*2011 International Conference on Computer Science and Information Technology (ICCSIT 2011)*
***IPCSIT vol. 51 (2012) © (2012) IACSIT Press, Singapore***
*DOI: 10.7763/IPCSIT.2012.V51.32*

# A Favorable Real Code Evolutionary Method for Multiple Criteria Problems

Wang Gang*, Yu Long, Wang Junbo

School of Automation Wuhan University of Technology Wuhan, P. R. China

**Abstract.** In this paper, we present some new real code crossover operator, mutation operator and selection strategy for multiple criteria problems. One of the most important performances of the evolutionary algorithms is the ability to approximate the true Pareto frontier. And the proposed methods can improve this performance compared with some conventional Multi-objective Evolutionary Algorithms (MOGAs), such as NSGA-II and SPEA2 and so on. Then we compare the performance of the algorithm with NSGAII and SPEA2 with the popular test problems DTLZ1 and DTLZ2 which have a changeable scale on objectives and variables. Finally the test results (table II and table III) show that the solutions in our methods are closer to the Pareto frontier. Furthermore, along with the number of criteria is increasing, this superiority becomes more obvious.

**Keywords:** Evolutionary Algorithm, real code, multiple criteria

## 1. Introduction

During the past several years, many efficient multi-objective evolutionary algorithms have been proposed, such as NSGA-II by Deb [1], Zitzler andThiele presented SPEA2 [2] and Micro-GA2 [3], and so on. However, most of the MOGAs are only widely and successfully applied to problems with two or three objectives. For example NSGA-II and SPEA2 for have been shown to be not suitable to solve optimization problems with more than three objectives, which have been termed many-objective problems by Farina and Amato [4]. [5] Studies the influence of the number of objectives of a continuous multi-objective optimization problem on its hardness for evolution strategies which is of particular interest for many-objective optimization problems. And unfortunately, most real-world problems in all fields (science, engineering and business management and so on) always involve with more than tree objectives simultaneously. Therefore, almost all hot issues in the design of MOEAs have related to the handling of many-objective optimization problems, and some approaches for improving the scalability of MOEAs for many-objective optimization problems have been proposed [6][7][8]. In [9], an alternative relation, called *Favour*, was proposed, and experiments had shown that Preferred clearly outperformed relation Dominates, but it is possible to have cycles in the relation graph of the elements of a search space. A fuzzy definition of optimality (k-optimality) was presented in [10] however; this definition was only used as a posteriori selection criteria within the huge Pareto-optimal front.

In this paper, we propose a whole set of evolutionary strategies to handle the optimization problems with large number objectives, which include the new simulated binary crossover operator, new mutation operator and new selection strategy. Compared with binary crossover put forward by Deb, our crossover method is improved as following two aspects: 1) Dynamic revising crossover parameter (i.e. there is no parameter given by user). 2) Setting some conditions for crossover candidates. Our mutation operator is also superior over classical polynomial mutation in some aspects. For instance, every child individual never exceeds the given variable ranges, and the users don't need to set the parameters. These kinds of adaptive strategy also

**E-mail address*: wanggang2005@yeah.net.

have been proposed by many studies. However, most of them focus on mutation or crossover probability revising, such as [11], [12].

This paper is organized as follows. In Section II, we introduce the new method in detail. Then we give the results of experiments on two kinds of test problems and the comparison between the proposed approach with NSGA-II and SPEA2 in Section III. Finally, we discuss our findings and future research directions in Section IV.

# 2. New Real Code Evolutionary Method

## 2.1. New Simulated Binary Crossover

In the simulated binary crossover (SBX) method, the crossover parameter setting is very important, because it directly influences the distributions of the children solutions. As the author said "*A large value of* η *gives a higher probability for creating near parent solutions and a small value of* η *allows distant solutions to be selected as children solutions*."(η is the crossover parameter). Therefore, how to set the parameter may determine the quality of the final results. Conventionally, this parameter is given by the users and remains constant during the procedure. Generally, it is not easy to set the parameter at a proper value. On the contrary, we dynamically calculate this parameter based on each individual in every generation (if all the crossover conditions are fulfilled) as the following:

$$A = \sum_{i=1}^{m} \left( f_i^i - z_i^* \right)^2 + \sum_{j=1}^{m} \left( f_j^j - z_j^* \right)^2$$

$$q = m * e \wedge (-\frac{1}{A})$$

If we note $z_1^*$ ; $z_2^*$ ; . . . ; $z_m^*$ as the individual minima of each respective objective function, the utopian solution $z^* = [z_1^*, z_2^* \ldots z_m^*]$ is the best theoretical solution which simultaneously minimizes all the objectives. Nevertheless, this utopian solution is rarely feasible because of the existence of constraints. Here we note $z^*$ as reference point, and m is the number of objectives.

Obviously, when A is large (i.e. the parents are far from the reference point), we can calculate a small crossover parameter (η). As a result, the children individuals may be far from their parents. Oppositely, while A is small, the crossover parameter (η) is large, then the children solutions are near to their parents.

On the other hand, we set a block to crossover parents. It means that don't allow individual to cross; even the probability condition is fulfilled. Firstly, we give a definition about "Relation-ship" as follows.

**Definition** (Relation-ship): Suppose that Pop is the population of the algorithm, $\forall\ i, j \in \mathrm{Pop}$, $\left[ f_i^1, f_i^2, \ldots f_i^m \right]$ and $\left[ f_j^1, f_j^2, \ldots f_j^m \right]$ are their objective vectors, then we can calculate the Relation-ship (RP) as the following: $RP = \left| f_i^1 - f_j^1 \right| + \left| f_i^2 - f_j^2 \right| + \ldots + \left| f_i^m - f_j^m \right|$. (*m* is the number of objectives).

After that, if the probability condition is met, we select the parents to crossover as the following steps.

**Step 1:** Randomly select *n* individuals from Pop as crossing candidates ( $3 \le n \le |Pop|$ ).
**Step 2:** Calculate the RP for each pair of the candidate individuals.
**Step 3:** Choose the two candidates as the crossover parents, whose RP value is maximum.

Evidently, if we select more crossing candidates (i.e. n is larger), the RP of crossover parents may be larger as well. That means that the two individuals are more proper to cross. But along with the number of candidate increases, the complexity of computer also becomes large.

## 2.2. New Mutation Method

The most popular real code mutation method is polynomial mutation (PLM), which is proposed by Deb and Coyal [13]. However, this scheme has a vital shortcoming: offsprings exceed some variable limits easily. Moreover, to use this approach, the users need to give two mutation parameters, which is certainly also a very tough work. To tackle with this dilemma, we propose a new mutation operator. The details about new mutation approach are as follows.

**Step 1:** Choose two random numbers $r_1, r_2 \in (0,1)$.

**Step 2:** Suppose the optimization problems have *n* decision variables, and the individual can be shown as $\mathrm{Pr} = [\mathrm{Pr}_1, \mathrm{Pr}_2, ..., \mathrm{Pr}_n]$ (*n* is the number of decision variables) in decision space. Assume the range of the $i^{\text{th}}$ variable is $[a_i, b_i]$, we can compute the children solutions using the following scheme.

$$\mathrm{Pr}_{i\_mut} = \begin{cases} \mathrm{Pr}_i + r_2 * \left(\mathrm{Pr}_i - a_i\right) * e \wedge (\phi(\mathrm{Pr})) & r_1 < 0.5 \\ \mathrm{Pr}_i + r_2 * \left(b_i - \mathrm{Pr}_i\right) * e \wedge (\phi(\mathrm{Pr})) & otherwise \end{cases}$$

$$\phi\left(\mathrm{Pr}\right) = -\frac{1}{\sum_{i=1}^{m}\left(f_i(\mathrm{Pr}) - z_i^*\right)^2}$$

Here, $z^* = [z_1^*, z_2^* ... z_m^*]$ is the reference point, and *m* is the number of objectives, and $f_i(\mathrm{Pr})$ stands for the $i^{\text{th}}$ objective value of individual Pr.

The pseudocode of new mutation method is illustrated as follows.

*Randomly choose p, r;*
*if p < mut_p // mut_p is the given mutation probability*
*for i = 1:n // n is the number of decision variable*
*if r<0.5*
$\mathrm{Pr}_{i\_mut} = \mathrm{Pr}_i + r * \left(\mathrm{Pr}_i - a_i\right) * e \wedge (\phi(\mathrm{Pr}))$;
*else*
$\mathrm{Pr}_{i\_mut} = \mathrm{Pr}_i + r * \left(b_i - \mathrm{Pr}_i\right) * e \wedge (\phi(\mathrm{Pr}))$;
*end if*
*end for*
*end if*

**Note:** when the parent individual is far from the reference point, the mutation offsprings will be far from the parent. On the contrary, if the parent is close to the reference point, the mutation solutions will be close to the parent. A very extreme situation is that the parent is the reference point, then $(\phi(\mathrm{Pr}_i)) \to -\infty$. Consequently, $\mathrm{Pr}_{i\_mut} = \mathrm{Pr}_i$.

## 2.3. New Elitist Preservation Strategy

The main reason for the Pareto-dominance-based evolutionary algorithms' inefficiency dealing with the Multiple Criteria Problems is that solutions are usually non-dominated with each other by using the Pareto dominance relation when the number of the objectives is big enough. This results from a very low selection pressure toward the Pareto front. In order to converge to the Pareto-optimal solutions with a wide diversity among the solutions, a new elitist preservation strategy is proposed in this section. Firstly, we give some new definitions of optimality for handling multiple criteria problems.

**Definition (**k-rank-matrix): Suppose optimizing m-Criteria Problem, the k-rank- matrix of one individual is combined by all the k-element subsets of the objectives non-dominated-sorting values (the number of the k-element subsets is $C_m^k$, $C_m^k = \frac{m!}{(m-k)!*(k)!}$).

Take 3-Criteria Problem for example. Suppose $i \in Pop$ （Pop is an evolutionary population）, and $\left[f_i^1, f_i^2, f_i^3\right]$ is its objective vector, and *k* = 2. There are three 2-element subsets of the objectives: $\left[f_i^1, f_i^2\right], \left[f_i^1, f_i^3\right], \left[f_i^2, f_i^3\right]$ ( $C_3^2 = 3$ ). And suppose their non-dominated-sorting values are2, 3, and 2. Last, we can get the individual i's *k*-rank-matrix is [2, 3, 2].

After getting each individual's k-rank-matrix, we can calculate each individual's fitness as the following formula:

$$fitness = \frac{5}{\min(k - rank\_array)} + \frac{1}{\sum_{\substack{i \in k-rank\_array \\ i \neq \min(k-rank\_array)}} i} + Dis$$

Where, *Dis* is the normalized crowding-distance (about crowding-distance, see [1] for more details).

Note: because of $\frac{5}{\min(k - rank\_array)} >> \frac{1}{\sum_{\substack{i \in k-rank\_array \\ i \neq \min(k-rank\_array)}} i}$,

We can deduce that the smaller the min of the k-rank-matrix is, the better the individual is.

Definition (difference of fitness ( *Dif_fit*)): Suppose that $fitness_i$ denotes the fitness of the $i^{th}$ individual, and the fitnesses of its two adjacent solutions are $fitness_{i-1}$and $fitness_{i+1}$ respectively, then we can compute the difference of fitness of individual *i* using the following formula: $Dif\_fit_i = |fitness_i - fitness_{i-1}| + |fitness_i - fitness_{i+1}|$.

We select next generation evolutionary population as follows: (N is the population size. and Q denotes the current generation evolutionary population, namely offspring population through selection, crossover and mutation, $R = P \cup Q$ ).

**Step 1)** Calculate the fitness value of R.

**Step 2)** If the number of individuals with a fitness value larger than five, is smaller than N, we fulfill the pool through picking up individuals from the remaining individuals. Otherwise, we need to calculate the *Dif_fit* values of R, and select N individuals with larger fitness and *Dif_fit* value.

## 3. Test Results and Analysis

In this paper, we select DTLZ1 and DTLZ2 as the test problems. And we mainly focus on the problems with four to eight objectives, i.e. *m*=4, 5,6,7,8, respectively. To validate the better performance of our methods, we performed quantitative comparisons (adopting three metric) among the proposed methods and NSGA-II) [1], Strength Pareto Evolutionary Algorithm2 [2]. We adopt the following three performance assessment metrics to make comparisons.

1) **Generational Distance (GD)**: GD represents the distance between $PF_{know}$ and $PF_{true}$. Its can be calculated as the following formula:

$$GD = \frac{(\sum_{i=1}^{n} d_i^p)^{1/p}}{n}$$

Where, *n* is the number of non-dominated vectors found by the algorithm, and $p = 2$, $d_i$ is the Euclidean distance between each of these and the nearest member of the true Pareto front. It should be clear that a value of GD = 0 indicates that all the elements generated are in the true Pareto front of the problem. Therefore, the value indicates how "far" they are from the global Pareto front of our problems.

2) **Two Set Coverage (CS):**

This strategy evaluates the quality of non-dominated solutions. Let $P', P'' \subseteq X$ be two sets of non-dominated solutions. The coverage metric is defined as follows:

$$CS(P', P'') = \frac{\left|\left\{a'' \in P'' \middle| \exists a' \in P', a' \succ a'' or a' = a''\right\}\right|}{|P''|}$$

Where the value CS (P′, P″) = 1 means that all individuals in P′ dominated P″ and CS (P′, P″) = 0 represents that no solutions in P″ are covered by P′. For the performance verification, we can suppose P′ was the set of proposed algorithm and P″ was the set of conventional MOGAs. To some extent, if $CS(P', P'') > CS(P'', P')$, we can believe that proposed algorithm is better than conventional MOGAs.

3) **Spacing (SP):**

Many different diversity metrics have been presented over the past decades. Here we choose the method proposed by Schott [14], described as the following spacing metric:

$$SP= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2}$$

Where, $d_i = \min_j\left(\left|f_1^i - f_1^j\right| + \left|f_2^i - f_2^j\right| + \cdots \left|f_m^i - f_m^j\right|\right)$ （$i,j$ =1,2,…$n$）, $\bar{d}$ is the mean of $d_i$, and $n$ is the population size, and $m$ is the number of objectives. A value of zero for this metric indicates all the non-dominated solutions obtained are equidistantly spaced.

In order to trim off all the elements that might blemish the outcomes of the experiments. Every parameter for each test functions and the number of objectives are stetted a proper value. Table I shows the genetic operators and the parameter set for the three algorithms. Run every algorithm for 10 ten times on each case, and get the average performance data as the final result. This could eliminate the disturbance brought by the different initial populations. Table II presents the GD values of the three algorithms at different situations; and table III shows the CS values for the three algorithms on each test problem with different objectives. Assume that X is the solution set gained by the proposed algorithm, N by NSGA-II, and P by SPEA2. So CS (X, N) represents the ratio of N covered by X, while CS (N, X) represents the ratio of X covered by N. Of course, CS (X, P) and CS (P, X) are the same. In table 5 we give the SP values for the three algorithms on each test problem with different objectives.

Table 1.MOEA Settings Adopted for This Study

| Parameter<br>Algorithm | Max<br>Generation | Max<br>Population | | Crossover<br>type | Crossover parameter | | Mutation<br>type | Mutation parameter | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | parameter | probability | | parameter | probability |
| Proposed | 500 | 50 | | New SBX | $n$ = 5 | 0.9 | New Mutation | —— | 0.1 |
| SPEA2 | 500 | pop 150 | NDSet 50 | SBX | q =2 | 0.9 | PLM | q=2 | 0.1 |
| NSGAII | 500 | 50 | | SBX | q=2 | 0.9 | PLM | q=2 | 0.1 |

Table 2.The GD Values for Each MOGA Over All The Test Problems And The Number of Objectives Considered

| *Test Problem* | **Metric** | ***m*** | **Proposed** | **NSGAII** | **SPEA2** |
|---|---|---|---|---|---|
| DTLZ1 | GD | 4 | 0.0315 | 0.0622 | 0.0538 |
| | | 5 | 0.0159 | 0.0758 | 0.0233 |
| | | 6 | 4.899e-5 | 0.0783 | 0.0170 |
| | | 7 | 0.00233 | 0.0803 | 0.0373 |
| | | 8 | 0.00429 | 0.0778 | 0.0405 |
| DTLZ2 | GD | 4 | 0.0267 | 0.0672 | 0.0352 |
| | | 5 | 0.0239 | 0.0739 | 0.0283 |
| | | 6 | 0.0059 | 0.0618 | 0.0312 |
| | | 7 | 0.0038 | 0.0662 | 0.0226 |
| | | 8 | 0.0033 | 0.0707 | 0.0345 |

Table 3.The CS Values for Each MOGA Over All The Test Problems And The Number of Objectives Considered

| *Test Problem* | **Metric** | ***m*** | **CS(X,N)** | **CS(N,X)** | **CS(X,P)** | **CS(P,X)** |
|---|---|---|---|---|---|---|
| DTLZ1 | CS | 4 | 0.04 | 0 | 0 | 0 |
| | | 5 | 0.02 | 0 | 0.02 | 0 |
| | | 6 | 0.2 | 0 | 0.16 | 0 |
| | | 7 | 0.12 | 0 | 0.02 | 0 |
| | | 8 | 0.26 | 0.14 | 0.2 | 0 |
| DTLZ2 | CS | 4 | 0.02 | 0 | 0.02 | 0 |
| | | 5 | 0.04 | 0.02 | 0.06 | 0 |
| | | 6 | 0.34 | 0.06 | 0.24 | 0.02 |
| | | 7 | 1 | 0.16 | 0.14 | 0 |
| | | 8 | 0.12 | 0 | 0.2 | 0 |

Table 4.The SP Values for Each MOGA Over All The Test Problems And The Number of Objectives Considered

| *Test Problem* | **Metric** | ***m*** | **Proposed** | **NSGAII** | **SPEA2** |
|---|---|---|---|---|---|
| DTLZ1 | SP | 4 | 0.0434 | 0.0588 | 0.0367 |
| | | 5 | 0.1080 | 0.112 | 0.0299 |
| | | 6 | 6.55E-6 | 0.0426 | 0.0338 |
| | | 7 | 0.000678 | 0.0552 | 0.0382 |
| | | 8 | 0.000685 | 0.0639 | 0.0367 |
| DTLZ2 | SP | 4 | 0.1574 | 0.1662 | 0.0872 |
| | | 5 | 0.2843 | 0.1934 | 0.1099 |
| | | 6 | 0.0699 | 0.0733 | 0.1021 |
| | | 7 | 0.1052 | 0.0997 | 0.1022 |
| | | 8 | 0.2704 | 0.1655 | 0.1177 |

Now, we start to analyze the test results. In the table II, as it can be observed, when the number of objectives is four and five, the proposed approaches have slightly better than NSGAII and SPEA2. However, when the number of objectives is larger than five, the proposed approaches are significantly better than NGSAII and SPEA2. So we could predict that along with the number of objectives is increasing, our approaches would be become more and more efficient than conventional MOGAs. The same conclusion also can be got from table III. In table III, we can see CS (X, N) > CS (N, X) for all test problems and all number of objectives considered. Similarly, along with the number of objectives increases, the CS (X, N) is larger than CS (N, X). There is the same situation to CS(X, P) and CS (P, X), of course. Those mean that the solutions set obtained by proposed methods maybe become more and more better compared with other algorithms, while the number of objectives increases. For DTLZ1, showed in table IV, the proposed methods have much better values than NSGAII and SPEA2, when the number of objectives is five to eight. Most of other situations, the proposed algorithm is similar with NSGAII, but slightly worse than SPEA2.

# 4. Conclusions

In this paper, a whole new set of evolutionary method is proposed. Using these methods, we can obtain much better solutions than conventional MOGAs. In order to verify this conclusion, two scalable benchmark problems are utilized. In addition, three criteria used to assess the algorithm performance include the closeness of non-dominated solutions to the true Pareto front (GD), the coverage rate of two solution sets and the distribution of solutions across the front (SP). The analysis indicates that the new MOGA can produce non-dominated solutions that are better than that generated by NGGAII and SPEA2. And moreover, when the number of objectives is large, the proposed algorithm can obtain even better solutions. However, our algorithm don't improve the diversity, and even some worse than SPREA2. We think change the last term (Dis) of fitness formula to density of grid, or some other strategies, such as [15], may improve diversity compared to the new MOGA.

# 5. References


[1] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II , Kanpur Genetic Algorithm Laboratory (KanGAL), Tech. Rep. 200001, 2000.

[2] Zitzler E, Laumanns M, Thiere L. SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-Report 103[R]. 2001.Strunk Jr W, White EB. *The elements of style*. 3rd ed. New York: Macmillan; 1979.

[3] Pulido G T, Collo C A C. The micro genetic algorithm 2: Towards online addaption in evolutionary multiobjective optimization[C]//EMO, 2003:252-266.

[4] M. Farina and P. Amato. “On the optimal solution denition for many-criteria optimization problems,” in Proc. NAFIPS-FLINT Int. Conf. J. Keller and O. Nasraoui, Eds., Jun. 2002, pp. 233–238.

[5] Schütze, O.; Lara, A.; Coello Coello, C. A. On the Influence of the Number of Objectives on the Hardness of a Multiobjective Optimization Problem . Evolutionary Computation, IEEE Transactions on Volume: PP , Issue: 99 Digital Object Identifier: 10.1109/TEVC.2010.2064321 Publication Year: 2010

[6] H. Sato, H. E. Aguirre, and K. Tanaka, “Controlling dominance area Of solutions and its impact on the performance of MOEAs,” in Proc. 4th IntConf. Evol. Multi-Criterion Optimization, S. Obayashi, K. Deb, C. Poloni T. Hiroyasu, and T. Murata, Eds. New York: Springer-Verlag, Mar. 2007vol. 4403, pp. 5–20.

[7] Banu Soylu and Murat Kksalan. “A Favorable Weight-Based Evolutionary Algorithm for Multiple Criteria Problems,” IEEE Transactions on evolutionary computation, vol. 14, no. 2, April 2010

[8] Garza-Fabre, M.; Toscano-Pulido, G.; Coello, C.A.C. “Two novel approaches for many-objective optimization” Evolutionary Computation (CEC), 2010

[9] N. Drechsler, R. Drechsler, and B. Becker, “Multi-objective optimization based on relation favour,” in Proc. 1st Int. Conf. Evol. Multi-Criterion Optimization, E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, Eds. New York: Springer-Verlag, 2001, vol. 1993, pp. 154–166.

[10] M. Farina and P. Amato, “A fuzzy denition of optimality for many criteria optimization problems,” IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 34, no. 3, pp. 315–326, May 2004.

[11] Jun Zhang; Henry Shu-Hung Chung; Wai-Lun Lo. Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms. Evolutionary Computation, IEEE Transactions on Volume: 11 , Issue: 3 Digital Object Identifier: 10.1109/TEVC.2006.880727 Publication Year: 2007 , Page(s): 326 - 335

[12] Raghuwanshi, M.M. (RCERT, Chandrapur, Maharashtra, India); Kakde, O.G. A robust real coded genetic algorithm with self adaptation of crossover probabilities. *Journal of the Institution of Engineers (India), Part CP: Computer Engineering Division*, v 89, NOV., p 31-37, 2008

[13] K.Deb, M.Goyal, “A combined genetic adaptive search for engineering design,” *Complex Science and Informatics*, vol. 26, no. 4, pp.30-45, 1996.

[14] Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995

[15] Martínez, S.Z.; Coello, C.A.C. “An archiving strategy based on the Convex Hull of Individual Minima for MOEAs”. Evolutionary Computation (CEC), 2010 IEEE Congress on Digital Object Identifier: 10.1109/CEC.2010.5586462 Publication Year: 2010 , Page(s): 1-8