International Journal of Hybrid Information Technology
Vol.8, No.10 (2015), pp.177-186
http://dx.doi.org/10.14257/ijhit.2015.8.10.16

An Improved Differential Evolution Algorithm for Solving High Dimensional Optimization Problem

Chunfeng Song and Yuanbin Hou

The Institute of Electric and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054 China

Abstract

In order to improve the weak situation of the global search ability, the stability and time consuming of optimization of differential evolution(DE) algorithm in solving high dimensional optimization problem, an improved differential evolution algorithm with multi- population and multi-strategy(MPMSIDE) is proposed to solve high dimensional optimization problem. Firstly, the different DE mutation strategies are studied. Then the MPMSIDE algorithm divides the population into several sub-populations, which evolve independently and communicate with each other at regular intervals by using different DE strategies, in order to save the computation time. And the improved mutation strategy and local optimization strategy are introduced to raise and balance the global searching ability and local searching ability, and improve the optimization efficiency. The self-adaptive update strategy is used to adjust the scaling factor and crossover factor for making the parameter sensitivity of DE algorithm and improving the stability and robustness. Finally, the proposed MPMSIDE algorithm is applied to standard test function optimization for verifying the effectiveness. The experimental results show that the proposed MPMSIDE algorithm has a relatively better optimization performance for solving complex optimization problem, and takes on remarkable optimizing ability, higher searching accuracy and faster convergence speed.

Keywords: *differential evolution, multi-population, multi-strategy, control parameter, complex optimization problem*

1. Introduction

The optimization method is divided into the traditional optimization methods and heuristic optimization methods. The traditional optimization methods mainly realize the order of single feasible solution and deterministic search based on the objective function gradient (or derivative) information. And the heuristic optimization methods are a kind of bionic algorithm, which realizes the parallel and stochastic optimization of multi solutions by using the heuristic strategy. The heuristic search algorithms do not require the continuous and differentiable information of the objective function, and take on better global search ability [1, 2]. So they has become a hot topic in the optimization area.

In these heuristic optimization methods, differential evolution (DE) algorithm is a heuristic random search algorithm based on population differences. The DE algorithm was proposed by Storn in 1995, to solve Chebyshev polynomial problem, and complex optimization problems subsequently [3]. The DE algorithm has a very special connection with evolutionary algorithms. The DE algorithm and particle swarm optimization (PSO) algorithm [4] are optimization algorithms based on the swarm intelligence theory. They use the generated swarm intelligence of the competition and cooperation between individuals within the population to guide optimization search. But compared to the evolutionary algorithm, the DE algorithm retains the global search strategy based on the population, uses the real-coding, simple differential mutation and competitive survival strategy to reduce the complexity of the genetic operation. At the same time, the unique

ISSN: 1738-9968 IJHIT

remembering ability of DE algorithm can dynamically track the current search, in order to adjust its search strategy. The DE algorithm has a strong global convergence and robustness, and does not require the character information of solving problem. It is suitable to solve the optimization problems in the complex environment, which can not be solved by using conventional mathematical programming methods. So the DE algorithm is an efficient parallel search algorithm, theory and an application research has important academic significance and engineering value.

However, the DE algorithm is like other evolutionary algorithms, it exists the low searching efficiency and premature convergence. In order to improve the performance of DE algorithm, many scholars have made useful research work. Su and Lee [5] proposed improved mixed-integer hybrid differential evolution (MIHDE) method for distribution systems in order to reduce power loss and enhance the voltage profile. Bhat, *et al.*, [6] proposed an improved differential evolution method (IDE) to apply for the evaluation of the parameters in the framework of the solution of an inverse problem. Swagatam, *et al.*, [7] described an application of DE to the automatic clustering of large unlabeled data sets. Nasimul and Hitoshi [8] proposed a crossover-based adaptive local search (LS) operation for enhancing the performance of standard differential evolution (DE) algorithm and a LS technique to solve this problem by adaptively adjusting the length of the search, using a hill-climbing heuristic. Swagatam and Amit [9] proposed an evolutionary-fuzzy clustering algorithm for automatically grouping the pixels of an image into different homogeneous regions. Lai and Cao[10] proposed an improved differential evolution algorithm(IDE) for solving a general mixed integer programming model of VRP-SPDTW. Zou, *et al.*, [11] proposed an improved differential evolution algorithm (IDE) to solve task assignment problem. The IDE is an improved version of differential evolution algorithm (DE), and it modifies two important parameters of DE algorithm: scale factor and crossover rate. Lee, *et al.*, [12] 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. Ramesh, *et al.*, [13] proposed an improved multi-objective generalized differential evolution (I-GDE3) approach to solve optimal reactive power dispatch (ORPD) with multiple and competing objectives. Baatar, *et al.*, [14] proposed an improved differential evolution algorithm adopting a new mutation strategy, 'DEλ-best1,' to increase the performance of global optimization. Elsayed, *et al.*, [15] proposed an improved differential evolution algorithm that uses a mix of different mutation operators. Basu [16] proposed an improved differential evolution to determine the optimal hourly schedule of power generation in a hydrothermal system. Tang, *et al.*, [17] 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. Tsai [18] proposed an improved differential evolution algorithm(IDEA) based on combining the Taguchi method with sliding levels and a differential evolution algorithm (DEA) to solve nonlinear programming and engineering design problems. Zhang and Duan [19] formulated the global route planning problem for the unmanned aerial vehicles (UAVs) as a constrained optimization problem in the three-dimensional environment and proposed an improved constrained differential evolution (DE) algorithm to generate an optimal feasible route.

These improved DE algorithms overcome the premature convergence and falling into local optimum problems. But the local search ability, convergence speed and optimization accuracy of the algorithm still require to be further strengthened. So an improved differential evolution algorithm with multi- population and multi-strategy (MPMSIDE) is proposed to solve high dimensional optimization problem is proposed in this paper.

2. Differential Evolution

The DE algorithm is a computational intelligent method by simulating biological evolution of natural population. The main idea is: the cooperation, competition and the generational evolution and reproduction of individuals in a population are used to improve the adaptation degree of individuals to the external environment, so as to approximate the optimal solution of the problem. In essence, the DE algorithm is a greedy genetic algorithm with high quality thought based on real coding. The definition of optimization problem and mathematical describing of the DE algorithm are given:

For optimization problem:

$$\min f(x_1, x_2, \cdots, x_d) \quad \text{s.t. } x_j^L \le x_j \le x_j^U \qquad (1)$$

where $j = 1,2,3,\cdots,D$, D is the dimension of the solution space. x_j^L and x_j^U respectively represent the upper and lower bounds of the j^{th} component x_j .

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

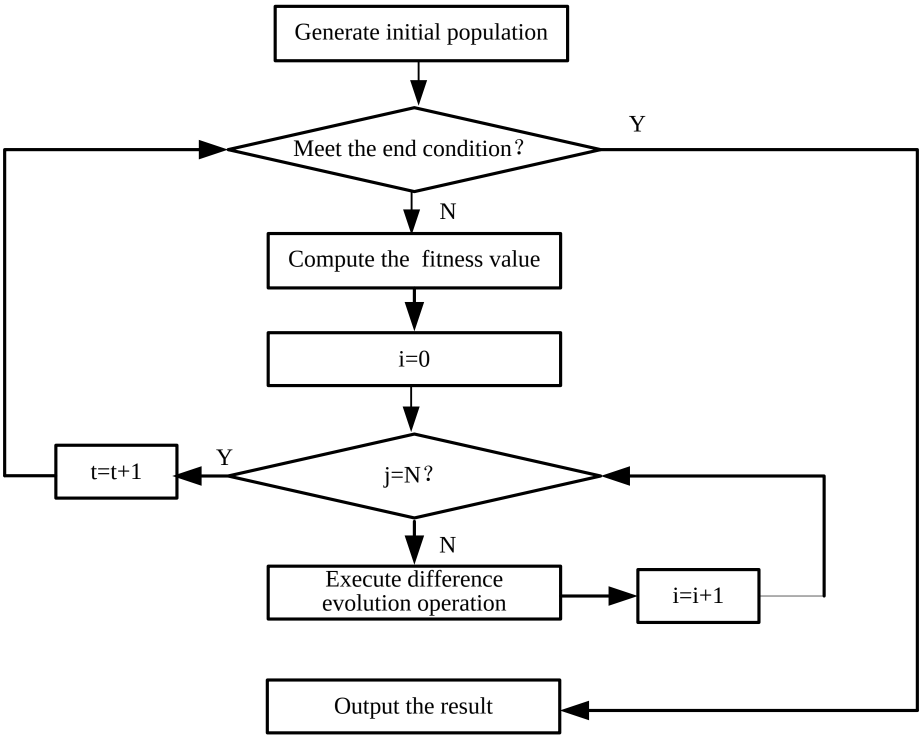

Figure 1. The Flow of DE Algorithm

2.1. Initialization

The key parameters of DE algorithm are initialized. These parameters include the population size (N),the mutation factor(F), the crossover rate(CR) and the stopping criterion(T).The initial generation counter is set as t =1. Each individual is encoded as a vector of floating-point numbers. And the prescribed upper and lower bounds of each decision variable with random generated values are initialized according to the uniform probability distribution in the N-dimensional problem space. The following equation is used to initialize the initial population of individuals:

$$x_j(0) = x_j^L + rand(0,1) \times (x_j^U - x_j^L)\ (j = 0,1,2,\cdots,D) \qquad (2)$$

2.2. Mutation

The DE algorithm executes the individual mutation for target vector by using differential strategy. The mutation component is the different vector of the parent. Each vector consists of two different individuals (x_{g1}^t, x_{g2}^t). According to the different generation method of mutation individual, there proposed some different DE algorithms.

（1）DE/rand/1/bin

$$x_k = x_{g3}^t + F \times (x_{g1}^t - x_{g2}^t) \qquad (3)$$

（2）DE/rand/2/bin

$$x_k = x_{g3}^t + F \times [(x_{g1}^t - x_{g2}^t) + (x_{g4}^t - x_{g5}^t)] \qquad (4)$$

（3）DE/best/1/bin

$$x_k = x_{gbest}^t + F \times (x_{g1}^t - x_{g2}^t) \qquad (5)$$

（4）DE/rand/2/bin

$$x_k = x_{gbest}^t + F \times [(x_{g1}^t - x_{g2}^t) + (x_{g4}^t - x_{g5}^t)] \qquad (6)$$

（5）DE/current-to-best/1

$$x_k = x_{g3}^t + F \times [(x_{g1}^t - x_{g2}^t) + (x_{gbest}^t - x_{g3}^t)] \qquad (7)$$

2.3. Crossover

The crossover operation is used to construct an offspring by mixing current components. In order to enhance the potential diversity of the population, crossover operation plays a key role. The essence of crossover operation is to execute 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). It includes the binomial cross method and index cross method. The binomial cross method is used. The specific crossover operator equation is given:

$$x_{Gj} = \begin{cases} x_{kj} & rand(0,1) \le CR\, or\ j = j_{rand} \\ x_{ij}^t & other \end{cases} \quad j = 1,2,\cdots,D \qquad (8)$$

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].

2.4 Selection

The selection operator is used to construct a population by selecting the trial vectors and their predecessors according to the better fitness value or the optimal value. The DE algorithm generates the offspring by using the greedy selection strategy. After the crossover operation is executed, the test individual will be competed between x_G and x_i^t. The individual with the better fitness value will be selected as the offspring. If the objective function is to be minimized, the selected operation equation is given:

$$x_i^{t+1} = \begin{cases} x_G & f(x_T) < f(x_i^t) \\ x_i^t & f(x_T) \geq f(x_i^t) \end{cases} \tag{9}$$

3. An Improved DE Algorithm with Multi-population and Multi-Strategy (MPMSIDE)

3.1. The Idea of MPMSIDE Algorithm

The DE algorithm a evolutionary algorithm based on real coding, the mutation operation, crossover operation and selection operation of the individual are used to constantly evolve the population until the algorithm terminates. Due to the weak situation of the global search ability, the stability and time consuming of optimization of differential evolution (DE) algorithm in solving high dimensional optimization problem, this paper proposes an improved differential evolution algorithm with multi- population and multi-strategy (MPMSIDE) for solving high dimensional optimization problem. In the proposed MPMSIDE algorithm, according the fitness value, standard deviation of fitness and distance of each individual, the population is divided into three different subpopulations, which are best population with the better fitness of individuals, worst population the poor fitness of individuals and general population with the rest individuals. The best population is responsible for local search and improves the convergence speed and precision. The worst population is responsible for global search, jumps out the local optimum and avoids premature convergence. The general population is responsible for balancing the global search ability and local search ability. The mutation is the key operation of DE algorithm; the selected mutation strategy determines population direction in the process of evolution. The improved mutation strategy is introduced to enhance the global optimization ability of the greedy algorithm, avoid possible stagnation in a local minimum value for dealing with complex functions with high dimension multimodal optimization problems, and the premature loss of population diversity. The local optimization strategy is used to avoid the local extreme point and improve the local hill-climbing ability in the local search. The self-adaptive update strategy determines the similarity between the best individual and the general individual according to the individual similarity coefficient for reducing the adverse effects of the linear adjusting scaling factors and making the parameter sensitivity of DE algorithm and improving the stability and robustness.

The improved mutation strategy DE/best/1 is given:

$$x_k = x_{gbest}^t + F_2 \times (x_{g1}^t - x_{g2}^t) + \alpha \times F_2 \times (x_{g3}^t - x_{g4}^t) \tag{10}$$

where α is correlation parameter of vector. The added α can give more choice space in this mutation. The improved mutation strategy can greatly enhance the searching space and the local search ability of the algorithm. The MPMSIDE algorithms increases a number of control parameters and the population, and effectively enhance the searching space the local search ability and provides a more flexible choice than standard DE algorithm.

3.2. The Flow of MPMSIDE Algorithm

On the basis of the above idea, the flow of an improved differential evolution algorithm with multi- population and multi-strategy (MPMSIDE) in solving high dimensional optimization problem is shown in Figure 2.

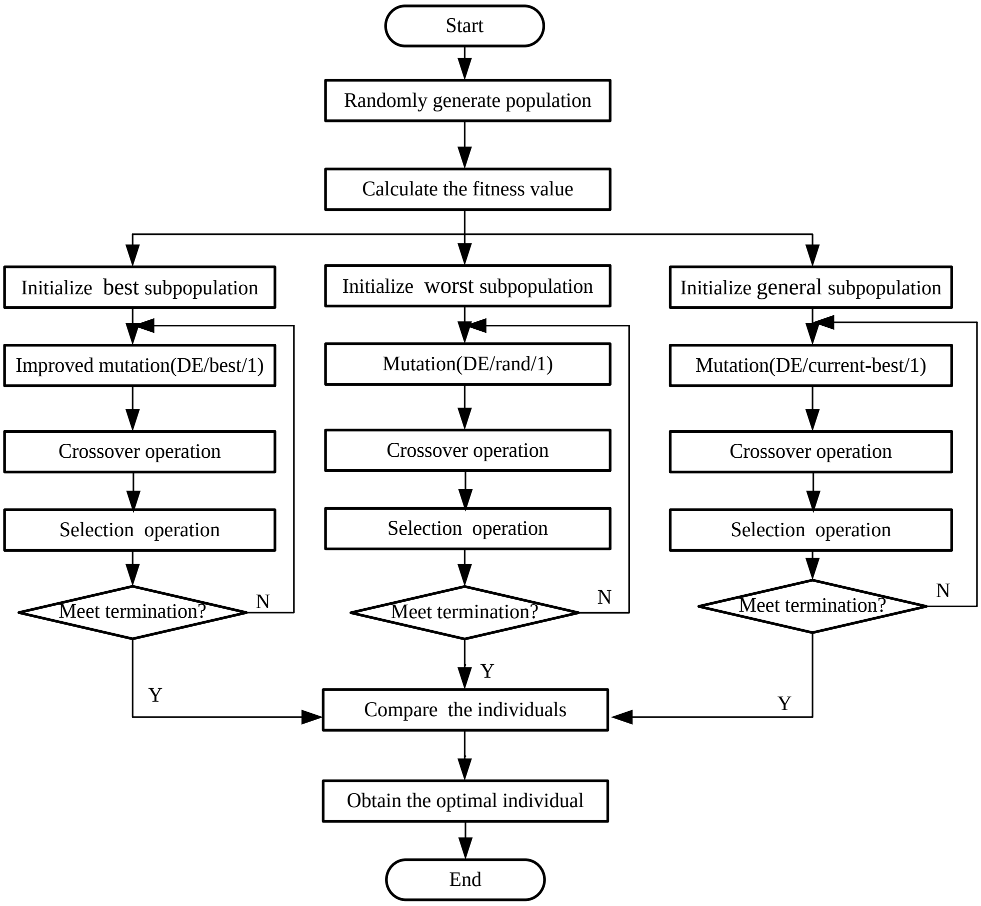

Figure 2. The Flow of the MPMSIDE Algorithm

4. The Results of Testing and Analyzing MPMSIDE Algorithm

The famous Benchmarks functions are selected to test the performance of the MPMSIDE algorithm in this paper. For example, Rosenbrock function, Noisy function, Schwefel1.2 function, Step function, Schwefel2.21 function and so on. In order to make the MPMSIDE algorithm with comparative results, the MPMSIDE algorithm is compared with the DE algorithm and CDE algorithm. The experiment works on Intel(R) Core i5-4200U,2.40GHz, 2G RAM, Windows 8 and Matlab 2012. The experimental parameters are given: population size N = 66, crossover probability factor $CR = 0.9$, scaling factor $F = 0.5$, the functional dimension is 30, the maximum evolution generation $T_{\max}$ =1000. The specific formula and variables' range of all functions are shown in Table 1.

Table 1. Benchmarks Testing Functions

index	Function name	Function expression	Variable Range
f_1	Rosenbrock	$f(x)=\sum_{i=1}^{n} 100(x_i - x_{i-1}^2)^2 + (x_{i-1}-1)^2$	[-30,30]
f_2	Noisy	$f(x)=(\sum_{i=i}^{n} ix_i^4 + rand[0,1]$	[-1.28,1.28]
f_3	Schwefell.2	$f(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i} x_j)^2$	[-30,30]
f_4	Step	$f(x)=\sum_{i=1}^{n}(\lfloor x_i + 0.5 \rfloor)^2$	[-100,100]
f_5	Schwefel2.21	$f(x)=\max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	[-30,30]

The MPMSIDE algorithm, DE algorithm and CDE algorithm run independently with 30 times for five functions. The maximum value, minimum value and mean value and standard deviation are used to evaluate three algorithms. The results are shown in Table 2.

Table 2. The Results for the DE, CDE and MPMSIDE Algorithms

Function	Algorithm	The optimal value	Maximum value	Minimum value	Mean best value	Standard deviation
	DE		3.145 46e+00	6.431 39e-02	4.237 29e-01	1.931 53e+00
f_1	CDE	0	3.942 16e-04	1.362 42e-07	3.432 17e-06	4.256 65e-04
	MPMSIDE		7.042 15e-12	5.632 67e-14	8.325 48e-13	4.453 15e-13
	DE		4.225 37e-01	6.242 48e-02	2.324 47e-02	3.310 24e-02
f_2	CDE	0	3.941 49e-05	8.054 74e-07	6.618 19e-06	2.691 74e-05
	MPMSIDE		5.132 18e-05	3.451 45e-06	9.452 31e-05	4.234 57e-05
	DE		4.942 08e-01	5.578 03e-02	9.558 64e-01	4.756 83e-01
f_3	CDE	0	4.341 64e-02	3.683 65e-04	7.053 48e-03	5.522 47e-03
	MPMSIDE		2.656 30e-03	1.468 37e-04	5.105 92e-04	3.471 37e-04
	DE		3.167 32e-20	6.857 26e-24	3.145 52e-21	4.203 06e-22
f_4	ACDE	0	6.846 45e-28	3.690 23e-32	4.645 90e-29	4.365 46e-24
	SPMDE		0.000 00e+00	0.000 00e+00	0.000 00e+00	0.000 00e+00
	DE		5.036 45e-02	7.999 26e-03	9.056 02e-02	1.698 39e-03
f_5	CDE	0	8.572 45e-04	2.068 34e-06	1.245 35e-05	3.896 73e-04
	MPMSIDE		6.643 47e-07	3.561 24e-08	9.395 46e-07	2.045 38e-04

As can be seen from the Table 2, for the five famous Benchmarks functions with the MPMSIDE algorithm, the MPMSIDE algorithm is better optimization performance than DE algorithm for solving f_1, f_2, f_3, f_4 and f_5 functions. The MPMSIDE algorithm is better optimization performance than CDE algorithm for solving f_1, f_3, f_4 and f_5 functions. For f_4 function, The MPMSIDE algorithm obtains the optimal value(zero). So the proposed MPMSIDE algorithm takes on the better global convergence ability and searching precision in solving high dimensional optimization problems.

5. Conclusion

DE algorithm is an efficient and powerful algorithm for solving complex optimization problems. It depends on the mutation strategy and crossover strategy, and the values of the associated control parameters. And for one complex optimization problem, the different mutation strategy and crossover strategy may be more effective than single mutation strategy and crossover strategy. So an improved differential evolution algorithm with multi- population and multi-strategy(MPMSIDE) is proposed to solve high dimensional optimization problem in this paper. In the MPMSIDE algorithm, the population is divided into best population, worst population and general population according the fitness value, standard deviation of fitness and distance of each individual. The improved mutation strategy is introduced to enhance the global optimization ability of the greedy algorithm, avoid possible stagnation in a local minimum value for dealing with complex functions with high dimension multimodal optimization problems. The local optimization strategy is used to avoid the local extreme point and improve the local hill-climbing ability in the local search. The self-adaptive method is used to automatically adjust the scaling factor and crossover factor during the running time. The performance of MPMSIDE algorithm is evaluated on benchmark problems and is favorably compared with DE, CDE algorithm in the literature. The results demonstrate that the proposed MPMSIDE algorithm is overall more effective and takes on better searching precision, convergence speed, and global convergence ability.

ACKNOWLEDGEMENTS

This research was supported by the Youth Science Foundation (51405381) and the Special Foundation for Scientists of Xi'an university of science and technology (201314).

References

[1] C. I. L. Lopez, L. G. van Willigenbury and G. van Straten, "Efficient differential evolution algorithm for multimodal optimal control problems", Applied Soft Computing, vol. 3, no. 2, **(2003)**, pp. 97-122.
[2] 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.
[3] 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.
[4] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", Proceeding of the IEEE International Conference on Neural Networks, Piscataway, NJ, **(1995)**, pp. 1942-1948.
[5] 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.
[6] T. 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.
[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] N. Nasimul and I. Hitoshi, "Accelerating differential evolution using an adaptive local search", IEEE Transactions on Evolutionary Computation, vol. 12, no. 1, **(2008)**, pp. 107-125.
[9] 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.
[10] 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, **(2006)**, pp. 188-195.
[11] 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.
[12] 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.
[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] 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.
[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, **(2012)**, pp. 89-99.
[16] M. Basu, "Improved differential evolution for short-term hydrothermal scheduling", International Journal of Electrical Power and Energy Systems, vol. 58, no. 6, **(2014)**, pp. 91-100.
[17] 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.
[18] J. T. Tsai, "Improved differential evolution algorithm for nonlinear programming and engineering design problems", Neurocomputing, vol. 148, **(2015)**, pp. 628-640.

[19] 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.

Authors

Chunfeng Song, He was born in Nanyang city, the province of Henan in July, 1977, who graduated from the University of Science and Technology and obtained the Master of Engineering degree in 2005. He is majoring in the research about the control theory and control engineering.

Yuanbin Hou, She was born in November, 1953, graduated from Xi'an Jiaotong University and obtained the Doctor of Engineering degree in 1997. She is majoring in the research about the control theory and control engineering.