JOURNAL OF PURE AND APPLIED MICROBIOLOGY, September 2013. Vol. 7(3), p. 2109-2116

Multi-Colony Bacterial Foraging Algorithm for Multi-Objective Optimization

Yichuan Shao[1]* Liwei Tian[2] and Wen jin[1]

[1]College of Information Engineering, Shenyang University, Shenyang - 110 014, China.
[2]Personnel Division, Shenyang University, Shenyang - 110 014, China.

(Received: 15 June 2013; accepted: 30 August 2013)

In this work, a novel approach called multi-objective multi-colony bacterial foraging algorithm for multi-objective optimization (M²BFO) is proposed. The proposed M²BFO extend original bacterial foraging optimization (BFO) algorithm to multi-objective and cooperative mode by combining external archive and cooperative search strategy. Our algorithm uses the concept of Pareto dominance to determine the swim direction of a bacterium and maintains nondominated solution vectors in external archive based on greedy selection and crowing distance strategies. With cooperative search approaches, the single population BFO has been extended to interacting multi-colony model by constructing colony-level interaction topology and information exchange strategies. Simulation experiment of M²BFO on a set of benchmark test functions are compared with other nature inspired techniques which includes nondominated sorting genetic algorithm II (NSGAll) and multi-objective particle swarm optimization (MOPSO). The numerical results demonstrate M²BFO approach is a powerful search and optimization technique for multi-objective optimization problems.

Key words: Multi-colony; BFO algorithm; Multi-objective Optimization; Coorperative coevolution.

In many real-world optimization applications, the decision maker (DM) is always involving simultaneous optimization of several objectives[1]. The solutions for these multi-objective optimization problems (MOP) often result from both the optimization and decision making process. Generally, these objective functions are noncommensurable and often competing and conflicting.

Generally, a solution x_1 of the multi-objective problem is said to be Pareto optimal if there does not exist another solution x_2, such that $f(x_1)$ dominates $f(x_2)$. These optimal solutions are called Pareto-optimal solutions. Multi-objective optimization problems consist of n decision variables, m objective functions, p equality constraints and q inequality constraints. It can be formulated as follows:

$$\text{Minimize } y = f(x) = [f_1(x), f_2(x), \cdots f_m(x)] \quad ...(1)$$

$$\text{Subject to: } \begin{cases} g_i(x) \le 0, & i = 1, 2, \cdots, p \\ h_i(x) = 0, & i = 1, 2, \cdots, q \end{cases} \quad ...(2)$$

where $x = (x_1, x_2, \ldots, x_n) \in D$ is a decision vector that represents a solution, $y = (f_1, f_2, \ldots, f_m)$-$Y$ represent objective functions, D is a n-dimensional search space for decision vectors, and Y is a m-dimensional search space for objective vectors. The set of optimal trade-offs forms the solution set which is called the Pareto set and it is denoted by P^*. The set $PF^* = \{ f(x) \mid x \in P^* \}$ is called the Pareto front.

* To whom all correspondence should be addressed.
E-mail: yichuan_shao@sina.com

MOP provides many challenges to optimization researchers, because the traditional techniques like linear and non-linear programming are unable to solve them efficiently. In recent years, many nature inspired methods were designed to deal with MOP problems. Over the past two decade, a lot of successful multi-objective algorithms based on such biologically inspired algorithms to optimize multi-objective problems were proposed in literature, such as Pareto-archived evolution strategy (PAES) [2], Pareto envelope-based selection algorithm (PESA)-II [3], nondominated sorting genetic algorithm II (NSGAII)[4], strength Pareto evolutionary algorithm (SPEA2)[5], indicator-based evolutionary algorithm (IBEA)[6], multi-objective particle swarm optimization (MOPSO)[7], multi-objective evolutionary algorithm based on Decomposition (MOEA/D)[8], multi-objective differential evolution (MODE) based on summation of normalized objective values. The primary reason for this is their ability to find multiple Pareto-optimal solutions in one single simulation run.

In recent years, chemotaxis (i.e. the bacterial foraging behavior) as a rich source of potential engineering applications and computational model has attracted more and more attentions. A few models have been developed to mimic bacterial foraging behavior and have been applied for solving some practical problems among them, Bacterial Foraging Optimization (BFO) is one population-based numerical optimization algorithm presented by Passino in the literature[9]. BFO is a simple but powerful optimization tool that mimics the foraging behavior of *E. coli* bacteria. Until now, BFO has been applied successfully to some engineering problems[10], such as optimal control, harmonic estimation, transmission loss reduction and machine learning. However, the original version of BFO algorithm is only able to optimize single objective problems.

The purpose of this paper is to develop a multi-colony bacterial foraging algorithm for solving the multi-objective problems. The proposed M^{2}BFO extends the single population BFO to interacting multi-colony model by combing multi-objective handling strategies and hierarchical interaction topologies. In the proposed model, the external archive, greedy selection, and crowding distance strategies are employed to evaluate the fitness of the food source positions and select nondominated solutions. In M^{2}BFO, we implement a hierarchical interaction topology that consists of two levels (i.e. individual level and colony level), in which information exchanges take place permanently. Each bacterium of the multi-colony model searches the food source based on the information integration of its colony members and its cooperative partners from other colonies. Therefore, the advantages of the proposed multi-colony model can be listed as: (1) it can improve the population diversity; (2) it can fasten the convergence speed; (3) it is easy to cooperate in hybrid with another search technique/strategy. In the experiment, we evaluate M^{2}BFO on a set of mathematical benchmark functions that including two two-objective problems and two three-objective cases, which have been widely employed by other researchers to evaluate their MO algorithms.

The remainder of this paper is structured as follows. Sectiona!introduces the original bacterial foraging algorithm. Section III describes the details of the proposed approach for multi-objective optimization. The numerical results of the experiments and discussions are presented in section c°. Finally, Sectiond V concludes the paper.

Bacterial Foraging Optimization

The classical Bacterial Foraging Optimization system consists of three principal mechanisms, namely chemotaxis, reproduction, and elimination-dispersal[9]. We briefly describe each of these processes as follows:

Chemotaxis

In the classical BFO, a unit walk with random direction represents a "tumble" and a unit walk with the same direction in the last step indicates a "run". Suppose $\theta^i(j,k,l)$ represents the bacterium at j^{th} chemotactic, k^{th} reproductive, and l^{th} elimination-dispersal step. $C(i)$ is the chemotactic step size during each run or tumble (run-length unit). Then in each computational chemotactic step, the movement of the i^{th} bacterium can be represented as

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad ...(3)$$

where $\Delta(i)$ is the direction vector of the j^{th} chemotactic step. When the bacterial movement is *run*, $\Delta(i)$ is the same with the last chemotactic step;

otherwise, $\Delta(i)$ is a random vector whose elements lie in [-1, 1].

With the activity of run or tumble taken at each step of the chemotaxis process, a step fitness, denoted as $J(i,j,k,l)$, will be evaluated.

Reproduction

The health status of each bacterium is calculated as the sum of the step fitness during its life, I.e. $\sum_{j=1}^{N_c} J(i,j,k,l)$, where N_c is the maximum step in a chemotaxis process. All bacteria are sorted in reverse order according to health status. In the reproduction step, only the first half of population survives and a surviving bacterium splits into two identical ones, which are then placed in the same locations. Thus, the population of bacteria keeps constant.

Elimination and Dispersal

The chemotaxis provides a basis for local search, and the reproduction process speeds up the convergence which has been simulated by the classical BFO. While to a large extent, only chemotaxis and reproduction are not enough for global optima searching. Since bacteria may get stuck around the initial positions or local optima, it is possible for the diversity of BFO to change either gradually or suddenly to eliminate the accidents of being trapped into the local optima. In BFO, the dispersion event happens after a certain number of reproduction processes. Then some bacteria are chosen, according to a preset probability P_{ed}, to be killed and moved to another position within the environment.

Multi-objective Multi-colony Bacterial Foraging Optimization

External Archive

As opposed to single-objective optimization, multi-objective EA and SI techniques usually maintain a nondominated solutions set. In multi-objective optimization, for the absence of preference information, none of the solutions can be said to be better than the others. Therefore, in MOABC algorithm, we use an external archive to keep a historical record of the nondominated vectors found along the search process[6-8].

In the initialization phase, the external archive will be initialized. After initializing the solutions and calculating the value of every solution, they are sorted based on nondomination. We compare each solution with every other solution in the population to find which one is nondominated solution. We then put all nondominated solutions into external archive *EA*. The external archive will be updated at each generation.

Greedy Selection Mechanism

In our algorithm, each bacterium will find a new solution in each generation. If the new solution dominates the original individual, then the new solution is allowed to enter the external archive. On the other hand, if the new solution is dominated by the original individual, then it is denied access to the external archive. If the new solution and the original bacterium do not dominate each other, then we randomly choose one of them to enter the external archive. That is, after producing new solutions in each generation, the greedy selection mechanism is applied to decide which solution enters *EA*.

Information Transfer Strategy

The whole bacterial community in M^{2}BFO is divided into several colonies, and each

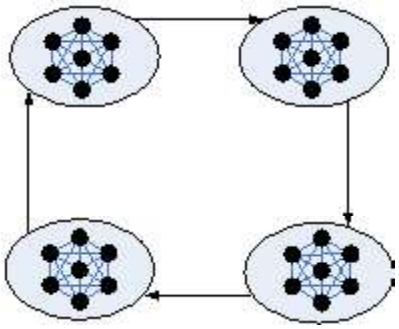

Fig. 1. The sequence of information transfer among colonies

colony performs a canonical paradigm. After some predefined generations of optimization, each colony will select some bacteria with superior information for information transfer. The selected bacteria comprise a list and the list will be sent to another colony. On the other hand, each colony prepares a replacement list comprised of bacteria which will be replaced by bacteria coming from other colonies. The sending list is prepared on the basis of the following rules.

Rule-1

The first consideration in the selection of bacteria is nondomination rank. The bacteria with

the lower rank are preferred.

Rule-2

If the number of bacteria in the first rank is greater than the predefined value of the size of the sending list, the average hamming distance and the crowding distance between each pair of bacteria in the colony will be calculated. The closest L bacteria to the individual that has the largest average hamming distance from others will first enter the sending list and then the bacteria with larger crowding distance will be selected. L is a value that depends on the size of the sending list K and the exchange factor $\delta(0 < \delta < 1)$:

$$L = \delta \times \frac{K}{3} - 1 \quad ...(16)$$

Rule-3

If the number of bacteria in the first rank is less than the predefined value of the size of the sending list, the bacteria in the first rank will first enter the sending list and then the remaining members of the sending list are chosen from subsequent nondominated fronts in the order of their ranking. If the bacteria are in the same rank, we will prefer those with larger crowding distance. This procedure is continued until no more individuals can be accommodated in the sending list.

The replacement list that each colony prepares is based on the nondomination rank and crowding distance in the colony. The replacement list is prepared on the basis of the following rules.

Rule-4: The bacteria in the last rank will be replaced first and then the remaining members of the sending replacement are chosen from previous nondominated fronts in the reverse order of their ranking.

Rule-5

If the bacteria are in the same rank, the bacteria which are located in a lesser crowded region will be replaced first.

Interaction Topology

The sequence of information transfer among colonies is shown in Fig.1, which is a ring sequential order among all colonies. Each colony can accept the sending list from an adjacent colony and the bacteria of its own replacement list will be replaced with the bacteria of the received list.

Obviously, in our multiple colony model, the interaction of bacteria occurred in a two-level hierarchical topology. Many patterns of connection can be used in different levels of our model.

EXPERIMENTAL

Settings and benchmark functions

To fully evaluate the performance of the M^{2}BFO algorithm without a biased conclusion towards some chosen problems, we employed two 2-objective and two 3-objective benchmark functions. The formulas of these functions are presented below.

ZDT1

This is a 30-variables (n=30) problem having a convex Pareto optimal set. The fucntions used are as follows:

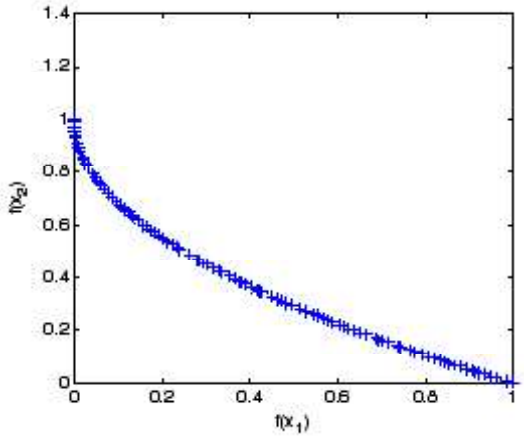

Fig. 2. M^{2}BFO on ZDT1

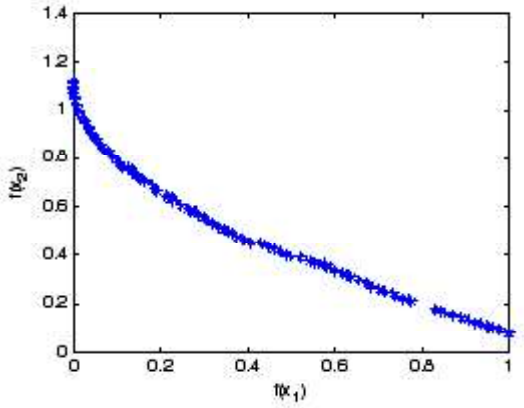

Fig. 3. NSGA-II on ZDT1

$$ZDT1:\begin{cases} Minimize & f_1(x) = x_1 \\ Minimize & f_2(x) = g(x)[1-\sqrt{x_1 / g(x)}] \\ & g(x) = 1+9(\sum_{i=2}^{n} x_i)/(n-1) \end{cases} \quad ...(4)$$

where all variables lie in the range [0, 1]. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for $i = 2,3,\ldots,30$.

ZDT2

This is also an n=30 variable problem having a nonconvex Pareto optimal set:

$$ZDT2:\begin{cases} Minimize & f_1(x) = x_1 \\ Minimize & f_2(x) = g(x)[1-(x_1 / g(x))^2] \\ & g(x) = 1+9(\sum_{i=2}^{n} x_i)/(n-1) \end{cases} \quad ...(5)$$

where all variables lie in the range [0, 1]. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for $i = 2,3,\ldots,30$.

DTLZ2

This test problem has a spherical Pareto-optimal front:

$$DTLZ2:\begin{cases} Minimize & f_1(x) = (1+g(x_M))\cos(x_1\pi/2)\cdots\cos(x_{M-1}\pi/2) \\ Minimize & f_2(x) = (1+g(x_M))\cos(x_1\pi/2)\cdots\sin(x_{M-1}\pi/2) \\ \vdots & \\ Minimize & f_M(x) = (1+g(x_M))\sin(x_1\pi/2) \\ Subject\ to & 0 \le x_i \le 1, \quad for\ i = 1,\ldots,n \\ Where & g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \end{cases} \quad ...(6)$$

where the Pareto-optimal solutions corresponds to $x_i^* = 0.5$ ($x_i^* \in x_M$) and all objective function values must satisfy the $\sum_{m=1}^{M} (f_m^*)^2 = 1$. As in the previous problem, it is recommended to use $k=|x_M|=10$. The total number of variables is $n=M+k$"1 is suggested.

DTLZ6

This test problem has 2^M-1 disconnected Pareto-optimal regions in the search space.

$$DTLZ6:\begin{cases} Minimize & f_1(x) = x_1 \\ Minimize & f_2(x) = x_2 \\ \vdots & \\ Minimize & f_{M-1}(x) = x_{M-1} \\ Minimize & f_M(x) = (1+g(x_M))h(f_1, f_2, \ldots, f_{M-1}, g) \\ Subject\ to & 0 \le x_i \le 1, \quad for\ i = 1,\ldots,n \\ Where & g(x_M) = 1+\frac{g}{|x_M|}\sum_{x_i \in x_M} x_i \\ & h = M - \sum_{i=1}^{M-1}[\frac{f_i}{1+g}(1+\sin(3\pi f_i))] \end{cases} \quad ...(7)$$

where the functional g requires $k=|x_M|$ decision variables and the total number of variables is $n=M+k$-1. It is suggested that k=20.

For the for benchmark functions used in this paper, all the tested algorithm parameters were

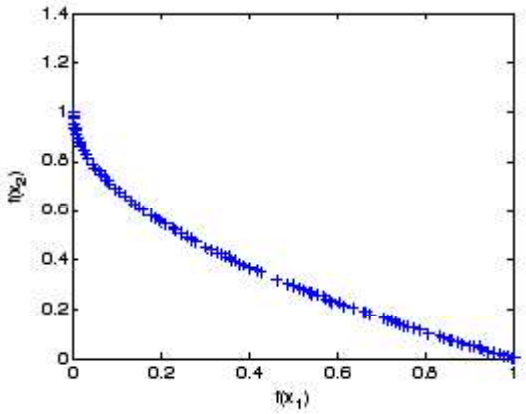

Fig. 4. MOPSO on ZDT1

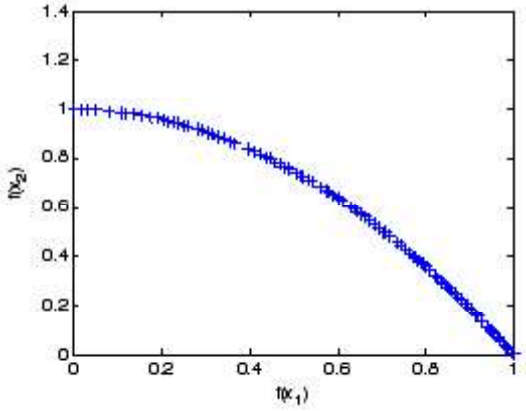

Fig. 5. M²BFO on ZDT2

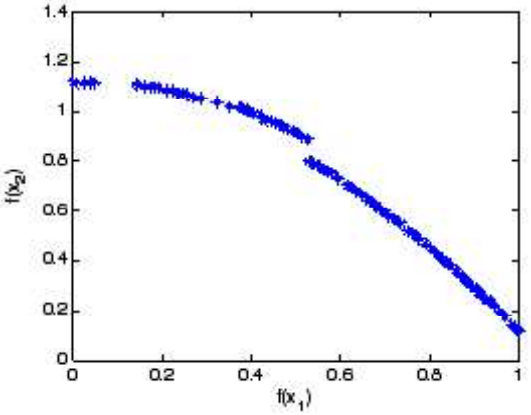

Fig. 6. NSGA-II on ZDT2

set as follows: for M²BFO, the number of colonies is set at 4, the population size of each colony N_i =20 and run step C(i)=0.1. For NSGA-II, the population size, crossover and mutation probabilities are selected as 400, 0.85 and 0.25, respectively for the five benchmark functions. For MOPSO, the population size, mutation rate and divisions for the adaptive grid are selected as 400, 0.5 and 30.

In order to facilitate the quantitative assessment of the performance of a multi-objective optimization algorithm, the convergence metric is taken into consideration.

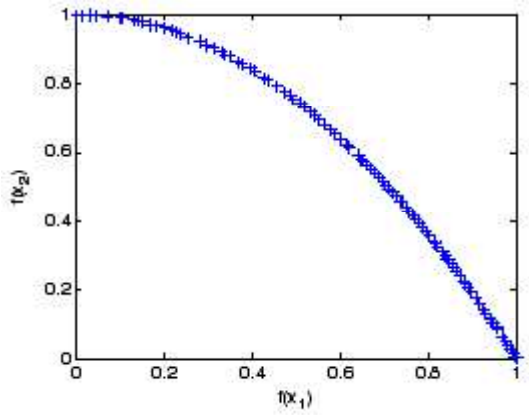

Fig. 7. MOPSO on ZDT2

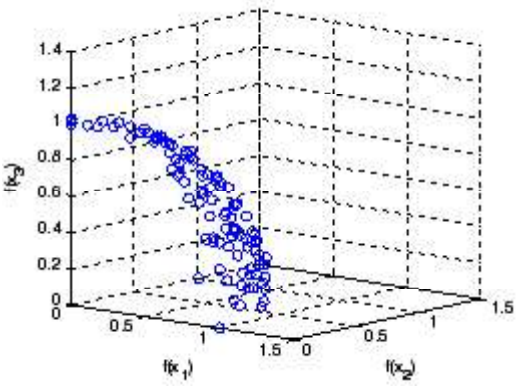

Fig. 8. M²BFO on DTLZ2

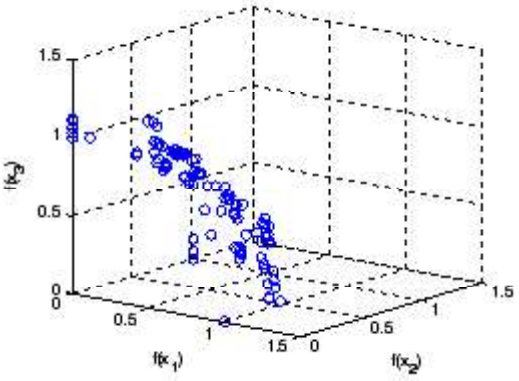

Fig. 9. NSGA-II on DTLZ2

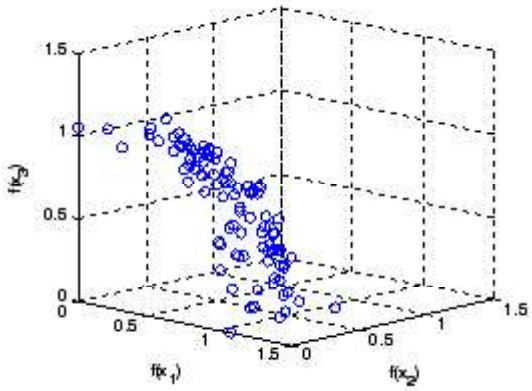

Fig. 10. MOPSO on DTLZ2

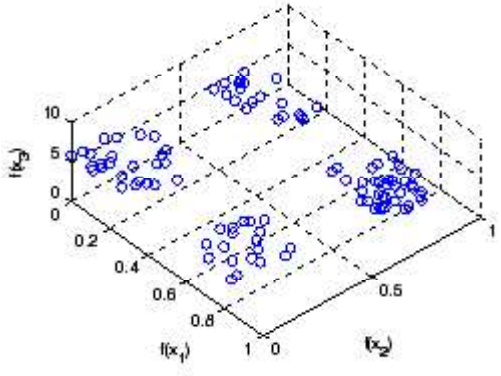

Fig. 11. M²BFO on DTLZ6

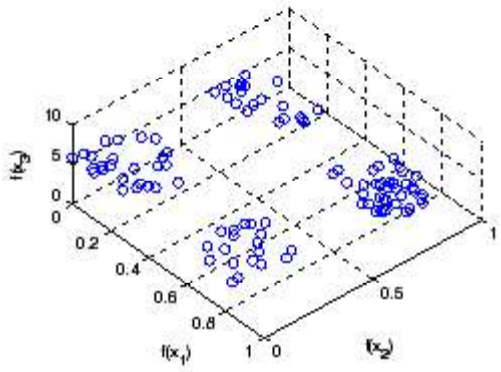

Fig. 12. NSGA-II on DTLZ6

RESULTS AND DISCUSSION

Simulations were conducted on the four benchmark functions to analyze the performances of the algorithms for multi-objective optimization. The results were obtained from thirty independent runs of M²BFO, NSGAII, and MOPSO. For the experiments, the maximal number of fitness function evaluations is 400000.

Figs.2-7 show the optimal front obtained by three algorithms for two objectives problems. These figures show that M²BFO can discover a well-distributed and diverse solution set for both ZDT1 and ZDT2 problems. However, NSGA-II only finds a sparse distribution, and it cannot archive the true Pareto front for ZDT2.

Figs.8-13 show the true Pareto optimal front and the optimal front obtained by three algorithms for DTLZ2 and DTLZ6. From the Figures, it can be seen that the fronts obtained from M²BFO are found to be uniformly distributed. However, NSGA-II algorithm is not able to cover the full Pareto front of DTLZ6.

Table 1. The Comparison Result of The covergence Metric

Function	Item	M²BFO	NSGAII	MOPSO
ZDT1	Avg.	63225e-004	1.4410e-001	2.1433e-002
	Min.	5. 5566e-004	7.2072e-002	1.7786e-002
	Max.	1.1110e-003	8.7348e-001	2. 3079e-002
	Std.	5. 4405e-005	2.3806e-001	2.4557e-003
ZDT2	Avg.	7.3525e-004	9.5537e-004	2.8850e-001
	Min.	3. 4626e-004	7.2183e-004	1. 6806e-001
	Max.	5. 2372e-003	1.5690e-003	7.8182e-001
	Std.	2.4879e-001	2.5380e-004	3.6122e-001
DTLZ2	Avg.	1.2342e-003	6.7337e-003	7. 4199e-002
	Min.	1. 2267e-003	4.9233e-003	5. 3109e-002
	Max.	2. 8106e-003	1.2708e-002	9.1301e-001
	Std.	1. 4501e-004	2.4500e-003	1. 4168e-002
DTLZ6	Avg.	1. 3350e-002	2.7952e-002	2. 6480e-001
	Min.	1. 3121e-002	1.9106e-002	1. 7264e-001
	Max.	2. 7114e-002	3.7997e-002	4. 9147e+000
	Std.	2. 3223e-003	6.7314e-003	8.2636e-001

Table 1 shows the optimization results of M²BFO, MOPSO and NSGA-II algorithms for all multi-objective problems. We can observe from Table 1 that the performances of M²BFO are one order of magnitude better than that of MOPSO and NSGA-II in both two and three objective problems.

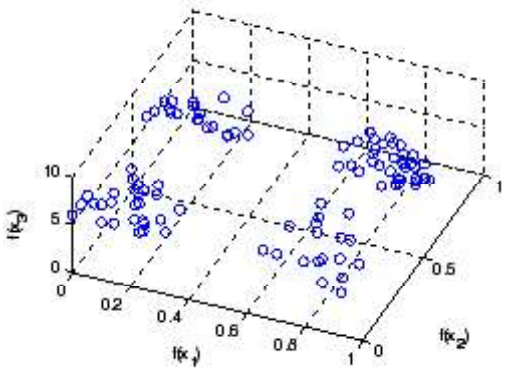

Fig. 13. MOPSO on DTLZ6

CONCLUSION

In this paper, we propose a novel multi-objective bacterial forging algorithm called M²BFO for multi-object optimization. The two novel performance strategies: external archive and information transfer are proposed and used in M²BFO in order to improve the algorithm's performance.

The four benchmark functions have been used to test M²BFO in comparison with NSGA II and MOPSO. It is seen from the comparison that

M^2BFO performs better than the other two algorithms on both two and three objective optimization problems.

ACKNOWLEDGMENTS

This work was supported in part by the International S&T Cooperation Program of China (ISTCP) under Grant 2011DFA91810-5 and Program for New Century Excellent Talents in University of Ministry of Education of China under Grant NCET-12-1012.

REFERENCES

1. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems. Norwell, MA: Kluwer, 2002.
2. J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evol. Comput.,* 2000; **8**(2): 149–172.
3. D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESAII: Region-based selection in evolutionary multiobjective optimization," in Proc. Genetic Evol. Comput. Conf. (GECCO), San Francisco, CA: Morgan Kaufmann, 2001, pp. 283–290.
4. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-a!," *IEEE Trans. Evol. Comput*., 2002; **6**(2): 182–197.
5. E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," in Proc. Eurogen 2001: Evolutionary Methods Design Optimization Control Appl. Ind. Problems, Athens, Greece, 2002; 95–100.
6. E. Zitzler and S. K¨unzli, "Indicator-based selection in multiobjective search," in Proc. Parallel Problem Solving Nature (PPSN), Lecture Notes in Computer Science 3242. Birmingham, U.K.: Springer-Verlag, 2004; 832–842.
7. Carlos A. Coello Coello, G. T. Pulido, and Maximino Salazar Lechuga, "Handling multiple objectives with particle swarm optimization," IEEE Transactions on Evolutionary Computation, 2004; **8**(3): 256-279.
8. Q. Zhang, W. Liu and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in P. of Congress on Evolutionary Computation (CEC 2009), Norway, 2009; 203-208.
9. K. M. Passino, "Biomimicry of bacterial foraging," *IEEE Control Systems Magazine,* 2002; 52–67.
10. W. Gao, S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters,* 2011; **111**: 871-882.