A Hybrid Harmony Search Algorithm with Genetic Algorithm for Unconstrained Multi-Objective Problems

[1]YINGLI LI，[2] CHAO LU，[3]BING ZENG

1, First Author YINGLI LI, Huazhong University of Science and Technology, lyl_elena@163.com
*2, Corresponding Author CHAO LU, Huazhong University of University of Science and Technology, luchao2006@163.com
3, BING ZENG, Huazhong University of Science and Technology, zengbing2016@126.com

Abstract

This paper presents a novel algorithm, i.e., hybrid multi-objective harmony search algorithm with genetic operation (HMOHS). In HMOHS, each harmony vector has two strategies (i.e. GA and HS) to generate a new vector of harmony among the process of evolution. The two strategies' cooperation is controlled by dynamical parameter HMCR, which can effectively choose an optimal strategy for maintaining a balance between exploration and exploitation during an evolution process. In addition, a new self-adaptive operator is applied to enhance the ability of global optimization in the earlier search stage, meanwhile, it enhances the ability of local optimization with fine-tuning operation in the later search process, and this adaptive operator can replace the role of parameter BW. HMOHS has been evaluated on benchmark problems, against its variants and three state-of-the-art multi-objective evolutionary algorithms in terms of spread, convergence, coverage and convergence speed. The obtained results indicate that the HMOHS is a promising approach for solving these types of problems.

Keywords：*Evolutionary Algorithm, Harmony Search, Multi-objective Optimization*

1. INTRODUCTION

Multi-objective optimization problems (MOPs) linked to more than one objective for minimization and/or maximization, simultaneously. Its result is a set of solutions, namely “non-dominated solutions” or “compromise solutions”, rather than a single one, which are optimal tradeoff candidates among all sub-objectives[1][2]. Each solution, in the non-dominated set, is a Pareto point. Multi-objective optimization aims to find Pareto front and non-dominated solutions. Since its search space is more large in multi-objective problems. Thus, MOPs takes more time to conduct search procedure which achieves optimization for sub-objective, simultaneously. Traditionally, MOPs have been solved by weighted sum approach, which converts MOP to a single objective optimization problem by giving fixed weights to sub-objectives. However, this method requires multiple runs, aims to get a set of non-dominated solutions and need much computational time resulting in a set of weak non-dominated solutions [3]. Recently, meta-heuristic multi-objective algorithms, proved to outperform the traditional approaches because of the ability to obtain Pareto optimal solutions in a single run, has been draw attention by many researchers and scholars. Meanwhile, some classic meta-heuristic multi-objective algorithms such as NSGA-II[4], SPEA2[5], and MOEA/D[6] have been widely employed for different applications.

HS invented by Geem *et al*. [7] is one of the population-based meta-heuristics in an analogy with improvisation process where music players improvise their instruments' pitches or notes to search a beautiful harmony. Improvisation process as the optimum design process which find optimal solution like each pitch of the harmony decides the quality of music. HS is useful to many engineering applications with its' few mathematical requirements and easy implementation [8]-[12]. Currently, many improved HS algorithms for single-objective problems have been proposed, such as, global-best harmony search (GHS) algorithm [14], self-adaptive global-best harmony search (SGHS) algorithm [15], local-best harmony search algorithm with dynamic subpopulations (DLHS)[16], the novel global harmony search (NGHS) algorithm [17], and an improved adaptive binary Harmony Search (ABHS) algorithm [18]. In optimization applications of multi-objective harmony search (MOHS), S. Sivasubramani *et al.* used MOHS for power flow problem [3] and Environmental/economic dispatch [19]; I. Landa-Torres *et al*. presented a novel multi-objective HS for the optimal distribution of 24-h emergency units [20]; S. Salcedo-Sanz et al. applied MOHS into urban traffic reconfiguration[21]; K. Nekooei et al. proposed an improved MOHS for placing Distributed Generators in radial distribution systems [22]. MOHS has been widely applied, however lack of theoretical research has existed. It may attribute to its certain advantage in terms of global search optimization, but it uses a stochastic random search which recede convergence in optimization applications. In addition, MOHS involves some parameters (i.e. HMCR, PAR, and BW) which have a profound impact on the performance of MOHS algorithm. And it is difficult to adjust these parameters since these parameters are problem-dependent.

To solve above issues and improve performance of MOHS, a hybrid multi-objective HS with genetic algorithm (GA [23]) is proposed, which inspired by hybrid mechanism with other EAs and self-adaptive operation such as in [24]-[26]. The characteristics of this algorithm: (1) a new harmony comes from GA operation with (1-HMCR) probability rather than random selection, which can improve the convergence toward Pareto optimal solutions, (2) dynamically fine-tuning parameter HMCR keep a balance between exploitation and exploration, (3) a new adaptive operator is introduced into the proposal, which eliminates the parameter BW and can enhance the ability of global search in initial evolution stage and local search in later evolution process. Problems addressed in this work can be summarized as follows:

- A hybrid MOHS with GA is proposed, where a new harmony stems from three approaches. (1) memory consideration (2) pitch adjustment and (3) GA operation rather than random selection which is not favor of convergence during the search.
- The proposed algorithm uses a new kind of adaptive mechanism to dynamically maintain harmony (or population) diversity.
- The proposed algorithm employs dynamical parameters for HMCR, to control GA and HS operation for a balance between exploration and exploitation, resulting in improving the convergence performance.
- The presented algorithm is evaluated with its variants and three state-of-the-art algorithms such as NSGA-II, SPEA2 and MOEA/D on a benchmark of MOPs.
- Convergence speed among above mentioned algorithms is compared and analyzed.

From Section 2 to 4, paper presents related background; an algorithm for unconstraint MOPs, HMOHS is proposed; experience results are presented and analyzed. Conclusion is drawn in Section 5.

2. BACKGROUND

The mathematical definition of a MOP for minimization can be described as follows:

$$\begin{aligned} &\min\ f(\mathbf{x}) = \min[f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})] \\ &\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Re^n \\ &s.t. \begin{cases} g_i(\mathbf{x}) > 0, i = 1, \ldots, K \\ h_j(\mathbf{x}) = 0, j = 1, \ldots, N \end{cases} \end{aligned} \tag{1}$$

where $f_m(\mathbf{x})$ indicates the m-th sub-objective function, $g_i(\mathbf{x})$ represents inequality constraints; $h_j(\mathbf{x})$ is equality constraints, K and N are the number of inequality and equality constraints respectively, $\mathbf{x}$ is a vector of the solution which need to satisfy above constraints and $\Re^n$ is decision variable space. The concepts of non-dominated and related terms are defined as follows:

Definition 1. The p-th solution $\mathbf{x}_p$ dominates the q-th solution $\mathbf{x}_q$ (denoted by $\mathbf{x}_p \prec \mathbf{x}_q$), if and only if $\forall i \in \{1, \ldots, m\}: f_i(\mathbf{x}_p) \le f_i(\mathbf{x}_q) \wedge \exists l \in \{1, \ldots, m\}: f_l(\mathbf{x}_p) < f_l(\mathbf{x}_q)$.

Definition 2. A solution $\mathbf{x}$ is said to be non-dominated by any other solution if and only if $\neg\exists \mathbf{x}^* \in \Re^n : \mathbf{x}^* \prec \mathbf{x}$.

Definition 3. For a given MOP, a Pareto optimal set (PS*) can be defined as below: $PS^* = \left\{ \mathbf{x} \middle| \neg\exists \mathbf{x}^* \in \Re^n : \mathbf{x}^* \prec \mathbf{x} \right\}$.

Definition 4. For a given MOP, if it's Pareto set is PS*, the corresponding Pareto front (PF*) can be defined as $PF^* = \left\{ f(\mathbf{x}) \middle| \mathrm{x} \in PS^* \right\}$.

Most MOPs may have infinite PF*, it is time-consuming to obtain all the PF*. So, many multi-objective evolutionary algorithms (MOEAs) are to yield a set of front (denoted as symbol PF) with finite size which are evenly distributed along the PF*, and thus good representatives of the entire PF*. Notice that Pareto optimal set (PS*) is always a non-dominated set according to definition while non-dominated solutions generated by an algorithm, which is denoted as symbol PS, may not be a subset of PS*.

3. HMOHS ALGORITHM

3.1 The Proposed Multi-Objective Harmony Search Algorithm

A novel multi-objective HS algorithm (HMOHS), for solving continuous MOPs is presented. The proposed HMOHS algorithm differs from the previous MOHS algorithms [19]-[22] in the following three aspects: First, a new type of adaptive operator is designed to dynamically adjust new pitch (note) rather than using BW during the entire search process; second, a dynamic parameter adjustment scheme is presented, which can dynamically update parameter HMCR to maintain a balance for exploration and exploitation in search solutions; third, a new harmony vector is not from the possible range of value but from genetic operation with (1-HMCR) probability, which can enhance convergence speed toward PF. The details of the proposed algorithm are given below.

3.1.1 Novel Self-adaptive operator

The parameter BW reveals an arbitrary distance bandwidth. It affects performance of the algorithm since BW is problem-dependent. More precisely, BW with large value can help the algorithm search solutions in a large step and improve the diversity of solutions, while with the BW value is in favor of fine-tuning the solution found in a small step [15]. Thus, there exists an inherent conflict between exploration and exploitation for a fixed BW value. Obviously, a fixed BW value is not suitable for MOPs during the entire search process. Thus, some scholars introduce some self-adaptive strategies into BW in literatures [13], [28], [29], but these strategies are still difficult to set the interval range of the BW parameter for solving MOPs. Therefore, in this work, a new self-adaptive operator is proposed. We integrate this self-adaptive operator into this proposed algorithm, which removes the influence of BW on performance of algorithm. Furthermore, this self-adaptive mechanism can improve ability of global optimization in the earlier search stage and enhance the ability of local optimization using fine-tuning operation in the later search process. Its mathematical expression is given as below:

$$x_{new}^{t'}(j) = x_{new}^{t}(j) + \delta_{new}^{t}(j) \tag{2}$$

Using random variation

$$\delta_{new}^{t}(j) = \begin{cases} \dfrac{x_{new}(j,\max) - x_{new}^{t}(j)}{T} \times (T-t) \times rand & random \le q_{new}^{t}(j) \\ \dfrac{x_{new}(j,\min) - x_{new}^{t}(j)}{T} \times (T-t) \times rand & otherwise \end{cases} \tag{3}$$

And the probability

$$q_{new}^{t}(j) = \frac{x_{new}^{t}(j) - x_{new}(j,\min)}{x_{new}(j,\max) - x_{new}(j,\min)} \tag{4}$$

where $x_{new}^{t}(j)$ donates the *j*-th pitch of a new harmony in current function evaluation number before self-adaptive operation and $x_{new}^{t'}(j)$ is the *j*-th pitch after self-adaptive operation, $x_{new}^{t}(j,\min)$ and $x_{new}^{t}(j,\max)$ respectively denote the minimum and maximum limits of the *j*-th pitch of the new harmony in current function evaluation and $x_{new}^{t}(j) \in [x_{new}^{t}(j,\min), x_{new}^{t}(j,\max)]$; *rand* is random distributed number in the interval [0, 1], T is the total number of function evaluations, t is current function evaluation number. This strategy provides global search with severely adjusting new pitch in initial phase of evolution process and it enhances local optimization by slight adjusting new pitch in latter stage of search process. Finally, it can improve solution diversity. Figure 1. presents distribution (symbol ‘*’ means new pitch after self-adaptive operation) of new pitch by this proposed self-adaptive operator when T=25000, $x_{new}^{t}(j) = 0.5$ (denoted by blue line) and in the interval [0, 1].

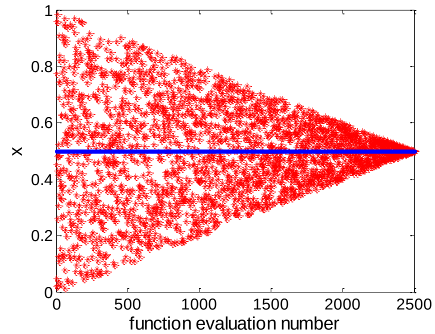


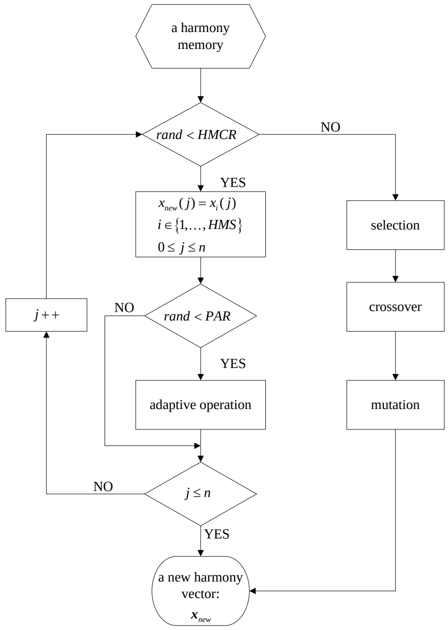
Figure 1. Distribution of new pitch after adaptive operation during the whole evaluations

3.1.2 Hybrid Mechanism and Parameter HMCR

MOHS is good at global optimization but the quality of the solutions obtained is not very high partially due to random selection with the probability 1-HMCR, which directly threatens convergence performance of the algorithm toward PF*. While GA can maintain good convergence by selecting better solutions on purpose, but it falls into premature convergence easily. Therefore, based on the merits of the GA and HS, we introduce genetic operator into HS to improve effectively convergence performance of the proposed HMOHS algorithm. This hybrid mechanism can yield a desirable candidate solution with the two cooperative strategies (i.e. GA and HS) according to the stage of evolution process. The flow chart of a new harmony, generated by this hybrid mechanism, is presented in Figure 2. We can observe from the flow chart that the parameter HMCR controls genetic operation and harmony search operation in HMOHS. Thus, it is very important to choose an appropriate value of HMCR for performance of the algorithm. R. Diao and Q. Shen[30] believe that adjustment of HMCR may contribute to better performance of algorithm for single objective problems. Here, HMCR is dynamically changed as follows:

$$\mathrm{HMCR}(t)=\mathrm{HMCR}_{\min}+\mathrm{HMCR}_{\max}\text{-}\left[\mathrm{HMCR}_{\min}+0.5\times\left(1+\sin\left(\frac{t\pi}{T}-\frac{\pi}{2}\right)\right)\times\left(\mathrm{HMCR}_{\max}-\mathrm{HMCR}_{\min}\right)\right] \tag{5}$$

Where $\mathrm{HMCR}(t)$ is the harmony memory considering rate in evaluation number t, $\mathrm{HMCR}_{\min}$ and $\mathrm{HMCR}_{\max}$ are the minimum and maximum considerate rate, respectively, $\mathrm{HMCR}_{\min}$ and $\mathrm{HMCR}_{\max}$ are set to 0.6 and 0.9 respectively, as our preliminary experiments indicate that HMOHS with this range performs better on most benchmark functions. For instance, when T is set to 25000, the variable curve of HMCR can be seen in Figure **3**. We can observe from the figure that if HMCR tends to be large value, it contributes to enhance harmony search operation and deter genetic operation in search process, and vice versa. In initial stage of evolution process, HMCR with large value can conduct HS operation to enhance global search, and HMCR with small value can execute GA operation to improve optimization quality of solutions in the later phase.



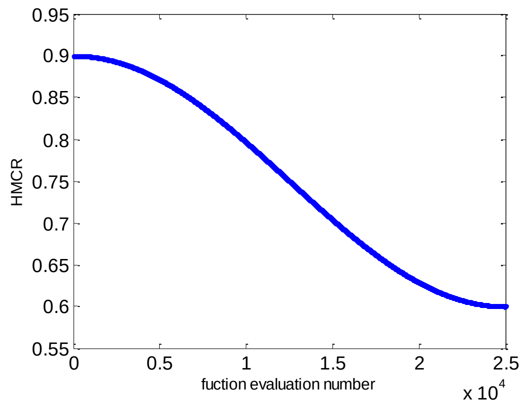


Figure 2. Generating a new vector **Figure 3. Variation of HMCR during the whole evaluations**

3.1.3 The Framework of HMOHS

The proposed HMOHS algorithm consists of 3 steps summarized as follows:

Input:

- MOP (1);
- A stopping condition;
- HMS: the size of harmony memory;
- HMCRmax, HMCRmin: the range of HMCR;
- PARmax, PARmin: the range of PAR;

Output:

Step 1) Initialization*:*

Step 1.1) Generate an initial harmony memory in the feasible range.

Step 1.2) Carry out a fast and non-dominated sorting operation on initial harmony memory, and compute current PS and PF of initial harmony memory.

Step 2) Stopping condition:

If stopping condition is satisfied, then stop and output PS and PF. Otherwise, go to step 3 as below.

Step 3) Update:

Create a random number equally distributed in the interval 0 and 1, denoted by R.

Step 3.1) Improvisation If R is less than HMCR, a pitch of new harmony will stem from existing harmony memory. And then create a random number $r \in [0,1]$, if r is less than PAR obtained by (6), apply self-adaptive operator on this new harmony pitch.

Step 3.2) Reproduction If R is more than HMCR, select two parents from harmony memory by binary tournament method; then apply genetic operators on two parents to generate a new harmony.

Step 3.3) Combination Incorporate a new harmony generated into the existing harmony memory to form (1+HMS) solutions. All the solutions in the harmony memory (HM) are sorted by the non-dominated sorting strategy which improves simultaneously the convergence and diversity in solutions.

Step 3.4) Elitist strategy: Each melody is associated with a rank equal to its non-dominance level (e.g. 1 for the one best level, 2 for the second one). Then within each level or rank a crowding distance, which indicates the sum of distances to the closest individual along each objective, is used to define an ordering among individuals. To achieve wide spread of the obtained Pareto fronts, melody with large crowding distance are preferred to melody with small crowding distance. Choose best solutions equal to HMS from combinational harmony memory to form new HM for the next loop.

Step 3.5) Update of PS and PF: The harmony solutions in the first level are considered as the PS found so far, and compute its corresponding Pareto front in objective space. Namely, the final internal population is returned as an approximation to the PF.

In the improvisation of the loop in step 3, a new pitch comes from existing harmony memory belonging to the first rank or level, which hopefully favors a new solution approximate to PS. The setting of parameter PAR should be dynamically variation, since the dynamic parameter PAR in some range contributes to keep a balance for diversification and intensification. To improve the performance of the proposed algorithm, PAR is changed during each generation as follows:

$$\mathrm{PAR}(t) = \mathrm{PAR}_{\min} + \frac{\mathrm{t}}{\mathrm{T}} \times \left(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min}\right) \tag{6}$$

where $\mathrm{PAR}(t)$ is the pitch adjusting rate of current evaluation number, $\mathrm{PAR}_{\min}$ and $\mathrm{PAR}_{\max}$ is the minimum and maximum pitch adjusting rate, respectively; t is the current evaluation number and T is the total number of the function evaluation.

In the combination of the loop in step 3, a new harmony generated is incorporated into the evolutionary loop. Notice that the size of HM is equal to HMS+1 rather than 2*HMS after combination. Though the time required by the strategy increases, it is worthwhile since it improves diversity of the solutions (see in section 4.2). To show difference between two strategies in HMOHS, HMOHS with 1+HMS combination strategy is called HMOHS and HMOHS with 2*HMS is denoted as HMOHSG.

4. EXPERIMENTS AND ANALYSIS

This section assesses the performance of HMOHS. Some benchmarks functions used in the paper are stated. Then the performance metrics measured the quality of obtained non-dominated by algorithms are described. Comparison with variants of MOHS is proved to improve effectively search ability by the hybrid with GA mechanism and tuning fine parameters. The chapter ends with a comparative analysis between our proposal and three state-of-art algorithms (NSGAII, SPEA2, and MOEA/D).

4.1 Benchmark functions

21 classical multi-objective benchmark test instances are selected in the experiment to evaluate the behavior of HMOHS. Test instances can be divided into three types: ZDT test suites [31], WFG test suites [32] and DTLZ test functions [33]. ZDT test suites are all bi-objective benchmarks, while DTLZ benchmarks are all three-objective test problems. The WFG suites allow users to define the benchmarks of MOPs with different properties. Nine problems' bi-objective version are adopted in this experiment.

4.2 Comparison HMOHS with variations of MOHS

The performance metrics can be grouped into three categories depending on whether they assess the convergence of the obtained Pareto front, the diversity in the obtained solutions, or combined performance measures [34]. Three metrics are used as follows: Epsilon (ε) [35], Spread (Δ) [4], and Hypervolume (HV) [36].

In this section, the performance of the proposed HMOHS algorithm is investigated by comparing with HMOHSG, MOGHS [3] and MOHS [20]. The simulated binary crossover (SBX) and polynomial mutation are used in HMOHS and HMOHSG and the distribution indexes in both SBX and the polynomial mutation are set to 20. The crossover rate is 0.9, and mutation rate is 1.0/L, where L denotes the number of decision variables. To have a fair comparison between these algorithms, the other parameter settings of the MOEAs should be set the same. Table I. describes the parameters used in these algorithms, where symbol 'CD' indicates crowding distance and 'DCD' represents dynamical crowding distance [39]. In MOGHS and MOHS, $\mathrm{BW} = \mathrm{BW}_{\max} \times \exp\left(\frac{\ln(\mathrm{BW}_{\min}/\mathrm{BW}_{\max})}{\mathrm{T}} \cdot \mathrm{t}\right)$, $\mathrm{BW}_{\min}$=0.2 and $\mathrm{BW}_{\max}$=2. Symbol'----' indicates no parameter setting.

Table I. Parameters setting of HMOHS, HMOHSG, MOGHS and MOHS Algorithms

	HMOHS	HMOHSG	MOGHS	MOHS
Stopping Condition	15,000 function evaluations			
Harmony Memory Size	100			
HMCR	Eq. (5)	Eq. (5)	0.85	0.85
PAR	Eq. (6)	Eq.(6)	Eq.(6)	Eq.(6)
BW		----	----	
Genetic operator				-----
Replacement	replace if better ranking and crowding distance			
Density estimator	CD	CD	CD	DCD
Combination strategy	HMS+1	2*HMS	2*HMS	2*HMS

In our experiment, the same number of function evaluations or NI has been seen as stopping criterion. These algorithms have been run 30 times independently for each benchmark problem. Table II-IV show the comparative results of these algorithms in terms of mean and standard deviation values (i.e. Spread, Epsilon and HV) obtained by 30 independent runs for each test benchmark. The values in bold represent the best results in these tables. Note: these metric data in tables are not absolute values but relative values.

Table II. shows the mean and standard deviation of the Spread-metric values. We can observe from the table that HMOHS outperforms the other three variant algorithms in terms of Spread-metrics on most test suite. Accurately, HMOHS gets the best values in 16 of 21 test suite, while HMOHSG obtain the best ones in 5 benchmark problems. Notice that the other two variants (i.e. MOGHS and MOHS) do not get best values in any test problems, which indicates the multi-objective harmony search algorithm with new adaptive operator has better diversity than its counterparts without this mechanism on most test suite.

Table III. presents the mean and standard deviation of the Epsilon-metric values. It is clear that HMOHS performs better than the other algorithms for most test suite. In other words, HMOHS is better than its counterparts in terms of convergence on all the benchmarks except WFG2, WFG8, DTLZ1, DTLZ3 and DTLZ6. It also should be noted that HMOHSG does not obtain the best value in any test problem, which suggests that different combination strategies in HMOHS and HMOHSG can have different performances in terms of convergence.

Table IV. reveals the mean and standard deviation of the HV-metric values. For HV-metric, HMOHS outperforms the compared algorithms since HMOHS can offer the best results (largest here) in 17 of 21 problems, whereas HMOHSG, MOGHS and MOHS provide the best ones in 0, 1 and 2 test functions, respectively. Regarding coverage, HMOHS is in the first place; MOEA/D comes in the second place; SPEA2 is the third; NSGA-II is rear.

Overall, HMOHS can generate better approximations than the other variants of multi-objective harmony search algorithms on most of the test problems in terms of diversity, convergence and coverage.

Table II. Mean and standard deviation value of spread metric on HMOHS, HMOHSG, MOGHS and MOHS

	HMOHS mean(std)	HMOHSG mean(std)	MOGHS mean(std)	MOHS mean(std)
ZDT1	**1.44e-01(1.4e-02)**	8.52e–01(5.0e−02)	4.76e-01(5.0e-02)	5.20e-01(6.9e-02)
ZDT2	**1.52e-01(1.4e-02)**	9.36e–01(3.6e−02)	5.36e-01(6.3e-02)	5.41e-01(8.6e-02)
ZDT3	**7.15e-01(7.7e-03)**	8.17e–01(4.8e−02)	7.23e-01(5.1e-02)	7.24e-01(4.1e-02)
ZDT4	**6.82e-01(1.4e-02)**	9.30e–01(4.8e−02)	7.37e-01(1.2e-01)	7.04e-01(1.0e-01)
ZDT6	**1.18e-01(1.1e-02)**	9.42e–01(2.1e−02)	9.47e-01(6.1e-02)	8.25e-01(1.9e-01)
WFG1	**5.56e-01(3.7e-02)**	1.08 (1.2e−01)	1.14(2.9e-02)	1.20 (2.9e-02)
WFG2	7.66e-01(7.2e-03)	**7.43e–01(1.0e−01)**	8.28e-01(5.3e-02)	8.59e-01(5.6e-02)
WFG3	**6.62e-02(1.0e-02)**	6.01e–01(8.1e−02)	3.22e-01(3.7e-02)	3.17e-01(3.0e-02)
WFG4	**1.32e-01(1.9e-02)**	6.32e–01(4.3e−02)	3.53e-01(3.0e-02)	5.46e-01(2.7e-02)
WFG5	**1.38e-01(1.5e-02)**	6.44e–01(7.3e−02)	3.88e-01(2.6e-02)	5.32e-01(2.4e-02)
WFG6	**1.78e-01(3.8e 02)**	6.37e–01(7.6e−02)	3.88e-01(3.3e-02)	5.63e-01(2.8e-02)
WFG7	**1.30e-01(1.4e-02)**	6.43e–01(7.0e−02)	3.56e-01(3.3e-02)	5.45e-01(2.6e-02)
WFG8	**6.30e-01(8.0e-02)**	6.84e–01(5.3e−02)	7.33e-01(5.3e-02)	7.37e-01(4.8e-02)
WFG9	**1.35e-01(1.8e-02)**	5.97e–01(7.8e−02)	3.36e-01(3.1e-02)	5.27e-01(3.0e-02)
DTLZ1	7.58e-01(4.2e-02)	**6.78e-01(6.9e-02)**	7.47e-01(4.8e-02)	9.10e-01(2.7e-01)
DTLZ2	6.58e-01(4.4e-02)	**5.94e-01(5.4e-02)**	6.19e-01(3.6e-02)	6.26e-01(4.7e-02)
DTLZ3	7.81e-01(1.3e-01)	**6.40e-01(6.0e-02)**	6.78e-01(6.1e-02)	6.50e-01(1.3e-01)
DTLZ4	**6.40e-01(1.3e 01)**	1.15 (1.1e-01)	6.81e-01(6.0e-02)	7.33e-01(6.7e-02)
DTLZ5	**1.78e-01(3.1e-02)**	6.26e-01(7.3e-02)	5.94e-01(3.3e-02)	6.15e-01(3.4e-02)
DTLZ6	**1.36e-01(1.3e-02)**	6.52e-01(3.8e-02)	6.73e-01(3.5e-02)	7.11e-01(3.0e-02)
DTLZ7	6.85e-01(4.0e-02)	**6.56e-01(6.9e-02)**	7.43e-01(3.7e-02)	8.36e-01(4.9e-02)
hit rate	16/21	5/21	0/21	0/21

Table III. Mean and standard deviation value of epsilon metric on HMOHS, HMOHSG, MOGHS and MOHS

	HMOHS mean(std)	HMOHSG mean(std)	MOGHS mean(std)	MOHS mean(std)
ZDT1	**8.74e-03(7.9e-04)**	2.19 (2.0e-01)	7.12e-02(7.0e-03)	7.35e-02(6.7e-03)
ZDT2	**8.90e-03(8.4e-04)**	3.84 (1.9e-01)	1.26e-01(2.6e-02)	1.35e-01(4.3e-02)
ZDT3	**1.86e-02(5.6e-02)**	2.48 (1.8e-01)	1.15e-01(2.3e-02)	1.04e-01(2.4e-02)
ZDT4	**1.13e-01(1.1e-01)**	6.88e+01(9.4e+00)	2.76e-01(1.5e-01)	2.79e-01(1.6e-01)
ZDT6	**3.18e-02(5.2e-03)**	6.95 (2.1e-01)	3.21e-02(9.6e-03)	3.47e-02(1.2e-02)
WFG1	**6.86e-01(2.7e-01)**	2.19 (3.8e-01)	1.07 (3.1e-02)	1.06 (4.1e-02)
WFG2	4.57e-01(3.9e-01)	1.12 (2.0e-01)	2.36e-01(3.3e-01)	**1.72e-01(2.8e-01)**
WFG3	**1.87e-02(2.3e-03)**	5.67e-01(9.3e-02)	4.15e-02(1.0e-02)	3.98e-02(8.3e-03)
WFG4	**1.66e-02(2.0e-03)**	9.79e-01(2.9e-01)	3.41e-02(5.5e-03)	5.08e-02(9.2e-03)
WFG5	**6.60e-02(3.6e-02)**	5.96e-01(1.1e-01)	7.31e-02(4.7e-03)	7.04e-02(6.3e-03)
WFG6	**2.98e-02(1.5e-02)**	6.62e-01(1.1e-01)	1.03e-01(2.7e-02)	1.25e-01(2.7e-02)
WFG7	**1.60e-02(1.7e-03)**	6.40e-01(1.9e-01)	3.69e-02(7.5e-03)	5.17e-02(1.6e-02)
WFG8	4.24e-01(1.2e-01)	1.08 (1.5e-01)	3.60e-01(5.0e-02)	**3.35e-01(5.4e-02)**
WFG9	**2.01e-02(3.5e-03)**	5.53e-01(1.2e-01)	7.19e-02(6.3e-03)	7.01e-02(5.8e-03)
DTLZ1	4.36e-01(2.5e-01)	3.39e+01(1.0e+01)	2.35e-01(1.2e-01)	**2.21e-01(1.1e-01)**
DTLZ2	**1.22e-01(1.3e-02)**	4.63e-01(3.2e-02)	1.24e-01(1.4e-02)	1.44e-01(3.0e-02)
DTLZ3	1.76e+01(7.3e+00)	3.46e+02(5.9e+01)	**4.61 (1.1)**	4.71 (1.1)
DTLZ4	**1.20e-01(1.2e-02)**	1.04 (1.8e-01)	1.44e-01(2.5e-02)	1.75e-01(4.7e-02)
DTLZ5	**4.95e-03(5.6e-04)**	3.58e-01(3.9e-02)	1.24e-02(2.0e-03)	1.53e-02(2.2e-03)
DTLZ6	1.39 (8.7e-02)	5.54 (2.5e-01)	**1.00e-02(1.4e-03)**	1.41e-02(2.5e-03)
DTLZ7	**1.42e-01(4.5e-02)**	9.76 (9.4e-01)	1.88e-01(6.0e-02)	2.59e-01(7.9e-02)
hit rate	16/21	0/21	2/21	3/21

TABLE IV. Mean and standard deviation value of HV metric on HMOHS, HMOHSG, MOGHS and MOHS

	HMOHS mean(std)	HMOHSG mean(std)	MOGHS mean(std)	MOHS mean(std)
ZDT1	**6.61e-01 (1.0e-4)**	0	5.93e-01(8.8e-03)	5.92e-01(7.0e-03)
ZDT2	**3.28e-01(1.9e-04)**	0	2.38e-01(1.3e-02)	2.38e-01(1.2e-02)
ZDT3	**5.13e-01(7.5e-04)**	0	4.67e-01(7.4e-03)	4.68e-01(8.4e-03)
ZDT4	**6.07e-01(6.5e-02)**	0	3.62e- 01(1.3e-01)	3.53e-01(1.4e-01)
ZDT6	**4.01e-01(3.9e-03)**	0	3.93e-01(2.2e-03)	3.93e-01(3.1e-03)
WFG1	**4.01e-01(9.2e-02)**	2.90e-03(5.1e-03)	1.73e-01(4.3e-03)	1.74e-01(5.5e-03)
WFG2	**5.54e-01(2.6e-03)**	3.59e-01(2.2e-02)	5.53e-01(2.3e-03)	5.51e-01(2.9e-03)
WFG3	**4.93e-01(8.1e-04)**	3.20e-01(1.5e-02)	4.91e-01(1.7e-03)	4.90e-01(1.9e-03)
WFG4	**2.10e-01(2.2e-04)**	1.32e-01(5.4e-03)	2.09e-01(4.2e-04)	2.08e-01(5.6e-04)
WFG5	2.12e-01(4.5e-05)	1.11e-01(6.6e-03)	2.14e-01(1.6e-03)	**2.16e-01(4.2e-03)**
WFG6	**2.01e-01(9.8e-03)**	7.40e-02(1.1e-02)	1.67e-01(1.5e- 02)	1.65e-01(1.3e-02)
WFG7	**2.10e-01(1.4e-04)**	1.08e-01(1.0e-02)	2.09e-01(3.8e -04)	2.08e-01(6.4e-04)
WFG8	**2.97-01(1.3e-02)**	1.17e-01(8.3e-03)	2.84e-01(2.0e -03)	2.84e-01(1.4e-03)
WFG9	**2.35e-01(1.6e-03)**	1.26e-01(2.0e-02)	2.27e-01(1.8e -03)	2.26e-01(1.8e-03)
DTLZ1	1.81e-01(2.7e-01)	0	2.47e-01(2.4e-01)	**2.92e-01(2.2e-01)**
DTLZ2	**4.11e-01(5.1e-03)**	2.80e-02(1.9e-02)	3.74e-01(5.0e-03)	3.76e-01(1.0e-02)
DTLZ3	0	0	0	0
DTLZ4	**4.16e-01(3.7e-03)**	0	3.84e-01(8.9e-03)	3.73e-01(9.6e-03)
DTLZ5	**9.34e-02(4.2e-05)**	1.23e-03(2.3e-03)	9.03e-02(6.6e-04)	8.79e-02(1.1e-03)
DTLZ6	0	0	**9.22e-02(1.3e-04)**	9.00e-02(4.6e-04)
DTLZ7	**2.99e-01(3.9e-03)**	0	2.91e-01(8.5e-03)	2.90e-01(7.4e-03)
hit rate	17/21	0/21	1/21	2/21

4.3 Comparison HMOHS with NSGA-II, SPEA2 and MOEA/D

To further evaluate the performance of HMOHS, it is compared with the current state-of-the-art algorithms such as NSGA-II [4], SPEA2 [5] and MOEA/D [6]. The characteristic feature of NSGA-II developed by Deb *et al.* is that it uses a fast and non-dominated sorting and crowding distance estimation procedure. SPEA2 was presented by Zitler *et al.* In SPEA2 algorithm each individual has a fitness value which is the sum of its strength raw fitness and density estimation based on the distance to the k-th nearest neighbor. MOEA/D and its variants [40]-[44] invented by Q. Zhang *et al.* decomposes a multi-objective problem into a number of scalar sub-problems and optimizes them simultaneously, and here we use MOEA/D [40] with differential evolution (DE [45]) strategy to compare with our proposal. To evaluate behavior of these algorithms with a fair comparison, the parameters of the algorithms should be set the same as possible as we can.

In our experimental, the stopping criteria is 15,000 function evaluations for HMOHS, NSGA-II, SPEA2 and MOEA/D on ZDT and WFG test suite, while the stopping criteria is increased to 25,000 function evaluations for DTLZ benchmarks since it is difficult to obtain exact PS for DTLZ test problems. The other parameter settings are shown in Table V. These algorithms are executed 30 runs for each test problem. Table VI-VIII present the comparative results of these algorithms in terms of mean and standard deviation values (i.e. Spread, Epsilon and HV) obtained by 30 independent runs. The values in bold represent the best results in Table VI-VIII.

Table V. Parameters setting of HMOHS, NSGA-II, SPEA2 and MOEA/D algorithms

Parameterization used in NSGA-II	
Population Size	100 for ZDT and WFG suites and 500 for DTLZ suites
Selection Scheme	binary tournament selection
Recombination	simulated binary, P_c =0.9, crossover distribution ratio =20
Mutation	polynomial, P_m = 1.0/L, mutation distribution ratio =20
Evaluate solution	Pareto based (i.e. rank and crowding distance)
Parameterization used in SPEA2	
Population Size	100 for ZDT and WFG suites and 500 for DTLZ suites
Selection Scheme	binary tournament selection
Recombination	simulated binary, P_c=0.9, crossover distribution ratio =20
Mutation	polynomial, P_m =1.0/L, mutation distribution ratio =20
Evaluate solution	Pareto based
Parameterization used in MOEA/D	
Population Size	100 for ZDT and WFG suites and 500 for DTLZ suites
Selection scheme	binary tournament selection
Recombination	Differential Evolution, CR=0.1, F=0.5
Mutation	polynomial, P_m = 1.0/L, distribution ratio =20
Evaluate solution	Tchebycheff approach
Parameterization used in HMOHS	
Harmony memory size	100 for ZDT and WFG suites and 500 for DTLZ suites
HMCR	$HMCR_{max}$=0.9, $HMCR_{min}$=0.6
PAR	PAR_{min}=0.1, PAR_{max}=0.5
Selection Scheme	binary tournament selection
Recombination	simulated binary, P_c =0.9, crossover distribution ratio =20
Mutation	polynomial, P_m = 1.0/L, mutation distribution ratio =20
Evaluate solution	Pareto based

where L represents the number of decision variables

Table VI presents the mean and standard deviation of the Spread-metric values in 30 runs. It is obvious from this table that HMOHS is much better than other three algorithms in terms of Spread-metrics on most test suite. Because HMOHS gets the best values in 17 of 21 test suite, while NSGA-II, SPEA2 and MOEA/D respectively obtain the best ones in 2, 2 and 3 benchmark problems. Furthermore, HMOHS outperforms the other compared algorithms on all bi-objective benchmarks, namely ZDT and DTLZ test suite. It indicates the front obtained by HMOHS algorithm has a more uniform distribution along the PF* than that by its counterparts on most test suite considered, especially for bi-objective problems.

Table VII reveals that concerning Epsilon-metric values, HMOHS performs better than the other algorithms for most test suite. It suggests that HMOHS is better than that of its counterparts in terms of convergence on these benchmarks except ZDT3, WFG2, DTLZ1-4, and DTLZ6. With regard to convergence performance of these algorithms, we can state that HMOHS ranks the top; MOEA/D takes second place, followed by SPEA2 and NSGA-II.

The HV-metric values, which are used to strengthen the results of the two metrics above, are listed in Table VIII. For HV-metric, HMOHS outperforms the compared algorithms since HMOHS can offer the best results (largest here) in 14 of 21 problems, whereas NSGA-II, SPEA2 and MOEA/D provide the best ones in 1, 2 and 4 test functions, respectively. Regarding coverage, HMOHS is in the first rank; MOEA/D comes in the second place; SPEA2 is the third; NSGA-II is rear.

It is also observed that differences are very small on some benchmarks with regard to HV-metric. However, these tiny differences generate discernible differences in the Pareto fronts [38]. To graphically show our results, we plot in Figure 4 four fronts obtained by HMOHS, NSGA-II, SPEA2 and MOEA/D for problem ZDT6. The fronts are those having the best coverage (the largest value of HV) in 30 runs for that problem. We can observe that the front generated by HMOHS achieves an almost uniform distribution along PF* and perfect convergence toward PF*. It is clearly noticed that SPEA2 has the worst spread and convergence on this problem, and NSGA-II obtains the second worst spread and convergence among these algorithms. Regarding MOEA/D, its front has good convergence to PF, but it has not good diversity when $f_1(\mathbf{x})$ is about equal to 0.6 where a gap exists.

Table VI Mean and standard deviation value of spread metric on HMOHS, NSGA-II, SPEA2 and MOEA/D

	HMOHS mean(std)	NSGA-II mean(std)	SPEA2 mean(std)	MOEA/D mean(std)
ZDT1	**1.53e-01(1.3e-02)**	3.87e-01(2.8e-02)	4.12e-01(7.4e-02)	7.16e-01(7.8e-02)
ZDT2	**1.43e-01(1.5e-02)**	3.75e-01(2.3e-02)	8.28e-01(1.5e-01)	9.42e-01(1.5e-01)
ZDT3	**7.50e-01(7.1e-03)**	7.77e-01(9.7e-03)	7.79e-01(2.9e-02)	1.01(4.1e-02)
ZDT4	**4.44e-01(1.6e-01)**	6.63e-01(2.0e-01)	9.38e-01(3.7e-02)	1.15(1.1e-01)
ZDT6	**2.52e-01(2.6e-02)**	5.00e-01(7.4e-02)	8.21e-01(7.8e-02)	3.13e-01(2.6e-01)
WFG1	**5.94e-01(1.1e-01)**	7.18e-01(6.8e-02)	8.99e-01(8.1e-02)	1.10(1.5e-01)
WFG2	**7.59e-01(4.8e-03)**	7.94e-01(1.2e-02)	7.82e-01(1.1e-02)	1.11(7.0e-03)
WFG3	**7.08e-02(1.2e-02)**	3.65e-01(2.6e-02)	1.68e-01(1.5e-02)	3.47e-01(1.8e-03)
WFG4	**1.29e-01(1.4e-02)**	3.63e-01(2.6e-02)	2.66e-01(1.5e-02)	5.79e-01(5.5e-02)
WFG5	**1.31e-01(1.1e-02)**	3.94e-01(3.4e-02)	2.69e-01(1.7e-02)	4.60e-01(2.0e-02)
WFG6	**1.45e-01(2.2e-02)**	3.87e-01(2.7e-02)	2.67e-01(2.8e-02)	4.20e-01(9.2e-03)
WFG7	**1.15e-01(1.2e-02)**	3.81e-01(3.0e-02)	2.53e-01(2.3e-02)	4.27e-01(1.0e-02)
WFG8	**5.80e-01(4.1e-02)**	6.67e-01(7.0e-02)	6.94e-01(6.4e-02)	6.59e-01(5.1e-02)
WFG9	**1.27e-01(1.8e-02)**	3.56e-01(2.5e-02)	2.54e-01(2.0e-02)	4.52e-01(1.5e-02)
DTLZ1	**1.12(1.1e-01)**	1.20(5.0e-02)	1.24(5.8e-02)	1.24(7.0e-02)
DTLZ2	6.79e-01(2.4e-02)	6.73e-01(2.9e-02)	**5.75e-01(1.1e-02)**	8.37e-01(1.9e-02)
DTLZ3	1.02 (3.1e-02)	1.01 (3.2e-02)	**9.84e-01(5.3e-02)**	1.28(1.1e-01)
DTLZ4	6.60e-01(1.5e-02)	6.61e-01(2.6e-02)	**5.82e-01(1.6e-02)**	1.03(7.9e-02)
DTLZ5	**3.96e-01(4.5e-02)**	6.21e-01(3.4e-02)	6.21e-01(5.6e-02)	9.01e-01(4.2e-02)
DTLZ6	8.32e-01(2.3e-02)	**8.24e-01(2.9e-02)**	8.66e-01(2.3e-02)	8.52e-01(1.6e-01)
DTLZ7	**7.85e-01(1.3e-02)**	7.87e-01(2.2e-02)	8.32e-01(2.4e-02)	1.11(4.6e-02)
hit rate	17/21	1/21	3/21	0/21

TABLE VII. Mean and standard deviation value of epsilon metric on HMOHS, NSGA-II, SPEA2 and MOEA/D

	HMOHS mean(std)	NSGA-II mean(std)	SPEA2 mean(std)	MOEA/D mean(std)
ZDT1	**8.82e-03(9.8e-04)**	1.63e-02(1.8e-03)	4.47e-02(1.1e-02)	1.76e-01(6.8e-02)
ZDT2	**9.21e-03(1.3e-03)**	1.68e-02(1.9e-03)	5.16e-01(4.0e-01)	5.44e-01(2.8e-01)
ZDT3	2.86e-02(7.7e-02)	**1.11e-02(1.9e-03)**	1.15e-01(1.0e-01)	4.63e-01(1.2e-01)
ZDT4	**1.25e-01(1.0e-01)**	1.84e-01(1.2e-01)	1.46e-01(4.5e-01)	1.84e-01(7.6e-01)
ZDT6	**3.49e-02(6.9e-03)**	9.79e-02(1.8e-02)	6.95e-01(9.5e-02)	2.27e-02(6.5e-02)
WFG1	**7.13e-01(3.6e-01)**	7.88e-01(2.4e-01)	1.34e-01(1.7e-01)	8.92e-01(1.3e-01)
WFG2	6.13e-01(3.3e-01)	3.80e-01(3.9e-01)	4.60e-01(3.9e-01)	**3.31e-02(5.1e-03)**
WFG3	**1.95e-02(2.8e-03)**	3.84e-02(5.3e-03)	2.78e-02(4.3e-03)	3.13e-02(3.5e-03)
WFG4	**1.62e-02(9.2e-04)**	3.57e-02(5.0e-03)	3.12e-02(8.3e-03)	8.39e-02(1.5e-02)
WFG5	**6.48e-02(2.2e-02)**	6.79e-02(1.6e-02)	7.78e-02(1.4e-02)	8.53e-02(2.9e-03)
WFG6	**3.83e-02(2.0e-02)**	4.49e-02(1.9e-02)	4.61e-02(1.7e-02)	2.44e-02(1.2e-03)
WFG7	**1.65e-02(2.0e-03)**	3.81e-02(8.6e-03)	3.55e-02(1.5e-02)	2.67e-02(8.1e-04)
WFG8	**3.89e-01(1.4e-01)**	4.43e-01(1.1e-01)	5.48e-01(9.7e-02)	3.84e-01(1.6e-01)
WFG9	**1.88e-02(2.8e-03)**	3.66e-02(5.4e-03)	3.63e-02(1.3e-02)	3.38e-02(1.4e-03)
DTLZ1	2.31(6.8e-01)	2.50 (7.9e-01)	1.49 (2.5e-01)	**9.17e-01(7.2e-01)**
DTLZ2	6.98e-02(8.6e-03)	7.37e-02(8.0e-03)	**4.07e-02(3.0e-03)**	5.83e-02(3.5e-03)
DTLZ3	3.66e+01(6.7)	4.27e+01(7.0)	4.48e+01(1.0e+01)	**9.31 (1.0e+01)**
DTLZ4	5.67e-02(1.1e-02)	6.26e-02(6.4e-03)	**4.21e-02(4.4e-03)**	1.33e-01(6.2e-02)
DTLZ5	**2.24e-03(2.3e-04)**	4.05e-03(4.8e-04)	5.47e-03(3.0e-03)	9.01e-03(7.3e-04)
DTLZ6	2.32 (1.2e-01)	2.73 (8.9e-02)	2.88 (8.9e-02)	**6.92e-03(2.5e-03)**
DTLZ7	**6.83e-02(1.7e-02)**	2.53e-01(4.4e-02)	4.93e-01(3.0e-02)	2.75 (5.2e-01)
hit rate	14/21	1/21	2/21	4/21

TABLE VIII. Mean and standard deviation value of HV metric on HMOHS, NSGA-II, SPEA2 and MOEA/D

	HMOHS mean(std)	NSGA-II mean(std)	SPEA2 mean(std)	MOEA/D mean(std)
ZDT1	**6.83e-01(9.3e-04)**	6.79e-01(9.3e-04)	6.46e-01(8.1e-03)	5.19e-01(5.5e-02)
ZDT2	**3.25e-01(1.0e-03)**	3.20e-01(1.8e-03)	1.51e-01(1.0e-01)	1.26e-01(7.9e-02)
ZDT3	**5.47e-01(8.1e-04)**	5.44e-01(7.6e-04)	5.16e-01(7.9e-03)	2.93e-01(6.3e-02)
ZDT4	**6.14e-01(7.1e-02)**	5.80e-01(8.6e-02)	1.76e-03(9.5e-03)	7.23e-03(2.2e-02)
ZDT6	**4.83e-01(4.6e-03)**	3.34e-01(1.4e-02)	3.11e-02(1.9e-02)	4.05e-01(3.6e-02)
WFG1	3.62e-01(9.3e-02)	**4.03e-01(7.7e-02)**	1.72e-01(7.0e-02)	8.52e-02(5.9e-02)
WFG2	**5.54e-01(2.6e-04)**	5.54e-01(2.5e-03)	5.53e-01(2.2e-03)	5.54e-01(4.7e-04)
WFG3	**4.93e-01(6.6e-04)**	4.92e-01(8.0e-04)	4.92e-01(1.2e-03)	4.93e-01(3.3e-04)
WFG4	**2.10e-01(1.8e-04)**	2.09e-01(4.0e-04)	2.09e-01(4.5e-04)	1.99e-01(1.9e-03)
WFG5	**2.11e-01(5.0e-05)**	2.11e-01(1.5e-03)	2.11e-01(1.5e-03)	2.10e-01(2.7e-04)
WFG6	1.92e-01(1.4e-02)	2.01e-01(7.1e-03)	1.96e-01(1.1e-02)	**2.08e-01(1.9e-04)**
WFG7	**2.10e-01(2.1e-04)**	2.09e-01(2.4e-04)	2.09e-01(3.8e-04)	2.08e-01(1.3e-04)
WFG8	2.06e-01(3.3e-03)	2.12e-01(1.4e-02)	2.02e-01(4.0e-03)	**2.22e-01(2.1e-02)**
WFG9	**2.26e-01(5.3e-04)**	2.23e-01(1.4e-03)	2.23e-01(1.8e-03)	2.21e-01(5.3e-04)
DTLZ1	5.35e-02(1.2e-01)	4.88e-02(1.0e-01)	**2.46e-01(1.3e-01)**	2.34e-01(3.1e-01)
DTLZ2	4.28e-01(2.1e-03)	4.24e-01(1.7e-03)	**4.37e-01(9.6e-04)**	4.33e-01(8.0e-04)
DTLZ3	0	0	0	**1.53e-01(1.9e-01)**
DTLZ4	**4.16e-01(1.5e-03)**	4.11e-01(1.7e-03)	3.50e-01(1.3e-01)	3.93e-01(3.3e-02)
DTLZ5	**9.53e-02(4.8e-05)**	9.47e-02(1.1e-04)	9.29e-02(1.5e-03)	9.37e-02(9.5e-05)
DTLZ6	0	0	0	**9.48e-02(4.1e-05)**
DTLZ7	**4.27e-01(2.5e-03)**	3.82e-01(6.6e-03)	3.00e-01(2.6e-02)	4.25e-02(4.5e-02)
hit rate	14/21	1/21	2/21	4/21

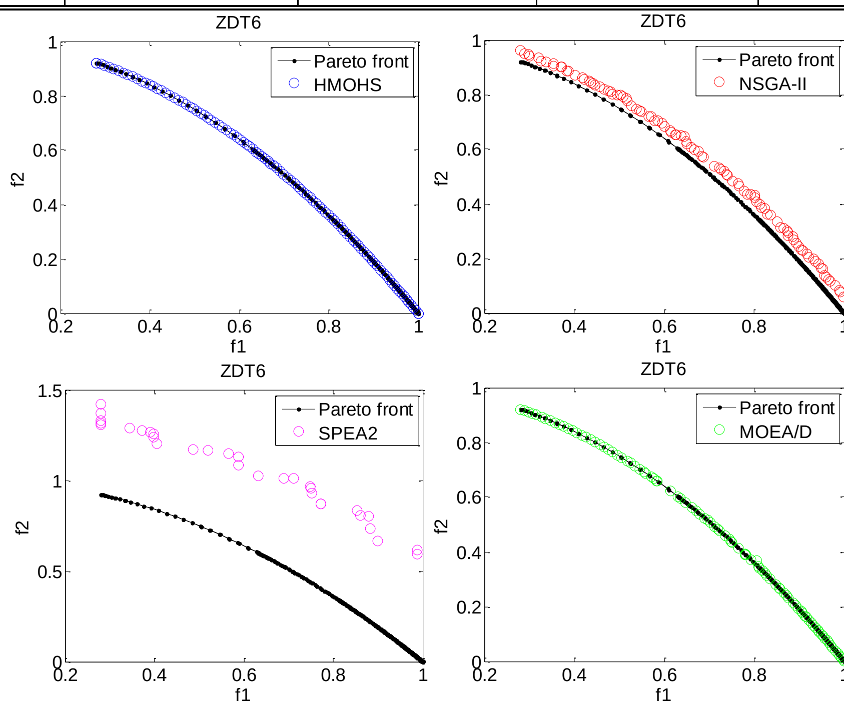


Figure 4. Approximated fronts of these algorithms when solving ZDT6

4.4 Analysis of convergence speed

In this section, we mainly focus on investigating convergence speed of HMOHS, NSGA-II, SPEA2 and MOEA/D when solving ZDT, WFG, and DTLZ test suites except DTLZ3, DTLZ5 and DTLZ6 since HV-metric values by algorithms on DTLZ3, DTLZ5 and DTLZ6 are close to zero value.

To make a fair comparison among different algorithms, we use the following parameters. The stopping condition is set to 25,000 function evaluation numbers for above algorithm on all the benchmarks, and the setting of the other parameters in these four algorithms is the same as in Table V. To evaluate quality of front obtained by each algorithm in our experiment, here we only employ HV-metric to measure the performance of each algorithm since this metric is a comprehensive metric of convergence and diversity in a sense. The above algorithms have been run 30 times independently on each benchmark problem and then HV-metric mean value of each function evaluation is calculated during the whole search. And we adopt the number of function evaluation to track trend of mean HV-metric in 30 runs. It should be noted that convergence speed represents speed of high quality solution obtained rather than only convergence such as Epsilon-metric.

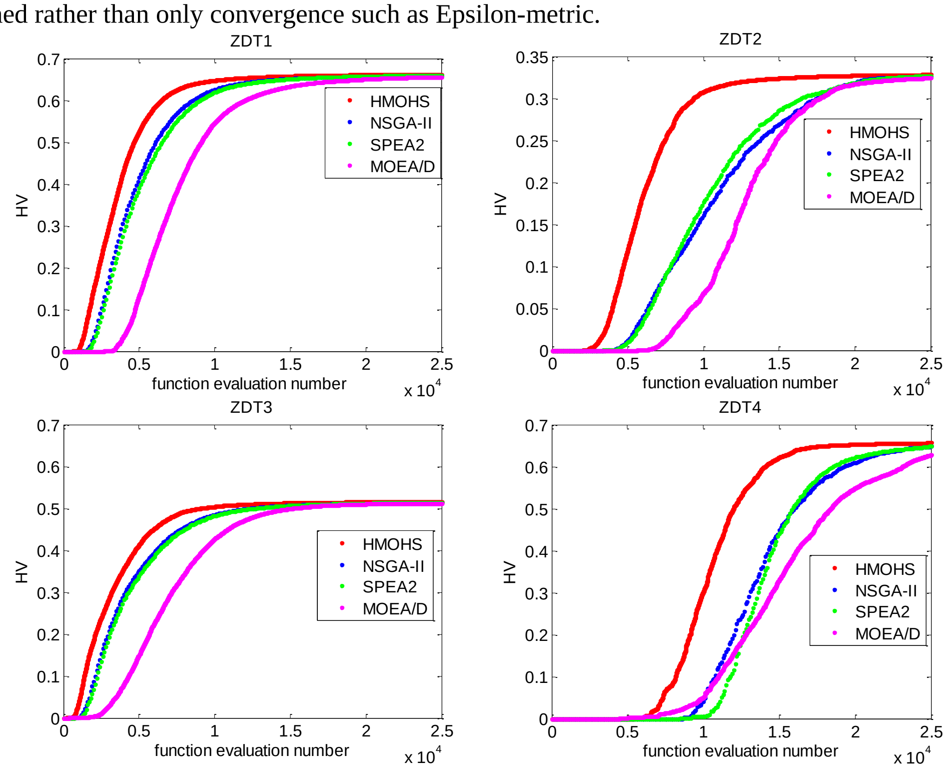


Figure 5. Convergence curve of these algorithms when solving benchmarks

Figure 5. presents the evolution of mean HV-metric value of the current population with the number of function evaluations in each algorithm for each test problem. These curves suggest that convergence speed of HMOHS, in terms of the number of the function evaluations, is faster than that of its counterparts in maximizing the HV-metric value for ZDT suite.

We can conclude from this observation that result of plots is consistent with our view that a hybrid MOHS by introducing GA operator effectively enhance convergence speed to PF*.

5 CONCLUSION

We have proposed HMOHS, a hybrid HS with GA algorithm to solve MOPs. In HMOHS, a new self-adaptive operator, which replaces role of BW, is used to HMOHS. This self-adaptive mechanism can make candidate solution automatically choose suitable value at an appropriate time during the entire evolution process. Moreover, each solution has two different features operation (i.e. GA and HS) to guide candidate solution to converge toward Pareto optimal front, explore new promising areas and exploit local areas found. In addition, dynamical HMCR is employed to control HS and GA operation in HMOHS for enhancing convergence speed to Pareto optimal front, and maintaining diversity in solutions.

HMOHS have been compared to its variants or previous version. Three metrics were used to assess the performance of the algorithms on 21 test functions. The obtained results suggest that HMOHS clearly outperforms its counterparts. To further assess how competitive the most promising HMOHS is, we have compared it against three state-of-the-art MOEAs such as NSGA-II, SPEA2 and MOEA/D for solving MOPs, and then convergence speed of each algorithm on test problems has been analyzed by using graph display. In the context of problems, metrics, and parameter settings used, HMOHS is better than the other three algorithms on most test problems. Finally, the HMOHS with other evolutionary strategies and its application to solve real-world applications are our future work.

REFERENCES

[1] M. Mateo and I. Alberto, "A mutation operator based on a Pareto ranking for multi-objective evolutionary algorithms," *J. Heuristics*, vol. 18, no. 1, pp. 53–89, Feb. 2012.

[2] O. Chikumbo, E. Goodman, and K. Deb, "Approximating a multi-dimensional Pareto front for a land use management problem: A modified MOEA with an epigenetic silencing metaphor," *In. Proc. CEC*, pp. 1–8, 2012.

[3] S. Sivasubramani and K.S. Swarup, "Multi-objective harmony search algorithm for optimal power flow problem," *Int. J. Elec. Power.* , vol. 33, no. 3, pp. 745–752, Mar. 2011.

[4] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 95-197, Apr. 2002.

[5] E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papailiou, and T. Fogarty, Eds., Athens, Greece, pp. 95-100, 2001.

[6] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712-731, Dec. 2007.

[7] Z.W. Geem, J.H. Kim and G.V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol.76, no. 2, pp. 60–68, Feb. 2001.

[8] Q. K. Pan, L. Wang and L. Gao, "A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers," *Appl. Soft Comput.*, vo. 11, no. 8, pp. 5270–5280, Dec. 2011.

[9] W. Sun, J. Wang and H. Chang, "Forecasting Annual Power Generation Using a Harmony Search Algorithm-Based Joint Parameters Optimization Combination Model," *Energies*, vol.5, no. 10, pp. 3948-3971, Oct. 2012.

[10] M. Aghaie, T. Nazari, A. Zolfaghari, A. Minuchehr, *et al*, "Investigation of PWR core optimization using harmony search algorithms," *Ann. Nucl. Energy*, vol. 57, pp. 1–15, Jul. 2013.

[11] L. Wang, R. Yang, P. M. Pardalos, L. Qian and M. Fei, "An adaptive fuzzy controller based on harmony search and its application to power plant control," *Int. J Elec. Power*, vol. 53, pp. 272–278, Dec. 2013.

[12] Y. Yuan, H. Xu, J. Yang, "A hybrid harmony search algorithm for the flexible job shop scheduling problem," *Appl. Soft Comput.*, vo. 13, no. 7, pp. 3259–3272, Jul. 2013.

[13] M. Mahdavi, M. Fesanghary and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput.*, vol. 188, no.2, pp.1567–1579, May. 2007

[14] M.G.H. Omran and M. Mahdavi, "Global-best harmony search," *Appl. Math. Comput.*, vol. 198, no. 2, pp. 643–656, May. 2008.

[15] Q. K. Pan, P. N. Suganthan, M. F. Tasgetiren and J.J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Appl. Math. Comput.*, vol. 216, no. 3, pp. 830–848, Apr. 2010.

[16] Q.K. Pan, P.N. Suganthan, J.J. Liang and M.F. Tasgtrien, "A local-best harmony search algorithm with dynamic subpopulations," *Eng. Optimiz.*, vol. 42, no. 2, pp. 101–117, 2010.

[17] D. Zou, L. Gao, J. Wu and S. Li, "Novel global harmony search algorithm for unconstrained problems," *Neurocomputing*, vol. 73, no. 16-18, pp. 3308–3318, Oct. 2010.

[18] L. Wang, R. Yang, Y. Xu, Q. Niu, P. M. Pardalos and M. Fei "An improved adaptive binary Harmony Search algorithm," *Inform. Sciences*, vol.232, pp.58-87, May. 2013.

[19] S. Sivasubramani and K.S. Swarup, "Environmental/economic dispatch using multi-objective harmony search algorithm," *Electr. Pow. Syst. Res.*, vol.81, no.19, pp. 1778–1785, Sep. 2011.

[20] I. Landa-Torres, D. Manjarres, S. Salcedo-Sanz, J. Del Ser and S. Gil-Lopez, "A multi-objective grouping Harmony Search algorithm for the optimal distribution of 24-hour medical emergency units," *Expert Syst. Appl.*, vol.40, no.2, pp. 2343–2349, May. 2013.

[21] S. Salcedo-Sanz, D. Manjarrés, D. Manjarrés, J. Del Ser, *et al.*, "One-way urban traffic reconfiguration using a multi-objective harmony search approach," *Expert Syst. Appl.*, vol.40, no.9, pp. 3341–3350, Jul. 2013.

[22] K. Nekooei, M. M. Farsangi, H. Nezamabadi-Pour and K. Lee, “An Improved Multi-Objective Harmony Search for Optimal Placement of DGs in Distribution Systems,” *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp.557–567, Mar. 2013.
[23] D. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning,” Addison-Wesley, Boston, MA, 1989.
[24] C. Li, S. Yang, and T. T. Nguyen, “A Self-Learning Particle Swarm Optimizer for Global Optimization Problems,” *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp.627-646, Jun. 2012.
[25] Z. Zhan, J. Zhang, Y. Li and H. S. Chuang, “Adaptive Particle Swarm Optimization,” *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 39, no. 6, Dec. 2009.
[26] Y. Wang and Z. Cai, “A dynamic hybrid framework for constrained evolutionary optimization,” *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no.1, pp. 203–217, Feb. 2012.
[27] J. H. Holland, “Adaptation in Natural and Artificial Systems,” University of Michigan Press, Ann Arbor, MI, USA, 1975.
[28] J. Chen, Q. K. Pan and J. Q. Li, “Harmony search algorithm with dynamic control parameters,” *Appl. Math. Comput.*, vol. 219, no.2, pp. 592–604, Oct. 2012.
[29] A. Kattan, and R. Abdullah, “A dynamic self-adaptive harmony search algorithm for continuous optimization problems,” *Appl. Math. Comput.*, vol. 219, no.16, pp. 8542–8567, Apr. 2013.
[30] R. Diao and Q. Shen, “Feature Selection with Harmony Search,” *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 6, pp. 1509–1523, Dec. 2012.
[31] E. Zitzler, K. Deb, and L. Thiele. “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results,” *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
[32] V. L. Huang, P. N. Suganthan, and J. J. Liang. “Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems,” *Int. J Intell. Syst.*, vol. 21, no. 2, pp. 209–226, Feb. 2006.
[33] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. “Scalable Test Problems for Evolutionary Multiobjective Optimization,” In *Evolutionary Multiobjective Optimization*, A. Abraham, L. Jain, and R. Golderg, Eds. London: Springer, USA, pp. 105-145, 2005.
[34] M. Helbig and A. P. Engelbrecht, “Performance measures for dynamic multi-objective optimization algorithms,” *Inform. Sciences*, vol.250, pp.61-81, Nov. 2013.
[35] J. Knowles, L. Thiele, and E. Zitzler, “A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers,” Technical Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2006.
[36] E. Zitzler L. Thiele, “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach,” *IEEE Trans. Evol. Comput.*, vol.3, no.4, pp.257–271, 1999.
[37] J. J. Durillo, “Metaheuristics for Multi-objective Optimization: Design, Analysis, and Applications,” University Malaga, 2011.
[38] A. J. Nebro, J. J. Durillo, F. Luna *et al.*, “Design Issues in a Multi-objective Cellular Genetic Algorithm,” *EMO, LNCS* 4403, pp. 126–140, 2007.
[39] B. Luo, J. Zheng, R. xie and J. Wu, “Dynamic crowding distance-A new diversity maintenance strategy for MOEAs”, in: *Natural Computation, 2008 ICNC ‘08*. Fourth International Conference on, vol. 1, pp.580–585, 2008.
[40] H. Li and Q. Zhang, “Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284-302, Apr. 2009.
[41] L. Ke, Q. Zhang and R. Battiti, “Using ACO in MOEA/D for Multiobjective Combinatorial Optimization,” *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, to be published.
[42] Q. Zhang, W. Liu, E. Tsang and B. Edward, “Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model,” *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456-474, Jun. 2010.
[43] H. Liu, F. Gu, and Q. Zhang, “Decomposition of a Multiobjective Optimization Problem into a Number of Simple Multiobjective Subproblems,” *IEEE Trans. Evol. Comput.*, to be published.
[44] S. Zhao, P.N. Suganthan and Q. Zhang, “Decomposed-Based Multiobjective Evolutionary Algorithm with an Ensemble of Neighbourhood Sizes,” *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442-446, Jun. 2012.
[45] K. Price, R. Storn, and J. Lampinen, “Differential evolution: a practical approach to global optimization,” Springer, Berlin, 2005.