Saudi Journal of Biological Sciences 27 (2020) 3743–3752

Contents lists available at ScienceDirect

Saudi Journal of Biological Sciences

journal homepage: www.sciencedirect.com

Original article

The hybrid bacterial foraging algorithm based on many-objective optimizer

Yang Liu, Liwei Tian, Linan Fan *

School of Information Engineering, Shenyang University, Shenyang 110044, China

ARTICLE INFO

Article history:
Received 10 April 2020
Revised 11 August 2020
Accepted 12 August 2020
Available online 19 August 2020

Keywords:
The improvement of many-objective problems
Bacterial foraging improvement
The hybrid strategy

ABSTRACT

A new multi-objective optimized bacterial foraging algorithm - Hybrid Multi-Objective Optimized Bacterial Foraging Algorithm (HMOBFA) is presented in this article. The proposed algorithm combines the crossover-archives strategy and the life-cycle optimization strategy, look for the best method through research area. The crossover-archive strategy with an external archive and internal archive is assigned to different selection principles to focus on diversity and convergence separately. Additionally, according to the local landscape to satisfy population diversity and variability as well as avoiding redundant local searches, individuals can switch their states periodically throughout the colony lifecycle with the life-cycle optimization strategy. all of which may perform significantly well. The performance of the algorithm was examined with several standard criterion functions and compared with other classical multi-objective majorization methods. The examiner results show that the HMOBFA algorithm can achieve a significant enhancement in performance compare with other method and handles many-objective issues with solid complexity, convergence as well as diversity. The HMOBFA algorithm has been proven to be an excellent alternative to past methods for solving the improvement of many-objective problems.

1. Introduction

Most improvement cases contain not only one objective, and sometimes the objectives conflict with one another, a situation that is referred as multi-objective optimization (MO). The improvement of many-objective issues refers to multi-objective issues with more than three objectives (Deb, 2001). Many classical evolutionary or genetic algorithm (GA)-based process have been developed to solve the improvement of many-objective issues, such as multi-objective evolutionary algorithms (MOEAs) (Dorigo et al., xxxx); nondominated sorting genetic algorithm (NSGAII) (Deb et al., 2001), multiple objectives particle swarm optimization (MOPSO) (Coello et al., 2004), and bacterial chemotaxis optimization (BCO) (Alejandra Guzmán et al., 2010).

Ecosystems in nature have been an abundant fountain of mechanisms for designing computing systems and artificial intelligence to work out complicated computer and engineering science improvement issues. The GA-based methods have attracted increasing attention; Biological operation processes inspirited scientist for a long time and have developed efficient optimization algorithms to simulate the unique certain organisms' architecture or certain actions. Such as, the genetic algorithm (GA) represent the evolutionary and biogenetic optimization model (Deb, 2001). Ant colony optimization (ACO) is invented according to the ant colony foraging nature movement (Deb, 2001; Dorigo et al., xxxx). Particle swarm optimization (PSO) use concept from the bird association nature movement and fish schools. (Coello et al., 2004). These algorithms outperform gradient-based or classical heuristics algorithms when dealing with complex functions that are multimodal, not differentiable, and discrete. At this stage, before achieving the Pareto optimal result, these naturally inspired paradigms are used to deal with the multi-objective improvement issue in the act of finding available non-dominated result widely. However, these nature-based algorithms have their drawbacks. For example, the GA can avoid local optima but experiences slow convergence. Compare with above algorithm, the particle swarm optimization algorithm can obtain a faster convergence rate but is more likely to fall into a locally optimal solution.

In recent years, as a rich resource for engineering applications and computational models, bacterial foraging behavior has

* Corresponding author.
E-mail address: linanfan@163.com (L. Fan).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.sjbs.2020.08.021
1319-562X/

attracted increasing attention. Models for replicating bacterial foraging nature movement have been established and put into use to work out practical issues (Alejandra Guzmán et al., 2010) (Tripathy et al., 2006, 2006). In all the models, bacterial foraging optimization (BFO) is a robust and straightforward improvement tool that can replicate the foraging nature movement Up to now, There are many specific engineering issues that should be solved effectively by using the BFO algorithm, for example, machine learning (ML), optimal control, transmission loss reduction and harmonic excitation ([6]).

Classical bacterial foraging improvement methods can be used to solve single-objective improvement issues. Many improvement methods are usually handled as multiobjective optimization problems with high-dimensional objectives, all values should be able to satisfy all functions, and the convergence, diversity, and complexity should be considered at the same time. Nonetheless, tests on multi-modal and complicated comparison functions show that the convergence result of the BFO method is not good enough, and its result becomes worse and worse with the more complicated matter and the increment of search dimension.

In order to get better improvement of BFO, the life-cycle optimization strategy and crossover-archives strategy are used to replace the simple descriptions of chemotaxis nature movement in the classical BFO algorithm. We operated a lot of experiment using a common set of benchmark functions and re-evaluated the result performance of the algorithm. The experiment results are very positive because the HMOBFA shows huge superior search performance while keep search result almost the same. If it is not superior, the result is in all benchmark functions for convergence speed, robustness and accuracy.

The main contributions of this article are as follows:

•A new algorithm with a hybrid strategy based on the original bacterial foraging improvement method is introduced to effectively deal with the improvement of many-objective issues.

•The introduction of a lifecycle mechanism and crossover-archives strategy are integrated to receive better diversity and faster convergence and avoid a redundant local search.

The other part of the paper is briefly described below. In the second section, the BFO method and bacterial chemotaxis are briefly reviewed, and the essential movement of bacteria in the BFO model is analyzed in depth. In Section III, the new strategy which is used by HMOBFA is introduced, and its mechanism is reviewed. This section will also introduce the HMOBFA method and explain their operation details. In Section IV, the experimental research on the proposed HMOBFA method and other method is performed, and the used test function, laboratorial results and test settings and are described. Finally, section V summarizes the full conclusion.

2. Classical bacterial foraging optimization method

The classical bacterial foraging optimization method (BFO) is a new kind of bionic method based on the chemotaxis foraging nature movement of *E. coli*. Just like multicellular organisms and other social structures, bacterial colonies use a variety of complex social nature movements to change their group structure and decision-making (Jacob et al., 2004). The cooperative nature movement of bacterial communities is considered a universal mechanism, and it has been adopted by many advanced organisms (Alejandra Guzmán et al., 2010; Tripathy et al., 2006, 2006). According to the concept of no dominance, strong bacteria contrast their present location with their eariler location by using their spatiotemporal memory. The consequence of the contrast is that every strong bacterium has two movement patterns: if this bacterium's location (present or earlier) control the other bacterium's location, the bacterium moves to a non-controlled location and start to be a tight tumble and the direction is random (short tumbling). Nonetheless, if either location (present or earlier) controls another location, the bacterium makes a larger roll (long tumble) in a random direction from its current location. When a nutrient source is discovered, the stronger bacteria send a signal to their colony members, and the weak individuals are notified.

The weak bacteria, following the patterns of bacterial communication in nature, will be notified by the strong bacteria when a nutrient source is discovered via a signal to their colony members. The advantages to weak bacteria that use this information exchange are as follows:weak bacteria recognize that if they approach strong bacteria, they can get the origin of nutrition. Thus, weak bacteria randomly choose a strong bacterium to move closer to, moving stepwise towards the nutrient-rich location. Once the chemotaxis strategy has been applied to each bacterium in the colony, the chemotaxis step is completed.

In the model of BFO, three characteristics of bacteria are simulated: chemotaxis, reproduction and migration.

(1) Chemotaxis: For bacteria, chemotaxis is the process of the aggregation to eutrophic areas. Bacterial motion patterns include both tumbling and swimming. The unit step length of the bacterium moving in any direction is defined as a flip. It will determine whether the new position is more or less desirable than the last in terms of a need to adapt. Then, the bacterium will maintain to move in a more eager tendency for a few steps until the threshold for adaptation is no more closed. The formula is simplified as follows:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{1}$$

Δ(i) is equal to the last step. When the bacterial movement pattern is tumbling, Δ(i) is a random vector which is [-1, 1]. The Δ(i)$\in R^n$ is the direction vector which is the jth chemotactic step. When the bacterial movement pattern is swimming,

(2) Swarming: As a new discovery in microbiology, this process allows bacteria to use secreted chemical signaling molecules called automatic inducers to look for related cells in the surrounding environment, and the mutual attraction and repulsion of these cells are combined. The value is rendered by calculating the spend function amount to the practical spend function to make it minimization, which is calculated by Eq. (2).

$$F_{cc}\left(\theta, P(j,k,l) = \sum_{i=1}^{S} F_{cc}^{i}\left(\theta, \theta^i(j,k,l)\right) = \sum_{i=1}^{S}\left[-d_a \exp\left(-w_a \sum_{m=1}^{P}\left(\theta_m, \theta_m^i\right)^i\right)\right] + \sum_{i=1}^{S}\left[h_r \exp\left(-w_r \sum_{m=1}^{P}\left(\theta_m, \theta_m^i\right)^2\right)\right] \tag{2}$$

In which P is the whole number of current parameters in the bacterium which need to be optimized. $d_a$and w_a are attractant coefficients that are to be chosen. h_r and w_r are repellent coefficients that are to be chosen.

If there are G (G = 1, 2. . .g) objectives to be solved for, the values of each function should be calculated.

$$F_g(i,j+1,k,l) = F_g(i,j+1,k,l) + F_{cc}\left(\theta^i(j+1,k,l), P(j+1,k,l)\right) \quad (3)$$

(3) Replication: After several chemotaxes, the bacteria propagate. The bacteria are classified according to their health. The fitness is calculated during life. The healthy half of the bacteria will divide, and each bacterium will form two new individuals in its original location. The freshly generated bacteria will take the biological feature of the parent bacteria and have the equal step size compare to the parent. The number of bacteria which is not strong enough will decrease. The whole number of bacteria still has the same level when duplication happened.

(4) Migration (dispersion): The procedure of chemotaxis guarantees the native seeking capacity of bacteria. The reproduction process improves the seeking rate of bacteria, but for complicated majorization, chemotaxis and reproduction cannot stop bacteria from slide into a native optimal solution. Thus, BFO introduces a dispersal mechanism. After propagation occurs a certain number of times, the bacteria perform a dispersal operation. The individual bacteria are scattered to a random location in the seeking area with a determinate possibility.

3. Many-objective hybrid bacterial foraging optimization

3.1. The optimization of many-objective problems

The optimization of many-objective problems (ManyOPs) are knows as problems in which several objectives should be simultaneously optimized, and a vector of decision variables can be found that both satisfy the restriction and optimize the vector function. Most issues in the actuality world include optimizing multiple objective functions simultaneously, and these objective functions often conflict with each other. The optimization of many-objective problems can be mathematically defined as follows ([6]):

Minimize:$F = \{f_1(x), f_2(x), \ldots f_n(x)\}\ i = 1, 2, \ldots, N$

Subject to: $g_i(x) \leqslant 0 j = 1, 2, \ldots, J$

$$h_k(x) = 0\, k = 1, 2, \ldots, K$$

$$x_i^L \leqslant x_i \leqslant x_i^U\ i = 1, 2, \ldots, N$$

where $\times$ is the vector of determined variables $x = (x_1, x_2 \ldots x_3) \in x$; x is the space of parameter, and F is the vector of the objective. $f_i(x)$ are the vector of majorization objective which meet the J inequalities$g_i(x)$ and the k inequalities $h_k(x)$ the functions of restrain. x_i^Lstand for the minimum space of adopted values, and x_i^U stand for the maximum space of adopted values. The J inequalities g_i and the K inequalities h_k stand for the function of restrain functions.

The traditional method used to deal with the optimization of many-objective problems is known as the Pareto optimal front (POF), which is not a sole best result, but a group of results that are nondominated and balance; however, on ManyOPs, with the total of targets expend, the convergence result become not so good. , In order to deal with this issue, a non-Pareto-based approach called an indicator-based approach is proposed. This approach takes an indicator as the fitness function, but because of its lack of diversity maintenance, the diversity of this approach is not satisfactory (Handing Wang et al., 2015).

3.2. The crossover-archives strategy

The major concept the crossover-archive policy is to divide the solution set into the external archive (EA) and the internal archive (IA), which have different updating rules for the two goals of ManyOPs. The sizes of EA and IA are fixed separately; the former maintains the diversity based on the Pareto control a high-dimensional nature space, and the latter encourages the convergence at a fast speed by the quality indicator I(ϵ+). The crossovers are made between EA and IA, and mutations occur in EA during reproduction. Offspring of IA and EA are selected independently of each other by different methods, and IA is used as the final output (Handing Wang et al., 2015).

3.2.1. The updating rule of the file

The updating rule of the external file is based on the Pareto dominance, which should both lead the seek to the global Pareto optimal area and keeps the population variety of the Pareto best front, all values must meet the whole functions requirement. The purpose of this is to find a nondominant front closing to the real Pareto front. The value of the objective functions is represented by a function space Which is evaluated in each position of the bacteria in the parameter area to simulate the environment of real bacteria, objective function values for each bacterium are calculated. Which on behalf of the amount of nutrients show in the first location, the current position of the bacteria is divided into Pareto optimal frontier (POF). The solution is represented by the location of the bacteria. According to the level of environmental conditions, the others are further classified into different POF sets. The bacteria in the most nutrient-rich environments are sorted of the initial series of the Pareto sell front solution in each iteration. To improve the diversity, the suitable distances are chosen.

Instead of the traditional Euclidean distance, the fractional (L_d norm) distances were adopted, which show a better result in a huge data dimensional area.

$$L_d(x,y) = \sum \left[\left(x^i - f(x)^i\right)^d\right]^{1/d} \quad (4)$$

where $d = 1/m$, m is the quantity of objectives.

3.2.2. The updating rule of the internal file

To increase the result of the convergence with ManyOPs. Select quality signal as selection tenet for IA. When the IA is updated, and the descendants are first added to the IA, the minimum distance is given as an indicator $I_{\epsilon+}$, which is one solution that needs to control another result in the objective area. It is defined as follows:

$$I_{\in +}(x_1, x_2) = \min_{\in} \{f_i(x_1) - \in \leqslant f_i(x_2), 1 \leqslant i \leqslant n\} \quad (5)$$

n is the quantity of the objectives. the indicator $I_{\in +}$ is dominance preserving, when x_1 dominates x_2, the indicator values become negative. According to the dominance preserving, the suitable result of each are assigned in IBEA as follows:

$$F(x_1) = \sum_{x_2 \in P\{\{x_1\}} -e^{-I_{\epsilon+}(x_2,x_1)/k} \quad (6)$$

where k should bigger than 0; we chose k = 0.05. According to the fitness, the extra solutions in IA are deleted. In each iteration, the indicator-based selection principle removes the solution with the minimum $I_{\in +} I_{\epsilon+}$ loss, updates the $I_{\in +} I_{\epsilon+}$ values of the remainder of IA, and obtains a settled quantity of result.

3.2.3. The pseudocode of the selection principles

The pseudocode of the selection principles is shown in Table 1, which includes the updating rule of the external archive and the internal archive.

Table 1
The pseudocode of the selection principles.

Step 1: Initialize parameters A_I, n_{IA}, $A_D^* A_D^*$, n_{DA} where
A_I-Pareto set approximation of IA
n_{IA}-the fixed population size of A_i
A_E^*-the overflowed EA
N_{EA}-the fixed population size of EA
A_E-Output
Step 2: Moderate calculation
Calculate the fitness values of individuals according to Eq. (3)
Step 3: Reproduction (crossover between IA and EA and mutation on IA only)
tep 4: Update IA by I(ϵ+)
a. While$|A_I| > n_{IA}$
b.$F(x^*)$Calculate out the suitable minimal x
c.Delete x^* from A_I
d.Update the remaining individual by
$F(x) = F(x) + \sum_{x_2 \in P\{\{x_1\}} - e^{-I_{\epsilon+}(x_2,x_1)/k}\ e^{-I_{\epsilon+}(x_2,x_1)/0.05}$
e. End
Step 5: Update EA by Pareto dominance
a. Set A_E as empty.
b. Seek out the maximal or minimal objective result in A_E^*Then move the result to A_E fromA_E^*
c. When the AE is not full
d.I in A_E^* for every individual
e.Similarity[i] = min(distance(i,j)),j∈A_E
f.End
g.I = argmax (Similarity), move result I from $A_E^*$to A_E
h. End
End

3.3. Life-cycle optimization strategy

3.3.1. The life-cycle transformation of the bacterial state

Within the biology area, the life cycle means the different stages a living body goes through. This work assumes that there are five states of transformation in the bacterial colony: birth, foraging, reproduction, migration and death. Every individual transforms its state in a different stage to adapt to the particular environment (Liu et al., ,2013). In the process of population initialization, all bacteria are born, and the then process to the seek food. During the foraging process, if the bacteria obtain enough nutrients from the environment, they will enter a state of reproduction (Liu et al., ,2013). To maximize foraging ability, the bacterial colony evolves in adaptive growth patterns, resulting in complex colony structures. Some replicate another individual in the original location. If bacteria live in places with low nutritional value for a long period of time, other bacteria will die and disappear from the population. In addition, if a bacterium has low nutritional value and does not die, it will randomly migrate with a certain probability to another random location. After the bacteria split and migrate, the new bacterium enters the state of birth and restarts its foraging process. In the lifecycle foraging optimization model, the state changes of bacteria during foraging are shown in Fig. 1.

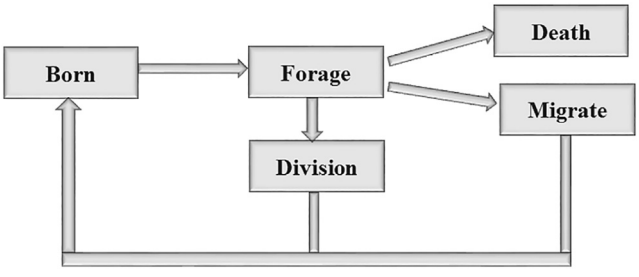

Fig. 1. The life-cycle transformation of the bacterial state.

3.3.2. The rules of lifecycle optimization strategy

When bacteria are foraging in their environment, they obtain nutrients from the environment and consume resources for energy in the foraging process. We define the total nutritional value of the bacteria as $H_i(t)$, *which* expresses the health degree of the bacterium in a chemotactic foraging transformation. When a superior position is found, the nutrient value of the individual bacterium increases by one; otherwise, the foraging action is considered unsuccessful, and the nutrient value decreases by one. We assume the definition of the resource of the energy value of the bacterium as R_i^t, which is calculated as follows:

$$R_i^t = \eta \frac{\frac{F(X_i^t) - F_{worst}^t}{F_{best}^t - F_{worst}^t}}{\sum_{j=1}^{S^t} \frac{F(X_i^t) - F_{worst}^t}{F_{best}^t - F_{worst}^t}} + (1 - \eta) \frac{H_i(t)}{\sum_{i=1}^{S^t} H_j(t)}, \eta \subset [0,1] \quad (7)$$

where $F(X_i^t)$ is the ith bacterium fitness, F_{best}^t is the best fitness of the whole bacterial colony, F_{worst}^t is the worst fitness of all the bacterial population (Liu et al., ,2013; Chen et al., 2014).

In order to prevent the size of the population from becoming too huge or too tiny, a growth principle of population was introduced to the adaptive modulation rules. R_d and R_e are adaptive population size adjustment parameters, which are used to control the threshold of the division and the threshold of the extinction. This strategy is more fit the order of the nature. that is, the evolution of the biological population will be affected by transformations to their living environment. When populations are overcrowded, competition is greater than cooperation, and populations tend to die off. When the population is less dense, cooperation between individuals is more important than competition, and it becomes easier to reproduce. Through this strategy, the lifecycle optimization model keeps the size of the bacterial population in a relatively stable range. In addition, if a bacterium has a low enough energy value but is not yet completely dead, the bacterium in the model will perform a migration operation with a certain probability. If the migration probability P_e is better than a random number, which is generated randomly, the bacterium moves randomly to another location in the foraging space . After the bacterium carries out its breeding or migration operation, the newly born bacterium enters its initial state, restarting the foraging process (Chen et al., 2014).

3.3.3. The pseudocode of the lifecycle optimization strategy

The pseudocode of lifecycle optimization strategy is illustrated in Table 2, which describes the whole transformation in the bacteria colony.

3.4. The proposed algorithm

The Hybrid Multi-Objective Optimized Bacterial Foraging Algorithm (HMOBFA) is briefly introduced step by step, the pseudocode is shown in Table 3, and the flow diagram of the HMOBFA method is showed in Fig. 2 (Chen et al., 2014)

4. Benchmark test

To evaluate the result of the HMOBFA, let us do some comparison with several classical many-objective algorithms, including MOEA/D, NSGA-III , MOMBI2 and VaEA. The involved test functions are selected from DTLZ and WFG . To be specific, these compared algorithms can be classified into 3 groups. 1) the decomposition approach (MOEA/D), 2) the indicator-based approach (MOMBI2) and 3) the Pareto-based approaches (NSGA-III and VaEA). The deal with particulars of MOEA/D, NSGA-III, and VaEA can be found in

Table 2
Pseudocode of the lifecycle optimization strategy.

```
Step 1: Transform the nutrition value
  If (F(X_i(t + 1) < F(X_i(t))) // Location improved
    H_i(t + 1) = H_i(t) + 1; // Nutritional value plus one
  Else
    H_i(t + 1) = H_i(t)-1; // Nutritional value minus one
  End
Step 2: Calculate the resource of the energy fitness values of individuals according to Eq. (7)
Step 3: Division operation
  If (R_i^t > max(R_d, R_d + ((S^t − S)/R_e))) // To achieve a splitting condition
    H_i = 0; // Return the nutritional value to zero
    X_{S+1} = X_i; // Split into a new individual
    S = S + 1; // Population size plus one
  End
Step 4: Extinction operation
  If () // Reach the conditions of death
    X_i = []; // Remove from population
    S = S-1; // Population size minus one
  End
Step 5: Migration operation
  If (Pe > rand())
    Xi(t + 1) = randposition; // Move randomly to another location in the foraging space.
  End
```

their original paper. MOMBI2 is a new algorithm on account of R2, which is a less-computation indicator (Deb et al., 2001).

4.1. Test functions

The initial 4 measurement instances (DTZL1 to DTLZ4) are selected from the DTLZ measurement suit (Deb et al., 2001). Following the advice in (Deb et al., 2001), the quantity of determine variate is initialized with $n = M + r - 1$, in which M means the objective quantity, $r = 5$ stand for DTLZ1, and $r = 10$ stand for DTLZ2 to DTLZ4.

The other measurement example (WFG1 to WFG6) are chosen from the WFG measurement suit , in which the quantity of determine variate is initialized with $n = k + l - 1$. Following the suggestion in (Deb and Jain, 2014), the relevant distance argument l is initialized with 10, and the relevant position argument k is initialized with 4, 10, and 7 for the measurement example with $M = 3$, 5, and 8, individually. The geometries of these test instances cover separability-no separability, unimodality-multimodality, and convex-concave characteristics.

To compare each algorithm's performance, the widely used IGD metric is adopted in our experiments . Its detailed description can be found in (Deb and Jain, 2014).

4.2. Experimental setting

The parameter values for the involved algorithms follow the recommended settings of their original designs, as indicated below.

1) Group size: for MOEA/D and NSGA-III, the Group magnitude is determined empirically based on the simplex-lattice layout element H and the objective quantity M. As for VaEA, the group magnitude is equal to the equal that of NSGA-III. Regarding HMOBFA and MOMBI2, the group magnitude is initial with to that of NSGA-III and MOEA/D, As regard to different objective quantity M.

2) Intersection and saltation: NSGA-III uses SBX and polynomial saltation to create offspring. Accordingly, the propagator key of intersection and saltation (n_c, n_m) are initialed with 20. In addition, the intersection probability and saltation probability pc and pm are initialed with 1.0 and $1/D$, independently, where D is the quantity of determine variate.

3) Finish criteria: Every calculation is executed for 20 runs self-governed on their measurement living example, and the maximum quantity of iterations is initialed with 4000.

4) Other arguments: MOEA/D take the Chebycheff come near. Regarding MOMBI2, its related arguments are initialed withε = 1-e-3 and $\alpha = 0.5$. As for HMOBFA and VaEA, their arguments are initialed with the equal that is NSGA-III.

4.3. Computational results

The computational results of all algorithms over 3-, 5-, and 8-test living example are showed in Tables 4 and 5. Showed with the tables, the average and criterion deviation of the HV measures obtained by these methods in 20 self-governed runs were documented.

From Table 4, we can see that HMOBFA obtains powerful performance, achieving the best results, ranking first on 5 test instances, and ranking second on 3 test instances out of the 12 instances. At the same time, MOEA/D also performs well, obtaining the best results in 4 test instances. For MOMBI2, it achieves the first ranks on 3- and 5-objective DTLZ3. Noticed that the shape of the PF of DTLZ3 consists of a serious of discrete fragment which is difficult to address. For VaEA, it does best on 2 test instances, which is still better than NSGA-III.

DTLZ1 has a simple linear PF. As shown, all the algorithms have promising performance. MOEA/D and HMOBFA perform relatively better than other methods on the whole DTLZ1 living example. The PF of DTLZ2 is a spherical concave. We can also observe that HMOBFA and MOEA/D achieve the best results on these DTLZ2 living example. Note that MOMBI2 also performs well on the 3-objective DTLZ2 instance. The PF of DTLZ4 has a strongly nonuniform PF, which is difficult to optimize. VaEA does well on all DTLZ4 instances, obtaining the best results on 3-objective and 5-objective measurement living example. However, HMOBFA acquire the best data on the high-dimensional DTLZ4 instance, obtaining the first rank on 8-objective DTLZ4.

From Table 5, it should be noticed that MOEA/D and HMOBFA get the best on most measurement living example. To be specific, HMOBFA is ranked first in terms of IGD on 8 out of 18 instances. In addition, MOEA/D is ranked first in 4 test instances. On 5- and 8-objective WFG1 and 3- and 5-objective WFG3, MOEA/D obtains better results than HMOBFA. For 5- and 8-objective WFG1, MOMBIII also obtains the best results.

Table 3
Pseudocode of the HMOBFA algorithm.

Step 1: Parameters initialization
Ns: steps of swim,
Nc: steps of chemotactic,
S: bacteria quantity,
Nre: steps of reproductive,
P: search space dimension,
Ped: elimination probability,
Ned: steps of elimination and dispersal,
C(i): the unit of run-length. (*i.e.*, the step size of chemotactic during every tumble or run).
Step 2: Loops for elimination and dispersal: $l = l + 1$.
Step 3: Loops for reproduction: $k = k + 1$.
Step 4: Loops for chemotaxis: $j = j + 1$.
a. For $i = 1 = 1, 2, \ldots, S$, take a bacteria chemotactic step i ,show below.
b. Calculate the fitness functions g, $F_g\ (i,j,k,l)$
g on behalf of the objectives quantity g = 1,2,. . .G
c. Let $F_{lastg} = F_g(i,j,k,l)$ to record the values because we could calculate out a better figure through next run.
d. Tumble: Develop a stochastic vector $\Delta(i) \in R^n$ with every element $\Delta m(i)$, m = 1, 2, . . ., S, a stochastic figure on [-1, 1].
e. Move: Update of its position as Eq. (1)
g. Compute each fitness function F_g(i,j + 1,k,l) (g = 1,2,. . .G) with θ^i(i,j + 1,k,l) on the basis of Eq. (2) and Eq. (3).
h. Swim:
(i) Make m = 0 (swim length counter).
(ii) While m < Ns (if the bacteria have not moved too far
Let m = m + 1.
• If$F_1(i, j+1, k, l) < F$
let $F_{last1} = F(i, j+1, k, l)$
If$F_2(i, j+1, k, l) < F_{last2}$
let$F_{last2} = F_2(i, j+1, k, l)$
If$F_g(i, j+1, k, l) < F_{lastg}$
let$F_{lastg} = F_g(i, j+1, k, l)$
Therefore, another size C(i) step in the equal orientation will be seen as Eq. (1) and use the latest updated θ^i(j + 1,k,l) to calculate the new
$F_{lasti} = F_i(i, j+1, k, l)(i = 1, 2, \ldots G)$
$F_{lasti} = F_i(i, j+1, k, l)$Construct the Change rules of the bacterial nutrition value as shown in Table 2.
• Else let m = Ns.
i. will turn into another new bacterium (i + 1): if i ≠ S turn into step (b) to calculate the new bacteria.
Step 5: If j < Nc, turn into step 3. Under the circumstances, chemotaxis need to go on because the bacteria's life is ongoing.
Step 6: Reproduction based on crossover-archives strategy
The procedure is shown in Table 1.
Step a Update EA by Pareto dominance:
According to the updating rule of the external archive, update EA by Pareto dominance.
Step b Update IA by I(ϵ+)
According to the updating rule of the external archive, update IA by the quality indicator I(ϵ+)
Step c Crossover between IA and EA
The crossovers are made between EA and IA, and the mutations occur in EA during reproduction. IA was used as the final output
Step 7: Reproduction and death operations based on the lifecycle mechanism:
The procedure is shown in Table 2.
Step a Calculate the information rate of each bacteria by Eq. (7), determine whether or not to reproduce.
Step b Division operation
If reproduction occurs, according to the adaptive bacterial splitting rules, the bacteria splits.
Step c Extinction operation
If the bacterium dies, according to the rule of adaptive bacterial extinction, it is removed from the population count.
Step d Migration operation
If the migration probability P_e > rand (), the bacterium moves randomly to another location in the foraging space.
Step 8: If k < Nre turn into step 2. Under the circumstances, the quantity of prescriptive duplication steps has not been satisfied, and another step is initiated in the next chemotactic cycle.
Step 9: Elimination and dispersal:
From i = 1, 2, . . ., S, and $P_{ed}P_{ed}$probability eliminate and disperse every bacteria, that makes the bacteria quantity in the population stable. In order to meet this requirement, if a bacterium is killed, we only need to distribute the bacterium to a stochastic position on the majorization dimension. If $l < N_{ed}$, and turn into step 2; if not, over.

On WFG2, whose PF includes a serial of disconnected convex segments, NSGA-III has the best data, ranked first on 3-objective and 5-objective WFG2 living example. HMOBFA is ranked second on the WFG2 instances. On WFG3 with a degenerated PF, HMOBFA and the other algorithms perform similarly to each other on the 3- and 5-objective test instances. It is noted that HMOBFA turn into more competent when the quantity of objectives raise on this test problem. On WFG4, whose PF has a set of local optima, HMOBFA exhibits significant advantage to other arithmetic, obtaining the first-rank results on 5-objective and 8-objective measurement living example. On the deceptive WFG5, HMOBFA performs

more powerfully on the 5-objective and 8-objective measurement living example, while NSGA-III also gets the best data on the 3-objective living example. On WFG6, which is nonseparable and reduced, NSGA-III obtains the best data on the 3-objective measurement living example, while HMOBFA is the best performer for the other WFG6 instances.

Figs. 3 and 4 illustrate the distributions of the final solutions found by all algorithms on the 8-objective DTLZ3, and WFG4, respectively, plotted by parallel coordinates. This chart shows that the last solutions obtained by HMOBFA are better in terms of both convergence and distribution than the other compared algorithms. Specifically, for 8-objective WFG4, it is clear that the solutions obtained by HMOBFA have an excellent distribution, while other algorithms perform relatively poorly.

Fig. 5 shows the evolution comparison of all algorithms on the basis of IGD data compare to the quantity of iterations. It can be found from Fig. 5 that HMOBFA exhibits a superior convergence

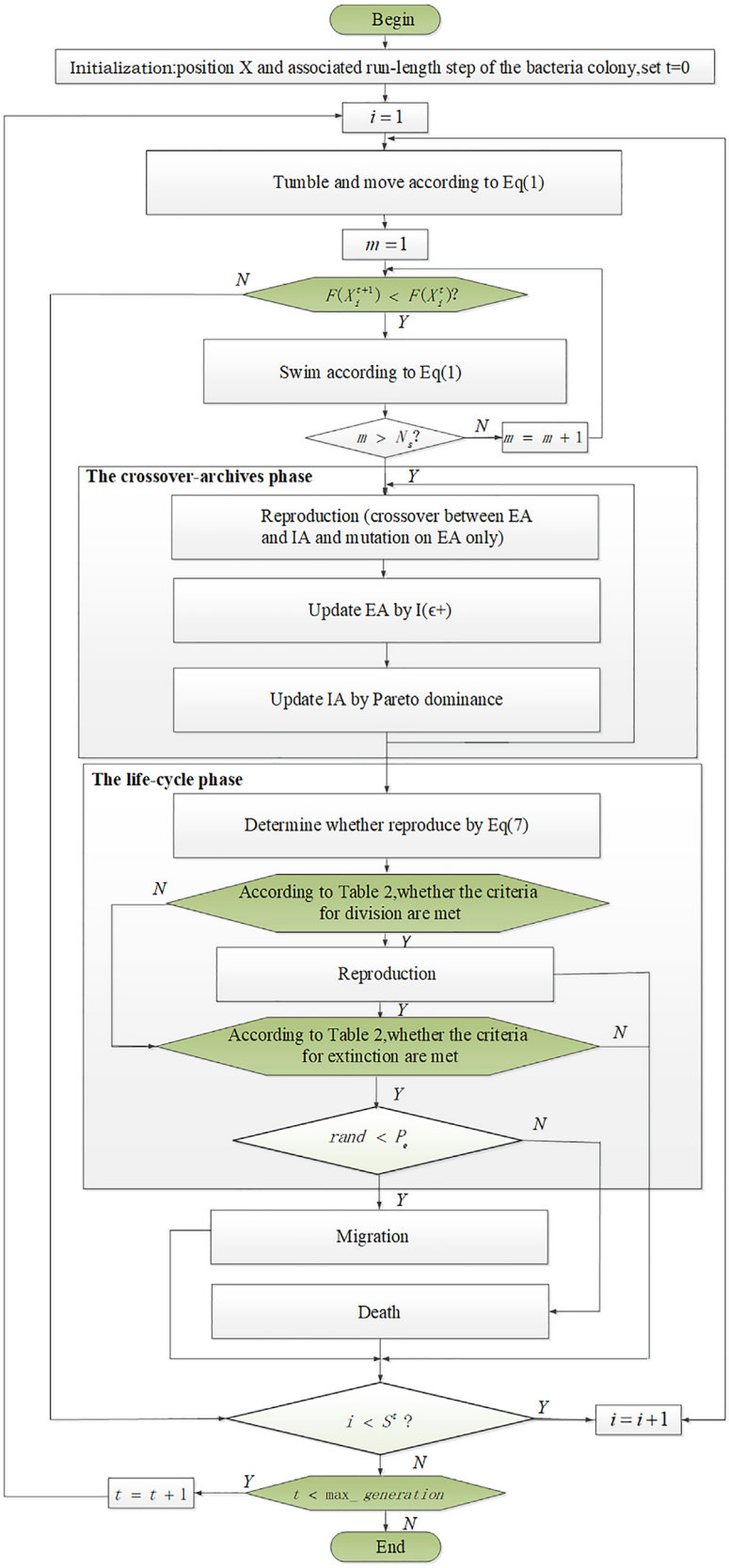

Fig. 2. The flowchart of HMOBFA algorithm.

Table 4
Mean and standard deviation results of IGD obtained by each algorithm on DTLZs (The best items are in bold).

Problem	M	HMBFO	MOEA/D	MOMBI2	VaEA	NSGA-III
DTLZ1	3	**1.4071e-1**	2.5778e-1	1.4709e-1	2.4410e-1	3.0772e-1
		1.5659e-1	3.1235e-1	1.8536e-1	2.6789e-1	2.7954e-1
	5	**3.2200e-1**	**3.200e-1**	3.2771e-1	6.9223e-1	7.2065e-1
		3.0437e-1	**1.8330e-1**	1.8335e-1	5.0824e-1	3.2442e-1
	8	4.4509e + 0	**3.1888e-1**	4.4583e-1	1.8616e + 0	1.1989e + 0
		3.2426e + 0	**3.2559e-1**	1.5333e-1	1.1021e + 0	4.1222e-1
DTLZ2	3	**5.4876e-2**	5.4880e-2	5.8297e-2	5.5361e-2	5.5035e-2
		2.1432e-4	2.3444e-4	1.5330e-3	8.8221e-4	2.3012e-4
	5	2.3842e-1	**2.1328e-1**	2.3833e-1	2.1542e-1	2.1709e-1
		9.6432e-3	**6.3335e-4**	6.0321e-3	2.6867e-3	9.3332e-4
	8	4.2917e-1	**3.8677e-1**	4.9435e-1	4.4337e-1	4.4703e-1
		1.7443e-2	**5.0437e-3**	1.0448e-1	1.4667e-2	8.0853e-2
DTLZ3	3	**7.3543e + 0**	1.8322e + 1	7.8677e + 0	1.1975e + 1	9.4713e + 0
		4.23e + 0	1.4343e + 1	3.6333e + 0	9.3218e + 0	1.7265e + 0
	5	1.9942e + 1	2.1443e + 1	**9.0922e + 0**	2.6953e + 1	2.2926e + 1
		9.8445e + 0	1.5430e + 1	**5.7556e + 0**	7.2642e + 0	6.6877e + 0
	8	1.2122e + 2	1.4351e + 1	**9.0659e + 0**	6.3809e + 1	3.3739e + 1
		3.4662e + 1	8.4440e + 0	**4.7343e + 0**	1.4558e + 1	1.4523e + 1
DTLZ4	3	1.5641e-1	3.4531e-1	1.0837e-1	**5.6028e-2**	2.4969e-1
		1.7872e-1	3.1437e-1	1.5532e-1	**6.7444e-4**	2.5134e-1
	5	2.7193e-1	5.8421e-1	3.5581e-1	**2.1869e-1**	2.4070e-1
		7.7334e-2	1.7559e-1	1.4541e-1	**2.5442e-3**	6.8452e-2
	8	**4.3427e-1**	9.0843e-1	5.0228e-1	5.0228e-1	5.0228e-1
		1.3531e-2	1.3739e-1	1.0449e-1	1.0449e-0	1.0439e-1

Table 5
Mean and standard deviation results of IGD obtained by each algorithm on WFGs (The best items are in bold).

Problm	M	HMBFO	MOEAD	MOMBIII	VaEA	NSGAIII
WFG1	3	**7.7454e-1**	1.1611e + 0	1.1815e + 0	9.6622e-1	1.1552e + 0
		7.5121e-2	1.1831e-1	6.8311e-2	1.1211e-1	9.5233e-2
	5	1.8722e + 0	1.6860e + 0	**1.5524e + 0**	1.6511e + 0	1.7212e + 0
		7.7023e-2	1.3812e-1	**3.8634e-1**	9.4922e-2	9.3011e-2
	8	2.5466e + 0	2.6683e + 0	**2.2024e + 0**	2.3059e + 0	2.4422e + 0
		5.6753e-2	2.3644e-1	**4.8884e-1**	1.0044e-1	1.8412e-1
WFG2	3	2.3276e-1	1.0646e + 0	3.5514e-1	2.3845e-1	**1.8911e-1**
		3.2845e-2	2.0877e-1	1.1312e-2	3.2764e-2	**8.6421e-3**
	5	8.7461e-1	5.5649e + 0	1.0281e + 0	8.0949e-1	**7.5233e-1**
		1.3642e-1	1.8066e-1	5.1632e-2	1.8112e-1	**8.7621e-2**
	8	2.3012e + 0	8.8609e + 0	3.2047e + 0	**1.8403e + 0**	2.4057e + 0
		3.1842e-1	6.0767e-1	7.9443e-1	**3.1577e-1**	5.8012e-1
WFG3	3	1.5266e-1	3.6434e-1	**9.1507e-2**	2.1064e-1	1.4536e-1
		1.2153e-2	1.2612e-1	**5.7965e-3**	8.9543e-2	1.2643e-2
	5	6.4788e-1	1.2716e + 0	**6.0476e-1**	8.9814e-1	8.1403e-1
		6.2465e-2	2.9923e-1	**1.0511e-1**	2.0966e-1	9.7287e-2
	8	**1.2046e + 0**	3.9616e + 0	6.2616e + 0	2.2322e + 0	1.3841e + 0
		1.5533e-1	1.4487e-1	2.6954e-1	9.0845e-1	2.9922e-1
WFG4	3	2.3291e-1	2.9417e-1	2.5834e-1	2.5710e-1	**2.3150e-1**
		3.3942e-3	1.8565e-2	3.6654e-3	1.0243e-2	**2.2511e-3**
	5	**1.1886e + 0**	2.6699e + 0	1.7487e + 0	1.3518e + 0	1.2276e + 0
		5.3043e-3	2.7222e-1	3.66e-2	2.7512e-2	4.6832e-3
	8	**3.2956e + 0**	8.3914e + 0	5.1749e + 0	3.6091e + 0	3.6423e + 0
		2.8154e-2	4.7532e-1	1.3544e + 0	4.0545e-2	1.4334e-1
WFG5	3	2.3967e-1	2.6903e-1	2.7281e-1	2.5807e-1	**2.3724e-1**
		2.6055e-3	6.5212e-3	4.1012e-3	9.5255e-3	**1.4133e-3**
	5	**1.1942e + 0**	2.6283e + 0	1.7806e + 0	1.3033e + 0	1.2058e + 0
		1.2243e-2	1.1411e-1	2.0874e-2	3.5421e-2	3.3322e-3
	8	**3.3528e + 0**	8.1319e + 0	3.8642e + 0	3.5609e + 0	3.5633e + 0
		3.4366–2	2.0711e-1	1.19153e-1	3.9955e-2	2.3521e-2
WFG6	3	2.8491e-1	3.2185e-1	2.8585e-1	3.1534e-1	**2.8277e-1**
		1.85e-2	3.6143e-2	1.1911e-2	1.0444e-2	**1.9212e-2**
	5	**1.2390e + 0**	3.1794e + 0	1.7851e + 0	1.3605e + 0	1.2422e + 0
		2.144e-2	2.5813'e-1	3.6923e-2	3.5354e-2	1.2254e-2
	8	**3.4423e + 0**	8.5397e + 0	4.3109e + 0	3.7680e + 0	3.6233e + 0
		2.7453e-2	9.2711e-2	1.2232e + 0	1.0712e-1	7.9044e-2

than the comparative arithmetic and acquire the best results on the basis of IGD on some measurement living example. These results show evidence to verify the effectiveness of the proposed scheme in HMOBFA.

5. Conclusions

A new Hybrid Multi-Objective Optimized Bacterial Foraging Algorithm (HMOBFA) integrating the crossover-archives strategy

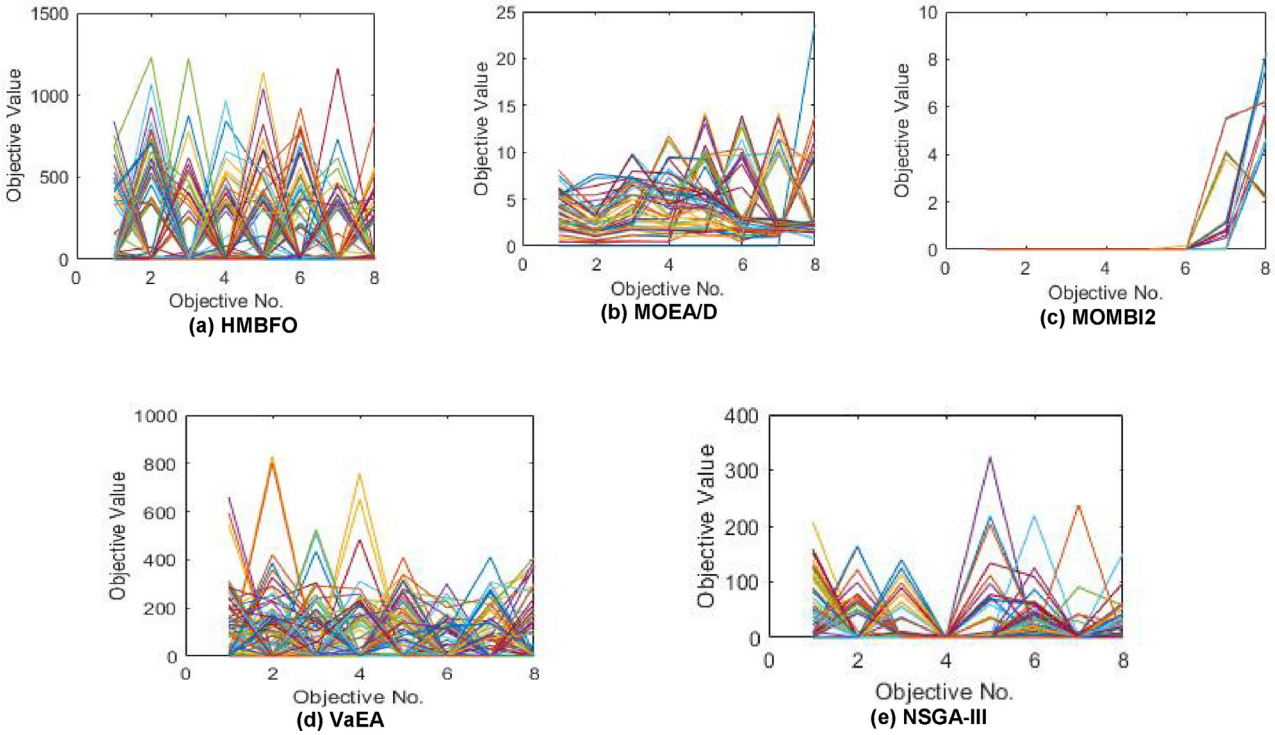

Fig. 3. Final solution set of involved algorithms on the 8-objective DTLZ3, shown by parallel coordinates.

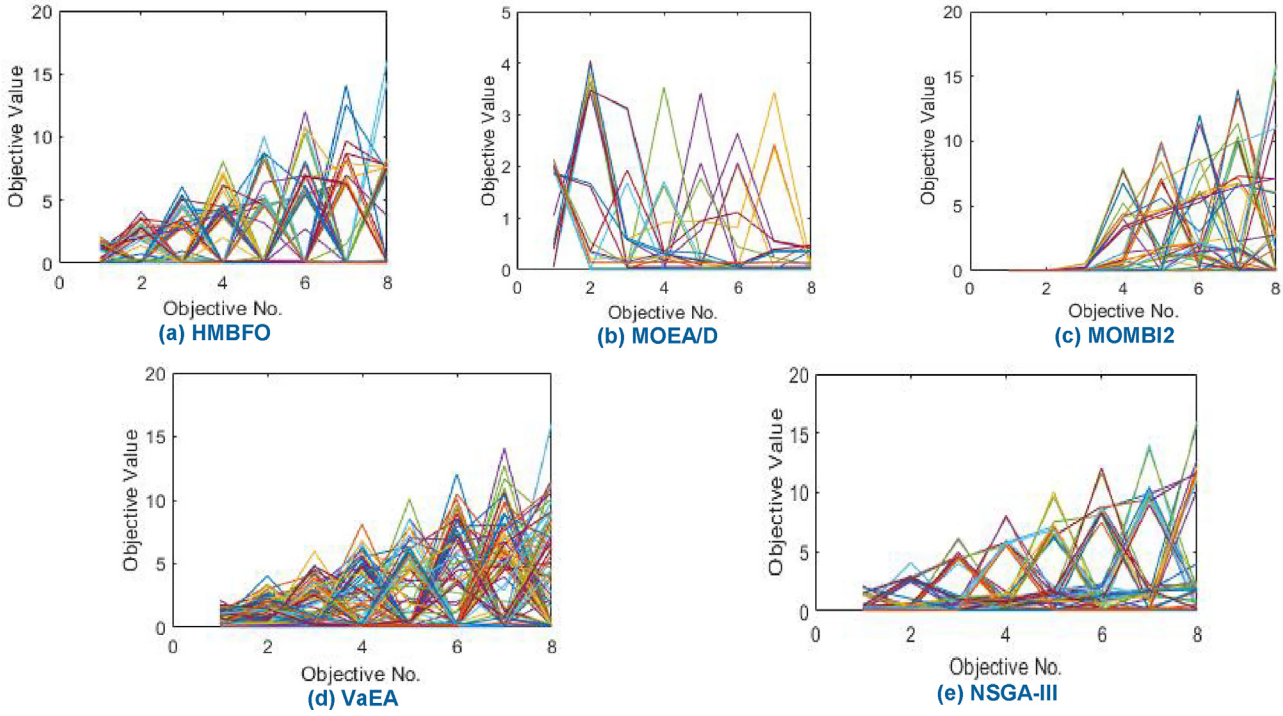

Fig. 4. Final solution set of involved algorithms on the 8-objective WFG4, shown by parallel coordinates.

and the life-cycle optimization strategy as a hybrid strategy is proposed. This combination successfully allows the population size to dynamically adapt to the complicacy of the target function to solve many-objective problems with satisfactory convergence, diversity and complexity. As such, the computational complicacy of the majorization course can be cut downed, and the accuracy and speed of the search can be improved at the same time. The typical high-dimensional test functions for ManyOPs have been implemented to test the HMOBFA algorithm in comparison with the other algorithms, which are all representative of ManyOPs. The simulation results are encouraging: the new algorithms are better than the existing algorithms. We believe that real-world problems should be dealed with using HMOBFA algorithm as an efficient tool. There are still ways to ameliorate our proposed arithmetic. In the future, more complicated and powerful HMOBFA variants should be adopted. The user-defined argument for the HMOBFA

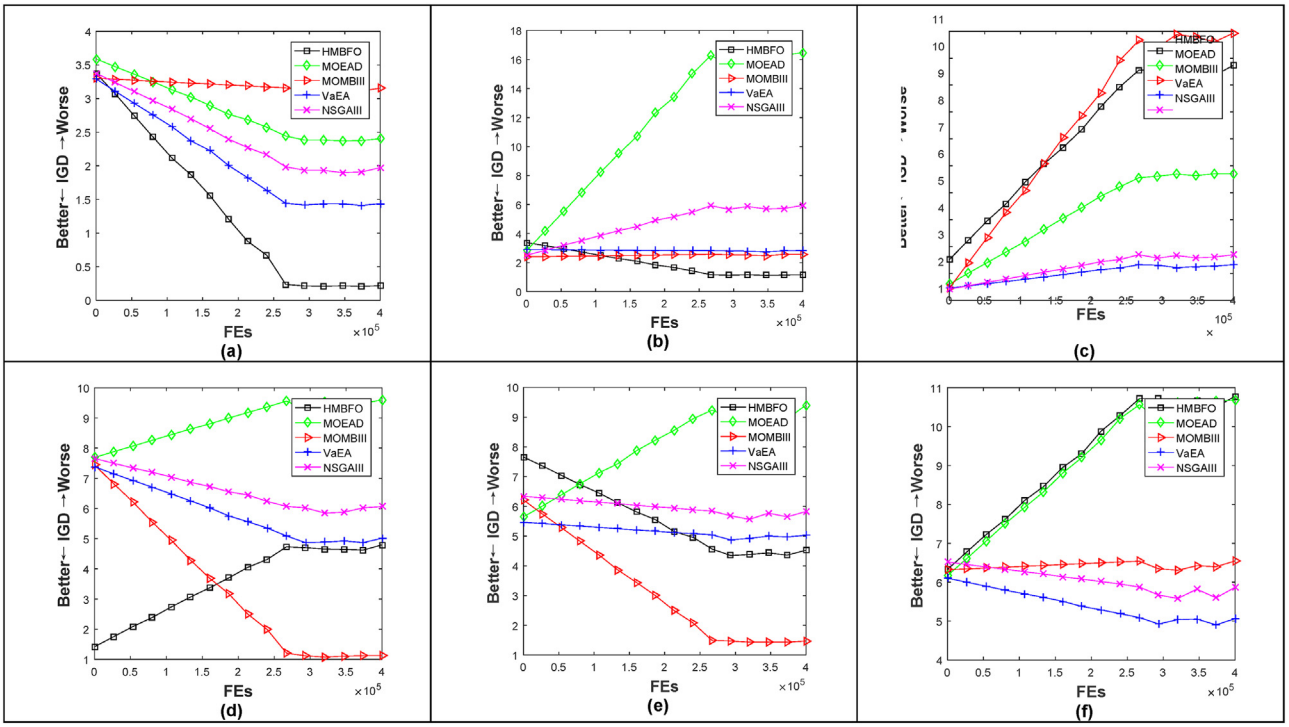

Fig. 5. Evolution of IGD on the 8-objective WFG test problems.

algorithm should be adjusted and optimized. We plan to concentrate on account of the extensive evaluation of many benchmark functions and real-world applications.

Acknowledgement

This work was supported in part by Liaoning Province Innovative Talent Support Program and Liaoning Science and Technology Plan Project.

References

Deb, K., 2001. Multi-Objective Optimization Using Evolutionary Algorithms, vol. 16. Wiley, New York, NY, USA.

Dorigo, M., Maniezzo, V., Colorni, A. The ant system: optimization by a colony of cooperating agents. IEEE Trans. on Systems Man and Cybernetics, Part B, 1996, vol.26, no1, pp.29–41.

Coello, C.A.C., Pulido, G.T., Lechuga, M.S., 2004. Handling multiple objectives with particle swarm optimization. IEEE Trans. Evol. Computat. 8 (3), 256–279.

Alejandra Guzmán, M., Delgado, A., De Carvalho, J., 2010. A novel multiobjective optimization algorithm based on bacterial chemotaxis. Engineering Applications of Artificial Intelligence 23 (3), 292–301.

Tripathy, M., Mishra, S., Lai, L.L. and Zhang, Q.P. Transmission Loss Reduction Based on FACTS and Bacteria Foraging Algorithm. PPSN 2006, pp.222-231,2006.

Deb.K A fast and elitist Multi-objective genetic algorithms. IEEE Transactions in Evolutionary Computation, vol.6, no.2, pp.182-187, 2002.

Jacob, Eshel Ben, Becker, Israela, Shapira, Yoash, Levine, Herbert, 2004. Bacterial linguistic communication and social intelligence. Trends in Microbiology 12 (8), 366–372.

Handing Wang, Licheng Jiao, Xin Yao, 2015. Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization. IEEE Trans. Evol. Computat. 19 (4), 524–541.

Liu, Y., Tian, L.W., Hu, K.Y., Zhu, Y.L., 013. Computational foraging in bacterial colony over composition environments. Journal of Pure and Applied Microbiology 7 (2).

Chen, Hanning, Zhu, Yunlong, Hu, Kunyuan, Ma, Lianbo, 2014. Bacterial colony foraging algorithm: Combining chemotaxis, cell-to-cell communication, and self-adaptive strategy. Information Sciences 273, 73–100.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, Scalable test problems for evolutionary multi-objective optimization, Dept. Comput. Eng. Netw. Lab., ETH Zurich, Zurich, Switzerland, TIK-Tech. Rep. 112, 2001.

Deb, Kalyanmoy, Jain, Himanshu, 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE Trans. Evol. Computat. 18 (4), 577–601.