

An Improved Adaptive Differential Evolution based on Hybrid Method for Function Optimization

Li Huang

*School of Software Engineering, Beijing Jiaotong University, Beijing 100044
China*

Abstract

In order to improve the optimum speed, crease the diversity of the population and overcome the premature convergence problem in differential evolution(DE) algorithm for solving the complex optimization problems, the chaotic optimization algorithm with powerful local searching capacity and multi- strategy are introduced into the DE algorithm in order to propose an improved adaptive differential evolution(COMSIADe) algorithm in solving function optimization problems. In the COMSIADe algorithm, the ergodicity, regularity and internal randomness of the chaotic sequence are used to overcome the shortcoming of premature local optimum to improve the global searching capacity of the DE algorithm. The multi-population with parallel evolution is used to preserve the diversity of the population at the initial generation. The self-adaptive crossover operator probability is used to improve the global convergence ability, the stability and robustness. Finally, in order to test and verify the effectiveness of the COMSIADe algorithm, several benchmark functions are selected in this paper. The experimental results indicate that the proposed COMSIADe algorithm can improve the global searching capacity and avoid falling into local optimum. And it takes on the higher searching precision and faster convergence speed in solving the complex optimization problem.

Keywords: *differential evolution, chaotic optimization, multi-strategy, self-adaptive, complex optimization problem*

1. Introduction

Differential evolution (DE) algorithm is a branch of genetic algorithm, it is proposed by Storn in 1995 [1]. The original conception is to solve the polynomial problem of Chebyshev, and the DE algorithm is found to be able to solve complex optimization problems. The DE algorithm is a parallel random search algorithm, which is simple in principle, less controlled parameters, and random, direct, parallel global search for the continuous space. In recent years, the DE algorithm has been widely applied due to the robustness, stability and strong global search ability on the real domain, and achieved good results [2]. However, the optimization performance of the DE algorithm is heavily influenced by the value of controlled the parameters and the type of differential evolution, the DE algorithm is easy to fall into the premature and convergence problem. With the increasing of iterations, the diversity of population will rapidly decrease and easily trapped into the local optimum [3].

In order to improve the performance of the DE algorithm, many scholars had proposed a lot of improved DE algorithms. Su and Lee [4] proposed an improved mixed-integer hybrid differential evolution (MIHDE) method for reducing power loss and enhance the voltage profile in distribution system. Bhat, *et al.*, [5] proposed an improved differential evolution method (IDE) to apply for the evaluation of the parameters. It has been demonstrated that the IDE is a variant of the original DE method. Coelho and Mariani [6] 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. The DE algorithms and its variants are validated for two test systems consisting of 6 and 15 thermal units. Swagatam, *et al.*, [7] proposed an improved differential evolution algorithm for developing partitioned clustering techniques and one popular hierarchical clustering algorithm. Coelho, *et al.*, [8] proposed an improved DE approach based on measure of population's diversity and cultural algorithm technique by using normative and situational knowledge sources to solve the economic load dispatch problems of thermal generators. Lai and Cao [9] proposed an improved differential evolution algorithm (IDE) for solving this problem. In the IDE, the novel decimal coding is adopted to construct an initial population, some improved differential evolution operators and an integer order criterion based on natural number coding method are used. And a self-adapting crossover probability is designed in the proposed IDE algorithm. Lee, *et al.*, [10] 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. The purpose of this work is to optimize the grinding variables such as wheel speed, work piece speed, depth of dressing, and lead of dressing, using a multi-objective function model with a weighted approach, simultaneously subject to a comprehensive set of process constraints. Wang and Cai [11] proposed a $(\mu+\lambda)$ -differential evolution and an improved adaptive trade-off model for solving constrained optimization problems. The proposed $(\mu+\lambda)$ -differential evolution adopts three mutation strategies (*i.e.*, rand/1 strategy, current-to-best/1 strategy, and rand/2 strategy) and binomial crossover to generate the offspring population. Iorio and Li [12] proposed a combinatorial sampling differential evolution (CSDE), which can sample vectors in two ways; highly correlated with the search space or around a 'better' individual. The CSDE approach can provide a similar number of samples as crossover, without being biased towards the principle coordinate axes of a decision space. Ramesh, *et al.*, [13] proposed an improved multi-objective generalized differential evolution (IGDE3) approach to solve optimal reactive power dispatch (ORPD) with multiple and competing objectives. The objective functions are minimization of real power loss and bus voltage profile improvement. Shi and He [14] proposed a novel improved differential evolution algorithm (IDE) for constrained optimization problems. Elsayed, *et al.*, [15] proposed an improved differential evolution algorithm that uses a mix of different mutation operators. In addition, the algorithm is empowered by a covariance adaptation matrix evolution strategy algorithm as a local search. Jia, *et al.*, [16] proposed an improved version of $(\mu+\lambda)$ -CDE, named ICDE, to solve constrained optimization problems (COPs). ICDE mainly consists of an improved $(\mu+\lambda)$ -differential evolution (IDE) and a novel archiving-based adaptive tradeoff model (ArATM). Padhye, *et al.*, [17] recollected the 'Unified Theory of Evolutionary Computation' from past and investigates four steps - Initialization, Selection, Generation and Replacement, which are sufficient to describe traditional forms of Evolutionary Optimization Systems such as Genetic Algorithms, Evolutionary Strategies, Evolutionary Programming, Particle Swarm Optimization and differential evolution (DE). Baatar, *et al.*, [18] proposed an improved differential evolution algorithm adopting a new mutation strategy, 'DE λ -best1,' to increase the performance of global optimization. The suggested mutation strategy guides the population to the feasible region of various constraint optimization problems. Wang and Ma [19] proposed several improved DE algorithms to find a balance between global and local search and get optimal solutions through rapid convergence. Meanwhile, a random mutation mechanism is adopted to process individuals that show stagnation behaviour. Wang, *et al.*, [20]

proposed an improved differential evolution algorithm (dn-DADE) based on dynamic adaptive strategy to complex optimization functions. A set of 14 benchmark functions is adopted to test the performance of the proposed algorithm. Dong, *et al.*, [21] proposed an optimization model to tune feature weights for improving performance of clustering via a minimization of uncertainty (fuzziness and non-specificity) of its similarity matrix among objects. The authors proposed an evolutionary search approach by integrating multiple strategies from both differential evolution and dynamic differential evolution. Then, the proposed method is applied to both weighted fuzzy c-means and weighted similarity-matrix-based transitive closure clustering. Tang, *et al.*, [22] proposed an improved differential evolution (DE) algorithm with a real-coded matrix representation for each individual of the population, a two-step method for generating the initial population, and a new mutation strategy. To further improve the efficiency and effectiveness of the solution process for dynamic use, an incremental mechanism is proposed to generate a new initial population for the DE whenever a real-time event arises, based on the final population in the last DE solution process. Kotinis [23] proposed in this article focuses on the development of a multi-objective optimization algorithm based on the differential evolution (DE) concept combined with Mamdani-type fuzzy logic controllers (FLCs) and K-medoids clustering. The FLCs are used for adaptive control of the DE parameters; K-medoids clustering enables the algorithm to perform a more guided search by evolving neighboring vectors, *i.e.*, vectors that belong to the same cluster. Tsai [24] proposed an improved differential evolution algorithm (IDEA) to solve nonlinear programming and engineering design problems. The proposed IDEA combines the Taguchi method with sliding levels and a differential evolution algorithm (DEA). The DEA has a powerful global exploration capability on macrospace and uses fewer control parameters. Bhadra and Bandyopadhyay [25] proposed an unsupervised feature selection algorithm based on improved version of a recently developed Differential Evolution technique.

2. Chaotic Optimization Algorithm (COA)

Chaos is a kind of characteristic, which has a bounded unstable dynamic behavior and exhibits sensitive dependence on its initial conditions [26]. It takes on many outstanding properties, such as randomness, ergodicity, sensibility and so on. So it is a significant interest to construct optimization methods based on chaotic search routines in the past 20 years. Chaotic optimization algorithm (COA) is a recently proposed population-based stochastic optimization algorithm, which is used by chaotic map. The basic steps of the COA are: (1) the COA use the ergodicity, regularity, initial sensitivity and topological transitivity to search all the points in turn and take the better point as the current optimum point. (2) The current optimum point is regarded as the center. A tiny chaotic disturbance is imposed and the careful search is performed in order to search the global optimum point with the high probability. Because the chaos is the non-repetition, the COA can execute the overall search at higher speeds. The COA has the easy implementation, short execution time and robust mechanisms of escaping from the local optimum.

Currently, several kinds of the COA based on chaotic characteristics are proposed, such as adaptive mutative scale COA, a mutative scale COA, chaotic harmony search algorithms, multi-objective chaotic ant swarm optimization, etc. Generally, the key problem of the COA is how to generate chaotic variables. In this paper, the logistic chaotic model is used to generate the chaotic variables. The map equation of the logistic model is shown:

$$X_{n+1} = L(\mu, X_n) = \mu X_n (1 - X_n) \quad \mu \in [0,4], n = 0,1,2,3, \dots \quad (1)$$

where μ is control variable, the value is $[0,4]$. When X_n is $[0,1]$, the logistic mapping work is in the chaotic state. That is, the generated sequences under logistic mapping function (the initial condition X_0) are not periodic and converge. Otherwise, the generated sequences must converge to one specific value.

3. Differential Evolution Algorithm

Differential evolution (DE) algorithm is a simple, fast and efficient population-based direct search algorithm to solve global optimization problems. Due to the advantages of the simple structure, easy use, robustness, and fast convergence, the DE algorithm has been widely used in many fields, and achieved good applied results. The DE algorithms use the basic framework of genetic algorithm to design a unique differential mutation operator. Their basic operation includes the mutation operation, crossover operation and selection operation. The flow chart of DE algorithm is shown in Figure 1.

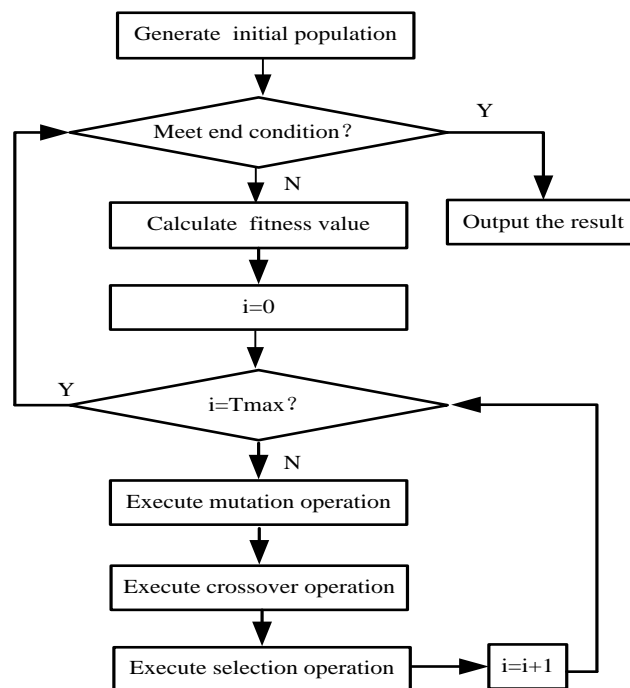


Figure 1. The Flow of DE Algorithm

The different variants of DE algorithm will be classified by the notation $DE/\alpha/\beta/\delta$. The α is a method to select parent chromosome. The β is a number of difference vectors. The δ is the recombination mechanism. The different mutation strategy and selection strategy are used to improve the convergence rate and accuracy of the DE algorithm.

4. Multi-strategy

4.1. Multi-population Strategy

The population is randomly divided into N sub-populations (far less than the number of individuals) according to the fitness value to enable the dynamic exchange of 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. Next, the above process is sequentially executed in order to keep the best individuals to achieve the dynamic exchange information. The multi-population strategy can keep the evolutionary stability of the best individuals and avoids the premature convergence in the evolution process.

Sss

4.2. Adaptive Crossover Probability (CR) Strategy

Adaptive crossover probability strategy is used to adaptively adjust the CR value according to the fitness value of current individual in the sub-population. When the CR value is larger, the more components will be got from the mutation vector X_k . And the X_k makes much contribution to the vector X_G . When the CR value is lesser, the more components will be got from the mutation vector X_i^t . And the X_i^t makes much contribution to the vector X_G . When the CR value is equal to one, the new individual completely comes from the mutation vector X_k . When the CR value is equal to zero, the new individual completely comes from the mutation vector X_i^t . This will be beneficial to the global search and keeping the population diversity. So the CR value in the initial stage should be smaller in order to improve the global search ability. The CR value in the later stage should be gradually increase in order to improve the local search ability. In this paper, a dynamic adaptive adjustment strategy is used to control the CR value. The expression of dynamic adaptive adjustment strategy is given:

$$CR(t) = \frac{CR_{\min}}{1 + \left(\frac{CR_{\min}}{CR_{\max}} - 1\right)e^{-\beta t}} \quad (2)$$

where the CR_{\min} and CR_{\max} indicate the minimum and maximum value of crossover probability. The β is the initial growth rate, $\beta \in [0, +\infty)$. And t is number of iterations. With the increasing of iteration, the β value is dynamically adjusted. When the CR is equal to CR_{\max} , the CR will stop increasing. In order to keep the balance between the diversity and convergence, the CR value is $[0.5, 1]$.

5. An Improved Adaptive Differential Evolution (COMSIAD E) Algorithm

Because the diversity of the population declined rapidly in the late evolution stage, the standard DE algorithm is easy to fall into local optimum and cause the premature phenomenon. However, the DE algorithm mainly caused the reason that all individuals within the population selected too large pressure in the early evolution, and resulted in very fast convergence on the mode. From an evolutionary sense, the biological population usually takes place the stagnation in the evolution under the closed geographic environment, then the individual with new gene pattern is required to participate in the competition. And the chaotic optimization algorithm with characteristics of randomness and ergodicity can effectively compensate for the

defects of the DE algorithm. Currently, some researchers considered two optimization methods for coupling in order to obtain a new optimization algorithm. But these coupling algorithms only uses chaotic map to search solution space, which outstands the global search ability. On this basis, the randomness and ergodicity of the chaotic optimization algorithm is further used. The chaotic disturbances are implemented respectively in the evolution and the end of evolution in order to strengthen the local search ability and balance the global search ability and local search ability. And the multi-population strategy and self-adaptive crossover operator probability strategy are introduced into the DE algorithm to further improve the DE algorithm. So an improved differential evolution (COMSIADe) algorithm based on the chaotic optimization algorithm and multi-strategy is proposed to solve function optimization problems. In the proposed COMSIADe algorithm, there are four basic points: (1) the chaotic map (Logistic map) generated the chaotic sequence, which are mapped into the solution space of the solved problem. These converted chaotic sequence are regarded as the initial solution of the DE algorithm. (2) The multi-population with parallel evolution is used to preserve the diversity of the population at the initial generation. It can maintain the evolutionary stability of the individuals, promote the parallel evolution of the various sub-populations and avoid the premature convergence phenomenon in the evolution process. (3) The obtained individuals by executing the mutation operation, crossover operation and selection operation are implemented the chaotic disturbance in order to obtain new individuals. The self-adaptive crossover operator probability is used to improve the global convergence ability, the stability and robustness. (4) The obtained global optimal solution is implemented the chaotic disturbance in order to execute the deeply local search.

According to the idea, the steps of the proposed COMSIADe algorithm in solving function optimization problems are given:

Step 1. Initialize all parameters.

Generate three subpopulations. The initial parameters include the subpopulation size N , the maximum value and minimum value of the scale factor F , the maximum value and minimum value of the crossover probability CR , the maximum iteration times T_{max} , the initial decay rate α , the initial growth rate β , the chaotic search times k , and so on.

Step 2. Introduce chaotic disturbance strategy

The characteristics of the randomness and ergodicity of chaotic optimization is mapped into the solution space of the solved problem. These converted chaotic sequence are regarded as the initial solution of the DE algorithm

Step 3. Evaluate each individual according to the fitness function in the subpopulation in order to search the best individual.

Step 4. Judge the optimum solution. If the optimum solution meets the termination criterion, the individual of the maximum fitness value in the population is outputted as the optimal solution. Otherwise, go to Step 3.

Step 5. Do the mutation operation. After the mutation operation is done, these individuals with meeting constraints will be chosen to the next generation. Otherwise, the parent individuals will be chosen to the next generation.

Step 6. Do the crossover operation. After the crossover operation is implemented, these individuals with meeting constraints will be chosen to the next generation. Otherwise, the parent individuals will be chosen to the next generation.

Step 7. Do the selection operation. The next generation population is obtained.

Step 8. After the new population is generated, $t = t + 1$ is implemented. Return to Step 3.

6. Numerical Experiment and Analysis

6.1. Testing Function

Some benchmarks functions are chosen to validate the performance of the proposed COMSIAD E algorithm. These typical test functions from benchmarks testing set include Ackley function, Rosenbrock function, Griewank function, Step function, Schwefel2.21 function, Schaffer function and Shekel function. The performance of the proposed COMSIAD E algorithm is compared with DE algorithm [27] and SACDE algorithm [28]. The specific function expression, optimum value, dimension, range is shown in Table 1.

Table 1. Benchmark Test Functions

index	Function	Opt.	Expression	Dim.	Range
F_1	Ackley	0	$f(x) = -20 \exp\left[\sum_{i=1}^n \frac{x_i^2}{n}\right] - \exp\left(\sum_{i=1}^n \cos(2\pi x_i)/n\right) + 20 + e$	30	[-32,32]
F_2	Rosenbrock	0	$f(x) = \sum_{i=1}^n 100(x_i - x_{i-1}^2)^2 + (x_{i-1} - 1)^2$	30	[-30,30]
F_3	Griewank	0	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]
F_4	Step	0	$f(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	30	[-100,100]
F_5	Schwefel2.21	0	$f(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[-30,30]
F_6	Schaffer	-1	$f(x) = \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5$	100	[-100,100]
F_7	Shekel	-10.54	$f(x) = \sum_{i=1}^{10} [(x - a_i)^T (x - a_i) + c_i]$	4	[0,10] ⁺

6.2. Parameter Setting and Run Environment

The all parameters of the DE algorithm, SACDE algorithm and COMSIAD E algorithm are initialized in the first. The parameters with best computational results are selected in this paper. The experimental parameters are obtained: population size $NP = 99$, the subpopulation number is 3. The initial crossover probability is $CR_{\min} = 0.50$ and $CR_{\max} = 1.0$, the initial scaling factor is $F_{\min} = 0.50$ and $F_{\max} = 1.0$. The iteration times $T_{\max} = 1000$, each algorithm is independently run 30 times, the chaotic search state is $\mu = 3$. The initial decay rate α and the initial growth rate β are 80. Run environment is: Matlab 2012a, running on the Pentium IV, 2.0GB RAM.

6.3. Experimental Results and Analysis

The experimental results are shown in Table 2.

Table 2. The Experimental Results

Fun.	Algorithm	Opt.	Worst value	Best value	Mean value	Sta.Dev.
F_1	DE		8.113E+00	2.761E-02	6.814E-01	4.235E-02
	SACDE	0	1.467E-11	2.562E-13	5.467E-12	4.543E-04
	COMSIAD E		1.246E-12	1.135E-14	5.248E-13	3.235E-05
F_2	DE	0	3.573E +01	6.815E-05	7.973E-01	5.693E+00

	SACDE		7.163E +00	5.892E-07	4.246E-01	4.862E-02
	COMSIAD		1.801E -13	4.308E-15	8.684E-14	4.451E-08
	E					
	DE		7.813E -01	8.031E-04	3.780E-02	5.892E-03
F_3	SACDE	0	2.473E -01	3.532E-06	4.645E-02	3.453E-04
	COMSIAD		1.391E -13	3.456E-15	8.597E-13	6.164E-08
	E					
	DE		4.763E -20	3.349E-24	5.547E-21	1.563E-10
F_4	SACDE	0	2.458E-22	2.759E-26	8.452E-24	1.671E-15
	ISDEMS		0.000E+00	0.000E+00	0.000E+00	0.000E+00
	E					
	DE		5.825E-02	6.967E-04	3.031E-03	1.962E-04
F_5	SACDE	0	1.657E-03	5.367E-06	3.462E-04	4.461E-06
	COMSIAD		3.105E-02	1.468E-03	8.428E-02	2.435E-04
	E					
	DE		-7.582E-00	-3.356E-00	-5.143E-00	2.254E-02
F_6	SACDE	-1	-6.459E-01	-8.268E-01	-7.665E-01	4.102E-03
	COMSIAD		-8.763E-01	-9.783E-01	-9.203E-01	2.343E-04
	E					
	DE		-9.561E+00	-9.901E+00	-9.667E+00	3.052E-03
F_7	SACDE	-10.54	-9.902E+00	-1.003E+01	-9.993E+00	1.480E-08
	COMSIAD		-1.021E+01	-1.050E+01	-1.038E+01	4.539E-10
	E					

In this paper, seven famous benchmarks functions are performed by the DE algorithm, the SACDE algorithm and the proposed COMSIAD algorithm with 30 times. The obtained worst value, the obtained best value, the obtained mean value and obtained standard deviation are selected to describe the performances. As can be seen in Table 2, the proposed COMSIAD takes on better optimization performance for solving the F_1 , F_2 , F_3 , F_4 , F_6 and F_7 . But the optimization performance of the F_5 is better than the proposed COMSIAD algorithm. And the proposed COMSIAD algorithm obtains the optimal value for the F_4 . Therefore, the proposed COMSIAD algorithm takes on the better global convergence ability in solving high dimensional optimization problems.

7. Conclusion

This paper proposes an improved differential evolution (COMSIAD) algorithm based on the chaotic optimization algorithm and multi-strategy to solve the complex optimization problems. The COMSIAD algorithm uses the ergodicity, regularity and internal randomness of the chaotic sequence to overcome the shortcoming of premature local optimum to improve the global searching capacity of the DE algorithm. Then the multi-population is used to preserve the diversity of the population at the initial generation. The self-adaptive crossover operator probability is used to improve the global convergence ability, the stability and robustness. The experimental results show that the proposed COMSIAD algorithm can improve the global searching capacity and avoid falling into local optimum. And the proposed COMSIAD algorithm has the higher searching precision and faster convergence speed for solving the complex optimization problem.

References

- [1] K. Storn, "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] 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.
- [3] 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.
- [4] 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.
- [5] R. Bhat, D. Venkataramani, V. Ravi and C. V. S. Murty, "An improved differential evolution method for efficient parameter estimation in biofilter modeling", *Biochemical Engineering Journal*, vol. 28, no. 2, (2006), pp. 167-176.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] M. Y. Lai and E. B. Cao, "An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows", *Engineering Applications of Artificial Intelligence*, vol. 23, no. 2, (2010), pp. 188-195.
- [10] 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.
- [11] Y. Wang and Z. X. Cai, "Constrained evolutionary optimization by means of $(\mu+\lambda)$ -differential evolution and improved adaptive trade-off model", *Evolutionary Computation*, vol. 19, no. 2, (2011), pp. 249-285.
- [12] A. W. Iorio and X. D. Li, "Improving the performance and scalability of Differential Evolution on problems exhibiting parameter interactions", *Soft Computing*, vol. 15, no. 9, (2011), pp. 1769-1792.
- [13] S. Ramesh, S. Kannan and S. Baskar, "An improved generalized differential evolution algorithm for multi-objective reactive power dispatch", *Engineering Optimization*, vol. 44, no. 4, (2012), pp. 391-405.
- [14] Y. C. Shi and Z. Y. He, "A novel improved differential evolution algorithm for constrained optimization problems", *Advances in Information Sciences and Service Sciences*, vol. 4, no. 12, (2012), pp. 250-258.
- [15] S. M. Elsayed, R. A. Sarker and D. L. Essam, "An improved self-adaptive differential evolution algorithm for optimization problems", *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, (2013), pp. 89-99.
- [16] G. B. Jia, Y. Wang, Z. X. Cai and Y. C. J. Yaochu, "An improved $(\mu + \lambda)$ -constrained differential evolution for constrained optimization", *Information Sciences*, vol. 222, (2013), pp. 302-322.
- [17] N. Padhye, P. Bhardawaj and K. Deb, "Improving differential evolution through a unified approach", *Journal of Global Optimization*, vol. 55, no. 4, (2013), pp. 771-799.
- [18] 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.
- [19] 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.
- [20] C. J. Wang, X. H. Wang, J. M. Xiao and Y. Ding, "Improved differential evolution algorithm based on dynamic adaptive strategies and control parameters", *International Journal of Control and Automation*, vol. 7, no. 9, (2014), pp. 81-96.
- [21] C. R. Dong, W. Y. Ng, X. Z. Wang, P. K. Chan and S. Daniel, "An improved differential evolution and its application to determining feature weights in similarity-based clustering", *Neurocomputing*, vol. 146, (2014), pp. 95-103.
- [22] L. X. Tang, Y. Zhao and J. Y. Liu, "An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production", *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, (2014), pp. 209-225.
- [23] M. Kotinis, "Improving a multi-objective differential evolution optimizer using fuzzy adaptation and K-medoids clustering", *Soft Computing*, vol. 18, no. 4, (2014), pp. 757-771.

- [24] J. T. Tsai, "Improved differential evolution algorithm for nonlinear programming and engineering design problems", *Neurocomputing*, vol. 148, (2015), pp. 628-640.
- [25] T. Bhadra and S. Bandyopadhyay, "Unsupervised feature selection using an improved version of Differential Evolution", *Expert Systems with Applications*, vol. 42, no. 8, (2015), pp. 4042-4053.
- [26] H. J. Lu, H. M. Zhang and L. H. Ma, "A new optimization algorithm based on chaos", *Journal of Zhejiang University SCIENCE A*, vol. 7, no. 4, (2006), pp. 539-542.
- [27] J. Brest, S. Greiner, B. Boscovic, M. Mernik and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems", *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, (2006), pp. 646-657.
- [28] Z. Y. Guo, B. Cheng, M. Ye and B. G. Cao, "Self-adaptive chaos differential evolution", *Lecture Notes in Computer Science*, vol. 4221, (2006), pp. 972-975.

Authors



Li Huang, He was born in Guangxi province, China in 1988. He received his B.E. degree in Mechanical Design Manufacture and Automation from Tongji University, Shanghai, China in 2011. Now he is pursuing his M.E. degree in Software Engineering in Beijing Jiaotong University, Beijing, China. His research interests concentrate on Software Development and Intelligent Information Process.