AMSE JOURNALS-2016-Series: Advances A; Vol. 53; N°1; pp 145-159
Submitted July 2016; Revised Oct. 15, 2016, Accepted Dec. 10, 2016

A Co-Evolutionary Particle Swarm Optimization with Dynamic Topology for Solving Multi-Objective Optimization Problems

Daqing Wu[a,b,c,d,e], Lixiang Tang[f*], Haiyan Li[a], LiJun Ouyang[a]
[a]Computer Science and Technology Institute, University of South China, Hangyang, Hunan, China,(dqw_1982@126.com)
[b]Antai College of Economics and Management of Shanghai Jiao Tong University, Shanghai 200240, China
[c] Artificial Intelligence Key Laboratory of Sichuan Province (Sichuan University of Science and Engineering) ,Zigong 643000 ,China
[d] Key Laboratory of Guangxi High Schools for Complex System & Computational Intelligence, Guangxi University for Nationalities, Nanning 530006, China
[e]Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, Hefei, Anhui Province 230039, China
[f]Department of Business Administration, Hunan University of Finance and Economics, Hunan 410205, China (Tanglx0731@126.com)

Abstract

This paper proposes a multi-objective with dynamic topology particle swarm optimization (PSO) algorithm for solving multi-objective problems, named DTPSO. One of the main drawbacks of classical multi-objective particle swarm optimization algorithm is low diversity. To overcome this disadvantage, DTPSO uses two dynamic local best particles to lead the search particles with multiple populations to deal with multiple objectives, and maintains diversity of new found non-dominated solutions via partitioned the searching space into fixed number of cells. The proposed DTPSO is validated through comparisons with other two multi-objective algorithms using established benchmarks and metrics. Simulation results demonstrated that DTPSO shows competitive, if not better, performance as compared to the other algorithms.

Key words

Multi-objective optimization, particle swarm optimizer, two local best solutions, diversity

1. Introduction

Recently, numerous works related to population-based metaheuristic algorithm designs for Multi-objective optimization problem. In a single-objective optimization [1], the best individual corresponds to the one having the lowest/highest fitness of the problem to be minimized/maximized, while using the multi-objective evolutionary algorithm (MOEA) to solve MOPs, the selection of these best individuals' present additional challenges, since two or more conflicting functions must be optimized simultaneously. It is difficult to say whether one individual is better than another if it is better on one objective but is worse on another objective. The reason is related to the fitness assignment problem and the non-dominance concept that applies to two solutions when none of them improves the other in all the objectives, if so then one dominates the other. Therefore, it is important that suitable mechanism have to be considered to assign an individual's fitness.

Now, the MOPSO method is becoming more popular due to its simplicity to implement and its ability to quickly converge to a reasonably acceptable solution for problems in science and engineering. Coello et al [2] proposed a MOPSO method which incorporates Pareto dominance and a special mutation operator to solve MO problems. Zielinski and Laur [3] presented an adaptive approach for parameter setting of an MOPSO. Multi-objective comprehensive learning particle swarm optimizer (MOCLPSO) [4], which extended the CLPSO [5] has also demonstrated competitive performance against several other MOEAs [6].B. B. Li et al[7] proposes a hybrid algorithm based on particle swarm optimization (PSO) for a multi-objective permutation flow shop scheduling problem. A decision support system based on the object-oriented design methodology is described with the multi-objective differential evolution as the core search engine[8],multi-objective particle swarm optimization[9] have been applied to solve economic environmental problem and get a set of solutions efficienently. Peng-YengYin and Jing-YuWang[10]used hybrid particle swarm optimization and adaptive resource bounds technique to optimal multiple-objective resource allocation, Location and allocation decisions for multi-echelon supply chain network was solved by a Multi-objective Hybrid Particle Swarm Optimization (MOHPSO) algorithm approach [11].

In this paper, a novel efficient multi-objective particle swarm optimizer with multiple-populations (DTPSO) is proposed, DTPSO don't handle all the objectives together as a whole in population. Instead, the population was divided into some subswarms, each subswarm being corresponded with one objective separately, and the MOP was solved with multiple subswarm based on the MPSO technique. Design aspects that are incorporated in the proposed DTPSO include the following:

DTPSO uses multiple populations to deal with multiple objectives, taking one objective is optimized by Each swarm into account, so DTPSO avoid the difficulty of fitness assignment. Then, different swarms will cooperate with each other to approximate the whole Pareto front efficiently. DTPSO adopt that a two local best based multi-objective particle swarm optimization algorithm which is integrated with superiority of feasible solution constraint handling method.

The rest of the paper is organized as follows: Section 2 introduces the background information of basic Multi-objective optimization (MO) concepts, Section 3 presents the proposed DTPSO algorithm, Section 4 presents the experimental results, Section 5 of the paper contains the conclusion.

2. Multi-objective optimization

Definition 1 (Multi-objective optimization problem). A multi-objective optimization problem (MOP) can be stated as follows:

$$\text{Minimize} \quad F(x) = (f_1(x), \ldots, f_m(x)), \qquad Subject\ to\ x \in X.$$

(1)

Where $x = (x_1, \ldots, x_n)$ is called decision (variable) vector, $X \subset R^m$ is the decision (variable) space , R^m is the objective space, and $F{:}X \to R^m$ consists of m($m \geq 2$) real-valued objective functions. $F(x)$ is the objective vector. We call problem a MOP (1).

Definition 2 (Pareto optimal).For(1) ,let $a = (a_1, \ldots, a_m), b = (b_1, \ldots, b_m) \in R^m$ be two vectors, a is said to dominate b if $a_i \leq b_i$ for all i=1,...,m, and $a \neq b$.A point $x^* \in X$ is called(globally) Pareto optimal if there is no $x \in X$ such that $F(x)$ dominates $F(x^*)$.Pareto optimal solutions are also called efficient ,non-dominated, and non-inferior solutions. The set of all the Pareto optimal solutions, denoted by PS, is called the Pareto set. The set of all the Pareto objectives vectors, $PF = \{F(x) \in R^m | x \in PS\}$ is called the Pareto front [12]. Illustrative example can be seen in Fig.1.

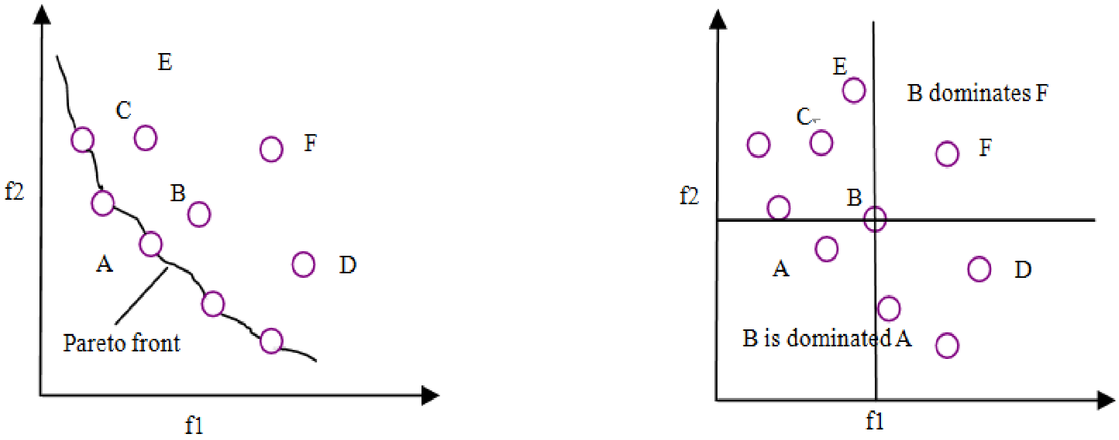


Fig.1. Example of Pareto optimality in objective space and the possible relations of solutions.

3. A multi-objective particle swarm optimizer with dynamic topology

This paper designs a multi-objective particle swarm optimizer with multiple populations to optimize different objective, which named DTPSO.A novel two local bests based multi-objective particle swarm optimizer is proposed to focus the search around small regions in the parameter space in the vicinity of the best existing fronts. The approach emphasizes the elitism at the expense of diversity when the size of the current set of non-dominated solutions in the external archive is small. Therefore, DTPSO is different from many existing multiple-population algorithms. In this section, details of the evolutionary process for each swarm and the information-cooperative mechanisms for all swarm are described. Later, the complete DTPSO process is presented.

A major problem in employing multiple-swarm concept is the need to exchange information to promote diversity among swarms, particularly if no mutation operator is incorporated. In this work, we adopted a two level PSO updating rule wherein the particles learn their local neighborhood experiences. The idea is to further enhance the information sharing among particles by incorporating the concept of neighborhood in the updating PSO equations. The new velocity and position equations are given as follow.

$$v_i^d(t+1) = w \times v_i^d(t) + c_1 \times rand_1^d \times \left(lbest_i^d(t) - x_i^d(t)\right) + c_2 \times rand_2^d \times \left(nbest_i^d(t) - x_i^d(t)\right)$$

(2)

$$x_{id}^m(t+1) = x_{id}^m(t) + v_{id}^m(t)$$

(3)

Where $v_{id}^m(t)$ is the jth dimensional velocity of swarm member i of swarm m in iteration t; $x_{id}^m(t)$ is the jth dimensional position of swarm member i of swarm m in iteration t, the inertial weight w

is randomly varied between 0.1 and 1 to encourage exploration and local search in different iteration counts;and c1, c2, and c3 are the acceleration constants. Firstly, assume that there are M objectives in the MOP, and therefore, there are m swarms working concurrently in DTPSO to optimize the MOP， and then each objective function range in the external archive is divided into a number of cells (n_cells). The lbest and nbest are chosen from the external archive members located in two neighbouring cells, so that they are near each other in the parameter space. In order to select the lbest for a particle, an objective is first randomly selected followed by a random selection of a non empty cell of the chosen objective. Within this cell, the archived member with the lowest front number and among these with the highest crowding distance is selected as the lbest. The nbest is selected from solutions in the neighbourhood non empty cells with the lowest front number and the smallest Euclidean Distance in the parameter space to the lbest. As each particle is guided by lbest and nbest from a neighbourhood in parameter and objective spaces with the smallest front number, the velocity updating of each particle will be in the direction between the positions of lbest and nbest to improve upon the current non-dominated solutions as shown in Figure 2. The proposed DTPSO algorithm updating rule promotes convergence, discovery, and diversity and improves good solutions.

In Fig.2, a two-objective minimization problem with objective1 and objective 2 is shown. Each objective range in the archive is equally divided. The archive has 22 archived solutions in three fronts, among them, archived solutions (AS) a to k are non-dominated front 1 solutions, l to r are front 2 solutions, s to v are front 3 solutions.

The two particles namely P1 to P2 fly in directions guided by their corresponding pbest, Nbest, external archive. For example, when selecting the lbest and Nbest for the particle P2, it is best to randomly select one objective and one cell for its lbest. Here, objective 2 and cell2 are assumed to be selected for lbest of P2. Among the two candidates e,f and g with the lowest front number in cell2, the g is chosen as the lbest as it has the larger crowding distance, The h is selected to be the nbest of P2 as it is nearest to the g in the neighborhood in the parameter space as well as the objective space with the lowest front number. Hence, the P2 will accelerate in direction D2 in the next iteration.

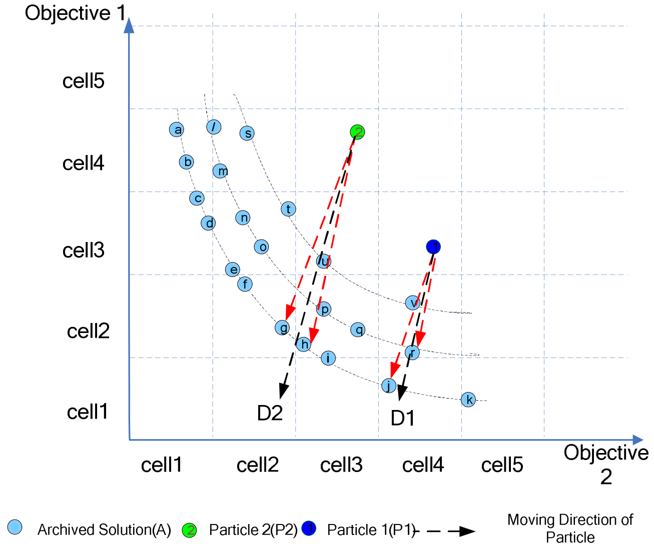


Fig.2. An illustration of Nbest based search process

The procedure of the DTPSO algorithm is shown in Table.1. The infrastructure and basic principle of the DTPSO algorithm is shown in Fig.3. The optimization process starts with the initialization of the different subswarms. After that, the cooperation mechanism is conducted to evaluate the particles in each subswarm.The archive is then updated based on the evaluated solutions.The new set of nondominated solutions in the archive is used as the reference for the calculation of the rank and niche count for each particle.

Table.1. Pseudocode of the DTPSO algorithm

The DTPSO algorithm

//Initialization process

s: population size of each swarm
m: swarms' number
n: objective range in the archive is equally divided into n cells
Max_gen: Max generations, stop criterion
NA:the maximal size of the external archive

Generate initial s*m particles and set parameters for each particle, initialize the position of all particles X,and their fitnesses,and the velocity of all particles V,evaluate all particles and assign lbest and Nbest, the external archive (EA), each objective range in the archive is equally divided into n cells.

//Evolutionary process

While stopping criterion is not met
  For each swarm
    For each Particle
    Step1. Select an archive solution randomly form external archive and assign lbest and nbest for each particle.
    Step 2. Update particle's position and velocity using the equation (2) (3).
    Step 3. Updates the external archive using dominate concepts.
    Step 4. lbest and nbest updating operation.
      End For
  End For
    Using crowding distance function select the solutions if the size of solution is larger than the maximal size of the external archive
End While
END

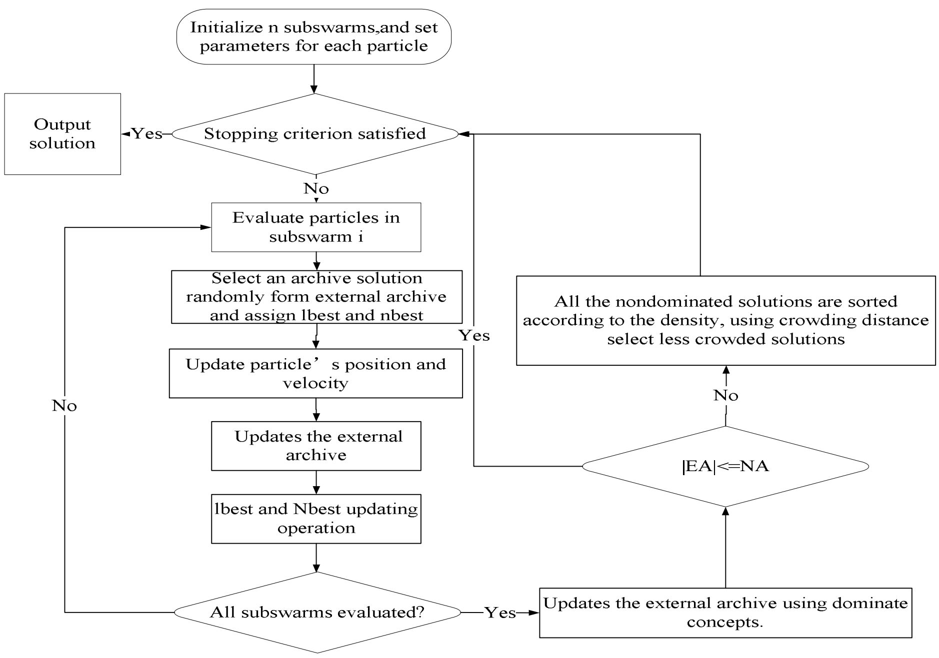


Fig.3. The infrastructure of the DTPSO algorithm

4. Research on numerical experiments

4.1 Testing function

Four benchmark problems, SCH, FON, ZDT1 and ZDT2 are used to examine and compare the performance of DTPSO with other two algorithms. These test functions have different problem characteristics [13], such as multi-modality, convexity, discontinuity and non-uniformity, which may challenge the MOEA's ability to converge and maintain population diversity. The definition of these test functions is summarized in Table.2.

Table.2. The definition of test functions

Problem	Dimension	Variable	Objective function
SCH	1	[-10^3,10^3]	$f_1(x)=x^2, f_2(x)=(x-2)^2$
FON	3	[-4,-4]	$f_1(x)=1-\exp(-\sum_{i=1}^{3}(x_i-1/\sqrt{3})^2)$, $f_2(x)=1-\exp(-\sum_{i=1}^{3}(x_i+1/\sqrt{3})^2)$
ZDT1	30	[0,1]	$f_1(x)=x_1,\ f_2(x)=g(x)[1-\sqrt{x_1/g(x)}]$, $g(x)=1+9(\sum_{i=2}^{n}x_i)/(n-1)$
ZDT2	30	[0,1]	$f_1(x)=x_1,\ f_2(x)=g(x)[1-\sqrt{x_1/g(x)}]$, $g(x)=1+9(\sum_{i=2}^{n}x_i)/(n-1)$

4.2 Performance metrics

In order to facilitate the quantitative assessment of the performance of a multiobjective optimization algorithm, two performance metrics are taken into consideration: (1) convergence metric; (2) diversity metric [14].

(1) Convergence Metric(CM)

This metric measures the extent of convergence to a known set of Pareto optimal solutions, as follows:

$$\lambda=\frac{\sum_{i=1}^{N}d_i}{N}$$

(3)

where N is the number of nondominated solutions obtained with an algorithm and *d* is the Euclidean distance between each of the nondominated solutions and the nearest member of the true Pareto optimal front. To calculate this metric, we find a set of H= 10000 uniformly spaced solutions from the true Pareto optimal front in the objective space. For each solution obtained with an algorithm, we compute the minimum Euclidean distance of it from H chosen solutions on the Pareto optimal front. The average of these distances is used as the convergence metric.

(2) Diversity Metric(DM)

This metric measure the extent of spread achieved among the obtained solutions. Here, we are interested in getting a set of solutions that spans the entire Pareto optimal region. This metric is defined as:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1)\bar{d}}$$

(4)

Where d_i is the Euclidean distance between consecutive solutions in the obtained nondominated set of solutions and N is the number of nondominated solutions obtained with an algorithm. d is the average value of these distances. d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained nondominated set.

4.3 Operation environment and parameter setting

In this paper, we compare the results obtained by DTPSO with NSGA-II [14], multiobjective comprehensive learning PSO (MOCLPSO) [4] These algorithms are chosen because NSGA-II and MOCLPSO are two state-of-the-art algorithms, these algorithms are representative and helpful to make the comparisons more comprehensive and convincing.

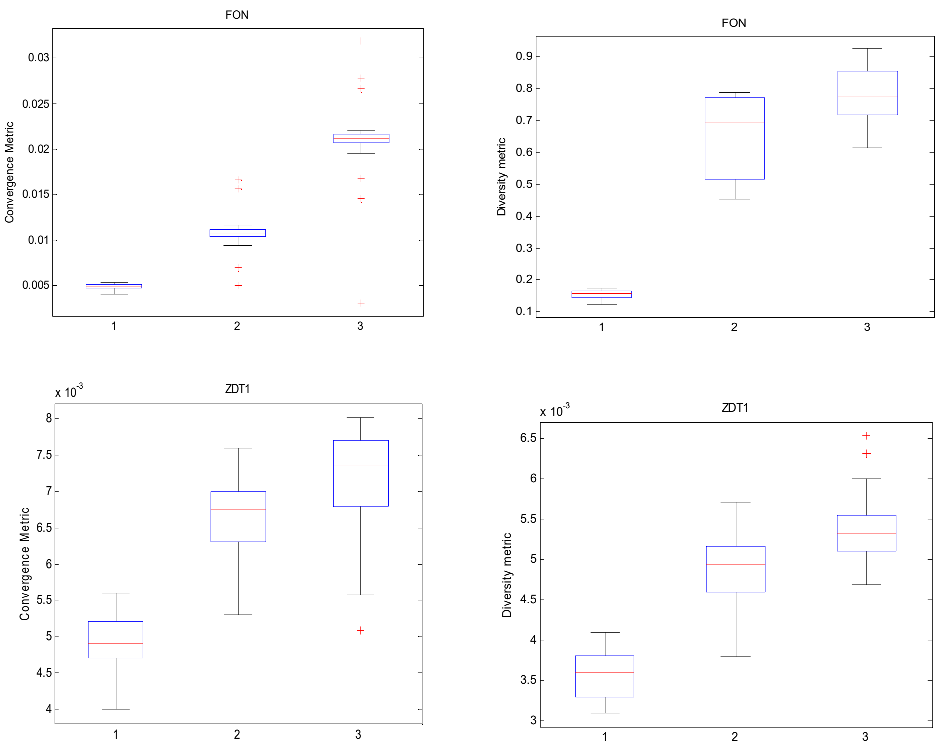
The parameters of the aforementioned algorithms are set according to the proposals in their corresponding references, as summarized in Table.3. In order to make the comparisons fair, all the four algorithms have the same archive size of 100.The maximal number of function evaluations (FEs) is set to be 5000. All the above parameter settings are exactly the same as in the other MOEA variants from the CEC07 Special Session and Competition [15]. Moreover, the experimental results are the average values of 30 independent runs. The best results are denoted by the bold font.

Table.3. Parameter settings of the algorithms

Algorithms	Parameters Settings
NSGA-II	N=100, Pc=0.9,Pm=1/D, η_c=20， and η_m=20
MOCLPSO	N=50,Pc=0.1,Pm=0.4, $\omega=0.9\rightarrow0.2$,c=2.0
DTPSO	N=100, $\omega=0.729, c_1=c_2=c_3=2.05$,cell=10

4.4 Influence of the parameter changes on the algorithm

The Nbest is chosen from the external archive members located in two neighboring cells, the neighboring cells are near each other in the parameter space. So each function range in the external archive is divided into a number of cells (n_cells). Fig.4 shows the performance of DTPSO over different settings Sn-cell = {10; 8; 4}. The size of the cells was varied while maintaining the total number of evaluations. From the box-plots, it is apparent that bigger cells sizes give rise to better convergence to the true Pareto front. Nonetheless, we note that lower values of DM denoting better diversity are achieved at higher Sn-cell settings in the case of FON, ZDT1. This is because, by maintaining a fixed number of evaluations, there is an inherent tradeoff between the diversity provided by a larger population size and the number of generations allowed for exploration. Thus, we note that cell size of 10 is sufficient to produce good results.



(a) (b)

Fig.4. Algorithm performance in (a) CM for FON and ZDT1 (1st row), (b) DM for FON and ZDT1 (2nd row)

4.5 Experimental results and analysis

The experimental results, including the best, median, worst, mean, and standard deviation of the convergence metric and diversity metric values found in 30 runs are proposed in Table. 4,5, 6 and 7, all algorithms are terminated after 10000 function evaluations, respectively. Fig.5, 6, 7 and 8 show the optimal front obtained by four algorithms for two-objective problems.

When given 10000 function evaluations for them, DTPSO and MOCLPSO algorithms improve converge metric. DTPSO get better value than other algorithms in diversity metric. From Fig. 5, it can be seen that the front obtained from DTPSO, and MOABC are found to be uniformly distributed. However, MOCLPSO NSGA-II algorithm not able to cover the full Pareto front. On the whole the DTPSO and MOABC algorithms are much better than MOCLPSO and NSGA-II on SCH problem.

Table.4. Comparison of performance on SCH

SCH		DTPSO	MOCLPSO	NSGA-II
Converge metric	Best	**1.13 e-004**	**1.14e-004**	6.29e-004
	Median	1.46 e-004	4.63e-004	1.81e-004
	Worst	2.57 e-004	7.58 e-004	**2.30e-004**
	Mean	**1.14e-004**	4.36e-004	1.81e-004
	Std	4.41e-005	**2.19 e-004**	3.32e-005
Diversity metric	Best	**2.79 e-001**	7.53e-001	7.28e-001
	Median	**2.39 e-001**	8.03e-001	7.57e-001
	Worst	**3.24e-001**	8.82 e-001	7.77e-001
	Mean	**2.42 e-001**	8.14e-001	7.56e-001
	Std	**1.47e-002**	5.33e-002	1.51e-002

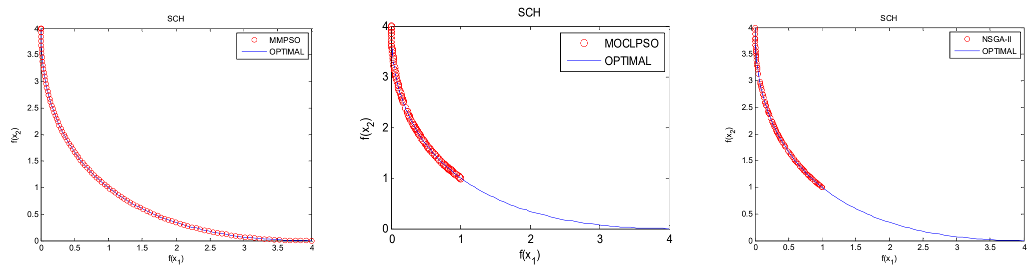


Fig.5.Pareto fronts obtained by DTPSO, MOABC, MOCLPSO, and NSGA-II on SCH

For FON problem, it can be observed from Table.5 that all algorithms perform very well in convergence metric. In diversity metric aspect, DTPSO, MOABC, and MOCLPSO algorithms can guarantee a good performance. On the other hand, even though NSGA-II is able to find the true Pareto front for this problem, it cannot maintain a relatively good diversity metric. It is only able to cover the half Pareto front, supporting the results of Fig.6.

Table.5. Comparison of performance on FON

FON		DTPSO	MOCLPSO	NSGA-II
Converge metric	Best	**1.79 e-003**	2.27 e-003	4.72e-003
	Median	**1.92 e-003**	2.56 e-003	1.99e-003

	Worst	**2.31 e-003**	2.86 e-003	3.12e-003
	Mean	**1.45 e-003**	2.54e-003	1.85e-003
	Std	**1.13e-004**	1.65e-004	9.59e-003
Diversity metric	Best	2.51 e-001	**2.45e-001**	6.90e-001
	Median	**2.65e-001**	2.94e-001	8.35e-001
	Worst	**2.85 e-001**	3.37e-001	9.25e-001
	Mean	**2.65e-001**	2.96e-001	7.96e-001
	Std	2.98e-004	2.35e-002	9.10e-002

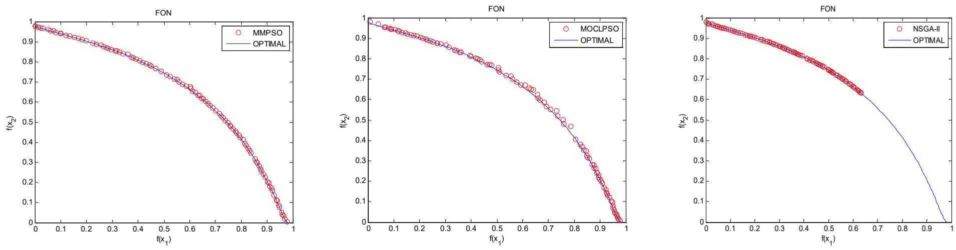


Fig.6.Pareto fronts obtained by DTPSO, MOABC, MOCLPSO, and NSGA-II on FON

On ZDT1 function, for convergence metric, one can note from Table.6 that DTPSO outperform other three algorithms. the performance of DTPSO in diversity metric is two orders of magnitude better than that of MOABC, MOCLPSO and NSGA-II. Fig.7 shows that DTPSO, MOABC and MOCLPSO can discover a well-distributed and diverse solution set for this problem. However, NSGA-II only finds a sparse distribution, and they cannot archive the true Pareto front for ZDT1.

Table.6. Comparison of performance on ZDT1

ZDT1		DTPSO	MOCLPSO	NSGA-II
Converge metric	Best	4.17 e-003	**1.72e-003**	7.20e-002
	Median	**5.23e-003**	2.41e-003	1.44e-001
	Worst	**5.60 e-003**	3.10e-003	8.73e-001
	Mean	3.01 e-003	**2.41e-003**	2.22e-001
	Std	**2.85e-004**	4.90e-004	2.38e-001
Diversity metric	Best	**3.10e-003**	2.64e-001	4.58e-001
	Median	**3.71 e-003**	2.92e-001	5.26e-001
	Worst	**5.56e-003**	3.50e-001	9.14e-001
	Mean	**4.64 e-003**	2.97e-001	5.93e-001
	Std	**3.86e-004**	2.45e-002	1.56e-001

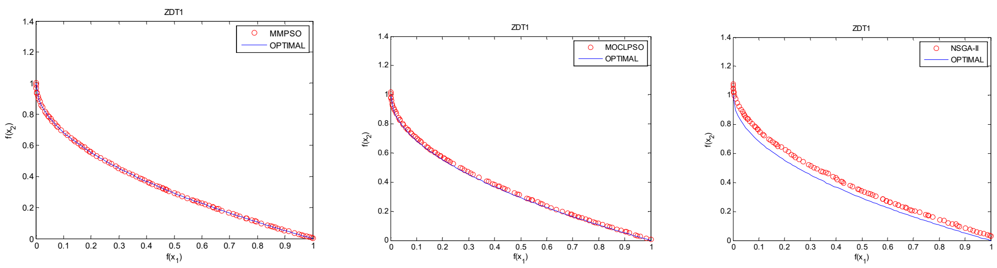


Fig.7.Pareto fronts obtained by DTPSO, MOABC, MOCLPSO, and NSGA-II on ZDT1

On ZDT2 function, from the Table.7, the results of the performance measures show that DTPSO and MOCLPSO have better convergence and diversity compared to the MOABC and NSGA-II. One can note that the performance of DTPSO in convergence metric is three orders of magnitude better than that of NSGA-II. Fig.8 shows that NSGA-II produces poor results on this test function and it cannot achieve the true Pareto front.

Table.7. Comparison of performance on ZDT2

ZDT2		DTPSO	MOCLPSO	NSGA-II
Converge metric	Best	**3.51e-004**	3.38e-004	1.06e-001
	Median	3.90e-004	**1.03e-003**	1.81e-001
	Worst	**4.32e-004**	1.42e-003	9.82e-001
	Mean	**4.21e-004**	9.73e-004	2.94e-001
	Std	**3.44e-004**	3.73e-004	2.62e-001
Diversity metric	Best	**3.11e-003**	2.67e-001	4.66e-001
	Median	**3.74e -003**	3.16e-001	7.29e-001
	Worst	**4.41e-003**	3.50e-001	1.05e-000
	Mean	**3.70 e-003**	3.17e-001	7.63e-001
	Std	**2.67e-004**	2.61e-002	2.35e-001

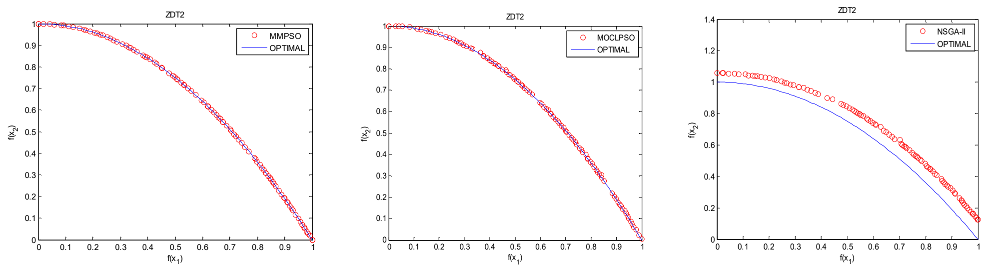


Fig.8.Pareto fronts obtained by DTPSO, MOABC, MOCLPSO, and NSGA-II on ZDT2

5. Conclusion

This paper proposes a novel coevolutionary particle swarm optimizer with multiple populations named DTPSO, for dealing with multi-objective problems. DTPSO uses multiple populations to deal with multiple objectives, and maintains diversity of new found non-dominated solutions via adopted a three-level PSO updating rule wherein the particles learn their experiences based on personal, neighborhood, and external archive to help particles explore more areas in the solution space, and comcelles advantages of wide-ranged exploration and extensive exploitations of PSO in the external repository with the improved jump strategy to enhance the solution searching abilities of particles. Seven test functions were adopted for testing through a reasonable average and the results are very authoritative. The experimental results proved that the proposed method can find better solutions when compared to other approaches.Simulation results obtained from the proposed approach have been compared with those from previous methods.

The comparison shows that DTPSO provides a competitive performance demonstrate that the proposed DTPSO outperforms the other techniques.Since DTPSO doesn't impose any limitation on objectives, it can be extended to more objectives problems.

Acknowledgments

The authors would like to thank all the reviewers for their constructive comments. The authors thank Mr. Suganthan for his thoughtful help and constructive suggestion. This research was supported by the Subject of Ministry of Education of HuNan province (No. 2016NK2135). Open Fund of Key Laboratory of Guangxi High Schools for Complex System &Computational Intelligence (No.15CI07Y).The open project program of artificial intelligence key laboratory of Sichuan province (No.2015RYJ01). The fund for National Statistical Science Research Project (No. 2015044),and the PAPD fund.

References

1. Daqing Wu, Jianguo Zheng.(2012).A Dynamic Multistage Hybrid Swarm Intelligence Optimization Algorithm for Function Optimization, Discrete Dynamics in Nature and Society,2012,Article ID 578064,22 pages,2012.doi:10.1155/2012/578064.
2. Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. Evolutionary Computation, IEEE Transactions on, 8(3):256-279.
3. Zielinski, K. and Laur, R.,(2007). Adaptive parameter setting for a multi-objective particle swarm optimization algorithm. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (pp. 3019-3026).
4. Huang, V. L., Suganthan, P. N., & Liang, J. J. (2006). Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. International Journal of Intelligent Systems, 21(2), 209-226.
5. Liang J J, Qin A K, Suganthan P N, et al. (2006).Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. Evolutionary Computation, IEEE Transactions on, 10(3): 281-295.
6. Knowles, J.D. and Corne, D.W.(2000). Approximating the nondominated front using the Pareto archive evolutionary strategy. Evolutionary Computation, 8 (2):149–172.

7. B. B. Li , L. Wang and B. Liu.(2008)An effective PSO-based hybrid algorithm for multiobjective permutation flow shop scheduling", IEEE Trans. Syst., Man, Cybern. A, Syst. Humans, 38(4):818 -831 2008
8. Xue, F., Sanderson, A. C., & Graves, R. J. (2009). Multiobjective evolutionary decision support for design–supplier–manufacturing planning. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 39(2):309-320.
9. Zhang, Y., Gong, D. W., & Ding, Z. (2012). A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. Information Sciences, 192:213-227.
10. Yin, P. Y., & Wang, J. Y. (2008). Optimal multiple-objective resource allocation using hybrid particle swarm optimization and adaptive resource bounds technique. Journal of Computational and Applied Mathematics, 216(1):73-86.
11. Latha Shankar, B., Basavarajappa, S., Chen, J. C., & Kadadevaramath, R. S. (2012). Location and allocation decisions for multi-echelon supply chain network-A multi-objective evolutionary approach. Expert Systems with Applications. 2013(40):551-562
12. H. Li and Q. Zhang, (2009).Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," IEEE Transactions on Evolutionary Computation, 13(2):284–302.
13. K. Deb. (1999)Multi-objective genetic algorithms: Problem difficulties and construction of test problem, Evolutionary Computation. 7 (3):205-230.
14. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182-197.
15. Huang,V.L., Qin, A.K., Deb, K., Zitzler, E., Suganthan, P.N., Liang, J.J ., Preuss, M. and Huband, S., 2007. Problem definitions for performance assessment on multi-objective optimization algorithms [online]. Technical Report. Nanyang Technological University, Singapore.