Multi-objective Differential Evolution Algorithm based on Adaptive Mutation and Partition Selection

Sen Zhao
College of Computer Science and Engineering, South China University of Technology, Guangzhou, China
Department of Computer Science, JiNan University, Guangzhou, China

Zhifeng Hao
College of Computer Science and Engineering, South China University of Technology, Guangzhou, China
School of Computer, Guangdong University of Technology, Guangzhou, China

Han Huang
School of Software Engineering, South China University of Technology, Guangzhou, China

Yang Tan
College of Computer Science and Engineering, South China University of Technology, Guangzhou, China

Abstract—**A multi-objective differential evolution algorithm based on adaptive mutation strategies and partition selected search is proposed based on classical differential evolution （DE） to further improve the convergence and diversity of multi-objective optimization problems. This algorithm improves mutation operation in DE, makes search oriented and ensures the convergence of algorithm by adaptively selecting mutation strategies based on the non-inferiority of the individuals of the population in evolution. In addition, a partition-based elitist preserving mechanism is applied to select the best individuals for the next generation, thus improving the selection operation in DE and maintaining the diversity of Pareto optimal set. The experiment on 5 ZDT test functions and 3 DTLZ test functions and comparison with and analysis of other classical algorithms such as NSGA-II and SPEA2 show that this algorithm converges the populations towards non-inferior frontier rapidly on the premise of maintaining the diversity of the populations. From the measure and graphs, it can be seen that this algorithm is feasible and effective in solving the multi-objective optimization problems.**

Index Terms—**multi-objective optimization, differential evolution, adaptive mutation, partition selection**

I. INTRODUCTION

In science and engineering practices, many problems are multi-objective problems with multiple different objectives, usually conflicting, needing to be achieved. Such problems are called multi-objective optimization problems(MOP)[1]. The solution to the MOP is not unique and there are many optimal solutions which forms a set. An element of that set is a Pareto optimal solution. Elements of the set are uncomparable with respect to all objectives.

A great number of studies show that evolutionary algorithms are very suitable for solving MOPs. Domestic and foreign scholars conducted many researches and proposed multiple handling strategies, with many effective multi-objective evolutionary algorithms (MOEA) developed[2].

Thanks to simple principle, few controlled parameters and random, parallel and direct global search and high understandability and easy implementation, Differential Evolution (DE)[3]is a very effective heuristic evolution algorithm. DE has been proved to be the fastest evolution algorithm[4]. Solving MOPs with DE naturally attracted great attention and some DE-based MOEAs have been successively proposed, such as PDE[5] proposed by Abbass et al in 2001, PDEA[6] proposed by Madavan et al in 2002, DEMO[7] proposed by Robic et al in 2005.OW-MOSaDE[8] with learning strategies introduced proposed by Huang, V.L, MODE-LD+SS[9] combined with partial domination and invariant selection mechanism proposed by Alfredo in 2010, etc.

A MOP solution must achieve two objectives: (1) approaching the Pareto frontier; (2) maintaining the diversity of the populations, and it is considerably difficult to simultaneously achieve the two objectives. Although the existing MOEAs have their respective advantages, they still have disadvantages such as low speed, complex optimization techniques, etc.

This paper introduces DE into the MOEA and thus improves the mutation and selection operations in DE. A multi-objective differential evolution algorithm which determines the mutation strategies based on the domination relationship between individuals in the evolution and employs partition selected search is

Manuscript received January 10,2013; revised March 15,2013; accepted March 26, 2013
Sen Zhao:e-mail:zssally2002@yahoo.com.cn.

doi:10.4304/jcp.8.10.2695-2700

designed to achieve rapid optimization and the method is relatively simple.

II. DIFFERENTIAL EVOLUTION

DE is a population-based random search algorithm. Its main principle is to generate variants based on differential vector, then cross breeding them to create test individuals and finally select the best individuals into the next generation. Basic DE operations include mutation, crossover and selection operations.

A. Mutation Operation

The DE algorithm applies mutation to each objective vector through differential strategies. A common differential strategy is that two different individual vectors are selected and deducted from each other to genreate a differential vector. After being weighted, that differential vector is added to the third randomly selected individual vector so as to generate a mutated vector.

Assuming the population size is NP, each individual has D dimensions and t is the number of evolutionary generations, then an individual of Generation t can be expressed as $x_i(t)$,i=1,2,…NP.

For each target individual, the mutated individual is generated by the following method:

$$v_i(t+1) = x_{i1}(t) + F \cdot (x_{i2}(t) - x_{i3}(t)) \qquad (1)$$

,where x_{i1}, x_{i2} and x_{i3} are three individuals randomly selected from the population, $i_1 \neq i_2 \neq i_3 \neq i$, and F is the differential vector scale factor which decides the size of the differential individual.

B. Crossover Operation

To increase the diversity of the populations, in DE the mutated individual is parameter hybridized with the target individual so as to generate a test individual and this process is called crossover. The test individual ui=[u_{i1},u_{i2},…u_{iD}] is generated from the following formula with hybridization probability CR∈[0,1]:

$$u_{ij}(t+1) = \begin{cases} v_{ij}(t+1), if\ randb \leq CR\ or\ j = randr \\ x_{ij}(t), if\ randb > CR\ or\ j \neq randr \end{cases}$$

$$j=1,2,\ldots,D \qquad (2)$$

, where randb is a random number belonging to [0,1], randr is a random integer belonging to [1,D]. This operation ensures at least an element must be obtained from $v_i(t+1)$ by u_i (t+1), otherwise no new individual will be generated and the population will not change.

C. Selection Operation

Like other evolution algorithms, DE applies Darwinian's rule of survival of the fittest . In DE, such rule of survival of the fittest is expressed as greedy strategy. If the fitness of the test individual is greater than that of the target individual, then the target individual will be replaced by the test individual which will be a member of the next generation; otherwise, the target individual remains in the population and will serve as the parent individual of the next generation. Selection is carried out through the following formula:

$$x_i(t+1) = \begin{cases} u_i(t+1), if\ f(u_i(t+1)) \leq f(x_i(t)) \\ x_i(t),\ otherwise \end{cases} \qquad (3)$$

, where f is the fitness evaluation function.

III. MULTI-OBJECTIVE DIFFERENTIAL EVOLUTION ALGORITHM BASED ON ADAPTIVE MUTATION STRATEGIES AND PARTITION SELECTED SEARCH

This paper proposes a multi-objective differential evolution algorithm based on adaptive mutation strategies and partition selected search (AM+PS-MODE) on the basis of analyzing basic features of the DE algorithm.

A. Adaptive Mutation Strategies

Depending upon method for generating the mutated individual, DE can have many different mutation strategies. To differentiate different DEs, DE is usually expressed as DE/x/y/z, where x represents whether the base vector is randomly selected or the best individual in the current population, y represents the number of differential vectors and z represents the crossover mode, which can be index crossover or binomial crossover.

Formula (1) is the most commonly used strategy and is expressed as DE /rand/l. Besides this strategy, there are also other mutation strategies, for example,

DE/best/1: $v_i(t+1) = x_{best}(t) + F(x_{i1}(t) - x_{i2}(t))$ (4)

DE/current to best /2:

$$v_i(t+1) = x_i(t) + F(x_{best}(t) - x_i(t)) + F(x_{i1}(t) - x_{i2}(t)) \qquad (5)$$

x_{best} (t) is the individual with the best fitness which is identified in Generation t.

Mutation strategies are one of core parts of the DE algorithm and each mutation strategy has its own advantages. DE/rand/1/bin (binomial crossover) is the mutation strategy which was proposed earliest and is considered a mutation strategy which has been most widely and most successfully applied. In such mutation strategy, the base vectors are selected randomly from the entire population and is well diversified. They have strong global search capability, are highly robust and uneasily get stuck in local optimum problem. But their search efficiency is relatively low due to that they do not provide guiding information about better regions. DE/ current to best /2/bin uses the best individual as the base vector and this enables all target individuals to be searched around the best individual and they have strong local search capability and fast convergence. But due to the rapid reduction of the diversity of the population, too early convergence of the algorithm will be easily caused.

From the above analysis, it can be known that a relatively better method is to combine different mutation strategies to ensure the base vectors in evolution have high quality as well as high diversity and thus better balance the global and local search capabilities of the algorithm. Based on this philosophy, a method for adaptively selecting mutation strategies based on the non-inferiority (sequential value) of the individuals in evolution is proposed to sufficiently take advantage of the features of each mutation strategy. The specific method is as follows:

In evolution, when the sequential value of the target individual is "1" (in the optimal frontier plane), the DE/rand/1/bin strategy will be adopted, otherwise the DE/current to best /2/bin will be adopted. As of matter of fact, during initial phase of evolution individuals are on different frontier planes and have different sequential values. Using the DE/current to best/2/bin strategy to ensure the generated mutated individual has a lower sequential value is beneficial to converge the algorithm to the non-inferior frontier. With evolution going on, the individuals in the population are on the optimal frontier plane. Using the DE/rand/1/bin strategy to calculate the differential vector from the individuals on the same frontier is beneficial to the scattering of the generated mutated individual along that frontier and the spreading of information to obtain a non-domination set with good distribution characteristic. Therefore, the global and local search capabilities of the algorithm can be balanced by adopting the above mutation strategies, thus ensuring both the convergence and distribution requirements of the algorithm.

In addition, there is no absolute optimal solution to a MOP and there is only a satisfactory non-inferior optimal solution. Its optimal solution is usually a set and it is difficult to determine which non-inferior optimal solution in the Pareto optimal set is x_{best}. In the algorithm, the non-inferior optimal solution with the smallest sequential value and having the shortest euclidean distance from the target individual in the target space is selected as x_{best}. The formula is as follows:

$$x_{best} = \min\left\{\sqrt{\sum_{j=1}^{m}(f_j(x_i) - f_j(p_k^*))^2}\right\} \tag{6}$$

, where m is the number of the target functions, $p_k^* \in P^*$, P^* is the set of individuals with the smallest sequential value in the Pareto optimal set obtained and $k=1,2,\ldots,q, q=|P^*|$.

B. Selection of Control Parameters

The DE algorithm has three control parameters: population size NP, differential vector scale factor F and crossover probability CR, of which F and CR have important impact over the optimization performance.

F is used for controlling the impact that the differential vector has over the mutated individual. When the value of F is relatively large, great disturbance will be resulted in and this is beneficial to maintaining the diversity of the populations and global convergence. When the value of F is relatively small, small disturbance will be resulted in and the scale factor plays a part of local fined search, which is beneficial to carrying out local search and accelerating convergence. According to such rule of F, a fixed F value is not used in evolution any more but F value is determined according to the current number of generations in evolution. During initial phase of evolution, a relatively large F value is used to carry out decentralized search and this is beneficial to searching for information in the unknown space and maintaining the diversities of solutions. With evolution going on, solution individuals approach convergence, F value gradually decreases and a small-range centralized search is carried out to accelerate convergence. For this reason, the parameter F is designed to be a linear decreasing function correlated to the number of generations in evolution:

$$F = F_u - (F_u - F_l) \cdot t/T \tag{7}$$

, where F_u=0.6, F_l=0.4, t is the current number of genreations in evolution and T is the maximum number of generations in evolution.

Usually, a real number between 0 and 1 is gotten in advance as the crossover factor CR to control the contribution made by $v_i(t+1)$ and $x_i(t)$ to $u_i(t+1)$. During initial phase of evolution, the CR value should be relatively large to enable the individual to accept the genes of more mutated individuals and thus accelerate the evolution of this individual. With the evolution going on, the CR gradually decreases to avoid the gene structure of the individual from being damaged.

We use the following method to make CR decrease with the number of generations in evolution linearly:

$$CR = CR_u - (CR_u - CR_l) \cdot t/T \tag{8}$$

, where CR_u=0.3, CR_l=0.1, t is the current number of genreations in evolution and T is the maximum number of generations in evolution.

C. Selection Operation

In common DE, the generated test individual $u_i(t+1)$ is compared with the target individual $x_i(t)$ with respect to domination relationship, and apparently such operation is not accurate for MOEA because the superiority of each individual in MOEA is related to the status of the entire population. To avoid the above mentioned problem, this algorithm employs a selection operation in which the generated test individual is directly incorporated into the temporary population and the temporary population and original population are integrated as a hybrid population after the mutation and crossover operations are completed and the partition-based elitist preserving mechanism is applied to achieve the selection of next-generation individuals.

Although the partition-based elitist preserving mechanism is beneficial to retaining superior individuals and increasing the overall evolution level of the population, it will easily generate many similar Pareto optimal solutions due to its consideration of superiority excluding the solution distribution and the Pareto optimal set with wide distribution cannot be easily generated. To enable the individuals on the current Pareto frontier plane to be expanded to the entire Pareto frontier plane and be distributed as uniformly as possible, a partition-based elitist preserving mechanism is proposed to achieve selection of elitist individuals by selecting target spaces in a partitioned manner.

The specific method is as follows: select a target function as the index from the target space, divide the target space uniformly into several sub-spaces and determine the number of individuals (s) that should be selected from each sub-space according to the size of child population (N) and the number of spaces divided (i): $s = \lfloor N/i \rfloor$.

A series of Pareto solution sets are obtained from each sub-space according to the domination relationship in the order of F_1, F_2 , …, with F_1 at the highest level. If the number of F_1 is larger than s, F_1 is crowd sorted and those individuals with long crowded distance will enter P_{t+1} with priority. If the number of F_1 is smaller than s, then all members of F_1 are selected into the population P_{t+1}, the remaining members of which will be selected from F_2, F_3,… until the number of P_{t+1} reaches s. If the number of the members of the selected set is larger than the number of individuals to be selected, crowd sorting is carried out to maintain the diversity of the populations and the individuals with long crowded distance will enter P_{t+1} with priority.

In selection, if the number of individuals of a sub-space is smaller than s, then the individuals remaining to be selected are allocated to the adjacent upper-layer sub-space. If the number of individuals that can be selected is smaller than the population size, individuals are randomly selected from the initial population to complement the size of the population.

D. Basic Algorithm Procedures

Based on the above, the basic procedures of AM+PS-MODE can be expressed as follows:

Step 1: Randomly generate the initial population P_t. The population size is N, the initial number of populations in evolution t=0 and the maximum number of populations in evolution is T;

Step 2: Sort P_t by non-inferiority and calculate the sequential value and crowded distance of each individual;

Step 3: Decide the sequential value of the current target individual. If the sequential value is "1", apply the DE/rand/1/bin mutation strategy and crossover operation to obtain the test individual, otherwise apply the DE/current to best /2/bin mutation strategy and crossover operation to obtain the test individual;

Step 4: Integrate the initial population and test individuals to generate the hybrid population R_t;

Step 5: Apply partition-based elitist preserving mechanism to R_t and select N individuals to P_{t+1};

Step 6: t=t+1, if t<=T, return to Step 2, otherwise end.

IV. Numerical Experiment

A. Test Functions and Parameter Setting

To test the performance of AM+PS-MODE, 5 ZDT problems [10] and 3 DTLZ problems[11] widely used in the EMO field are employed to test the performance of the algorithm in this paper. The ZDT problems are 2-D objectives and there are 30 decision variables for ZDT1~ZDT3 and there are 10 decision variables for ZDT4 and ZDT6. The DTLZ problems are 3-D objectives and there are 12 decision variables for DTLZ1~DTLZ3.

In experiment, the parameters are set as follows: coding is in the real number mode. In case of 2-D objectives, the population size is set to be 100 and the number of generations in evolution is 250. In case of 3-D objectives, the population size is set to be 150, the number of generations in evolution is 500,F_u=1.0, and F_l=0.8.

B. Evaluation Criteria for Algorithm Performance

To evaluate the performance of multi-objective algorithm, γ criteria corresponding to the multi-objective algorithm are employed to measure the convergence of the populations and Δ criteria are employed to measure the diversity of the populations.

Assuming Z is a non-inferiority solution set obtained through a certain algorithm from the target space and Z′ is a set uniformly distributed on the Pareto frontier.

(1) Convergence criteria γ(Z,Z′)[12]: measure the degree at which Pareto frontier is approached. The definition is as follows:

$$\gamma(Z,Z') = \frac{1}{|Z|}\sum_{z\in Z}\min\{\|z-z'\|, z'\in Z'\} \tag{9}$$

A smaller γ value means a higher degree at which the algorithm approaches the Pareto optimal set.

(2) Diversity criteriaΔ(Z,Z′)[12]: measure the diversity of the distribution of solutions in the obtained set in the target space. The definition is as follows:

$$\Delta(Z,Z') = \frac{d_f + d_l + \sum_{i=1}^{|Z|-1}|d_i - \bar{d}|}{d_f + d_l + (|Z|-1)\bar{d}} \tag{10}$$

d_i is the euclidean distance between two continuous vectors, $\bar{d}$ is the average of all d_i, d_f and d_l are the euclidean distance between the extremal vector and the boundary vector obtained through the algorithm.

Δ indicator reflects whether the non-inferior solutions can be uniformly distributed on the entire frontier plane. A smaller *Δ* value indicates better diversity of non-inferior solutions.

C. Experiment Results and Analysis

Figure 1 and Figure 2 respectively compare the experimental Pareto curve with theoretical Pareto curve of 5 ZDT test functions and the experimental Pareto curve with theoretical Pareto curve of 3 DTLZ test functions.

From the experimental curve, it can be intuitively seen that the solutions obtained through AM+PS-MODE algorithm can converge to the optimal region of the problems and solutions of the set can uniformly cover the entire optimal region of the problems and the set has very good convergence and diversity.

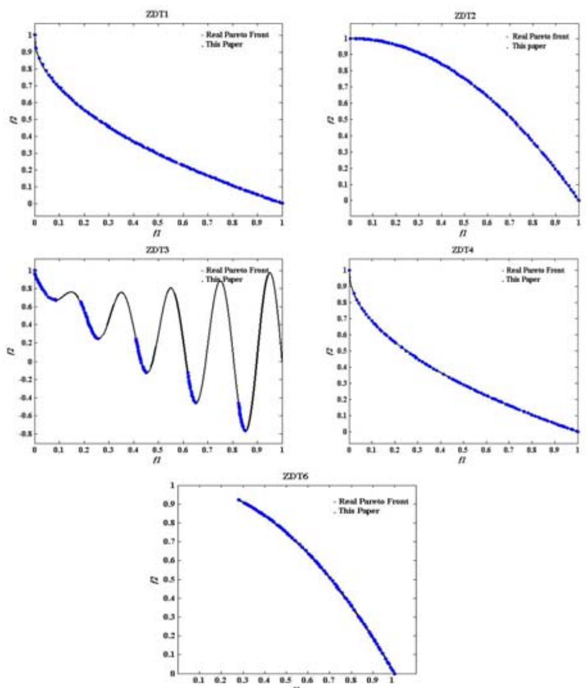

Fig.1 Pareto curve of five ZDT test functions

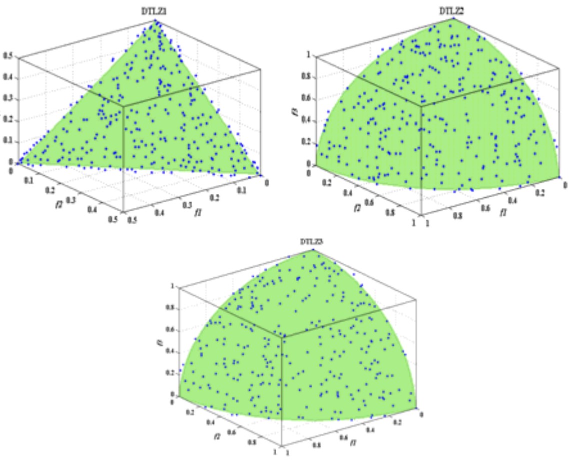

Fig.2 Pareto curve of three DTLZ test functions

In addition to the Pareto frontier graphic comparison of 5 ZDT test functions, to further quantitatively evaluate the performance of the AM+PS-MODE algorithm, the AM+PS-MODE algorithm is compared with such representative multi-objective evolution algorithms as NSGA-II[12], SPEA2[13], DEMO in terms of convergence and diversity.

To reduce the impacts that random factors have, the AM+PS-MODE algorithm runs each problem separately 10 times and each iteration involves 250 generations. Table 1 and Table 2 provide the mean and variance of the convergence and diversity. In each cell, the mean is in the first row and the variance is in the second row.

TABLE 1
MEAN AND VARIANCE OF THE CONVERGENCE METRIC Γ

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
AM+PS -MODE	0.000901 0	0.000741 0	0.0052 0	0.000695 0	0.0043 0
NSGA-II	0.033482 0.004750	0.072391 0.031689	0.114500 0.007940	0.513053 0.118460	0.296564 0.013135
SPEA2	0.023285 0	0.16762 0.000815	0.018409 0	4.9271 2.703	0.232551 0.004945
DEMO	0.001083 0.000113	0.000755 0.000045	0.001178 0.000059	0.001037 0.000134	0.000629 0.000044

TABLE 2
MEAN AND VARIANCE OF THE DIVERSITY METRIC Δ

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
AM+PS -MODE	0.3135 0.000195	0.2647 0.000145	0.6456 0.000418	0.3612 0.000672	0.6455 0.000193
NSGA-II	0.390307 0.001876	0.430776 0.004721	0.738540 0.019706	0.702612 0.064648	0.668025 0.009923
SPEA2	0.154723 0.000873	0.33945 0.001755	0.4691 0.005265	0.8239 0.002883	1.04422 0.158106
DEMO	0.325237 0.030249	0.329151 0.032408	0.309436 0.018603	0.359905 0.037672	0.442308 0.039225

From Tables 1 and 2, it can be seen that the AM+PS-MODE algorithm is superior to NSGA-II, SPEA2, DEMO both convergence and diversity for ZDT1 and ZDT2. For ZDT3, the AM+PS-MODE algorithm s superior to NSGA-II and SPEA2 in terms of convergence, and is superior to NSGA-II in terms of diversity. For ZDT4, the AM+PS-MODE algorithm is superior to NSGA-II, SPEA2, DEMO in terms of convergence but is superior to NSGA-II, SPEA2 and is similar to DEMO in terms of diversity. For ZDT6, the AM+PS-MODE algorithm is obviously superior to NSGA-II and SPEA2 in terms of convergence and diversity but is slightly inferior to the DEMO.

In addition, from table it can be easily seen that the standard variance that AM+SS-MODE has against 5 test functions is obviously smaller than the standard variance that NSGA-II, SPEA2, DEMO have against 5 test functions in terms of convergence and diversity of the solution set and this indicates that the performance of the algorithm in this paper is more stable.

The above mentioned quantitative measure results also indicate that applying adaptive mutation strategies and partition selected search in search not only enhances the search capability of the algorithm but also helps maintain the diversity of the populations.

V. CONCLUSIONS

This paper uses DE to solve the MOPs and this improves the mutation and selection operations in DE. And the differential vector scale factor F and crossover probability CR are designed to be the linear decreasing functions correlated with the number of generations in evolution. This algorithm preserves the DE's advantages of easy operation and fast convergence and is improved in terms of both convergence and diversity. The results from the numerical simulation of classical test functions indicate that the AM+PS-MODE shows good effects in terms of both approximation and uniformity of the solution set and has good stability. This indicates that the multi-objective differential evolution algorithm based on adaptive mutation strategies and partition selected search has certain advantages in solving MOPs and is a practical and effective method to solve MOPs.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (61003066,61070033,61070164), Doctoral Program of the Ministry of Education (20090172120035), Science and Technology Planning Project of Guangdong Province, China (2010B010600025), Natural Science Foundation of Guangdong Province, China (S2012010008767, 815106 32010000022).

REFERENCES

[1] K. Deb, "Multi-Objective Optimization Using Evolutionary Algorithms", Chichester: John Wiley & Sons ,2001.

[2] CA. Coello Coello, GB. Lamont, and DA. Van Veldhuizen, "Evolutionary Algorithms for Solving Multi-Objective Problems", Springer, New York, second edition, September 2007.

[3] R. Storn, K. Price, "Differential evolution-A simple and Efficient heuristic for global optimization over continuous spaces", Journal of Global ptimization, vol.11, No.4,1997, pp.341-359.

[4] K. Price, "Differential evolution vs. the functions of the 2nd ICEO", Proc of the 1997 IEEE International Conference on Evolutionary Computation, Indianapolis, 1997,pp.153-157.

[5] H.A. Abbass, R. Sarker, C. Newton,"PDE: A Pareto-Frontier differential evolution approach for Multi-objective optimization problems", In Proc of the Congress on Evolutionary Computation, Piscataway, 2001,pp.971-978.
[6] N.K. Madavan, ".Multiobjective Optimization using A Pareto Differential Evolution Approach", In: Proc of the congress on Evolutionary Computation (CEC'2002), Piscataway: IEEE Service Center, 2002, pp.1145-1150.
[7] T. Robic, B. Filipic, "DEMO: Differential Evolution for Multiobjective Optimization", In: Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO2005),Guanajuato, Springer, 2005,pp. 520-533.
[8] V.L. Huang, S.Z. Zhao, R. Mallipeddi, P.N. Suganthan, "Multi-objective optimization using self-adaptive differential evolution algorithm", In:Proc of the congress on Evolutionary Computation(CEC'2009),pp.190-194.
[9] A.A. Montaño, C.A.Coello Coello, E. Mezura-Montes, "MODE-LD+SS: A Novel Differential Evolution Algorithm Incorporating Local Dominance and Scalar Selection Mechanisms for Multi-Objective Optimization", In: Proc of the congress on Evolutionary Computation (CEC'2010),pp.1-8.
[10] E. Zitzler, K. Deb, L. Thiele, "Comparison of multi-objective evolutionary algorithms: Empirical results", Evolutionary Computation, vol.8, No.2, 2000, pp.173-195.
[11] K. Deb, L. Thiele, M. Lsumanns, E. Zitzler , "Scalable multi-objective optimization test problems". In: Fogel DB,ed. Proc. of the IEEE Congress on Evolutionary Computation, CEC 2002, Piscataway: IEEE Service Center,2002,pp.825-830.
[12] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan ,"A Fast and Elitist Multi-objective Genetic Algorithms: NSGA-II ",IEEE Transactions on Evolutionary Computation, vol.6, No.2,2002,pp.182-197.
[13] E. Zitzler, M .Laumanns, and L.Thiele,"SPEA2: Improving the Strength Pareto Evolutionary Algorithm", Swiss Federal Institute of Technology,Lausanne, Switzerland, Technical Report TIK Rep 103, May 2001.
[14] Wei-Ping Lee,Chang-Yu Chiang,"A Self-Adaptive Differential Evolution Algorithm with Dimension Perturb Strategy",Journal of Computers ,vol.6, No.3,2011.
[15] Lei Peng,Yuan zhenWang , Guang ming Dai, Zhong sheng Cao,"A Novel Differential Evolution with Uniform Design for Continuous Global Optimization", Journal of Computers ,vol.7, No.1,2012.
[16] Jing feng Yan,Chao feng Guo,Wen yin Gong,"Hybrid Differential Evolution with Convex Mutation", Journal of Software, vol.6, No.7,2011.