An Improved Clonal Algorithm in Multiobjective Optimization

Jianyong Chen, Qiuzhen Lin and Qingbin Hu
Department of Computer Science and Technology, Shenzhen University, Shenzhen, P.R. China
Email: jychen@szu.edu.cn, qiuzheng658@163.com, huqb@szu.edu.cn

***Abstract*—In this paper, we develop a novel clonal algorithm for multiobjective optimization (NCMO) which is improved from three approaches, i.e., dynamic mutation probability, dynamic simulated binary crossover (D-SBX) operator and hybrid mutation operator combining with Gaussian and polynomial mutations (GP-HM operator). Among them, the GP-HM operator is controlled by the dynamic mutation probability. These approaches adopt a cooling schedule, reducing the parameters gradually to a minimal threshold. By this means, they can enhance exploratory capabilities, and keep a desirable balance between global search and local search, so as to accelerate the convergence speed to the true Pareto-optimal front. When comparing NCMO with various state-of-the-art multiobjective optimization algorithms developed recently, simulation results show that NCMO evidently has better performance.**

***Index Terms*—multiobjective optimization; immune algorithm; clonal selection; hybrid mutation**

I. Introduction

In some real world problems, there exist many multiobjective optimization problems (MOPs) in practical engineering and scientific applications. Without any further information about the relationship among the objectives, it is possible to obtain a set of optimal solutions in which each solution is equally preferable when regarding all criteria considered. As a result, in order to provide multiple candidate selections for different applications, we aim at as many representative optimal solutions as possible.

Evolutionary algorithms have the ability to process sets of solutions in parallel and explore big search spaces in reasonable time. Therefore, Evolutionary algorithms have been recognized to be well suited to MOPs. The ability to handle complex problems, involving features such as discontinuities, multimodality, disjoint feasible spaces and noisy function evaluations, reinforces the potential effectiveness of evolutionary algorithms in MOPs [1]. In the last few years, numerous competent Evolutionary algorithms have been proposed as the state-of-the-art algorithms for MOPs. For example, NSGA-II [2] was proposed with a fast nondominated sorting technology, elitism and crowding-distance assignment. SPEA-II [3] was proposed with a fine-grained fitness assignment strategy, an enhanced archive truncation method and a new density estimation technique. The pareto archived evolution strategy (PAES) [4] was proposed with simple (1+1) evolution strategy. All of them tried to design effective and efficient technologies to improve the abilities of the convergence and the diversity.

On the other hand, Artificial Immune Systems (AIS) have been developed since 1990s as a new branch in computational intelligence, which simulate the defense of the human immune system against bacteria, viruses and other invaders [5]. A number of AIS models have been found applications in various fields such as machine-learning and pattern-recognition tasks [6], network security [7], scheduling [8], data mining [9] and others [10, 11]. In recent years, AIS are also applied in MOPs and studies show their remarkable performances. For example, Coello Coello and Cortes [12] presented a multiobjective immune system algorithm (MISA) based on the clonal selection principle. Freschi and Repetto [13] proposed a vector artificial immune system (VAIS) based on the artificial immune network. Gong and Jiao et al. [14] proposed a nondominated neighbor-based immune algorithm (NNIA), by using a novel nondominated neighbor-based selection technique, proportional cloning, heuristic search operators and elitism.

For MOPs, one of the key points is to find a uniformly distributed approximation set that is as close as possible to the Pareto-optimal front. Based on the notion that the preservation of representative solutions can effectively drive the algorithm toward the Pareto-optimal front [15], most of current studies pay much attention to fitness assignment and population maintenance such as crowding-distance [2], grid mechanism [16] and clustering technology [17]. However, rare studies pay attention to search operators, which can obviously accelerate the convergence speed.

In this paper, we propose a novel clonal algorithm for MOPs (NCMO) based on the improvement of search operators, aiming to speed up the convergence. Three novel approaches, i.e., dynamic mutation probability, dynamic simulated binary crossover (D-SBX) operator and hybrid mutation operator combining with Gaussian and polynomial mutations (GP-HM operator), are proposed in this paper. Similar to the temperature parameter in simulated annealing, these approaches also

Manuscript received January 1, 2009. The work was supported by the National Natural Science Foundation of China (Grant No: 60703112).

Jianyong Chen, Qiuzhen Lin and Qingbin Hu are with the Department of Computer Science and Technology, Shenzhen University, Shenzhen, P.R. China (e-mail: jychen@szu.edu.cn; qiuzheng658@163.com; huqb@sz u.edu.cn)

doi:10.4304/jsw.4.9.976-983

adopt a cooling schedule [17], reducing the parameters gradually to a minimal threshold. When NCMO is compared with various multiobjective optimization algorithms developed recently, simulation results illustrate that NCMO has evidently remarkable performance.

The remainder of the paper is organized as follows. Section 2 briefly describes the definition of MOPs and some important terms used in MOPs. In section 3, we describe NCMO in details and introduce some important operators. Besides that, complexity analysis is also given in this section. Section 4 shows the simulations of NCMO comparing with various multiobjective optimization algorithms. Finally, we present some conclusions.

II. MULTIOBJECTIVE OPTIMIZATION PROBLEMS

In general, taking minimization problems for example, the purpose of multiobjective optimization is to find a parameter set P that satisfies the following equation.

$$\mathrm{Min}_{\mathrm{P}\in\Omega} F(x) = (f_1(x), f_2(x), \cdots, f_m(x))^T \quad (1)$$

where Ω is set of the decision vector and m is the number of the function objectives.

For better understanding MOPs, the following four concepts are important [18]:

1. Pareto dominance: A vector $x^0 = (x_1^0, x_2^0, \cdots, x_n^0)$ is said to dominate another vector $x^1 = (x^1_1, x^1_2, \cdots, x^1_n)$ (noted as $x^0 \succ x^1$) if and only if

$$\forall i: f_i(x^0) \le f_i(x^1) \wedge \exists i: f_i(x^0) \prec f_i(x^1), i \in [1, m] \quad (2)$$

2. Pareto-optimal: A solution x^0 is said to be Pareto-optimal if and only if

$$\neg \exists x^1 \in \Omega : x^1 \succ x^0 \quad (3)$$

3. Pareto-optimal set: the set P includes all Pareto-optimal solutions:

$$P = \{x^0 \mid \neg \exists x^1 \in \Omega : x^1 \succ x^0\} \quad (4)$$

4. Pareto-optimal front: The set PF includes values of all objective functions corresponding to the solutions in P:

$$PF = \{f(x) = (f_1(x), f_2(x), \cdots, f_m(x))^T \mid x \in P\} \quad (5)$$

Because the Pareto-optimal set may be infinite, so it is unpractical to find out all solutions in Pareto-optimal set. In general, the aim of multiobjective optimization algorithms is to find a set of some representative solutions, which are distributed uniformly and approximate the Pareto-optimal front as close as possible.

III. A NOVEL CLONAL ALGORITHM FOR MOPS

It is well known that multiobjective optimization algorithms have two fundamental goals: one is to minimize the distance of the generated solutions to fit the Pareto-optimal set; the other is to maximize the diversity of the archive Pareto-set approximation [19]. In this study, in order to accelerate the convergence speed, the proposed algorithm uses three novel approaches, i.e., dynamic mutation probability, D-SBX operator and GP-HM operator. Besides that, the main operators of NCMO also include proportional cloning operator and population selection operator. Moreover, elitism mechanism is used here and an archive is used to preserve nondominated solutions in order to prevent the loss of elitists. Firstly, we give the corresponding pseudo-code of NCMO in Fig.1. In this figure, our novel approaches are marked with boldface.

Step 1. Randomly generate an antibody population, identify the nondominated antibodies and add them to the archive. Set parameter *gen=0*.
Step 2. If the *gen* reaches the predefined maximum of the generations, go to step 7; otherwise go to step 3.
Step 3. Choose part of greater crowding-distance antibodies to do proportional cloning and then the child population is generated.
Step 4. Based on crossover probability, **D-SBX** operator is performed on the child population.
Step 5. Calculate **dynamic mutation probability**, and perform **GP-HM** operator on the child population.
Step 6. Identify the nondominated antibodies in mating pool by combining the archive and the child population. Select a number of nondominated antibodies with greater crowding-distance values as the next generation population and update the archive. Set *gen* = *gen* +1. Go to step 2.
Step 7. Output the external archive as the approximate Pareto-optimal set. Stop the algorithm.

Figure 1. the pseudo-code of NCMO

A. Proportional Cloning Operator

In biological immune system, cloning means that a group of identical cells is generated from a single common ancestor and only antibodies with high affinity will be cloned to attack the pathogens. In NCMO, only part of antibodies with greater crowding-distance values is selected to do proportional cloning and the number of clones is generated based on their crowding-distance values. By this means, nondominated antibodies with greater crowding-distance values will have more clones. Therefore, the less-crowded regions possess more clones, which encourages exploring the complete Pareto-optimal front. Assuming that $A = \{a_1, a_2, \cdots, a_n\}$ is the population that is going to do cloning. The proportional cloning operation is defined as follows:

$$T_C(A) = [T_C(a_1), T_C(a_2), \cdots, T_C(a_n)]^T \quad (6)$$

where $T_C(a_i) = q_i \times a_i \ (i = 1,2,\cdots,n)$. The value of $q_i \ (i = 1,2,\cdots,n)$ is the number of clones for each antibody $a_i \ (i = 1,2,\cdots,n)$, with as follows:

$$q_i = \left\lceil n_C \times \frac{\zeta(a_i, A)}{\sum_{j=1}^{|A|} \zeta(a_j, A)} \right\rceil \quad (7)$$

where n_C is the given value of the clone population size and $\zeta(a_i, A)$ is the crowding-distance value of the antibody $a_i \ (i = 1,2,\cdots,n)$ with as follows:

$$\zeta(a_i, A) = \sum_{j=1}^{m} \frac{\zeta_j(a_i, A)}{f_j \max - f_j \min} \quad (8)$$

where $f_j \max$ and $f_j \min$ is the maximum and the minimum value of the *j*-th objective respectively and the definition of $\zeta_j(a_i, A)$ can be found in (9).

Noted that when calculating the number of clones for each antibody, it is possible that the crowding-distance value is ∞ when the antibody is in boundary. In this case, it is set as double maximum crowding-distance value except the boundary solutions.

$$\zeta_j(a_i, A) = \begin{cases} \infty, if\ (f_j(a_i) == \min\{f_j(a_k)\})\ or\ (f_j(a_i) == \max\{f_j(a_k)\}), k = 1,2...n; \\ \min\{f_j(a_k) - f_j(a_l)\}, (k, l = 1,2...n)\ and\ (k \neq l \neq i), otherwise; \end{cases} \quad (9)$$

B. D-SBX Operator

The recombination operator has the ability to escape from local optimal, and share gene segments from parent chromosomes. Simulated binary crossover (SBX) [20] is one of main recombination operators that have been used in various real-coded multiobjective algorithms [2,3,14]. However, the SBX operator is always used with fix probability (usually set as 0.5) of the variables to get crossed. It may destroy good gene segments at the end of algorithm running because too many genes get crossed at this stage. Therefore, in NCMO, dynamic SBX (D-SBX) is proposed as an improved recombination operator. Because the selection operator selects antibodies according to their Pareto dominance and diversity estimation measured by crowding-distance value. As a result, if it is executed with more generations, the antibodies are becoming closer to the Pareto-optimal front. At the beginning of the algorithm execution, when the generated antibodies are far away from the Pareto-optimal front, the dynamic probability is set with relative large value in order to share good gene segments. By this means, it is easy to generate better antibodies in high probability. With the algorithm running, the antibodies are becoming closer to the Pareto-optimal front, so the dynamic probability is becoming small gradually to a minimal threshold. Otherwise, the good genes segments may be destroyed and the antibodies can't explore closer to the Pareto-optimal front.

Ordinarily, SBX has three controllable parameters [15]: (1) *pc*: the probability for a pair of parent solutions to do recombination; (2) η : the magnitude of the expected variation from the parent values; (3) *pv*: the probability of the variables to get crossed. In D-SBX, the genes have dynamic probability *pv* to do crossover according to the number of generations. The dynamic probability of genes *pv* is defined as:

$$pv = pvx - \frac{(pvx - pvy) * gen}{maxgen} \quad (10)$$

where *pvx* and *pvy* is the predefined probability of the variables to get crossed at the beginning and at the end of the algorithm execution respectively, *gen* is the number of iteration, *maxgen* is the predefined maximum number of generations.

C. GP-HM Operator

Polynomial mutation has been used in many real-coded multiobjective optimization algorithms [2, 3, 14]. Ordinarily, Polynomial mutation has two controllable parameters: mutation probability and distribution index. However, with the fixed distribution parameter, it is not efficient at searching the global Pareto-optimal front. For example, at large search space, the convergence speed of polynomial mutation is very slow with the relative large distribution parameter and it is easy to stagnate if many local Pareto-optimal fronts exist. In this paper, we propose GP-HM operator, which combines with Gaussian and polynomial mutations.

For antibody $X = (x_1, x_2, \cdots, x_n)$, the GP-HM operator is defined as:

$$x_i' = x_i + delta \times (yu - yd), \quad i = 1,2,\cdots,n \quad (11)$$

where x_i' and x_i is the *i*-th decision variables after and before mutation respectively, *yu* is the upper bound of the *i*-th decision variables, *yd* is the lower bound of the *i*-th decision variables. If polynomial mutation is selected, *delta* is a small variation which is obtained from polynomial distribution by using:

$$delta = \begin{cases} [2r_i + \delta]^{\frac{1}{\eta+1}} - 1, if\ \ r_i < 0.5 \\ 1 - [2(1 - r_i) + \delta]^{\frac{1}{\eta+1}}, if\ \ r_i \geq 0.5 \end{cases} \quad (12)$$

where:

$$\delta = (1 - 2r_i) * (\frac{\max(yu - x_i, x_i - yd)}{yu - yd})^{\eta+1} \quad (13)$$

r_i is an uniformly sample random number between (0,1) and η is the mutation distribution index. The plot of *delta* obtained from polynomial distribution (assuming η =20, $x_i = (yu - yd)/2$) can be seen in left part of Fig 2. It is noted that when x_i is equal with *yu* or *yd*, x_i' will

be generated randomly between *yu* and *yd*. Otherwise, Gaussian mutation is selected and *delta* is a small variation obtained from Gaussian distribution by using:

$$delta = 0.1 \times N(0,1) \tag{14}$$

where $N(0,1)$ is a random Gaussian number with mean zero and standard deviation one. The plot of *delta* obtained from Gaussian distributions can be seen in right part of Fig 2.

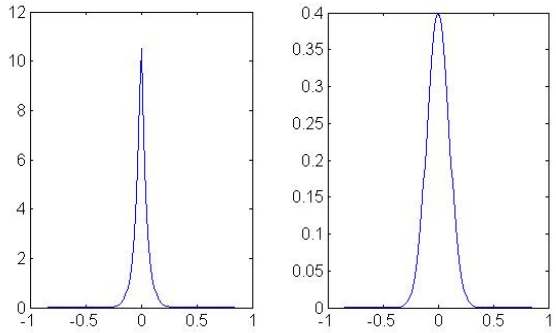


Figure 2. Plot of *delta* obtained from the polynomial and Gaussian distributions respectively.

Seen from the Fig 2, Gaussian mutation has higher probability to generate an offspring further away from its parent than polynomial mutation due to its long flat tails and low hill. In other words, it has a higher probability to escape from a local optimum. On the other hand, it also indicates that polynomial mutation has stronger fine-grained search ability than Gaussian mutation in small regions. In GP-HM, we combine the advantages of two mutations by a switching parameter. The switching parameter is used as a threshold value to control the switching of two mutations. When a randomly generated number is smaller than the switching parameter, the Gaussian mutation is deployed. Otherwise, the polynomial mutation is deployed. The switching parameter *sp* is proposed as:

$$dsp = spx - \frac{(spx - spy) * gen}{maxgen} \tag{15}$$

where *spx* and *spy* is the predefined probability to do Gaussian mutation at the beginning and at the end of the algorithm execution respectively. In general, there is high probability to do Gaussian mutation at the beginning of algorithm, which can make the proposed algorithm converge fast toward the Pareto-optimal front. When nondominated solutions are approaching the Pareto-optimal front, the probability with Gaussian mutation gradually decreases while the probability with polynomial mutation gradually increases. By this means, the algorithm gradually performs more fine-grained search. It is an effective and high-speed search operator, which well keeps the balance of global search and local search.

It is noted that the mutation probability should be changed dynamically according to the adaptive ability of the immune system. In this study, the mutation probability is only changed according to the number of generations in the first half part of algorithm running as follows:

$$pm = (1+p) * minpm - 2p * minpm * (\frac{gen}{maxgen}) \tag{16}$$

where *minpm* is the predefined minimal mutation probability that guarantees every antibody with one gene on average to do mutation and *p* is a predefined parameter that adjusts the mutation scale. While in the last half part of algorithm running, *pm* is fixed as *minpm*.

D. Population Selection And Archive Update

Reference [21] has shown that elitism can speed up the performance of GA significantly and elitism scheme has been well adopted by many state-of-the-art EAs [2, 3, 14]. The elitism mechanism is also deployed here. Nondominated antibodies are preserved in an archive, which is helpful to prevent the loss of good solutions once they are found. When the number of nondominated antibodies exceeds the archive size, a nondominated neighbor-based selection mechanism developed by Gong [14] is used to evaluate the nondominated antibodies. The selection mechanism can evidently extend the diversity of population and mainly focuses on less-crowded regions. The procedures can be described as follows. After the child population is generated from proportional cloning operator, mutated with D-SBX and GP-HM, nondominated antibodies are identified in mating pool that contains the archive and the child population. Then, part of nondominated antibodies with greater crowding-distance values is selected by using the nondominated neighbor-based selection mechanism as the next generation population and the archive is updated with the new population.

E. Complexity Analysis

The complexity analysis of NCMO is provided in this section. Assuming that the population size, the clone population size and the archive size are N, the number of function objectives is M. The basic operators and their worst time complexity are given as follows:

1. Population initialization: $O(M \times N)$.
2. Calculate the fitness value: $O(M \times 2N \times \log(2N))$.
3. Proportional cloning operator: $O(M \times N)$.
4. D-SBX operator and GP-HM operator: $O(M \times N)$.
5. Procedure to identify the nondominated individuals in the mating pool that contains the archive and the child population: $O(M \times (2N)^2)$.
6. New population selection and archive update: $O(2N \times \log(2N))$.

Therefore, the worst total time complexity is:

$$O(M \times N^2) \tag{17}$$

Note that the worst time complexity of NSGA-II, PAES, SPEA, SPEA2 and NNIA is $O(M \times N^2)$, $O(M \times N)$, $O(M \times N^2)$, $O(M \times N^3)$ and $O(M \times N^2)$ respectively.

IV. EXPERIMENTS

In this section, we make some simulations to present the performance of NCMO. At first, we give the related knowledge of the simulations such as benchmark functions and performance metrics used in this paper. Next, we give the simulation results of NCMO comparing with various multiobjective optimization algorithms.

A. Benchmark Functions

In this study, we use ten benchmark functions to valuate the performance of NCMO. The functions include five ZDT functions and five three-objective DTLZ functions without any inequality and equality constraints. Definitions of the ZDT and DTLZ functions can be found in Ref. [21] and Ref. [22] respectively. It is noted that the test problems are characterized with convexity, discontinuity and non-uniformity and some of them have many local Pareto-optimal fronts. By this means, they are suitable to test the comprehensive performance of multiobjective optimization algorithms.

B. Performance metrics

As we mention above, MOPs have two goals. In order to compare the performance of NCMO with various multiobjective optimization algorithms, we adopt the convergence metric and the diversity metric suggested by Ref. [2], and the spacing metric suggested by Ref. [23]. These metrics are designed based on the goals. The first metric measures the extent of convergence to a known subset of the Pareto-optimal front. The last two metrics measure the extent of diversity of an approximation set.

C. Comparison of NCMO with various algorithms

In the follow sections, we compare the performance of NCMO with various multiobjective optimization algorithms, i.e., NSGA-II (real-coded), PAES, SPEA2, MOEO [19] and NNIA. The simulation results are discussed respectively in the following. The parameters setting for NCMO and NNIA are tabulated in Table I.

TABLE I. PARAMETERS SETTING

parameters	NCMO	NNIA
Crossover probability pc	0.9	0.9
Distribution index for SBX	20	20
Mutation probability pm	/	1/n
Distribution index for polynomial mutation	20	20
Population size	100	100
Clone population size	100	100
Population(selected to clone) size	20	20

Besides that, the values of *pvx* and *pvy* in (10) are set as 0.5 and 0.25 respectively. The values of *spx* and *spy* in (15) are set as 0.1 and 0.02 respectively. The minimal mutation probability *minpm* and parameter *p* in (16) is set as *1/n* (*n* is the number of decision variables) and 0.2 respectively. These values of parameters have been determined after an intensive preliminary test phase of the algorithm on different benchmark functions. The parameters setting of NSGA-II, PAES, SPEA2, and MOEO are found in Ref. [2] and Ref. [19]. The maximum of generations is 250. With these parameters setting, NCMO has the same simulation conditions with the other algorithms.

Table II shows the mean and variance of the convergence metric and the diversity metric obtained by using NCMO, NNIA MOEO, NSGA-II, PAES and SPEA2 in solving five ZDT functions. Table III shows the mean and variance of the convergence metric and the spacing metric obtained by using NCMO and NNIA in solving five DTLZ functions. In this study, all the experimental results of MOEO, NSGA-II, PAES and SPEA-II come from Ref. [1] and Ref. [19], whereas the experimental results of NCMO and NNIA come from our simulations. The source code of NNIA can be found in author's web site [24]. All the simulation results can be found in Table II and Table III. It is noted that the best result for each benchmark function is marked with boldface.

TABLE II. MEAN (FIRST ROWS) AND VARIANCE (SECOND ROWS) OF THE CONVERGENCE METRIC AND DIVERSITY METRIC

Algorithm	Convergence					Diversity				
	ZDT1	***ZDT2***	***ZDT3***	***ZDT4***	***ZDT6***	***ZDT1***	***ZDT2***	***ZDT3***	***ZDT4***	***ZDT6***
NCMO	**0.000714**	**0.000708**	**0.001276**	0.003132	**0.000943**	0.345656	0.335635	**0.516642**	0.312922	0.431769
	0.000027	0.000030	0.000117	0.001656	0.000042	0.029055	0.024773	0.013411	0.035242	0.017527
NNIA	0.000728	0.000720	0.001319	**0.002921**	0.000965	0.348140	0.349681	0.521568	0.425531	0.444571
	0.000032	0.000047	0.000072	0.001596	0.000101	0.021056	0.034588	0.025105	0.311671	0.028471
MOEO	0.001277	0.001355	0.004385	0.008145	0.000630	**0.32714**	**0.285062**	0.965236	**0.275664**	**0.225468**
	0.000697	0.000897	0.00191	0.004011	3.26E-05	0.065343	0.056978	0.046958	0.183704	0.033884
NSGA-II	0.033482	0.072391	0.114500	0.513053	0.296564	0.390307	0.430776	0.738540	0.702612	0.668025
	0.004750	0.031689	0.007940	0.118460	0.013135	0.001876	0.004721	0.019706	0.064648	0.009923
PAES	0.082085	0.126276	0.023872	0.854816	0.085469	1.229794	1.165942	0.789920	0.870458	1.153052
	0.008679	0.036877	0.00001	0.527238	0.006664	0.004839	0.007682	0.001653	0.101399	0.003916
SPEA2	0.001448	0.000743	0.003716	0.028492	0.011643	0.472254	0.473808	0.606826	0.705629	0.670549
	0.000317	8.33E-05	0.000586	0.047482	0.002397	0.097072	0.0939	0.191406	0.266162	0.077009

TABLE III. MEAN (FIRST ROWS) AND VARIANCE (SECOND ROWS) OF THE CONVERGENCE METRIC AND SPACING METRIC

Algorithm	Convergence					Spacing				
	DTLZ1	***DTLZ2***	***DTLZ3***	***DTLZ4***	***DTLZ6***	***DTLZ1***	***DTLZ2***	***DTLZ3***	***DTLZ4***	***DTLZ6***

NCMO	0.039971	**0.007059**	**0.349003**	**0.006821**	**0.013286**	**0.024681**	0.061449	**0.082850**	**0.058814**	0.074760
	0.106255	0.000075	0.474796	0.002523	0.002956	0.010031	0.004640	0.042446	0.003253	0.006578
NNIA	0.**039407**	0.008560	4.440519	0.007805	0.014096	0.041189	0.**059426**	0.752856	0.060584	0.**072492**
	0.103240	0.000095	7.793263	0.001075	0.002331	0.045407	0.005330	1.344594	0.004668	0.004531

1) Comparison with some state-of-the-art EAs

It can be observed from Table II that when regarding convergence metric and diversity metric, NCMO is much better than the state-of-the-art EAs i.e., NSGA-II, PAES and SPEA2 in all five ZDT functions. It apparently illustrates the effectiveness of NCMO. In addition, in all cases with NCMO, NCMO is the smallest at the variance of the convergence metric and the diversity metric in many test functions in ten runs except PAES. These simulations also illustrate the robustness of NCMO.

2) Comparison with MOEO

When regarding the diversity metric, simulations from Table II show that MOEO performs better than NCMO except in ZDT3. While regarding the convergence metrics, NCMO is better than MOEO except in ZDT6. In addition, NCMO is smaller than MOEO regarding the variance of the convergence metric and the diversity metric in many test functions in ten runs. It is very difficult for the proposed algorithm to obtain better performance in every metric. It fits the saying that most currently best multiobjective optimization evolutionary algorithms do not outperform each other, but perform similarly or are preferable than respect to different performance indicators [15]. Based on the above discussion, it is concluded that NCMO and MOEO have their own advantages. NCMO has faster convergence speed and more robust than MOEO, while MOEO gets more uniformly distributed solutions than NCMO.

3) Comparison with NNIA

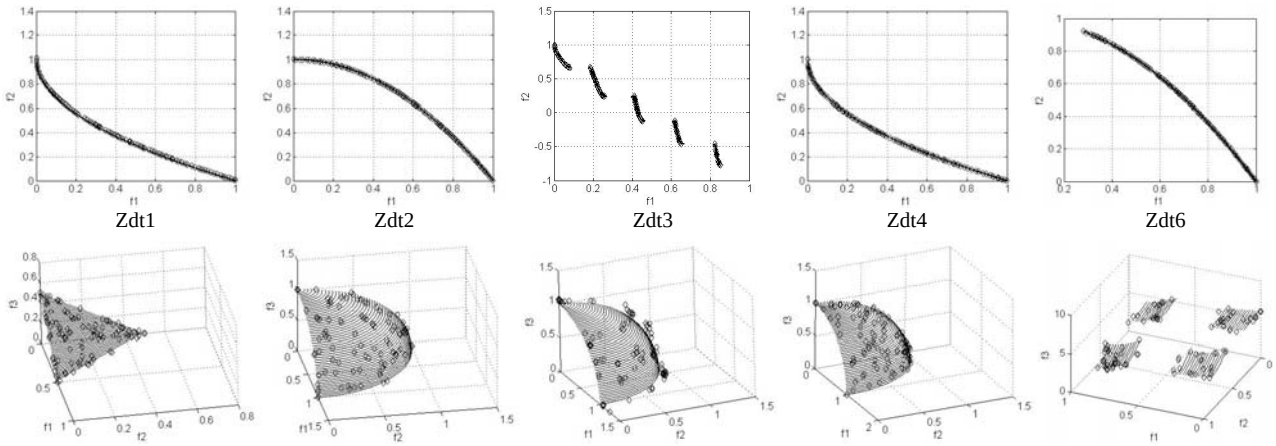
Observed from Table II and Table III, NCMO is better than NNIA in four ZDT functions and four DTLZ functions when regarding the convergence metric. Especially in DTLZ3, NCMO performs much better than NNIA. Because DTLZ3 have $(3^{|x_M|}-1)$ local Pareto-optimal fronts, it is very difficult to be optimized by many state-of-the-art multiobjective optimization algorithms. It is noted that DTLZ1 also has many local Pareto-optimal fronts, but NCMO performs a litter worse than NNIA. When regarding the diversity of solutions, although both algorithms use the same selection mechanism, NCMO gets better results than NNIA in five ZDT functions and three DTLZ functions. These simulation results evidently show the advantages of our novel approaches in this paper.

From the above comparison, we tabulate the performance results of NCMO comparing with various multiobjective algorithms in Table IV.

TABLE IV. THE PERFORMANCE RESULTS OF COMPARISON BETWEEN NCMO WITH VARIOUS MULTIOBJECTIVE ALGORITHMS

	Convergence metric	**Diversity metric**	**Spacing metric**
NCMO/NSGA-II	Better	Better	\
NCMO/PAES	Better	Better	\
NCMO/SPEA	Better	Better	\
NCMO/SPEA-2	Better	Better	\
NCMO/MOEO	Better	Worse	\
NCMO/NNIA	Better	Better	Better

In the above table, "Better" means that the performance of NCMO is generally better than other algorithms in the corresponding metric. "\" indicates that there is no comparison between NCMO and other algorithms. "Worse" means that the performance of NCMO is generally worse than other algorithms in the corresponding metric. In conclusion, it is evident that NCMO has two main advantages with the three improvement approaches. Firstly, NCMO can converge fast to the Pareto-optimal front and has the ability to escape from local Pareto-optimal fronts. Therefore, NCMO is robust when searching the global Pareto-optimal front in big search space with many local Pareto-optimal fronts. Secondly, NCMO is capable of keeping the desirable diversity of solutions. To graphically show the performance of NCMO, we show the images of the simulations obtained by NCMO in solving all benchmark functions respectively in Fig. 3. It is noted that the black line is the Pareto-optimal front and the diamond symbol is the corresponding front of the approximation set obtained by using NCMO. From the simulation images, it is easy to get the result that NCMO can get the approximation set that is distributed uniformly and close to the Pareto-optimal front.



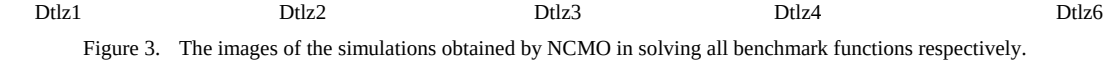


Figure 3. The images of the simulations obtained by NCMO in solving all benchmark functions respectively.

4) Simulations for the novel approaches

In order to further investigate the performance of the three improvement approaches in proposed algorithm, we make 30 independent runs of algorithm only with one of three approaches, i.e., dynamic mutation probability (SBX operator and polynomial mutation operator), D-SBX operator (static mutation probability and polynomial mutation operator) or GP-HM operator (static mutation probability and SBX operator) respectively. The parameters setting are the same with the above setting except that the crossover probability is set as 1.0. The results are shown in Table V and Table VI below. For increasing the precision of the convergence value, we find a set of more than 10000 uniformly spaced solutions from the Pareto-optimal front to calculate the convergence metric for every test functions. The number in brackets means times the algorithm converges to single approximate Pareto-optimal.

TABLE V. THE AVERAGE VALUE OF CONVERGENCE METRIC OBTAINED BY ALGORITHM WITH ONE OF THREE NOVEL APPROACHES AND NCMO WITH COMBINED APPROACHES

	Dynamic mutation probability	**D-SBX**	**GP-HM**	**Combined (NCMO)**
ZDT1	1.5040E-05	1.9335E-05	**9.6427E-06**	9.7176E-06
ZDT2	4.7625E-05(9)	4.8806E-05(5)	4.3416E-05	**4.2830E-05**
ZDT3	**2.3015e-005**	2.4277E-05	2.3778e-005	2.3290E-05
ZDT4	0.0028(9)	**0.0022(11)**	0.0025	0.0025
ZDT6	5.6340e-04	5.8041e-04	**5.1238e-04**	5.4041e-04
DTLZ1	0.0850	0.7040	0.0151	**0.0110**
DTLZ2	0.0084	0.0081	0.0078	**0.0072**
DTLZ3	7.9052	2.9255	2.4040	**1.4917**
DTLZ4	0.0085(1)	0.0074(4)	0.0077(2)	**0.0073**
DTLZ6	0.0147	**0.0131**	0.0148	**0.0131**

TABLE VI. THE AVERAGE VALUE OF SPACING METRIC OBTAINED BY ALGORITHM WITH ONE OF THREE NOVEL APPROACHES AND NCMO WITH COMBINED APPROACHES.

	Dynamic mutation probability	**D-SBX**	**GP-HM**	**Combined (NCMO)**
ZDT1	0.0071	**0.0070**	0.0074	0.0072
ZDT2	0.0073(9)	**0.0071(5)**	0.0074	0.0072
ZDT3	0.0081	**0.0077**	0.0079	**0.0077**
ZDT4	0.0067(9)	0.0066(11)	0.0068	**0.0063**
ZDT6	0.0056	0.0055	0.0056	**0.0053**
DTLZ1	0.0561	0.2613	0.0226	**0.0222**
DTLZ2	0.0580	0.0580	**0.0574**	0.0584
DTLZ3	3.7108	1.8189	0.2348	**0.1996**
DTLZ4	0.0591(1)	0.0575(4)	**0.0565(2)**	0.0582
DTLZ6	**0.0714**	0.0728	0.0738	0.0746

Observed from Table V, as far as the convergence metric is concerned, the algorithm performs best in ZDT3 if only the dynamic mutation probability is used. If only the D-SBX operator is used, the algorithm performs best in ZDT 4 and DTLZ6. If only the GP-HM operator is used, the algorithm performs best in ZDT1 and ZDT 6. When all novel approaches are used, the algorithm performs best in ZDT2, DTLZ1, DTLZ2, DTLZ3, DTLZ6 and DTLZ4.

Observed from Table VI, as far as the spacing metric is concerned, the algorithm performs best in DTLZ6 if only the dynamic mutation probability is used. If only D-SBX is used, the algorithm performs best in ZDT1, ZDT2 and ZDT3. If the GP-HM is used, the algorithm performs best in DTLZ2 and DTLZ4. If all novel approaches are used, the algorithm performs best in ZDT3, ZDT4, ZDT6, DTLZ1 and DTLZ3.

From the above results, it is concluded that the algorithm with one of three novel approaches has its own advantage in different test functions, but NCMO with all novel approaches performs best in most cases, no matter considering the convergence metric or the spacing metric. It illustrates that algorithm with D-SBX or GP-HM is suitable to solve the test functions with many local Pareto-optimal fronts. When solving ZDT3 with discreteness features, algorithm with dynamic mutation probability performs best. It is rational that whether these approaches can work well or not is relatively decided by the features of the test functions. Moreover, in solving ZDT2, ZDT4 and DTLZ4, algorithm with one of three novel approaches may only converge to single Pareto-optimal occasionally, while NCMO can get multiple solutions every times. It means that these three novel approaches can be well incorporated to preserve the population diversity. Only use one improved approach may perform better than NCMO when the optimized functions have some special features, but in real life problems, it is difficult to know the details of the optimized function. Therefore, it is hard to choose the improved approaches. However, if NCMO is used to solve the optimized functions, it can get better performance in general.

IV. CONCLUSIONS

In this paper, we proposed a novel clonal algorithm for MOPs, which uses three novel approaches with dynamic mutation probability, D-SBX operator and GP-HM operator. The notion of these approaches is straightforward and easy to be implemented. When comparing with various state-of-the-art algorithms and recently proposed algorithms, simulations show that NCMO with these approaches is robust. It can not only evidently accelerate the convergence speed, but also keep the desirable diversity.

REFERENCES

[1] C.M. Fonseca and P.J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization", *Evolutionary Computation*, vol. 3, pp. 1-16, 1995.

[2] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp.182-197, 2002.

[3] Eckart Zitzler, Marco Laumanns and Lothar Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm", Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
[4] Joshua Knowles and David Corne, "Approximating the non-dominated front using the pareto archived evolution strategy", *Evolutionary Computation*, vol. 8, No. 2, pp.149-172, 2000.
[5] Dipankar Dasgupta, "Advances in Artificial Immune Systems", *IEEE Computational Intelligence Magazine*, vol. 1, pp. 40-49, 2006.
[6] L.N. De Castro and F.J. Von Zuben, "Learning and Optimization Using the Clonal Selection", *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 239-251, 2002.
[7] P.K. Harmer, P.D. Williams, G.H. Gunsch and G.B. Lamont, "An Artificial Immune System Architecture for Computer Security Applications", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 252-280, 2002.
[8] Hong-Wei Ge, Liang Sun, Yan-Chun Liang and Feng Qian, "An Effective PSO and AIS-Based Hybrid Intelligent Algorithm for Job-Shop Scheduling Systems", *IEEE Transactions on Man and Cybernetics*, vol. 38, no. 2, pp. 358-368, 2008.
[9] A.A. Freitas and J. Timmis, "Revisiting the Foundations of Artificial Immune Systems for Data Mining", *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 4, pp. 521-540, 2007.
[10] Canova, A., Freschi, F. and Tartaglia, M., "Multiobjective Optimization of Parallel Cable Layout", *IEEE Transactions on Magnetics*, vol. 43, no. 10, pp. 3914-3920, 2007.
[11] A.M. Whitbrook, U. Aickelin and J.M. Garibaldi, "Idiotypic Immune Networks in Mobile-Robot Control", *IEEE Transactions on Systems, Man, and Cybernetics* , vol. 37, no. 6, pp. 1581-1598, 2007.
[12] C.A. Coello Coello and N.C. Cortes, "Solving multi-objective optimization problems using an artificial immune system", *Genetic Programming and Evolvable Machines*, vol. 6, pp. 163-190, 2005.
[13] F. Freschi and M. Repetto, "Multi-objective optimization by a modified artificial immune system algorithm", In *Proceedings of the 4th International Conference on artificial immune systems*, 3627 of Lecture Notes in Computer Science, pp. 248-261, 2005.
[14] Maoguo Gong, Licheng Jiao, Haifeng Du and Liefen Bo, "Multi-objective Immune Algorithm with Nondominated Neighbor-based Selection", *Evolutionary Computation* (MIT Press), vol. 16, pp. 225-255, 2008.
[15] R.C. Purshouse and P.J. Fleming, "On the evolutionary optimization of many conflicting objectives", *Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770-784, 2007.
[16] D.W. Corne, J.D. Knowles and M.J. Oates, "The pareto-envelope based selection algorithm for multi-objective optimization", In *proceedings of 6-th international conference Parallel Problem Solving from Nature (PPSN VI)*, pp. 839-848, 2000.
[17] S. Bandyopadhyay, S. Saha, U. Maulik and K. Deb, "A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA", *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269-283, 2008.
[18] P.A.N. Bosman and D. Thierens, "The Balance Between Proximity and Diversity in Multiobjective Evolutionary Algorithms", *Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174-188, 2003.
[19] Chen Min-Rong and Lu Yong-Zai, "A novel elitist multiobjective optimization algorithm: Multiobjective extremal optimization", *European Journal of Operational Research*, Vol. 188, No. 3, pp.637-651, 2008.
[20] Kalyanmoy Deb and Ram Bhusan Agrawal, "Simulated binary crossover for continuous search space", *Complex Systems*, Vol. 9, No. 6, pp.115-148, 1995.
[21] Eckart Zitzler, Kalyanmoy Deb and Lothar Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results", *Evolution Computation*, Vol. 8, No. 2, pp.173-195, 2000.
[22] K. Deb, L. Thiele, M. Laumanns and E. Zitzler, "Scalable multi-objective optimization test problems", Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) ,Zurich, Switzerland, 2001.
[23] J.R. Schott, "Fault tolerant design using single and multictiteria gentetic algorithm optimization", Master thesis, Massachusetts Institute of Technology, 1995.
[24] http://see.xidian.edu.cn/iiip/mggong/Projects/NNIA.htm.

Jianyong Chen is associate professor in Department of Computer Science and Technology, Shenzhen University. He got his PhD from City University of Hong Kong, Hong Kong, China, in 2003. He is interested in Artificial Intelligence and Information Security.

He worked for ZTE Corporation as senior engineer of network technology from 2003 to 2006. After that, he joined the Shenzhen University as associated professor. He is vice-chairman of International Telecommunication Union-Telecommunication (ITU-T) SG17 since 2004 and editor of three recommendations developed in ITU-T SG17. He has published more than 10 papers and applied more than 30 patents in the field of Artificial Intelligence and Information Security.

Qiuzhen Lin is a graduate student in the college of Information Engineering, Shenzhen University. He got his bachelor's degree in Zhaoqing University. He is interested in Artificial Intelligence and network Security.