An Improved Harmony Search Algorithm for Constrained Multi-Objective Optimization

Yuelin Gao
Institute of Information and System Science, Beifang University for Nationalities, Yinchuan, China
Email: gaoyuelin@263.net

Xia Chang and Yingzhen Chen
Institute of Information and System Science, Beifang University for Nationalities, Yinchuan, China
Email: changxia0104 @163.com, chenyingzhen_1987@yahoo.com.cn

***Abstract*—An improved harmony search algorithm for constrained multi-objective optimization problems is proposed in this paper. Inspired by Particle Swarm Optimization, an inductor particle is introduced to speed up the convergence rate of the CMOHS. Two populations are adopted to increase the opportunity of finding the optimal solutions. Numerical experiments are divided into two parts: the first one compares the CMOHS with NSGA-II, and the other one compares the CMOHS with the algorithm without the inductor individual. The results show that the CMOHS is more effective than NSGA-II, and the induction mechanism improved the convergence and diversity of the algorithm.**

***Index Terms*—Multi-objective optimization, Harmony search algorithm, Double populations, Density**

I. Introduction

Evolutionary algorithm has been used in the fields of academia and engineering to solve optimization problems successfully. The unconstrained optimization problems and the constrained optimization problems all need to be solved. The difficulty of solving constrained optimization problems is that the restrictions on the constraint variables in practical problems are hard to meet. The optimal solution of constrained optimization problems must satisfy all the constraints and have the best performance for the objective function among all the feasible solutions simultaneously. For the constrained multi-objective optimization problems, the constraints and all the objective functions should be taken into consideration. So it is more difficult to find the optimal solution of the constrained multi-objective optimization problems. In practical application, inevitably, most problems are constrained multi-objective optimization problems. so how to resolve the constrained multi-objective optimization problems has been paid particular attention.

The traditional approach to dealing with constraints is by using the penalty function. That is to translate the constrained optimization problem into the unconstrained optimization problem. Michalewicz [1] has done a comprehensive review on the various constraints approach based on the current evolutionary algorithm, in which the penalty function method is widely used. Woldesenbet [2] proposed an adaptive penalty function. The key of the penalty strategy is how to design a penalty function and how to select the penalty coefficient. The penalty coefficient is often related to the practical problem, so it is difficult to set. In recent years, the method that changing the constrained optimization problem into the multi-objective optimization problem has aroused great attention. The main idea of this approach is to change the constraints into the multi-objective functions. Then the existing algorithm is used to solve the multi-objective optimization. This method does not require parameters setting, but with the constraints increase, the dimension of objective function and the difficulty of computation will increase. As the constrained multi-objective evolutionary algorithm is hard to be treated, the research in this area is still in developing stage at present. There are many representative researches. Deb [3] proposed the constraints dominance relationship to deal with constraints. Binh [4] combined the objective function and the constraint violation degree of the infeasible solution to compute the individual fitness, and sorted them based on the distance between the individual and the feasible boundary. Mu [5] proposed a technique similar to simulated annealing genetic algorithm. Infeasible solutions can be divided as the acceptable solutions and the unacceptable solutions by the infeasible degree in Mu's method. And the infeasible solutions close to the feasible region and change into the feasible solutions gradually with the evolution. Wang [6] proposed a new genetic algorithm, in which the neighborhood comparison and the archiving operation are utilized to smooth the

Manuscript received December 17, 2012; revised March 18, 2013; accepted April 9, 2013.

The work is supported by the Foundations of National Natural Science China (No.60962006, No.61102008) and by the Foundations of research projects of State Ethnic Affairs Commission (No.12BFZ005).

Corresponding author: Yuelin Gao.

doi:10.4304/jsw.8.10.2554-2561

conflicting objectives. Infeasibility degree selection is used to handle constraints. Meng [7] proposed a new algorithm based on the double populations for constrained multi-objective optimization Problem (CMOP). In the proposed algorithm, two populations are adopted, one is for finding the feasible solutions during the evolution, and the other is for finding infeasible solutions with better performance which are allowed to participate in the evolution with the advantage of avoiding constructing penalty function and deleting infeasible solutions directly.

Harmony search algorithm [8] is a new intelligent optimization algorithm, proposed by Geem in 2001. As the genetic algorithm imitates biological evolution, the simulated annealing algorithm mimics the physical annealing mechanism, and the PSO imitates the predation of birds, the harmony search algorithm simulates the principles of playing music. The harmony search is widely used in many fields. Sadik [9] proposed a developing harmony search-based algorithm to determine the minimum cost design of steel frames with semi-rigid connections and column bases under displacement, strength and size constraints. Afkousi [10] proposed a harmony search algorithm to solve the unit commitment (UC) problem. Zeblah [11] used a harmony search meta-heuristic optimization method to solve the multi-stage expansion problem for multi-state series-parallel power systems.

This paper is structured as follows. Section II states the multi-objective constrained optimization problem. Section III describes the harmony search algorithm for constrained multi-objective optimization problems. Section IV shows the experimental results. Conclusions are drawn in Section V.

II. Basic Concepts of CMOP

The constrained multi-objective optimization problem can be mathematically described as:

$$\begin{cases} \min \ y = f(x) = [f_1(x), f_2(x), \cdots, f_m(x)] \\ \text{s.t.} \ \ g_i(x) \le 0, \ i = 1, 2, \cdots, p \\ \qquad h_i(x) = 0, \ i = 1, 2, \cdots, q \end{cases} \tag{1}$$

where m is the number of target vectors, p is the number of inequality constraints, q is the number of equality constraints, $x = (x_1, x_2, \ldots, x_n) \in D$ is the decision variables, $y = (f_1, f_2, \ldots, f_m) \in Y$ represents the target vector, $g(x) \le 0$ are the inequality constraints, $h(x) = 0$ are the equality constraints, D is the decision space, and Y represents the target space.

In general, the equality constrain $h_i(x) = 0$, can be transformed into two inequality constraints viz. $h_i(x) \ge 0$ and $h_i(x) \le 0$. The $h_i(x) \ge 0$ can be transformed into $-h_i(x) \le 0$. Thus, all constraints can be completely converted into $g(x) \le 0$. Therefore, constrained multi-objective optimization problem model can be described as:

$$\begin{cases} \min \ y = f(x) = [f_1(x), f_2(x), \cdots, f_m(x)] \\ \text{s.t.} \ \ g_i(x) \le 0, \ i = 1, 2, \cdots, c \end{cases} \tag{2}$$

where m is the number of target vectors, c is the number of constraints, $x = (x_1, x_2, \ldots, x_n) \in D$ is the decision variables, $y = (f_1, f_2, \ldots, f_m) \in Y$, represents the target vector, $g(x) \le 0$ are the constraints, D is the decision space, Y represents the target space.

There are several basic concepts which are often used in multi-objective optimization [12]:

(1) Pareto dominate: A decision vector x^0 is said to dominate a decision vector x^1 (also written as $x^0 \succ x^1$) if and only if $\forall i \in \{1,...,m\}$: $f_i(x^0) \ge f_i(x^1)$, $\wedge \ \exists j \in \{1, \cdots, m\}$: $f_i(x^0) > f_i(x^1)$.

(2) Pareto optimal solution: A decision vector x^0 is said to be non-dominated, if and only if there is no decision vector x^1 which dominates x^0, formally: $\neg\exists x^1: x^1 \succ x^0$.

(3) Pareto optimal set: The Pareto optimal set P_S is defined as $P_S = \{x^0 | \neg\exists x^1 \succ x^0\}$ also called non-dominated optimal set.

(4) Pareto optimal front: The Pareto front P_F is defined as $P_F = \{ f(x) = (f_1(x), f_2(x), \ldots, f_m(x)) | x \in P_S \}$.

III. Harmony Search Algorithm for CMOP

Basic harmony search algorithm is described as follows.

Step 1. Set the basic parameters of harmony search.

Step 2. Initialize the harmony memory.

Step 3. Generate new solutions. A new solution is generated through three mechanisms. (1) Keep the vectors in the harmony memory. (2) Randomly generate. (3) Make disturbance to some components of (1) and (2).

Step 4. Update harmony memory. If the new solution is better than the worst solution in the harmony memory, replace the worst solution with the new solution, and get the harmony memory.

Step 5. If the maximum iteration is reached, stop and output optimal solution, otherwise return to Step3.

The following is the improved harmony search algorithm to deal with multi-objective constrained optimization problems.

A. Improvement of the Harmony Memory Consideration Rate and the Pitch Adjusting Rate

In the basic harmony search algorithms, the harmony memory consideration rate (HMCR) and the pitch adjusting rate (PAR) are fixed value. However, in the actual study we found that the amount of the next evolution solutions should be the same as that of their parents as possible when the number of feasible solutions in the population is large. The HMCR should be increased, and the PAR should be reduced. On the contrary, when the number of feasible solutions in the population is small, the HMCR should be reduced, and the PAR should be increased. The HMCR and PAR are calculated as follows:

$$HMCR = \frac{pop_f}{pop} \tag{3}$$

$$PAR = \frac{pop_c}{pop} \tag{4}$$

where $HMCR$ is the harmony memory consideration rate, PAR is the pitch adjusting rate, pop is the population size, pop_f is the number of feasible solution in the population, and pop_c is the number of infeasible solution in the population.

B. Generate New Solutions

It is different from the basic harmony search algorithm. To deal with constrained multi-objective optimization problems, infeasible solutions are also participated in the evolution. The harmony memory is divided into a feasible solution set $HM1$ and an infeasible solution set $HM2$. Feasible solution and infeasible solution evolve by different methods which are described as follows.

For the feasible solution set $HM1$:

$$x'_{new} = \begin{cases} HM1_{round(rand(hm1-1)+1)} & r_1 \le HMCR \\ rand(ub - lb) + lb & r_1 > HMCR \end{cases} \tag{5}$$

$$x_{new1} = HM1(i) + \lambda(x_{best} - x'_{new}), \quad i = 1, 2, \cdots hm1 \tag{6}$$

where r_1 is a random number between 0 and 1, $HM1$ is the feasible solution set, $hm1$ is the size of the feasible solution set, and ub and lb is the lower bound and upper bound respectively. If $r_1 \le HMCR$, solutions in feasible solution set are selected randomly. If $r_1 > HMCR$, generated solutions randomly. x_{best} is the optimal solution in the external populations, the specific selection method is described later.

For the infeasible solution set $HM2$:

$$x_{new1} = \alpha HM2(i) + (1-\alpha)x_{best}, \quad i = 1, 2, \cdots hm2 \tag{7}$$

where $HM2$ is the infeasible solution set, $hm2$ is the size of the infeasible solution set, and x_{best} is the optimal solution in the external populations.

For all solutions in the feasible solution set and infeasible solution set, if $rand \le PAR$, the solution generated needs for adjusting, which is described as:

$$x_{new} = x_{new1} + \beta(ub - lb) \tag{8}$$

C. Calculate Individual Density

For individuals in the external population, it is difficult to recognize which one is the best. The density is adopted to find the best individual. This paper presents a new distance measurement method, and the density function is described as:

$$d_{i,j} = \begin{cases} 0 & |d_i - d_j| > 2\sigma \\ 1 - (\frac{d_i - d_j}{2\sigma}) & |d_i - d_j| \le 2\sigma \end{cases} \tag{9}$$

$$D_i = \sum_{j=1}^{M} d_{i,j} \tag{10}$$

where $|d_i - d_j|$ is the Euclidean distance between individual i and individual j. δ is the standard deviation of the distance that the individual i to all the individual in the population. $d_{i,j}$ is the impact value of the individual i to the individual j. D_i is the density of the individual i. Formula (9) means that if the distance between individual i and individual j beyond 2δ, then the individual j has no effect on the individual i. When calculating the density of the individual i, individual j can not be considered. If the distance between individual i and individual j is less than 2δ, then individual j impacts individual i. The distance is smaller, the impact is greater, and the density is greater. Calculate the influence that all the individual impose on the individual i and sum them, which are the density of the individual i.

D. Select the Best Individual

The x_{best} is from particle swarm optimization [13-15]. In the PSO, all particles have to study the best particle in the population. This can increase information sharing and mutual cooperation between particles. The x_{best} is selected in the external population A. Calculate the density of each individual in external population by formula in III.C, the minimum density of the individual is selected as the global optimal solution.

E. Select the Evolutionary Population

The new solution produced by harmony search algorithm will also be stored in the harmony memory. Then there will be a larger population, which is divided into feasible solution set $HM1$ and the infeasible solution set $HM2$. The feasible solution set $HM1$ is sorted by the dominance relationship, and is described as follows. Select all non-dominated solutions in the population and define their order as 1. Then select the non-dominated solutions in the rest of the individuals and define them order as 2. Rank individuals from small to large until all the individuals are assigned an order. If the individuals have the same order, rank them by density, the one who has the smaller density is in the front. For the infeasible solution set $HM2$, sort them according to their constraint violation degree, the one who has the smaller constraint violation value is in the front. Thus we re-sort the harmony memory and the front HM individuals are selected as the next evolution generation.

F. Steps of Harmony Search Algorithm for CMOP

Step 1. Set the basic parameters of harmony search. Set the external population $A = \varnothing$.

Step 2. Initialize the harmony memory. Put the feasible non-dominate solutions in the external population until $A \ne \varnothing$.

Step 3. Generate new solutions. The harmony memory is divided into feasible solution set HM1 and infeasible solution set HM2. They generate new individuals according to the formula (5), (6), and (7).

Step 4. Update harmony memory. Add the newly generated individuals to the harmony memory and select the next evolution generation according to the III.E.

Step 5. If the maximum iteration is reached, stop and output optimal solution, otherwise return to Step 3.

IV. EXPERIMENTAL RESULTS

A. Performance Index

The dispersion of the solutions measured using the following formula

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

$$d_i = \min_{j\in(1,n)}\left(\sum_{k=1}^{n}\left|f_k^i(x) - f_k^j(x)\right|\right) \quad (11)$$

$$i, j = 1, 2, \cdots, n, i \neq j$$

where n is the number of obtained solutions, d_i is the distance between the individual i and its nearest individual, $\bar{d}$ is the average distance of d_i. The expression $SP = 0$ shows that the solution evenly distributed in the Pareto front. The indicator reflects the uniformity of the solutions.

The quality evaluation of the solutions for multi-objective optimization problems is mainly concerned in the distance between the obtained non-dominated solutions set and the Pareto optimal set, and the diversity of the obtained non-dominated solutions set. Here the generational distance [16] (Van Veldhuizen and Lamont, 1998) is adopted to measure the distance between the obtained non-dominated solutions set and the Pareto optimal set. And the extension indicator (Zhou, 2006) is adopted to evaluation the diversity of the solutions. The generational distance is defined as:

$$GD = \frac{\left(\sum_{i=1}^{n} d_i^P\right)^{\frac{1}{P}}}{n} \quad (12)$$

where n is the number of vectors in the set of non-dominated solutions found so far, d_i is the Euclidean distance (measured in objective space) between each of these solutions and the nearest member of the Pareto optimal set, and p is a positive integer. $p = 1$ or $p = 2$ usually. A smaller value of GD demonstrates a better convergence to the Pareto front. It is clear that a value $GD = 0$ indicates that all the generated elements are in the Pareto front.

The extend indicator is proposed by Deb,which is modified as

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1}|d_i - \bar{d}|}{d_f + d_l + (n-1)\bar{d}} \quad (13)$$

where the parameter d_i is the Euclidean distance of neighboring solutions in the obtained non-dominated solutions set and $\bar{d}$ is the mean of all d_i. The parameters d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set. A value of zero for this metric indicates all members of the Pareto optimal set are equidistantly spaced. When the non-dominated solutions found so far are complete distribute evenly on the balance surface, namely, $d_f = 0$, $d_l = 0$, and $d_i = \bar{d}$, so $\Delta = 0$ .The symbol Δ reflects the distribution and the diversity of the non-dominated solutions.

B. Parameters Setting and Results Analysis

We compared the CMOHS algorithm with NSGA-II in order to know the efficiency of the CMOHS algorithm. Four benchmark functions are chosen to test the performance of the algorithms, which are commonly used in the constrained multi-objective evolutionary algorithm test. They are belegundu, binh (2), srinivas and tanaka. In the CMOHS algorithm and the NSGA-II algorithm, the population size is set as 50 and the number of iterations is set as 100. Algorithms run 15 times independently. The maximum, minimum, average, standard deviation of the indicator convergence, diversity and spread are summarized in Table I to Table IV.

Table I to Table IV shows that the convergence and spread of CMOHS is significantly better than that of NSGA-II. The diversity of CMOHS is better than that of NSGA-II in binh (2) and srinivas, but weaker in belegundu and tanaka. It can be seen from the standard deviation that CMOHS is more stable than NSGA-II. Fig. 1. ~ Fig. 4. are the Pareto fronts of CMOHS for the four test functions.

In order to test the impact of the guide individual to CMOHS, we compared CMOHS with the algorithm without guide individual (signed alg1). The alg1 is described as follows.

Step 1. Set the basic parameters of harmony search. Set the external population $A = \varnothing$.

Step 2. Initialize the harmony memory. Put the feasible non-dominate solutions in the external population until $A = \varnothing$.

Step 3. Generate new solutions. The harmony memory is divided into feasible solution set HM1 and infeasible solution set HM2. They generate new individuals according to the following formula:

$$x_{new1} = \begin{cases} HM1_{round(rand(hm1-1)+1)} & r_1 \le HMCR \\ rand(ub-lb)+lb & r_1 > HMCR \end{cases} \quad (14)$$

$$x'_{new} = HM1(i) + \lambda x_{new1}, \quad i = 1, 2, \cdots hm1 \quad (15)$$

$$x_{new} = x'_{new} + \beta(ub-lb) \quad (16)$$

Step 4. Update harmony memory. Add the new generated individuals to the harmony memory, select the next evolution generation according to section III.E.

Step 5. If the maximum iteration is reached, stop and output optimal solution, otherwise return to Step3.

Six benchmark functions are chosen to test the performance of the two algorithms. The four benchmark functions are the same with that discussed above, and parameters setting is not changed. The parameters of obayashi are set as that, the population size is 100 and the number of iterations is 200. The parameters of jimenez are set as that, the population size is 100 and the number of iterations is 500.

From Table V to Table X, it can be found that the convergence, spread and diversity of CMOHS are significantly improved relative to alg1 in belegundu, binh, and srinivas. For the benchmark tanaka, the convergence and spread of CMOHS is better than alg1 and the diversity of CMOHS is worser than alg1. For the benchmark jimenez, the convergence and spread of CMOHS is better than alg1, the diversity of CMOHS is the same as that of alg1. For the benchmark obayashi, the convergence, spread and diversity of CMOHS is the same as that of alg1. Overall, the guide individual improves the performance of CMOHS.

TABLE I
EVALUATION RESULTS OF THE BENCHMARK BELEGUNDU

Evaluation index	GD		Δ		SP	
Algorithm	NSGA-II	CMOHS	NSGA-II	CMOHS	NSGA-II	CMOHS
maximum	0.0190	9.2000e-4	1.6219	1.6010	0.0158	0.0018
minimum	0.0025	9.6800e-5	1.1480	0.8990	0.0050	1.0000e-4
average	0.0095	1.7367e-4	1.3419	1.0693	0.0095	5.6000e-4
SD	0.0048	2.0721e-4	0.1211	0.1998	0.0029	5.8530e-4

TABLE II
EVALUATION RESULTS OF THE BENCHMARK BINH (2)

Evaluation index	GD		Δ		SP	
Algorithm	NSGA-II	CMOHS	NSGA-II	CMOHS	NSGA-II	CMOHS
maximum	0.0907	0.0367	1.2986	0.9745	0.2921	0.1157
minimum	0.0205	0.0079	1.0527	0.6546	0.0511	0.0029
average	0.0325	0.0139	1.1675	0.7762	0.1328	0.0331
SD	0.0169	0.0074	0.0821	0.1131	0.0681	0.0335

TABLE III
EVALUATION RESULTS OF THE BENCHMARK SRINIVAS

Evaluation index	GD		Δ		SP	
Algorithm	NSGA-II	CMOHS	NSGA-II	CMOHS	NSGA-II	CMOHS
maximum	0.1362	0.0060	1.2443	0.6607	0.3171	0.0351
minimum	0.0215	0.0026	1.0539	0.5886	0.0845	0.0115
average	0.0708	0.0037	1.1197	0.6188	0.1486	0.0171
SD	0.0297	8.3512e-4	0.0582	0.0222	0.0534	0.0065

TABLE IV
EVALUATION RESULTS OF THE BENCHMARK TANAKA

Evaluation index	GD		Δ		SP	
Algorithm	NSGA-II	CMOHS	NSGA-II	CMOHS	NSGA-II	CMOHS
maximum	0.0032	0.0024	1.3633	1.2264	0.0120	0.0062
minimum	1.6995e-4	3.0000e-4	1.0341	0.9672	2.1437e-4	4.0000e-4
average	9.5232e-4	7.5333e-4	1.1689	1.0225	0.0029	0.0010
SD	7.8863e-4	6.0694e-4	0.0981	0.0669	0.0032	0.0014

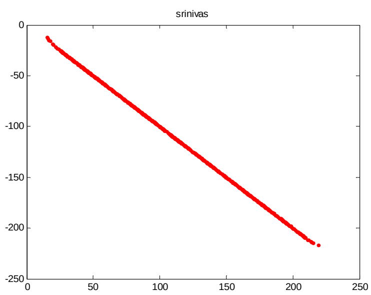


Fig. 1. belegundu

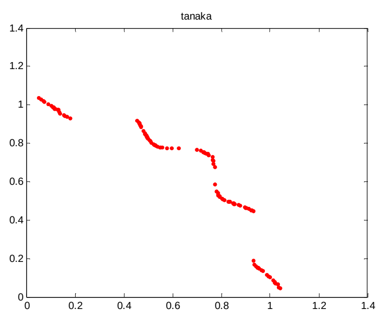


Fig. 2. binh (2)

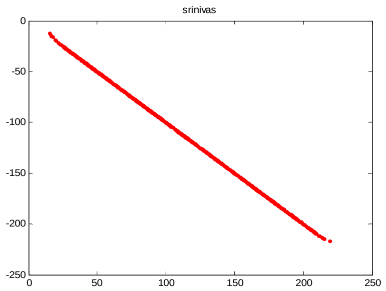


Fig. 3. srinivas

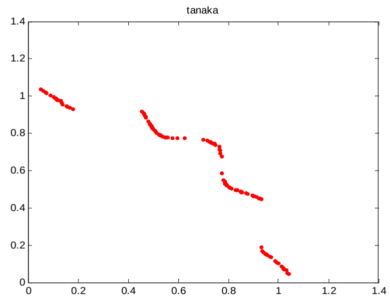


Fig. 4. tanaka

TABLE V
EVALUATION RESULTS OF THE BENCHMARK BELEGUNDU

Evaluation index	GD		Δ		SP	
Algorithm	alg1	CMOHS	alg1	CMOHS	alg1	CMOHS
maximum	0.0625	9.2000e-4	1.0991	1.6010	0.0149	0.0018
minimum	0.0067	9.6800e-5	0.6953	0.8990	0.0036	1.0000e-4
average	0.0281	1.7367e-4	0.8960	1.0693	0.0076	5.6000e-4
SD	0.0167	2.0721e-4	0.1103	0.1998	0.0035	5.8530e-4

TABLE VI
EVALUATION RESULTS OF THE BENCHMARK BINH (2)

Evaluation index	GD		Δ		SP	
Algorithm	alg1	CMOHS	alg1	CMOHS	alg1	CMOHS
maximum	0.1948	0.0367	1.3659	0.9745	0.1591	0.1157
minimum	0.0155	0.0079	0.6919	0.6546	0.0095	0.0029
average	0.0447	0.0139	0.8564	0.7762	0.0499	0.0331
SD	0.0436	0.0074	0.1916	0.1131	0.0339	0.0335

TABLE VII
EVALUATION RESULTS OF THE BENCHMARK SRINIVAS

Evaluation index	GD		Δ		SP	
Algorithm	alg1	CMOHS	alg1	CMOHS	alg1	CMOHS
maximum	0.6290	0.0060	1.0807	0.6607	0.3901	0.0351
minimum	0.0107	0.0026	0.6719	0.5886	0.0350	0.0115
average	0.1352	0.0037	0.8252	0.6188	0.1494	0.0171
SD	0.1731	8.3512e-4	0.1156	0.0222	0.1163	0.0065

TABLE VIII
EVALUATION RESULTS OF THE BENCHMARK TANAKA

Evaluation index	GD		Δ		SP	
Algorithm	alg1	CMOHS	alg1	CMOHS	alg1	CMOHS
maximum	0.0801	0.0024	1.0509	1.2264	0.2118	0.0062
minimum	0.0123	3.0000e-4	0.6774	0.9672	0.0155	4.0000e-4
average	0.0435	7.5333e-4	0.8856	1.0225	0.0934	0.0010
SD	0.0195	6.0694e-4	0.0946	0.0669	0.0558	0.0014

TABLE IX
EVALUATION RESULTS OF THE BENCHMARK JIMENEZ

Evaluation index	GD		Δ		SP	
Algorithm	alg1	CMOHS	alg1	CMOHS	alg1	CMOHS
maximum	3.4588	0.0656	1.4137	1.3821	12.2186	0.1464
minimum	0.0584	0.0351	0.9940	1.1620	0.1207	0.0478
average	0.9188	0.0460	1.2958	1.2864	2.6269	0.0934
SD	1.1532	0.0076	0.1031	0.0654	3.8965	0.0286

TABLE X
EVALUATION RESULTS OF THE BENCHMARK OBAYASHI

Evaluation index	GD		Δ		SP	
Algorithm	alg1	CMOHS	alg1	CMOHS	alg1	CMOHS
maximum	0.0013	0.0011	0.9284	0.9017	0.0015	0.0026
minimum	4.0000e-4	4.0000e-4	0.7883	0.7840	7.0000e-4	6.0000e-4
average	8.1333e-4	5.8667e-4	0.8550	0.8611	0.0010	0.0013
SD	2.5317e-4	1.9952e-4	0.0449	0.0397	2.2824e-4	7.8982e-4

V. CONCLUSION

This paper proposes an improved harmony search algorithm to solve constrained multi-objective optimization problems. There are two different populations to store feasible solution and infeasible solution respectively. Feasible solutions and infeasible solutions adopt different evolutionary mechanisms. The extreme global ideas improve the performance of harmony search algorithm. Six benchmark functions in the experimental results show that the convergence, spread and diversity of CMOHS are significantly superior to that of NSGA-II. But in the test we also found that the performance of CMOHS highly depends on the parameters. We will research deeply in the future work.

ACKNOWLEDGMENT

The work is supported by the Foundations of National Natural Science China under Grant No.60962006 and No.61102008 and by the Foundations of research projects of State Ethnic Affairs Commission under Grant No.12BFZ005.

REFERENCES

[1] Z. Michalewicz, D. Dasgupta, R. G. L. Riche, and M. Schoenauer, “Evolutionary algorithms for constrained engineering problems,” *Comput. Ind. Eng.*, vol. 30, no. 4, pp. 851–870, September 1996.

[2] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, “Constraint handling in multi-objective evolutionary optimization,” *IEEE Trans. Evolut. Comput.*, vol. 13, n.3, pp. 514–525, June 2009.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm：NSGA-II,” *IEEE Trans. Evolut. Comput.*, vol. 6, no. 2, pp. 182–197, April 2002.

[4] T. T. Binh and U. Korn, “MOBES: A multi-objective optimization evolution strategy for constrained Optimization Problems,” in *the Third International Conference on Genetic Algorithms*, pp. 176–182, 1997.

[5] S. Mu, H. Su, W. Mao, Z. Chen, and J. Chu, “A new genetic algorithm to handle the constrained optimization problem,” in *Proc. 41st IEEE Conference of decision and control*, pp. 739–740, 2002.

[6] Y. X. Wang, L. C. Wu, and S. J. Mou, “Constrained multi-objective optimization evolutionary algorithm,” *Tsinghua Science and Technology*, vol. 45, no. 1, pp. 103–106, January 2005. (China)

[7] H. Y. Meng, X. H. Zhang, and S. Y. Liu, “A Differential Evolution Based on Double Populations for Constrained Multi-Objective Optimization Problem,” *Chinese Journal of Computer*, vol. 31, no. 2, pp. 225–234, February 2009. (China)

[8] Z. W. Geem, J. H. Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: Harmony search,” *Simulation*, vol. 76, no. 2, pp. 60–80, 2001.

[9] S. O. Degertekin and M. S. Hayalioglu, “Harmony search algorithm for minimum cost design of steel frames with semi-rigid connections and column bases,” *Struct. Multidiscip. O.*, vol. 42, no. 5, pp. 755–768, November 2010.

[10] M. Afkousi-Paqaleh, M. Rashidinejad, and M. Pourakbari-Kasmaei, “An implementation of harmony search algorithm to unit commitment problem,” *Electr. Eng.*, vol. 92, no. 6, pp. 215–225, November 2010.

[11] A. Zeblah, S. Hadjeri, E. Chatelet, and Y. Massim, “Efficient harmony search algorithm for multi-stages scheduling problem for power systems degradation,” *Electr. Eng.*,vol. 92, no. 3, pp. 87–97, September 2010.

[12] K. Deb, *Multi-Objective optimization using evolutionary algorithm*, Chichester: John Wiley&Sons, 2001.

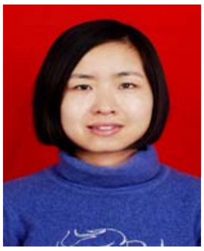
[13] J. Kennedy, R. Eberhart, “Particle swarm optimization,” in *Proc. IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[14] L. Wang, X. Wang, J. Fu, and L. Zhen, “A novel probability binary particle swarm optimization algorithm and its application,” Journal of Software, vol. 3, no. 9, pp. 38–35, December 2008.

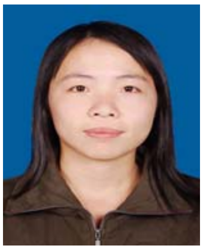
[15] Q. Niu and X. Huang, “An improved fuzzy C-mean clustering algorithm based on PSO,” Journal of Software, vol. 6, no. 5, pp. 873–879, May 2011.

[16] D. A. Van Veldhuizen and G. B. Lamont, “Evolutionary computation and convergence to a pareto front,” In John R Koza (editor). Late Breaking Papers at the genetic Programming 1998 Conference. Stanford University, California. Stanford Bookstore.1998:221–228.

Yuelin Gao, born in 1963, Ph. D., professor, Master supervisor. His main research interests currently include theory, methods and applications of optimization, intelligent information processing, financial computation and financial engineering.
Email:gaoyuelin@263.net



Xia Chang, born in 1982, Ph. D., lecturer. Her research interests include image sparse presentation, intelligence computation, and image understanding.
Email:changxia0104@163.com



Yingzhen Chen, born in 1987, Master's degree, Institute of Information and System Science, Beifang University for Nationalities, Yinchuan 750021, China. His main research interests currently include theory, methods of optimization, and intelligent information processing.

Email:chenyingzhen_1987@yahoo.com.cn