International Journal of Smart Home
Vol. 9, No. 11, (2015), pp. 313-322
http://dx.doi.org/10.14257/ijsh.2015.9.11.30

Research on an Improved Differential Evolution Algorithm based on Three Strategies for Solving Complex Function

Hao Jia[1,2,3,4]

[1]*Department of Electrical Engineering, Dalian Institute of Science and Technology, Dalian 116052 China*
[2]*Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning 530006 China*
[3]*The State Key Laboratory of Mechanical Transmissions, Chongqing University ,Chongqing 400044 China*
[4]*Traction Power State Key Laboratory of Southwest Jiaotong University,Chengdu 610031 China*

Abstract

For the shortcomings of differential evolution algorithm(DE), such as the low convergence rate in the late evolution, easy to trap into the local optimal solution, and weak situation of the global search ability and the stability of optimization, an improved differential evolution algorithm based on multi-population and dynamic local search(MPDLSDE) is proposed in this paper. In the MPDLSDE algorithm, different populations select different mutation operation model in order to obtain superiority reciprocity between different models in the process of evolution. And the random selected method and small probability perturbation are used to increase the diversity of population and balance the exploitation ability and exploration ability of the algorithm. Then dynamic local search method is used to solve the current optimal solution in order to speed up the convergence rate. Several well-known benchmark functions are selected to validate the efficiency of the MPDLSDE algorithm. The simulation experiment and comparative analysis results show that the MPDLSDE algorithm can enhance the global convergence ability and get the high accuracy solution in high dimensional complex optimization problems.

Keywords*: Differential evolution algorithm, multi-population, dynamic local search, mutation operator, convergence rate, complex function*

1. Introduction

Evolutionary algorithm(EA) a kind of modern stochastic search method by simulating the evolutionary process in the nature, which has been successfully used to solve all kinds of optimization problems. The typical evolutionary algorithms include differential evolution(DE)[1], genetic algorithm (GA)[2], particle swarm optimization(PSO) algorithm[3], and ant colony optimization(ACO) algorithm[4] and so on. These evolutionary algorithms do not require derivative information and the state of the function, and they have a wide range and strong robustness.

The DE algorithm is a stochastic global search algorithm and was proposed by Stron and Price in 1995. It is simple, less controlled parameters and takes on good robustness and global optimization ability. With the more and more perfecting of the theoretical research, the DE algorithm has gradually become the most commonly technology for solving the optimization problem[5]. The DE algorithm has been widely used in computer science, communication, biological information, chemical engineering, environmental protection, mechanical engineering and operation research and so on. However, some relevant literatures show that the performance of DE algorithm is highly dependent on the

ISSN: 1975-4094 IJSH

generation strategies of new individual, such as mutation operation and crossover operation, and the corresponding control parameters, such as population size, gain constant and cross probability. Because different optimization problems have different mathematical properties, the inappropriate generation strategy of new individual and parameters can not only result in the high computational cost and slow convergence rate, but also tend to premature convergence and trap into local optimal solution. In addition, the different evolution strategies and parameters are used to improve the performance of the algorithm in the different evolution stages.

For the above problems, a lot of scholars have proposed some strategies and adaptive mechanism for parameters. Su and Lee[6] proposed an effective method of network reconfiguration to reduce power loss and enhance the voltage profile by the improved mixed-integer hybrid differential evolution (MIHDE) method for distribution systems. Coelho and Mariani[7] proposed an improved DE algorithms for solving economic load dispatch problems that take into account nonlinear generator features such as ramp rate limits and prohibited operating zones in the power system operation. Swagatam et al.[8] proposed an improved differential evolution algorithm for the automatic clustering of large unlabeled data sets. Coelho et al.[9] proposed a DE approach based on measure of population's diversity and cultural algorithm technique using normative and situational knowledge sources to solve the economic load dispatch problems of thermal generators. Wang et al.[10] proposed a novel mixed-integer ant direction hybrid differential evolution algorithm, called MIADHDE, to solve this kind of objective function with constraints consisting of the damping factor and damping ratio. Swagatam and Amit[11] proposed an improved variant of the differential evolution (DE) algorithm to determine the number of naturally occurring clusters in the image as well as to refine the cluster centers. Swagatam and Sudeshna[12] proposed a modified differential evolution (DE) algorithm for clustering the pixels of an image in the gray-scale intensity space. The algorithm requires no prior information about the number of naturally occurring clusters in the image. Li et al.[13] proposed an Improved Differential Evolution (IDE) algorithm for optimization problems. With the novel mutant operation adopting sub-optimal individual, the convergence of Differential Evolution (DE) algorithm is accelerated without increasing the risk of premature. Lee et al.[14] proposed an improved differential evolution algorithm, named the Taguchi-sliding-based differential evolution algorithm (TSBDEA) to solve the problem of optimization for the surface grinding process. Musrrat and Millie[15] proposed a modified differential evolution (MDE) that enhances the convergence rate without compromising with the solution quality for optimization of real-valued, multimodal functions. Zou et al.[16] proposed an improved differential evolution algorithm (IDE) to solve task assignment problem. The IDE modifies two important parameters of DE algorithm: scale factor and crossover rate. Arnob et al.[17] proposed a very simple yet very much effective adaptation technique for tuning Scale Factor (F) and Crossover Rate (Cr). The adaptation strategy is based on the objective function value of individuals in the DE population. Jia et al.[18] proposed an improved version of ($\mu + \lambda$)-CDE, named ICDE, to to overcome the main drawbacks of ($\mu + \lambda$)-constrained differential evolution (($\mu + \lambda$)-CDE) and solve constrained optimization problems (COPs). Deng et al.[19] proposed an improved self-adaptive differential evolution algorithm with multiple strategies (ISDEMS) algorithm using a different search strategy and a parallel evolution mechanism. Nikhil et al.[20] proposed an improving differential evolution through borrowing of operations from a benchmark solver G3-PCX. Baatar et al.[21] proposed an improved differential evolution algorithm adopting a new mutation strategy, 'DEλ-best1,' to increase the performance of global optimization. The suggested mutation strategy guides the population to the feasible region of various constraint optimization problems. Huang[22] proposed an improved differential evolution algorithm based on loglinear model and implemented to avoid to fall into local optimum value and improve the slow convergence rate. Wang and Ma[23] proposed several improved DE algorithms to find a balance between global and local search and get optimal solutions through rapid convergence. Kordestani et al.[24] proposed a new variant of Differential Evolution (DE), called ADE-Grid, which adapts the mutation strategy, crossover rate (CR) and scale factor (F) during the run. Gong et al.[25] proposed an improved constrained differential evolution (DE) method to efficiently optimize the constrained engineering problems. Costa and de Oliveira[26] proposed a Differential Evolutionary Clustering Search (DECS) for the majority continuous optimization functions benckmarks. Zhang and Duan[27] proposed improved constrained differential evolution (DE) algorithm to generate an optimal feasible route. for the unmanned aerial vehicles (UAVs) in the three-dimensional environment. Ho-Huu et al.[28] proposed a novel discrete variables handling technique and integrating it into original ICDE to give a so-called Discrete-ICDE (D-ICDE) for solving layout truss optimization problems. Yi et al.[29] proposed a novel differential evolution algorithm based on adaptive differential evolution algorithm by implementing pbest roulette wheel selection and retention mechanism.

These improved DE algorithms overcome the premature convergence and falling into local optimum problems. But there exists some shortcomings, such as the low convergence rate in the late evolution and weak situation of the global search ability and the stability of optimization and so on. So an improved differential evolution algorithm based on multi-population and dynamic local search(MPDLSDE) based on multi-population, random mutation operation and dynamic local search method is proposed in this paper. To validate the effectiveness of the proposed MPDLSDE algorithm, some Benchmark functions are selected and tested.

2. Differential Evolution Algorithm

The DE algorithm is a random search algorithm based on introducing the novel and unique differential mutation operators. It has the advantages of simple structure, easy implementation, efficient and fast speed, and good robustness. The main idea is to use the individual difference (difference vector) of the current population for constructing the differential mutation operator of mutation individual. The operator differences and zooms two different individual vector in the same population, then adds the third individual vector in this population in order to obtain a mutation individual vector. The mutation individual vector crosses the parent individual vector according to the certain probability in order to generate trial individual vector. Finally, the trial individual vector and parent individual vector implement the paired league selection by using greedy selection mechanism, the better individual vector is saved to the next generation.

The flow chart of DE algorithm is shown in Figure 1.

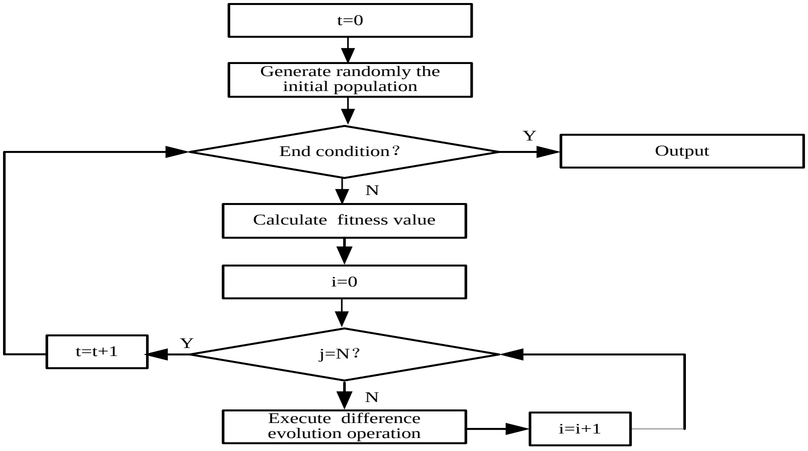

t=0
Generate randomly the initial population
End condition?
Y
Output
N
Calculate fitness value
i=0
t=t+1
Y
j=N?
N
Execute difference evolution operation
i=i+1

Figure 1. The Flow Chart of DE Algorithm

The different mutation and selection strategies are used to improve the convergence rate and the accuracy of DE algorithm. The $DE/rang/1/bin$ is selected in this paper, the steps of DE algorithm is described as follows:

Step1. Initialize Population

The key parameters of the population size(N), the mutation factor(F), the crossover rate(CR) and the stopping criterion(T) in the DE algorithm are initialized. Each individual is encoded as one vector of floating-point number with the same length of decision parameter. The upper bound and lower bound of each decision variable are initialized according to the uniform probability distribution in the D-dimension problem space. The following equation is used to initialize the individuals of population.

$$X_i^t = [x_1^t, x_2^t, \cdots, x_{DP}^t]$$

where D represents the dimension of individual variables. Each individual in the specified value $[x_{j\min}, x_{j\max}]$ of objective function is randomly generated:

$$x_{ji}(0) = x_{j\min} + rand(0,1) \times (x_{j\max} - x_{j\min}) \; (j = 0,1,2,\cdots,D) \quad (1)$$

where $x_{j\min}$ and $x_{j\max}$ respectively denote the lower bound and upper bound of the individuals in the population.

Step2. Mutation Operation

The different mutations of DE algorithm will be classified by the following notation: $DE/\alpha/\beta/\delta$. α is the method of selecting the parent chromosome. β is the number of difference vectors, and δ is the recombination mechanism. The bin acronym indicates that the recombination is controlled. DE algorithm realizes the individual mutation for each target vector by using differential strategy. Each vector includes two different individuals (x_{g1}^t, x_{g2}^t) of the parent. According to the different generation strategies of mutation individual, many different DE algorithms are proposed in the different literatures.

(1) DE/rand/1/bin approach

$$x_k = x_{g3}^t + F * (x_{g1}^t - x_{g2}^t) \quad (2)$$

(2) DE/rand/2/bin approach

$$x_k = x_{g3}^t + F * [(x_{g1}^t - x_{g2}^t) + (x_{g4}^t - x_{g5}^t)] \quad (3)$$

(3) DE/best/1/bin approach

$$x_k = x_{gbest}^t + F * (x_{g1}^t - x_{g2}^t) \quad (4)$$

(4) DE/rand/2/bin approach

$$x_k = x_{gbest}^t + F * [(x_{g1}^t - x_{g2}^t) + (x_{g4}^t - x_{g5}^t)] \quad (5)$$

(5) DE/current-to-best/1 approach

$$x_k = x_{g3}^t + F * [(x_{g1}^t - x_{g2}^t) + (x_{gbest}^t - x_{g3}^t)] \quad (6)$$

where t is the generation, $x_{g1}^t, x_{g2}^t, x_{g3}^t, x_{g4}^t$ and x_{g5}^t are separate random individuals. x_{gbest}^t denotes the best fitness function value of individual of g generation in the population. F is scale factor (F> 0), which controls the amplification of the difference between two individuals.

Step3. Crossover Operation

The goal of crossover operation is to obtain an offspring by mixing current components. In order to enhance the potential diversity of the population, the crossover operation plays a key role. The essence of crossover operation is to realize the uniform crossover between the generated individual (x_k) in the mutation and the i^{th} individual (x_i^t) in the population in order to compensate mutation search in the previous step to generate test vector (x_G). This operation includes the binomial crossover method and index crossover method. The binomial l crossover method is used in this paper. The crossover operator expression is:

$$x_{Gj} = \begin{cases} x_{kj} & rand(0,1) \le CR\, or\ j = j_{rand} \\ x_{ij}^t & other \end{cases} \quad j = 1,2,\cdots,D \tag{7}$$

where $j_{rand} \in \{1,2,\cdots,D\}$ is a random integer, which is used to ensure that at least one component in target individuals x_i^t makes the crossover operation. x_{ij}^t stands for the i^{th} individual of j^{th} real-valued vector, x_{kj} stands for the i i^{th} individual of j^{th} real-valued vector of a mutant vector, $rand(0,1)$ is the j^{th} evaluation of a uniform random number generation with[0,1], CR (crossover rate) is [0,1].

Step4. Selection operation

The goal of selection operator is to construct one population by selecting the trial vectors and their predecessors (target vectors). The DE algorithm generates the offspring by using the greedy selection strategy. The test individual will be competed between x_G and x_i^t after the crossover operation is executed. The better fitness value of the individual is selected as the offspring. If the objective function is to be minimized, the selected operation equation is:

$$x_i^{t+1} = \begin{cases} x_G & f(x_T) < f(x_i^t) \\ x_i^t & f(x_T) \ge f(x_i^t) \end{cases} \tag{8}$$

Step5. Termination condition

If the optimal solution is required or the max number of iterations is exceeded, then the solution is stopped. Otherwise, repeat Step1.~ Step4.

3. An improved differential evolution algorithm

3.1. Multi-population Strategy

The population is randomly divided into N sub-populations according the fitness value to enable the dynamic exchange information among these individuals. In each iteration, the parallel evolution strategy is executed in each sub-population. And all individuals are sorted according to their fitness values. The best individual is obtained to randomly replace the worst individual in each sub-population. Then the fitness values of these individuals are computed again in order to complete the exchange information among these individuals. A new population is obtained by combining the N sub-populations. The multi-population strategy can keep the evolutionary stability of the best individuals and avoids the premature convergence in the evolution process.

3.2. Random Mutation Operation Strategy

The mutation operation is a very important operation in the differential evolution algorithm. With the increase of the evolution iteration, the difference between individuals is gradually becoming smaller, so that the algorithm can converge. In order to avoid premature convergence and increase the diversity of population, the random mutation operation method is used in this paper. The specific expression is given as follows:

$$x_{i,g+1} = \begin{cases} x_{g0} + rand_1 * (x_{g1} - x_{g2}) & if \;\; rand > 0.5 \\ x_{gbest} + rand_2 * (x_{g1} - x_{g2}) & if \;\; rand \le 0.5 \end{cases} \tag{9}$$

where $rand$, $rand_1$ and $rand_2$ are uniform random number on [0,1], x_{g0} , x_{g1} and x_{g2} are randomly selected individuals from the current population. x_{gbest} is optimal individual of current population. The random mutation operation method can effectively expand the search performance of the algorithm, take into account the exploration ability and development ability in order to achieve better results.

3.3. Dynamic Local Search Strategy

The dynamic search technology is proposed based on the random search technology, which is composed of general search and local search. It mainly carried out the random search in the neighborhood of the current solution, and the search step size is gradually reduced with the number of iterations to make the current solution and its surrounding neighborhood for obtaining a full search. The standard DE algorithm is lack of local search ability for leading to the slow convergence of the algorithm. In order to make up for this shortcoming, the dynamic local search operator is introduced into the DE algorithm in order to make use of its powerful exploitation ability to search the optimal solution for many times, so that it can quickly converge to the optimal solution.

4. Experiment Results and Analysis

4.1. Experiment Functions

In order to test the performance of proposed MPDLSDE algorithm for solving complex problem, some benchmarks functions are chosen in this paper. These typical experiment functions from benchmarks testing set, include Step function, Schwefel function, Rosenbrock function, Ackley function and Griewank function. The specific function expression, optimum value, dimension, range is shown in Table 1.

Table 1. Benchmark Test Functions(30D)

Index	Function	Expression	Range
f_1	Step	$f(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	[-100,100]
f_2	Schwefel	$f(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	[-30,30]

f_3	**Rosenbrock**	$f(x)=\sum_{i=1}^{n}100(x_i - x_{i-1}^2)^2 + (x_{i-1}-1)^2$	**[-30,30]**
f_4	**Ackley**	$f(x)=-20\exp[\sum_{i=1}^{n}\frac{x_i^2}{n}]-\exp(\sum_{i=1}^{n}\cos(2\pi x_i)/n)+20+e$	**[-32,32]**
f_5	**Griewank**	$f(x)=\frac{1}{4000}\sum_{i=n}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	**[-600,600]**

4.2. Experiment Environment

The standard DE algorithm, ADE algorithm and ACDE algorithm are selected to compare with MPDLSDE algorithm. In the first, all parameters of these algorithms are initialized. The values of parameters of these algorithms could be a complicated problem itself, the change of parameters could affect the optimum value. So the selected ones are those that gave the best computational results concerning both the quality of the solution and the run time needed to achieve this solution. The obtained initial values of these parameters are: population size NP = 66, the sub-population number is 3, the function dimension is 30, the maximum evolution generation $T_{\max}$ =500, each algorithm is independently run 20 times. Run environment is: Matlab 2010, running on the Pentium IV, 4.0GB RAM.

4.3. Experiment Results

The average optimal adaptive value is used as the basis of performance comparison. Finally, the experiment results are shown in Table 2.

Table 2. The Experimental Tested Results (30D)

Fun	Algorithm	Opt. value	Max value	Min value	Average value
f_1	**DE**	**0**	3.453 305E -19	3.427 681E-24	4.603 472E-20
	ADE		4.431 053E -22	3.957 648E-27	3.356 745E-23
	ACDE		3.427 241E-23	7.734 074E-29	2.035 943E-24
	MPDLSDE		**0.000 000E+00**	**0.000 000E+00**	**0.000 000E+00**
f_2	**DE**	**0**	3.452 064E-02	4.903 746E-05	6.312 045E-03
	ADE		4.103 862E-03	2.240 847E-06	7.045 374E-04
	ACDE		1.486 235E-03	3.457 903E-07	5.350 743E-04
	MPDLSDE		3.420 841E-05	2.438 075E-09	4.047 845E-06
f_3	**DE**	**0**	3.045 934E +01	6.034 741E-05	4.394 047E-01

	ADE		6.731 641E +00	3.432 483E-06	3.582 046E-02
	ACDE		3.350 892E +00	8.047 865E-07	2.462 005E-02
	MPDLSDE		2.470 791E -14	**3.057 814E-16**	**7.847 074E-15**
f_4	**DE**	**0**	4.048 784E+00	3.647 037E-03	6.042 973E-01
	ADE		2.341 846E-05	5.158 547E-03	2.975 749E-06
	ACDE		3.401 632E-11	7.475 375E-14	3.736 704E-12
	MPDLSDE		1.468 863E-12	**5.147 974E-16**	**5.746 036E-13**
f_5	**DE**	**0**	3.940 762E -01	6.047 815E-04	4.730 732E-02
	ADE		2.362 046E -02	5.048 869E-06	2.632 714E-03
	ACDE		8.441 973E -04	3.837 861E-07	5.047 135E-05
	MPDLSDE		2.338 407E -13	**7.483 936E-16**	**7.473 043E-14**

In this paper, the obtained max value, the obtained min value and the obtained average value are selected to describe the performances. As can be seen in Table 2, we easily discover that the proposed MPDLSDE algorithm can find the best solution for Step function, Schwefel function, Rosenbrock function, Ackley function and Griewank function by observing the experiment results. So the proposed MPDLSDE algorithm takes on better optimization performance than the standard DE algorithm, ADE algorithm and ACDE algorithm. It is more capable to search for the global optimization solution and overcome the premature phenomenon for the high-dimensional function problems. Therefore, the proposed MPDLSDE algorithm takes on the better global convergence ability in solving high dimensional optimization problems.

5. Conclusion

The DE algorithm is a stochastic, population-based, evolutionary search algorithm. It is an efficient and powerful optimization algorithm, which widely applied in scientific research and engineering field. But it exists premature convergence, poor search ability and easy falling into local optimal solutions in solving complex problem. In order to improve the performance of DE algorithm, an improved differential evolution algorithm based on multi-population and dynamic local search(MPDLSDE) algorithm is proposed in this paper. The MPDLSDE algorithm uses different populations to obtain superiority reciprocity between different models in the process of evolution. The dynamic local search method is used to solve the current optimal solution in order to speed up the convergence rate. The random mutation operation method can effectively expand the search performance of the algorithm, take into account the exploration ability and development ability in order to achieve better results. Several well-known benchmark functions are selected to validate the efficiency of the MPDLSDE algorithm. The simulation results show that the proposed MPDLSDE algorithm can greatly improve the convergence speed and accuracy. And the optimization performance is better than the standard DE algorithm, ADE algorithm and ACDE algorithm.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (U1433124,51475065), the Liaoning Natural Science Foundation of China(2015020013), the Open Project Program of State Key Laboratory of Mechanical Transmissions (Chongqing University)(SKLMT-KFKT-201416, SKLMT-KFKT-201513), the Open Project Program of the Traction Power State Key Laboratory of Southwest Jiaotong University (TPL1403), Open Project Program of Guangxi Key laboratory of hybrid computation and IC design analysis(HCIC201402), Open Project Program of Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS1326),. The program for the initialization, study, training, and simulation of the proposed algorithm in this article was written with the tool-box of MATLAB 2010b produced by the Math-Works, In.

References

[1] K. S. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces”, Journal of Global Optimization, vol.11, no. 4, **(1997)**, pp. 341-359.

[2] W. Deng, R. Chen, J. Gao, Y. J. Song and J. J. Xu, “A novel parallel hybrid intelligence optimization algorithm for a function approximation problem”, Computers and Mathematics with Applications, vol. 63, no. 1, **(2012)**, pp. 325–336.

[3] W. Deng, R. Chen, B. He, Y. Q. Liu, L. F. Yin and J. H. Guo, “A novel two-stage hybrid swarm intelligence optimization algorithm and application”, Soft Computing, vol. 16,no. 10, **(2012)**, pp. 1707-1722.

[4] W. Deng, H. M. Zhao, J. J. Liu, X. L. Yan, Y. Y. Li, L. F. Yin and C. H. Ding, “An improved CACO algorithm based on adaptive method and multi-variant strategies”, Soft Computing, vol. 19, no. 3, **(2015)**, pp. 701-713.

[5] R. Kicinge, T. Arciszewski and K. De Jong, “Evolutionary computation and structural design:A survey of the state-of-the-art”, Computers & structures, vol. 83, no. 23-24, **(2015)**, pp.1943-1978.

[6] C. T. Su and C. S. Lee, “Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution”, IEEE Transactions on Power Delivery, vol. 18, no. 3, **(2003)**, pp.1022-1027.

[7] L. S. Coelho and V. C. Mariani, “Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints”, Energy Conversion and Management, vol. 48, no. 5, **(2007)**, pp. 1631-1639.

[8] D. Swagatam, A. Ajith and K. Amit, “Automatic clustering using an improved differential evolution algorithm”, IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans, vol. 38, no.1,**(2008)** ,pp. 218-237.

[9] L. S. Coelho, R. C. T. Souza and V. C. Mariani, “Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems”, Mathematics and Computers in Simulation, vol. 79, no. 10, **(2009)**, pp. 3136-3147.

[10] S. K. Wang, J. P. Chiou and C. W. Liu, “Parameters tuning of power system stabilizers using improved ant direction hybrid differential evolution”, International Journal of Electrical Power and Energy Systems, vol. 31, no. 1, **(2009)**, pp. 34-42.

[11] D. Swagatam and K. Amit, “Automatic image pixel clustering with an improved differential evolution”, Applied Soft Computing Journal, vol. 9, no. 1, **(2009)**, pp. 226-236.

[12] D. Swagatam and S. Sudeshna, “Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm”, Information Sciences, vol. 180, no. 8, **(2010)**, pp. 1237-1256.

[13] R. Li, L. Xu, X. W. Shi, N. Zhang and Z. Q. Lv, “Improved differential evolution strategy for antenna array pattern synthesis problems”, Progress in Electromagnetics Research, vol. 113, **(2011)**, pp. 429-441.

[14] K. M. Lee, M. R. Hsu, J. H. Chou and C. Y. Guo, “Improved differential evolution approach for optimization of surface grinding process”, Expert Systems with Applications, vol. 38, no. 5, **(2011)**, pp. 5680-5686.

[15] A. Musrrat and P. Millie, “Improving the performance of differential evolution algorithm using Cauchy mutation”, Soft Computing, vol. 15, no. 5, **(2011)**, pp. 991-1007.

[16] D. X. Zou, H. K. Liu, L. Q. Gao and S. Li, “An improved differential evolution algorithm for the task assignment problem”, Engineering Applications of Artificial Intelligence, vol. 24, no. 4, **(2011)**, pp. 616-624.

[17] G. Arnob, D. Swagatam, C. Aritra and G. Ritwik, “An improved differential evolution algorithm with fitness-based adaptation of the control parameters”, Information Sciences, vol. 181, no. 18, **(2011)**, pp. 3749-3765.

[18] G. B. Jia, Y. Wang, Z. X. Cai and Y. C. Jin, "An improved differential evolution algorithm with fitness-based adaptation of the control parameters", Information Sciences, vol. 222, **(2013)**, pp. 302-322.

[19] W. Deng, X. H. Yang , L. Zou, M. Wang, Y. Q. Liu and Y. Y. Li, "An improved self-adaptive differential evolution algorithm and its application", Chemometrics and Intelligent Laboratory Systems, vol. 128, **(2013)**, pp. 66-76.

[20] P. Nikhil, B. Piyush and D. Kalyanmoy, "Improving differential evolution through a unified approach", Journal of Global Optimization, vol. 55, no. 4, **(2013)**, pp. 771-799.

[21] N. Baatar, D. H. Zhang and C. S. Koh, "An improved differential evolution algorithm adopting λ-best mutation strategy for global optimization of electromagnetic devices", IEEE Transactions on Magnetics, vol. 49, no. 5, **(2013)**, pp. 2097-2100.

[22] Z. H. Huang, "An improved differential evolution algorithm based on statistical log-linear model", Sensors and Transducers, vol. 159, no. 11, **(2013)**, pp. 277-281.

[23] S. Q. Wang and J. L. Ma, "Multi-objective optimization based on improved differential evolution algorithm", Telkomnika (Telecommunication Computing Electronics and Control), vol. 12, no. 4, **(2014)**, pp. 977-984.

[24] J. K. Kordestani, A. Ahmadi and M. R. Meybodi, "An improved Differential Evolution algorithm using learning automata and population topologies", Applied Intelligence, vol. 41, no. 4, **(2014)**, pp. 1150-1169.

[25] W. Y. Gong, Z. H. Cai and D. W. Liang, "Engineering optimization by means of an improved constrained differential evolution", Computer Methods in Applied Mechanics and Engineering, vol. 268, **(2014)**, pp. 884-904.

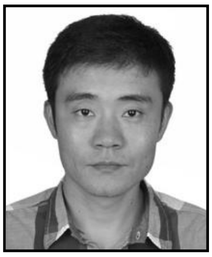
[26] T. S. Costa and A. C. M. de Oliveira, "Artificial bee and differential evolution improved by clustering search on continuous domain optimization", Soft Computing, vol. 19, no. 9, **(2014)**, pp. 2457-2468.

[27] X. Y. Zhang and H. B. Duan, "An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning", Applied Soft Computing Journal, vol. 26, no. 1, **(2015)**, pp. 270-284.

[28] V. H. Huu, T. N. Thoi, M. H. N. Thoi and L. L. Anh, "An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures", Expert Systems with Applications, vol. 42, no. 20, **(2015)**, pp. 7057-7069.

[29] W. B. Yi, Y. Z. Zhou, L. Gao, X. Y. Li and J. H. Mou, "An improved adaptive differential evolution algorithm for continuous optimization", Expert Systems with Applications, vol. 44, no. 2, **(2016)**, pp. 1-12.

Author



Hao Jia

The main research directions: Artificial Intelligence, Electrical Engineering.