Adaptive Multi-objective Differential Evolution with Stochastic Coding Strategy

Jing-hui Zhong and Jun Zhang (Corresponding Author)
Department of Computer Science, Sun Yat-sen University
Key Laboratory of Digital Life, Ministry of Education
Key Laboratory of Software Technology, Education Dept. of Guangdong Province, P.R. China
junzhang@ieee.org

ABSTRACT
Many real-world applications can be modeled as multi-objective optimization problems (MOPs). Applying differential evolution (DE) to MOPs is a promising research topic and has drawn a lot of attention in recent years. To search high-quality solutions for MOPs, this paper presents a robust adaptive DE (termed AS-MODE) with following two features. First, a stochastic coding strategy is used to improve the solution quality. This coding strategy represents each individual by a stochastic region, which enables the algorithm to fine-tune solutions efficiently. Second, a probability-based adaptive control strategy is utilized to reduce the influence of parameter settings. The adaptive control strategy associates each parameter with a candidate value set. Better candidate values would have higher selection probabilities to generate new individuals. The performance of the proposed AS-MODE is compared with several highly regarded multi-objective evolutionary algorithms. Simulation results on ten benchmark test functions with different characteristics reveal that AS-MODE yields very promising performance.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]:Problem Solving, Control Methods and search-Heuristic methods; G.1.6 [Numerical Analysis]: Optimization-Global optimization

General Terms
Algorithms, Design, Experimentation.

Keywords
Adaptive parameter control, differential evolution, evolutionary algorithm, multi-objective optimization, stochastic coding.

1. INTRODUCTION
Many real-world applications require optimizing multiple conflicting objectives simultaneously [1]. These problems are called multi-objective optimization problems (MOPs), which can be mathematically expressed as

$$\begin{aligned} &\text{Min}\{f_1(x), f_2(x), ..., f_m(x)\} \\ &\quad \text{subject to } x \in \Omega \end{aligned} \tag{1}$$

where Ω is the search space and $f_i : \Omega \rightarrow R$ is the *i*-th objective function. As objectives in MOPs are conflicting, no single solution is able to optimize all objectives at the same time. Hence in most cases, MOP is addressed by finding a set of alternative solutions which provide different trade-offs among the objectives. The recent years have seen an increasing interest in applying evolutionary algorithms (EAs) to MOPs. EAs are very suitable to solve MOPs, because they are population-based algorithms that can provide multiple alternative solutions simultaneously [1]-[5].

Differential evolution (DE) is a new class of evolutionary algorithm proposed by Storn and Price for single-objective continuous optimization [6]. With a simple principle and few parameters, DE has become one of the most popular optimization techniques and has been successfully applied to a wide range of applications such as the power flow problem [7] and the microwave filter design [8]. Over the past few years, extending DE for multi-objective optimization has drawn a lot of attention. In[9], Abbass *et al* proposed a Pareto differential evolution (PDE) for multi-objective optimization, where only non-dominated solutions are utilized to produce offspring. PDE is examined on two test problems and is reportedly better than the Strength Pareto evolutionary algorithm (SPEA), a state-of-the-art multi-objective evolution algorithm (MOEA). In[10], Babu *et al.* proposed a new multi-objective differential evolution (MODE). The authors utilized the weighting factor method and penalty function method to solve bi-objective problems. Later, Lampinen extended the selection operator in the basic DE algorithm and formed the generalized differential evolution (GDE) [11] for MOPs. The GDE is reported to have good convergence properties, but the distribution of alternative solutions need to be improved. Later, an enhanced GDE (GDE2) was presented in [12] where the crowdedness was taken into account in the selection process. The third version of GDE (GDE3) was proposed in [13], where new individuals were stored in the population. At the beginning of the next generation, the non-dominated sorting strategy and crowding-distance operations are carried out to reduce the population size. To reduce the impacts of parameter settings, Huang *et al*. [14] presented a self-adaptive MODE, termed MOSaDE, which adaptively adjusted the parameter settings. The MOSaDE was further improved by using an objective-wise learning strategy in [15].

Existing researches have shown good potential in applying DE to MOPs. However, there are still two significant and challenging tasks to be addressed. First, the solution quality is not high enough, especially for problems containing linkages [13]. Second, the parameter settings have significant influences on the performance of MODEs [14][15]. To address these two issues, this paper presents an adaptive MODE with stochastic coding strategy (termed AS-MODE).

GECCO'11, July 12–16, 2011, Dublin, Ireland.

The proposed AS-MODE improves the solution quality by using a stochastic coding strategy [16]. In the stochastic coding strategy, each individual is represented by a stochastic region in the search spaces. The stochastic region is defined by a normal distribution. The center of the region is used for fitness evaluation, while the size of the region determines the step sizes for local refinement. By adaptively adjusting the centers and sizes of stochastic region, individuals are more sensible to their surrounding regions. In this way, the algorithm can explore the search space in a region-by-region manner which is very efficient to fine-tune the solutions. Meanwhile, the AS-MODE reduces the influences of parameter settings by using a novel probability-based adaptive control strategy. In AS-MODE, the value of each parameter is randomly selected from a related discrete set before generating a new individual. When all new individuals are created, the efficiency of each candidate value in generating promising individuals is measured. Better candidate values would have higher selection probabilities in the next generation. The proposed AS-MODE is validated by testing ten benchmark functions used in CEC09 competition [17]. The experimental results reveal that the performance of the proposed AS-MODE is very promising.

The rest of the paper is organized as follows. Section II describes the framework of DE. Section III illustrates the implementations of the proposed AS-MODE. The experiment studies are presented in Section IV. At last, Section V draws the conclusions.

2. DIFFERENTIAL EVOLUTION

Differential Evolution (DE) is a stochastic evolution algorithm (EA) which is first proposed by Storn and Price. Like many other EAs, DE initializes a set of individuals at the beginning, and then generates new individuals by combination of other individuals which are randomly selected from the current population.

At each generation, a mutation operation is carried out to generate a set of new individuals, as expressed by

$$y_{G+1}^{i} = x_G^a + F \cdot (x_G^b - x_G^c), i = 1,2,...,N \quad (2)$$

where N is the population size, $a,b,c \in [1,N]$ are random integers with $a \neq b \neq c$, F is the scaling factor, G represents the current generation. To bring in more diversity, a set of trial individuals $u_G^i, i = 1,2,...,N$ are generated by combines x_G^i and y_{G+1}^i . The j-th variable value of u_G^i can be obtained by

$$u_G^i(j) = \begin{cases} y_{G+1}^i(j), & \text{if } rand(0,1) < CR \text{ or } j = k \\ x_G^i(j) & \end{cases} \quad (3)$$

where $CR \in [0,1]$ is the crossover rate, k is a random integer, *rand* (0,1) returns a random number uniformly distributed between 0 and 1. Finally, the selection operation creates a new population by using the following rule.

$$x_{G+1}^i = \begin{cases} u_G^i, & \text{if } f(u_G^i) \leq f(x_G^i) \\ x_G^i, & \text{otherwise} \end{cases} \quad (4)$$

where $f(u_G^i)$ is the objective value of u_G^i . It should be noted that the above scheme is just one of the simple and effective DE frameworks. There are various DE variants which use different mutation strategies. More details can be found in [6] and [18].

3. THE PROPOSED AS-MODE

3.1 Coding Mechanism

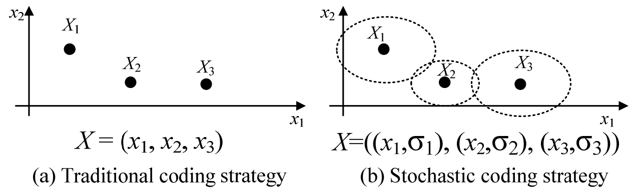


Figure 1. Coding mechanism

In most traditional EAs, each individual represents a feasible solution in the search space, as shown in Fig. 1 (a). As the individuals contain no information about its surrounding area, it may become difficult and inefficient in fine-tuning the values of decision variables to produce the best results.

To improve the fine-tuning capability, the proposed AS-MODE adopts a stochastic coding strategy. As shown in Fig. 1(b), each individual is represented by a stochastic region. The stochastic region is defined by a normal distribution, thus we can use the following formulary to express an individual.

$$X=((x_1,\sigma_1), (x_2,\sigma_2), \cdots, (x_D,\sigma_D)) \quad (5)$$

The mean vector $[x_1, x_2 ,\dots, x_D]$ represents the center of the region for fitness evaluation, while the variance vector $[\sigma_1, \sigma_2 ,\dots, \sigma_D]$ defines the size of the region for local refinement. During the evolution, individuals would be updated by sampling solutions within the stochastic region. In this way, individuals are more sensible to their surrounding regions and the algorithm can fine-tune solutions efficiently.

3.2 Adaptive Strategy

Finding suitable parameter settings is a critical issue in DE designing. Frequently, parameter setting has significant influence on the behavior of the algorithm and the best parameter setting is problem dependent. Traditional trial-and-error method requires multiple optimization runs, which may not be convenient in practices. In this section, we present a probability-based strategy to adaptively control the mutation factor F and the crossover probability CR.

First of all, two discrete sets are created as

$$\begin{cases} S_F = \{a_1, a_2, ..., a_A\}, 0 \leq a_1 < a_2 < ... < a_A \leq 2 \\ S_{CR} = \{b_1, b_2, ..., b_B\}, 0 \leq b_1 < b_2 < ... < b_B \leq 1 \end{cases} \quad (6)$$

where A and B were respectively the number of elements in S_F and S_{CR}. Before generating a new individual, the values of F and CR are respectively selected from these two sets. Our goal is to make better parameter values have higher probability of being selected. Assuming that new individuals produced when using better parameter values are more likely to survive, we set the selection probability of each parameter value according to its efficiency in producing good individuals surviving in the population. Specifically, the selection probability of each parameter value is computed by

$$\begin{cases} p_{Fi} = c_{Fi} / \sum_{j=1}^{A} c_{Fj}, i = 1,2,...,A \\ p_{CRi} = c_{CRi} / \sum_{j=1}^{B} c_{CRj}, i = 1,2,...,B \end{cases} \quad (7)$$

where c_{Fi} and c_{CRi} are respectively the number of individuals in the current population which are generated by using the i-th candidate value in S_F and S_{CR}. For example, if 20% of the individuals in the current population are generated with $CR = 0$, then $CR = 0$ would have a probability of 0.2 to be chosen to generate new individuals in the next generation.

3.3 Algorithm Framework

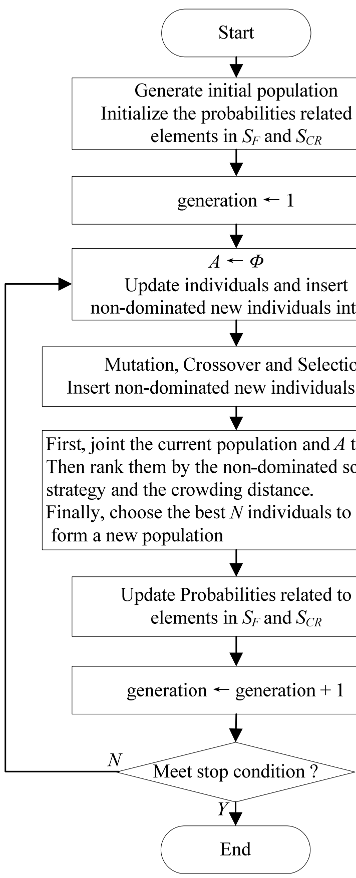


Figure 2. Flowchart of AS-MODE.

The flowchart of the proposed algorithm is shown in Fig. 2, and the implementation details are described as follows.

1) Step 1 - Initialization

This step aims to generate an initial population. Let $P(0)$ be the initial population. It contains N individuals $\{I_1, I_2, \ldots, I_N\}$, as expressed by

$$P(0) = \begin{bmatrix} I_1 \\ I_2 \\ \ldots \\ I_N \end{bmatrix} = \begin{bmatrix} (x_{11},\sigma_{11}),(x_{12},\sigma_{12}),...,(x_{1D},\sigma_{1D}) \\ (x_{21},\sigma_{21}),(x_{22},\sigma_{22}),...,(x_{2D},\sigma_{2D}) \\ \ldots \\ (x_{N1},\sigma_{N1}),(x_{N2},\sigma_{N2}),...,(x_{ND},\sigma_{ND}) \end{bmatrix} \quad (8)$$

For each individual I_i, i=1,2,…,N, the mean vector is randomly sampled from the search space, as expressed by

$$x_{ij} = rand(L_j, U_j) \quad (9)$$

where L_j and U_j are respectively the lower bound and the upper bound of the j-th variable, $rand(a,b)$ returns a random number uniformly distributed within a and b. Notice that, the initial mean vector could be set according to problem-specific knowledge, so as to improve the search efficiency. As for the variance vector, we empirically initialize it by

$$\sigma_{ij} = (U_j - L_j)/10 \quad (10)$$

After generating the mean vector and variance vector of an individual, values of F and CR associated with it are initialized by randomly choosing values from the candidate sets.

2) Step 2 – Updating operation

The main objective of this process is to fine-tune the individuals by sampling neighbor solutions within their stochastic regions. For individual $I_i = ((x_{i1},\sigma_{i1}),(x_{i2},\sigma_{i2}),...,(x_{iD},\sigma_{iD}))$, its neighbor $I_i^{'} = ((x_{i1}^{'},\sigma_{i1}^{'}),(x_{i2}^{'},\sigma_{i2}^{'}),...,(x_{iD}^{'},\sigma_{iD}^{'}))$ is generated by (11) and (12).

$$x_{ij}^{'} = \begin{cases} x_{ij} + N(0,1)\cdot\sigma_{ij}, & \text{if } rand(0,1) < p \text{ or } l = j \\ x_{ij}, & \text{otherwise} \end{cases} \quad (11)$$

$$\sigma_{ij}^{'} = \sigma_{ij} \quad (12)$$

where p is a parameter within [0,1], l is a random integer between 1 and D (D is the dimension of the problem), $N(0,1)$ returns a random number with a standard normal distribution. The random number l is used to ensure that there is at least one variable in $I_i^{'}$ that is different from the one in I_i. The parameter values associated with $I_i^{'}$ are set the same as those with I_i. If $I_i^{'}$ dominates I_i, then I_i would be replaced by $I_i^{'}$ immediately. Otherwise, if $I_i^{'}$ and I_i are non-dominated by each other, $I_i^{'}$ would be inserted into the archive A. This process is repeated for M times. If none of these M neighbors can dominate I_i, the variance values of I_i would be reduced. Otherwise, the variance values of I_i would be extended, as expressed by

$$\sigma_i^{'} = \begin{cases} \sigma_i \cdot \lambda, & \text{reduce case} \\ \sigma_i / \lambda, & \text{extend case} \end{cases} \quad (13)$$

where $\lambda \in (0,1)$ is the reducing rate. In order to reduce computational cost, we use the Roulette wheel selection strategy to select K better individuals to generate neighbors.

3) Step 3 –Mutation，Crossover and Selection

This step applies the DE operators to generate N new individuals. For each target individual $I_i = ((x_{i1},\sigma_{i1}),(x_{i2},\sigma_{i2}),...,(x_{iD},\sigma_{iD}))$ in the current population, the mutation and crossover operations generate a new individual $I_i^{'} = ((x_{i1}^{'},\sigma_{i1}^{'}),(x_{i2}^{'},\sigma_{i2}^{'}),...,(x_{iD}^{'},\sigma_{iD}^{'}))$ by

$$x'_{ij} = \begin{cases} x_{aj} + F \cdot (X_{bj} - X_{cj}), & \text{with probability } CR \\ x_{ij}, & \text{with probability } 1 - CR \end{cases} \quad (14)$$

$$\sigma'_{ij} = \begin{cases} \sigma_{aj} + F \cdot (\sigma_{bj} - \sigma_{cj}), & \text{with probability } CR \\ \sigma_{ij}, & \text{with probability } 1 - CR \end{cases} \quad (15)$$

where F and CR are two parameter values sampled according to (7); a, b, c with $a \neq b \neq c$, are three random individual indexes. These three indexes are selected by using the tournament selection strategy. We have made additional experiments to compare the effect of different selection strategies and we find that the tournament selection strategy performs better that the traditional random method. If x'_{ij} exceeds the search range, it would set to be the nearest value in the search range. Meanwhile, if the variance value σ'_{ij} is larger than $\sigma_{\max}$, σ'_{ij} would set equal to $\sigma_{\max}$. Here $\sigma_{\max}$ is computed by

$$\sigma_{\max} = (\frac{U - L}{10}) \cdot \frac{MAXEVALS - evals + 1}{MAXEVALS} \quad (16)$$

where $MAXEVALS$ is the maximum number of evaluations, $evals$ is the number of evaluations in the current state. In Eq.(16), we reduce the variance values as the evolution goes on, so as to fine-tune solutions more efficiently.

After obtaining the mean vector and the variance vector of a new individual, its fitness value is evaluated. If the new individual dominates I_i, I_i would be replaced by the new individual. Otherwise, if I_i and the new individual are non-dominated by each other, the new individual would be inserted into the extra set A.

4) Step 4 – Create New Population

The aim of this step is to choose N promising individuals to form a new population. First of all, a pool of individuals U is created by unioning the current population and A. Then the non-dominated sorting strategy and the crowding-distance operation [4] are carried out to rank all individuals in U and the best N individuals are chosen from U one by one.

5) Step 5 – Update probabilities of candidate values.

In this step, the adaptive control strategy given in subsection 3.2 is utilized to update the probability of each candidate values. To ensure that each candidate value has at least a small probability to be selected, if the value of c_{Fi} (or c_{Cri}) exceeds the range of $[C_{\min}, C_{\max}]$, it would set to be the nearest value in the range.

There is a repetition from *Step2* to *Step5* until the termination criteria are met.

4. EXPERIMENTS AND COMPARISONS

4.1 Experimental Settings

In this section, the proposed AS-MODE is validated by testing ten benchmark functions used in CEC 2009 multi-objective competition, including seven 2-objective functions (i.e., UF1-UF7) and three 3-objective functions (i.e., UF8-UF10) [17]. These test problems contains variable linkages and are very difficult for traditional MOEAs to solve. We choose them to investigate the performance of AS-MODE in finding high-quality solutions for complicated multi-objective problems. Moreover, these functions have different features and need different parameter settings of MODE to solve them. Thus they are suitable to examine the effectiveness and the efficiency of the proposed adaptive control strategy.

The initial parameter settings of the proposed AS-MODE are listed in Table 1. We compare the proposed algorithm with two multi-objective differential evolution algorithms. The first one is the well-known generalized differential evolution 3 (GDE3), which is a simple but effective MODE with fixed parameter settings [13]. The second is the OW-MOSaDE whose parameter settings is adaptively adjusted by using an objective-wise learning strategy [15]. We also compare AS-MODE with NSGA-II-ls [19] and MOEA/D [17][20]. As evolution algorithms are stochastic algorithms that provide different results in different runs, we run AS-MODE for 30 times on each test problem and use the average inverted generational distance (IGD) for comparison.

The IGD is defined as follows [5]: *Suppose P* is a set of Pareto solutions which is uniformly distributed along the Pareto front of a MOP, and P is a set of approximation Pareto solutions. The IGD from P* to P is defined as:*

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (17)$$

where $d(v, P)$ represents the distance between v and the points in P. $|P^*|$ denotes the number of solutions in P^*. It has been shown that the IGD can measure both the diversity and convergence of P, when $|P^*|$ is set large enough. In the experiment, the algorithms terminate when the number of function evaluations reaches 3×10^5. The number of points selected to compute IGD are 100 for 2-objective functions and 150 for 3-objective functions.

Table 1. Parameter settings of AS-MODE

parameter	*value*	*summary*
N	200	Population size
T	10	Tournament size
K	40	Number of individuals to be updated
M	5	Number of update attempts
λ	0.5	Stochastic region shrink rate
$[C_{min}, C_{max}]$	[1, 50]	Lower and upper bounds of c_{Fi} and c_{CRi}
S_F	{0.5,1,1.5}	Candidate values of F
S_{CR}	{0, 0.5, 1}	Candidate values of CR

4.2 Comparison Results

In this subsection, we compare AS-MODE with the GDE3, OW-MOSaDE, NSGA-II-ls and MOEAD. The experimental results are listed in Table 2, where "Mean" represents the average IGD values of the 30 runs and "Std." represents the standard deviations of the IGD values. It can be observed that the proposed AS-MODE has found the best (lowest) IGD values on seven of the ten functions (i.e., UF1, UF2, UF3, UF4, UF7, UF9, and UF10). Moreover, the results provided by AS-MODE on UF5, UF6, and UF8 are also very promising. According to the standard deviations, the proposed AS-MODE performs very stably in finding high quality solutions. Overall, AS-MODE generally outperforms other algorithms, in terms of the IGD values and the stability of performance. Fig. 3 shows the best approximation

Table 2 IGD values of GDE3, OW-MOSade, NSGA-II-ls, MOEAD, and AS-MODE on 10 test problems

F	GDE3		OW-MOSaDE		NSGA-II-ls		MOEAD		AS-MODE	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
UF1	0.00534	0.000342	0.0122	0.0012	0.01153	0.0073	0.00435	0.00029	**0.00405**	**5e-005**
UF2	0.01195	0.001541	0.0081	0.0023	0.01237	0.009108	0.00679	0.00182	**0.00438**	**0.000261**
UF3	0.10639	0.012900	0.103	0.0190	0.10637	0.06864	0.00742	0.00589	**0.00392**	**0.000138**
UF4	0.0265	0.000372	0.0513	0.0019	0.0584	0.005116	0.06385	0.00534	**0.02378**	**0.001148**
UF5	**0.03928**	**0.003947**	0.4303	0.0174	0.5657	0.1827	0.18071	0.06811	0.10634	0.026610
UF6	0.25091	0.019573	0.1918	0.0290	0.31032	0.19133	**0.00587**	**0.00171**	0.08633	0.059950
UF7	0.02522	0.008891	0.0585	0.0119	0.02132	0.01946	0.00444	0.00117	**0.00393**	**0.000216**
UF8	0.24855	0.035521	0.0945	0.0244	0.0863	0.01243	**0.0584**	**0.00321**	0.09796	0.039460
UF9	0.08248	0.022485	0.0983	0.0885	0.0719	0.04504	0.07896	0.05316	**0.04959**	**0.035446**
UF10	0.43326	0.012323	0.743	0.0384	0.84468	0.1626	0.47415	0.07360	**0.26915**	**0.069972**

Table 3 IGD values of A-MODE, OW-MOSaDE, S-MODE and AS-MODE on 10 test problems

F	A-MODE	OW-MOSaDE	S-MODE (0.5, 1)	S-MODE (0.5, 0)	S-MODE (0.5, 0.5)	S-MODE (0.1, 0.9)	S-MODE (0.9, 0.1)	AS-MODE
UF1	0.00646	0.0122	**0.00401**	0.01107	0.02331	0.00732	0.02484	0.00405
UF2	0.00778	0.0081	0.00562	0.00985	0.01029	0.00828	0.0104	**0.00438**
UF3	0.00655	0.103	0.00508	0.11268	0.11273	0.07294	0.10708	**0.00392**
UF4	0.03197	0.0513	0.03046	**0.02284**	0.02971	0.03156	0.02809	0.02378
UF5	0.1174	0.4303	0.22590	0.13193	0.1659	0.23067	0.16492	**0.10634**
UF6	**0.06805**	0.1918	0.13442	0.08476	0.11751	0.17653	0.13815	0.08633
UF7	0.00771	0.0585	**0.00384**	0.01105	0.01458	0.00721	0.03213	0.00393
UF8	0.11115	**0.0945**	0.11841	0.17481	0.1598	0.11318	0.12712	0.09796
UF9	0.09879	0.0983	0.16010	0.05996	0.06908	0.15571	0.09127	**0.04959**
UF10	0.32457	0.743	0.31780	0.30163	0.37881	0.33450	0.33106	**0.26915**

Pareto fronts found by the AS-MODE in 30 runs. The results demonstrate that the proposed AS-MODE can find good approximation Pareto fronts on UF1, UF2, UF3, UF4, UF7, UF8 and UF9, but performs relatively poor on UF5, UF6, and UF10. Nevertheless, compared with the published results in [15][13], [19], and [20], the approximation Pareto fronts found by AS-MODE on these three functions are also very promising.

4.3 Adaptive Control Strategy Investigation

In this subsection, we investigate the influence of the adaptive parameters control strategy and the stochastic coding strategy. First, we remove the updating operation in AS-MODE and form a simplified A-MODE. Then we remove the adaptive control strategy from AS-MODE to form a simplified S-MODE. S-MODE with different parameter settings are carried out and compared with AS-MODE. The final results are listed in Table 3, where S-MODE (a, b) means S-MODE with $F = a$ and $CR = b$.

It can be observed that A-MODE outperforms OW-MOSaDE on eight of the ten problems. Nevertheless, without updating operation, A-MODE generally performs worse than AS-MODE. Hence, the stochastic coding strategy is necessary and effective in improving the algorithm performance. Meanwhile, results of S-MODE indicate that parameter settings have significant impacts on the algorithm performance and the best parameter settings are problem dependent. For example, S-MODE with F=0.5 and CR = 1 performs much better than the one with F =0.5 and CR =0 on UF1, UF2, UF3, and UF7. However, the latter performs significantly better than the former on UF4, UF6 and UF9. By using the adaptive control strategy, AS-MODE can always obtain the best or near best IGD values. These results demonstrate that, the adaptive control strategy is also necessary and effective in improving the algorithm performance.

Fig. 4 and Fig. 5 show the evolution trend of selection probability related to each candidate parameter value. From Fig. 4, we can see that AS-MODE generally increases the selection probability of F = 0.5 on UF3, while the select-probabilities of F=0.5, F=1.0 and F = 1.5 are not much different on UF4. Meanwhile, Fig. 5 (a) demonstrates that the selection probability of CR = 1 increases dramatically on UF3, while the selection probability of CR = 0 increases quickly on UF4. According to the results in Table 3, CR = 1 and CR = 0 are respectively good parameter setting for UF3 and UF4. The above results demonstrate that the proposed adaptive control strategy is effective to make the algorithm select promising parameter values to generate offspring.

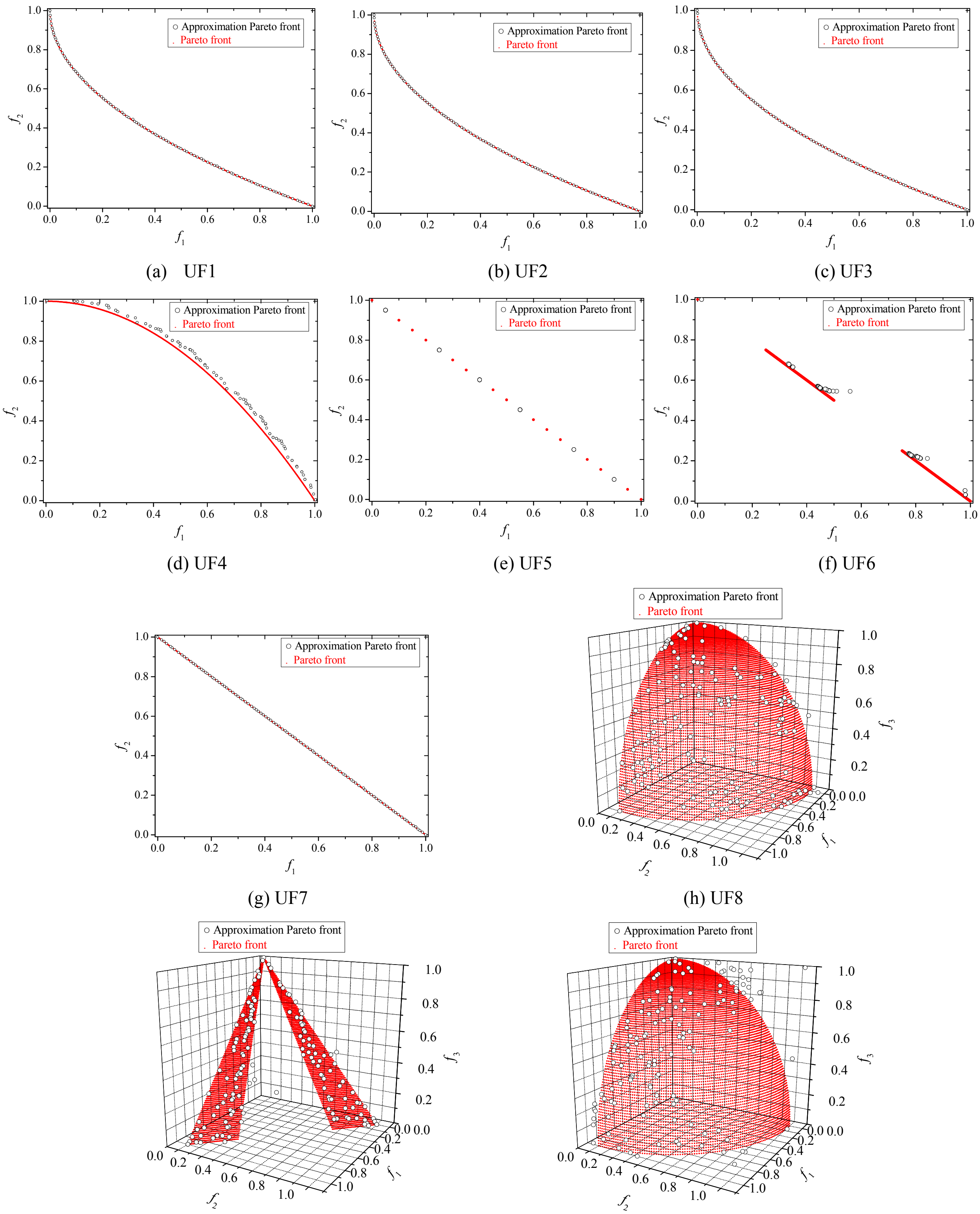
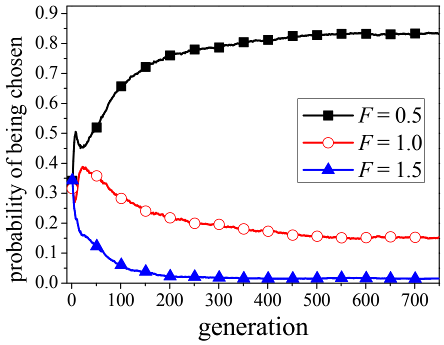
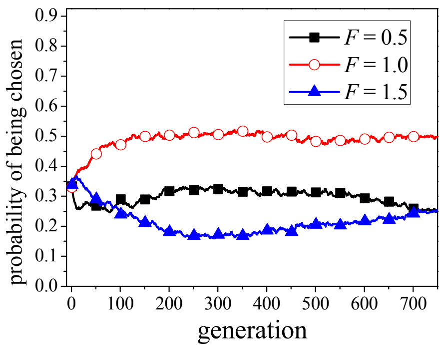


Figure 3. The approximation Pareto front with the lowest IGD value for each test problem

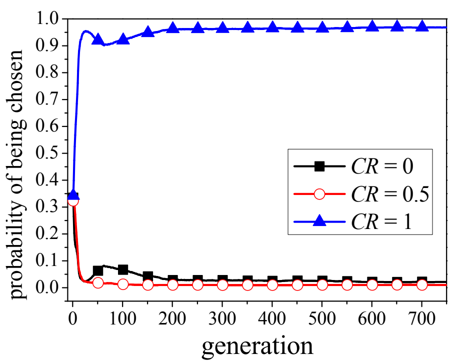


(a) UF3

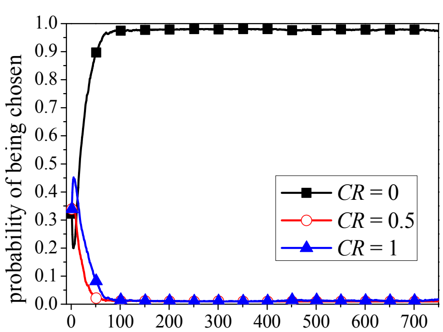


(b) UF4

Figure 4 Evolution trends of selection probability related to *F*



(a) UF3



(b) UF4

Figure 5. Evolution trends of selection probability related to *CR*

5. CONCLUSIONS

This paper presents a novel adaptive multi-objective differential evolution algorithm with stochastic coding strategy, namely AS-MODE, for multi-objective optimization. The stochastic coding strategy encodes each individual with a stochastic region which defined by a normal distribution. The stochastic coding strategy makes individuals more sensible to their surrounding region and improves the efficiency of the algorithm in fine-tuning solutions. In order to reduce the influence of parameter settings, a probability-based adaptive control strategy is also proposed. The main idea of the control strategy is to make promising parameter values have higher probability of being selected to generate offspring. The proposed AS-MODE has been validated over a suite of ten benchmark test functions, including seven 2-objective functions and three 3-objective functions. The experimental results show that AS-MODE outperforms GDE3, OW-MOSaDE, NSGA-II-ls and MOEA/D on most of the test functions. Moreover, the effectiveness and efficiency of the proposed adaptive control strategy has also examined to demonstrate the advantages of AS-MODE.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (NSFC) No. 61070004, and by NSFC Joint Fund with Guangdong under Key Project U0835002. The authors are with the Key Laboratory of Digital Life, Ministry of Education, China, and also with the Key Laboratory of Software technology, Education Department of Guangdong Province. The corresponding author is Jun Zhang, email: junzhang@ieee.org.

7. REFERENCES

[1] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, 2001.

[2] E. Zitzler, E. and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.

[3] D. J. Knowles and W. D. Corne, "Approximating the non-dominated front using the Pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp.149-172, 2000.

[4] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2. Pp. 183-197, April 2002.

[5] Q. Zhang, H. Li, "MOEA/D: a multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no.6, pp.712-731, Dec. 2007.

[6] K. Price, R. Storn, J. Lampinen, *Differential Evolution*. Springer, Heidelberg (2005).

[7] M. Varadarajan, K. S. Swarup, "Solving multi-objective optimal power flow using differential evolution," *Generation, Transmission & Distribution, IET* , vol.2, no.5, pp.720-730, September 2008.

[8] S. K. Goudos, J. N. Sahalos, "Pareto Optimal Microwave Filter Design Using Multiobjective Differential Evolution,"

IEEE Transactions on Antennas and Propagation, vol.58, no.1, pp.132-144, Jan. 2010.

[9] H. A. Abbass, R. Sarker, C. Newton, "PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems," *Proceedings of the 2001 Congress on Evolutionary Computation*, vol.2, no., pp.971-978, 2001.

[10] B.V. Babu, M. M. L. Jehan, "Differential evolution for multi-objective optimization," *The 2003 Congress on Evolutionary Computation, CEC '03.* , vol.4, no., pp. 2696-2703 Vol.4, 8-12 Dec. 2003.

[11] J. Lampinen, "DE's selection rule for multiobjective optimization," Lappeenranta University of Technology, Department of Information Technology, Technical Report, 2001.

[12] S. Kukkonen and J. Lampinen, "An extension of Generalized Differential Evolution for multi-objective optimization with constraints," in *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Birmingham, England, Sept 2004, pp.752-761.

[13] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of Generalized Differential Evolution," in *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, pp.443-450, Vol.1, 5-5, Edinburgh, UK, Sept. 2005.

[14] V. L. Huang, P. N. Suganthan, and M. F. Tasgetiren, "Multi-objective Optimization based on Self-adaptive Differential Evolution Algorithm", *IEEE Congress on Evolutionary Computation, 2007. CEC 2007.* , vol., no., pp.3601-3608, 25-28 Sept. 2007.

[15] V. L. Huang, S. Z. Zhao, R. Mallipeddi, P. N. Suganthan, "Multi-objective optimization using self-adaptive differential evolution algorithm," *IEEE Congress on Evolutionary Computation, CEC '09.* , vol., no., pp.190-194.

[16] T. Zhenguo, L. Yong, "A robust stochastic genetic algorithm (StGA) for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol.8, no.5, pp. 456- 470, Oct. 2004.

[17] Q. F. Zhang, A. Zhou, S. Z. Zhao, P. N. Suganthan, W. Liu and S. Tiwari, "Multiobjective Optimization Test Instances for the CEC2009 Special Session and Competition," Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report, University of Essex, Colchester, UK and Nanyang Technological University, Singapore, 2008.

[18] S. Das, P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol.PP, no.99, pp.1-28

[19] K. Sindhya, A. Sinha, K. Deb and K. Miettinen, "Local Search Based Evolutionary Multi-Objective Optimization Algorithm for Constrained and Unconstrained Problems", *IEEE Congress on Evolutionary Computation, 2009.* vol., no., pp.2919-2926, 18-21 May 2009.

[20] Q. Zhang, W. Liu, H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," *IEEE Congress on Evolutionary Computation, 2009.* vol., no., pp.203-208, 18-21 May 2009.