.137994	-1.137994	-1.137994	1.728	$4.11e - 21$
GACA	-1.1391185	-1.1391185	-1.1391185	1.646	$3.95e - 21$
CM-SFLA	-1.140243	-1.140243	-1.140243	1.011	$3.98e - 21$
AGCM-SFLA	-1.1413675	-1.1413675	-1.1413675	0.925	$2.31e - 21$

TABLE 4: Simulation results of function f_3 .

Algorithm	Optimal results	Worst result	Result	Time/s	Variance
PSO	6.56667	44.87252	19.48122	13.244	$3.90e + 02$
QACA	$5.24447e - 11$	2.18889	0.65692	4.522	$9.89e - 01$
GACA	$7.49716e - 11$	$5.66808e - 10$	$9.00064e - 10$	3.671	$6.91e - 01$
CM-SFLA	$2.66002e - 11$	$10.78605e - 11$	$7.58483e - 11$	2.191	$4.54e - 21$
AGCM-SFLA	$10.31712e - 12$	$9.76371e - 11$	$5.70064e - 11$	1.216	$3.89e - 21$

(1) Seen from the best result, the worst result, and the average result, the optimization search effect of PSO is the worst, followed by QACA, GACA, and CM-SFLA. AGCM-SFLA is the best. Although the calculation results of f_2 by reserving four decimal points are the same, AGCM-SFLA is closer to the global optimum value from the perspective of the actual calculation precision. This is because the species initialization of the reserve study mechanism is adopted by the algorithm, which increases the opportunity for individuals to get close to the optimum solution. Moreover, the stable tendency of the cloud model can well protect the optimum individual, thus realizing the adaptive positioning of the surrounding optimum value. The individual diversity can be maintained, while the evolution speed of algorithm acceleration and the search of optimum efficiency can be

achieved by randomness. Meanwhile, the introduced random uniform design (RUD) can help the algorithm to avoid local optimum in the later period, which is conducive to achieving the global optimum solution and preventing defects such as earlier restraint, slow restraint, and regional optimum of some complex functions.

(2) Seen from the average restraint and iterative algebra, the running time, and the variance, SFLA iteration time is less than that of PSO. If the iteration algebra is considered inside the frog leap algorithm memplex, the iteration algebra of SFLA is more. However, seen from the running time, SFLA is faster. This is because only the worst particles are renewed in iteration by SFLA. Under the best circumstance, once iteration is only calculated once, the worst situation is only calculated three times. Relative to the particle groups, the

calculation numbers of all particles in each time of iteration renewal are fewer, so the speed can be fast. Furthermore, the operation time and variance of AGCM-SFLA are both superior to the previous algorithms. The calculation result of function f_3 is optimal, indicating the AGCM-SFLA has strong adaptability for the solution of high-dimensional multipeak functions.

4. Conclusions

By using the cloud model algorithm, this paper refines the restraint area of the “elite frog” group to discover the global optimum position; the RUD is used to overcome the restrictive advantages caused by the irrational position distribution in the intelligent algorithm. Global optimum search is conducted for the “ordinary frogs” space along with the avoidance of the global optimum position. In this way, the optimum solution restraint can be done quickly, thus avoiding problems such as the local optimum solution and early restraint caused by classic frog leap algorithm. According to the simulation result, the proposed algorithm is with advantages such as high precision and fast restraint speed. The effective integration of the cloud model and the evolution calculation can help to expand the application field of the cloud model, which also promotes the new exploration and trial for the research about the evolution calculation.

Conflict of Interests

The authors declare no conflict of interests.

Authors' Contribution

All authors contributed to the theory and analysis developed in the paper and to finalizing the paper. All authors have read and approved the final paper.

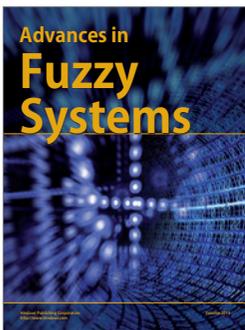
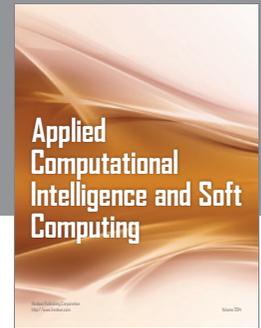
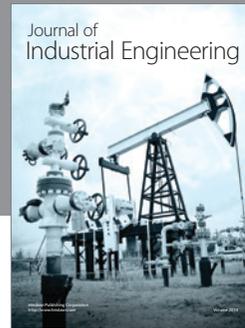
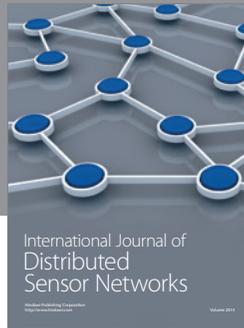
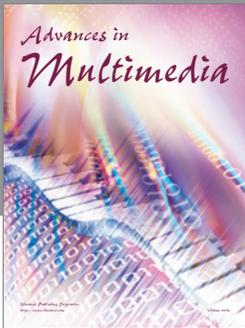
Acknowledgments

The present work is supported by Shandong Province Crucial R&D Plan Project, China (no. 2015GGX105008), and Shandong Provincial Science and Technology Development Plan Project, China (no. 2014GGX105001).

References

- [1] M. M. Eusuff and K. E. Lansey, “Optimization of water distribution network design using the shuffled frog leaping algorithm,” *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210–225, 2003.
- [2] E. Elbeltagi, T. Hegazy, and D. Grierson, “Comparison among five evolutionary-based optimization algorithms,” *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.
- [3] B. Amiri, M. Fathian, and A. Maroosi, “Application of shuffled frog-leaping algorithm on clustering,” *International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1-2, pp. 199–209, 2009.
- [4] A. Alghazi, S. Z. Selim, and A. Elazouni, “Performance of shuffled frog-leaping algorithm in finance-based scheduling,” *Journal of Computing in Civil Engineering*, vol. 26, no. 3, pp. 396–408, 2012.
- [5] M. T. Vakil Baghmisheh, K. Madani, and A. Navarbaaf, “A discrete shuffled frog optimization algorithm,” *Artificial Intelligence Review*, vol. 36, no. 4, pp. 267–284, 2011.
- [6] H. Liu and J. Yang, “Sliding-mode synchronization control for uncertain fractional-order chaotic systems with time delay,” *Entropy*, vol. 17, no. 6, pp. 4202–4214, 2015.
- [7] M. Eghbal, T. Saha, and K. Hasan, “Transmission expansion planning by meta-heuristic techniques: a comparison of shuffled frog leaping algorithm, PSO and GA,” in *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 1–8, Detroit, Mich, USA, July 2011.
- [8] E. Elbeltagi, T. Hegazy, and D. Grierson, “A modified shuffled frog-leaping optimization algorithm: applications to project management,” *Structure and Infrastructure Engineering*, vol. 3, no. 1, pp. 53–60, 2007.
- [9] C. Fang and L. Wang, “An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem,” *Computers & Operations Research*, vol. 39, no. 5, pp. 890–901, 2012.
- [10] M. Gitizadeh, A. A. Vahed, and J. Aghaei, “Multistage distribution system expansion planning considering distributed generation using hybrid evolutionary algorithms,” *Applied Energy*, vol. 101, pp. 655–666, 2013.
- [11] M. Gomez-Gonzalez, F. Jurado, and I. Pérez, “Shuffled frog-leaping algorithm for parameter estimation of a double-cage asynchronous machine,” *IET Electric Power Applications*, vol. 6, no. 8, pp. 484–490, 2012.
- [12] A. Jafari, E. Bijami, H. R. Bana, and S. Sadri, “A design automation system for CMOS analog integrated circuits using new hybrid shuffled frog leaping algorithm,” *Microelectronics Journal*, vol. 43, no. 11, pp. 908–915, 2012.
- [13] A. Kavousifard and H. Samet H, “A novel method based on modified shuffled frog leaping algorithm and artificial neural network for power system load prediction,” in *Emerging Intelligent Technologies in Industry*, pp. 35–46, Springer, Heidelberg, Germany, 2011.
- [14] A. Khorsandi, A. Alimardani, B. Vahidi, and S. H. Hosseinian, “Hybrid shuffled frog leaping algorithm and Nelder-Mead simplex search for optimal reactive power dispatch,” *IET Generation, Transmission & Distribution*, vol. 5, no. 2, pp. 249–256, 2011.
- [15] A. Kimiyaghalam, A. Ashouri, and M. Beykverdi, “Application of IBSFLA and BSFLA approaches for locating of fault indicators in distribution networks,” in *Proceedings of the 17th Conference on Electrical Power Distribution Networks (EPDC '12)*, pp. 1–7, IEEE, Tehran, Iran, May 2012.
- [16] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [17] J. Li, Q. Pan, and S. Xie, “An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems,” *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9353–9371, 2012.
- [18] X. Li, J. Luo, M.-R. Chen, and N. Wang, “An improved shuffled frog-leaping algorithm with extremal optimisation for continuous optimisation,” *Information Sciences*, vol. 192, pp. 143–151, 2012.
- [19] Z. Y. Li, C. X. Yu, and Z. J. Zhang, “Optimal algorithm of shuffled frog leaping based on immune evolutionary particle swarm

- optimization,” *Advanced Materials Research*, vol. 268–270, pp. 1188–1193, 2011.
- [20] Z. Li and Y. Wang, “An improved shuffled frog leaping algorithm for TSP” in *Advances in Multimedia, Software Engineering and Computing*, vol. 129 of *Advances in Intelligent and Soft Computing*, pp. 139–144, 2011.
- [21] Q. Zhang and P. C. Chi, “Adaptive grouping chaotic cloud model shuffled frog leaping algorithm for continuous space optimization problems,” *Control and Decision*, vol. 30, no. 5, pp. 923–928, 2015.
- [22] J. Lin, Y. Zhong, and J. Zhang, “A modified discrete shuffled frog leaping algorithm for RNA secondary structure prediction,” in *Advances in Control and Communication*, vol. 137 of *Lecture Notes in Electrical Engineering*, pp. 591–599, Springer, Berlin, Germany, 2012.
- [23] J. Lin, Y. Zhong, and J. Zhang, “Discrete shuffled frog leaping algorithm for RNA secondary structure prediction,” in *Proceedings of the 7th International Conference on Natural Computation (ICNC '11)*, vol. 3, pp. 1489–1493, IEEE, Shanghai, China, July 2011.
- [24] J. Liu, Z. Li, X. Hu, Y. Chen, and F. Liu, “Multi-objective dynamic population shuffled frog-leaping biclustering of microarray data,” *BMC Genomics*, vol. 13, no. 3, article S6, 2012.
- [25] A. R. Malekpour, S. Tabatabaei, and T. Niknam, “Probabilistic approach to multi-objective Volt/Var control of distribution system considering hybrid fuel cell and wind energy sources using improved shuffled frog leaping algorithm,” *Renewable Energy*, vol. 39, no. 1, pp. 228–240, 2012.
- [26] T. Niknam, E. A. Farsani, M. Nayeripour, and B. Bahmani Firouzi, “A new tribe modified shuffled frog leaping algorithm for multi-objective distribution feeder reconfiguration considering distributed generator units,” *European Transactions on Electrical Power*, vol. 22, no. 3, pp. 308–333, 2012.
- [27] H. Pakraveshe and A. Shojaei, “Optimization of industrial CSTR for vinyl acetate polymerization using novel shuffled frog leaping based hybrid algorithms and dynamic modeling,” *Computers & Chemical Engineering*, vol. 35, no. 11, pp. 2351–2365, 2011.
- [28] I. G. Sardou, M. Banejad, R. Hooshmand, and A. Dastfan, “Modified shuffled frog leaping algorithm for optimal switch placement in distribution automation system using a multi-objective fuzzy approach,” *IET Generation, Transmission & Distribution*, vol. 6, no. 6, pp. 493–502, 2012.
- [29] E. Sayedi, M. M. Farsangi, M. Barati, and K. Y. Lee, “A modified shuffled frog leaping algorithm for nonconvex economic Dispatch problem,” in *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 1–8, IEEE, San Diego, Calif, USA, July 2012.
- [30] M. Tavakolan, “Applying the shuffled frog-leaping algorithm to improve scheduling of construction projects with activity splitting allowed,” in *Proceedings of the Management and Innovation for a Sustainable Built Environment*, pp. 1–9, Amsterdam, The Netherlands, June 2011.
- [31] M. Tavakolan and B. Ashuri, “Comparison of evolutionary algorithms in non-dominated solutions of time-cost-resource optimization problem,” in *Proceedings of the 48th ASC Annual International Conference*, pp. 1–9, 2012.
- [32] L. Wang and C. Fang, “An effective shuffled frog-leaping algorithm for multi-mode resource-constrained project scheduling problem,” *Information Sciences*, vol. 181, no. 20, pp. 4804–4822, 2011.
- [33] L. Xu, M. Fei, T. Jia, and T. C. Yang, “Bandwidth scheduling and optimization using non-cooperative game model-based shuffled frog leaping algorithm in a networked learning control system,” *Neural Computing & Applications*, vol. 21, no. 6, pp. 1117–1128, 2012.
- [34] Y. Xu, L. Wang, G. Zhou, and S. Wang, “An effective shuffled frog leaping algorithm for solving hybrid flow-shop scheduling problem,” in *Advanced Intelligent Computing*, vol. 6838, pp. 560–567, Springer, Berlin, Germany, 2011.
- [35] C. Yammani, S. Maheswarapu, and S. Matam, “Multiobjective optimization for optimal placement and size of dg using shuffled frog leaping algorithm,” *Energy Procedia*, vol. 14, pp. 990–995, 2012, Proceedings of the 2nd International Conference on Advances in Energy Engineering.
- [36] X. Zhang, Y. Zhang, Y. Shi, L. Zhao, and C. Zou, “Power control algorithm in cognitive radio system based on modified shuffled frog leaping algorithm,” *AEU—International Journal of Electronics and Communications*, vol. 66, no. 6, pp. 448–454, 2012.
- [37] B. Fu, D. G. Li, and M. K. Wang, “Review and prospect on research of cloud model,” *Application Research of Computers*, vol. 28, no. 2, pp. 420–426, 2011.
- [38] M.-Z. Liu, X. Zhang, M.-X. Zhang, M.-G. Ge, and J. Hu, “Rescheduling decision method of manufacturing shop based on profit-loss cloud model,” *Control and Decision*, vol. 29, no. 8, pp. 1458–1464, 2014.
- [39] Y. J. Zhang, S. F. Shao, and J. Niyongabo, “Cloud hyper mutation particle swarm optimization algorithm based on cloud model,” *Pattern Recognition and Artificial Intelligence*, vol. 24, no. 1, pp. 90–94, 2011.
- [40] G. W. Zhang, R. He, Y. Liu, D. Y. Li, and G. S. Chen, “An evolutionary algorithm based on cloud model,” *Chinese Journal of Computers*, vol. 31, no. 7, pp. 1082–1090, 2008.
- [41] J. G. Jiang, L. Zhang, Q. Su, L. Deng, and M. Liu, “Shuffled frog leaping algorithm using dynamic searching strategy,” *Journal of Xidian University*, vol. 41, no. 4, pp. 51–57, 2014.
- [42] R. Zhe, “A Genetic algorithm based on random uniform design point set for solving MUCP,” *Pattern Recognition and Artificial Intelligence*, vol. 23, no. 2, pp. 284–288, 2010.



Hindawi

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

