Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 91 (2016) 1001 – 1010

Procedia Computer Science

Information Technology and Quantitative Management (ITQM 2016)

A novel Ranking-based Optimal Guides Selection Strategy in MOPSO

Li Li[a], Wanliang Wang*[a], Weikun Li[a], Xinli Xu[a], Yanwei Zhao[b]

[a] *College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, 310023, P.R.China*
[b] *College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, 310023, P.R.China*

Abstract

A challenging issue with multi-objective particle swarm optimization (MOPSO) is the mechanism to select the optimal guides. This paper presents a new strategy based on ranking dominance and integrates into MOPSO. By using the ranking information and incorporating the chebychev distance of particle in objective space, we implement the selection of gbest and pbest simply and elegantly. On the basis of ranking, we propose a new maintenance strategy for updating the external archive which can obtain a more diverse and uniform distribution. Furthermore, a qualitative and quantitative analysis in terms of convergence analysis over some benchmarks is presented, providing a basis for conclusions about the proposed method. showing that the proposed method performs better than the adopted algorithms.

© 2016 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the Organizing Committee of ITQM 2016

Keywords: Ranking-based Optimal, Guides Selection, MOPSO

1. Introduction

Particle swarm optimization algorithm [1] is a kind of swarm-based intelligence optimization method which simulates the behavior of birds flocking. The particles move in a given objective space searching for the promising solutions. Moreover, the particles communicate each other to optimize by using their previous best (*pBest*) and global best (*gBest*) information in swarm. PSO is an efficient algorithm for solving single objective problems [2][3][4] and achieving approving results. However, in the numerical optimization problems and practical engineering applications, there always exist multiple objectives that need to be met at the same time, which are too complex to simply apply the PSO to solve.

Recently, Multi-objective evolutionary algorithm (MOEA) has appeared to resolve such problems with desired results [5] [6]. It does not depend on the problem itself. By using randomized initial individuals in the solution space, the MOEA can find an ideal solution set. But the simple extension of the PSO to the multi-objective optimization problem will face some difficulties:

*Corresponding author. Tel.: +86-571-85290172.
E-mail address: wwl@zjut.edu.cn

1877-0509 © 2016 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ITQM 2016
doi:10.1016/j.procs.2016.07.135

a. Relaxed dominance. By enlarging the dominant space to improve the weakness of Pareto dominance, and it was claimed by Laumanns *et al.* [7]. To improve the effect of ϵ-dominance, Batista *et al.* [8] raised cone ϵ-dominance. Also, Zou *et.al* [9] proposed L-dominance which is a variation from Pareto dominance.
b. Non-Pareto dominance. Such as Weighted sum [10], weighted Tchebycheff [11], decomposition-based approach [12], Average ranking [13], Global Detriment [14] etc.

As for the optimal guides selection, there are several proposals in the literature for $gBest$ and $pBest$ selection. By using the roulette-wheel selection method, which is a random selection, Coello *et al.* [15] presents the MOPSO to choose the best local guide. Yen *et al.* [16] raised an adaptive cell-based technique guides selection. Mostaghim [17] uses the sigma for the selection of $gBest$. For a bi-objective problem, sigma method can be drawn as follows:

$$\sigma = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2} \tag{1}$$

This paper uses the ranking-based approach, by incorporating the distribution information of individuals in objective space, we put forward a new optimal guides selection strategy and apply it to the MOPSO. Different from the aforementioned complex $gBest$ $pBest$ selection, this paper uses the ranking strategy and density information of each particle in the whole population to gain the $gBest$ and $pBest$ which is elegant and efficient.

The paper is structured as follows. In section 2, some related works including MOPSO and ranking-based methods are briefly introduced. Section 3 proposes and details the proposed ranking method. Section 4 shows the performance assessment and statistical results among some classic MOEA and MOPSOs. Finally, section 5 draws the conclusion and proposes the future work.

2. Related work

2.1. PSO algorithm

In PSO, each particle represents a solution in objective space. Two best guides exist in it, namely, the historical best solution discovered by the particle itself ($pBest$) and the historical best solution identified by the entire population ($gBest$). These two optimal guides are tracked and updated by the particles accordingly. The updating mechanism is described as Eq. (2) and Eq. (3):

$$v_i(k+1) = \omega v_i(k) + c_1 r_1 [P_i - x_i(k)] + c_2 r_2 [P_g - x_i(k)] \tag{2}$$

$$x_i(k+1) = v_i(k+1) + x_i(k) \tag{3}$$

where, $i = (1, 2, 3, \cdots N)$ is the index of particle, N is the size of population; $v_i(k)$ represents the velocity of particle i in k-th iteration, similarly, $x_i(k)$ is the position of particle i in k-th iteration. w is the inertia weight and the constants $c1, c2$ are the acceleration coefficients, $r1$ and $r2$ are random values uniformly distributing in interval [0,1]; P_i is the personal best performance found so far by particle i while P_g is the global best position discovered in the swarm.

2.2. Multi-objective optimization problems

Without loss of generality, a general minimized multi-objective problem can be stated as follows:

$$\min F_i(X) = (f_1(X_i), f_2(X_i), \cdots, f_M(X_i))^T \tag{4}$$

$$\begin{aligned} &X = (x_1, \cdots x_n) \\ &Y = F(X) \\ &s.t.x = (x_1, \cdots x_n) \in X \subset R^n \end{aligned}$$

where the decision vector x claims to the decision space X, the objective function vector $F(X)$ consists $M(M \geq 2)$ objectives, $Y \subset \mathbb{R}^M$ represents the objective space, and $f : R^n \rightarrow R^M$ is the objective mapping function.

2.3. Ranking-based methods

The existing ranking-based methods mainly utilize a ranking function to gain the sort information in solution space so as to distinguish the merits of the individuals. Ranking-based methods have the advantage in some ways, they do not rely on Pareto dominance to push the population toward the Pareto front. In addition, as pinpointed by Mendez *et.al* [18], these ranking-based methods have good results in less time compared to some Pareto-based approaches. The existing methods mainly include the following.

Maximin: The maximin fitness function was proposed by Richard Balling and Scott Wilson in [19], where the maximin fitness of design i is defined to be:

$$fitness^i = \max_{j \neq 1} (\min_k (f_k^i - f_k^j)) \tag{5}$$

Adriana *et.al* [18] analyzed the merits and faults on maximin and carried out some improvements for it.

Average Ranking (AR): This method was proposed by Bentley and Wakefield [13]. AR selects one objective and builds a ranking list using the fitness of each solution for such objective. The AR is given as follows:

$$AR(X_i) = \sum_{m=1}^{M} r_m(X_i) \tag{6}$$

where $r_m(X_i)$ is the rank of X_i for the m-*th* objective. M is the number of objectives. It can be seen from the definition of AR, it is easy to make individuals concentrated in a particular region especially the region close to the objective.

Global Detriment (GD) On the basis of analyzing the deficiency of Pareto-based method, Garza *et al.* [14] put forward Global Detriment and successfully applied to high-dimensional problems with a satisfactory results. The GD fitness of a solution X_i is calculated as follows:

$$GD(X_i) = \sum_{X_j \neq X_i} \sum_{m=1}^{M} \max(f_m(X_i) - f_m(X_j), 0) \tag{7}$$

A solution X_i is better than X_j if it holds that $GD(X_i) < GD(X_j)$.

Particularly, the GD method can get approving results to some extent. However, it can be seen from the definition that it will assign large ranking value for the extreme points to be eliminated accordingly, whereas these extreme points are a significant reference for decision-maker [20], as well as the aspect of the distribution.

3. The proposed Algorithm

3.1. Fusion ranking approach

Since the Average Ranking is easy to aggregate on the edge of optimal front while Global Detriment is easy to exclude the extreme points, this paper integrates the strengths and weaknesses of AR and GD, and proposes a new ranking scheme called fusion ranking which is described as follows.

$$FR(X_i) = \omega_1 AR(X_i) + \omega_2 GD(X_i) \tag{8}$$

where, ω_1 and ω_2 are two coefficients in interval of [0,1] to coordinate the weight on AR and GD. We adopt 0.4 and 0.6 specifically. The lower of FR, the better the domination. Obviously, FR integrates the sequence of solutions which favor the objective with the relative information between individuals. So that individuals in the solution space can be more comprehensively assessed to guide the algorithm to converge.

The following is a small example to illustrate this case. For an optimization problem with two objectives, Fig. 1 shows the six individuals from $p1 - p6$ and Table **??** lists the ranking values obtained by AR, GD, FR respectively.

As can be seen from the Fig.1, AR gains the same two sort values and $\{p1, p2, p6\}$ are three top points, whereas GD put the $\{p1, p2, p3, p4\}$ into the front seats. However, FR combines both advantages to obtain a relatively satisfactory sort.

Table 1. Ranking values obtained by AR, GD and FR

	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$
AR	6	4	7	7	10	6
GD	2	0.4	1.2	2.3	3.5	2.7
FR	3.6	1.84	3.52	4.18	6.1	4.02

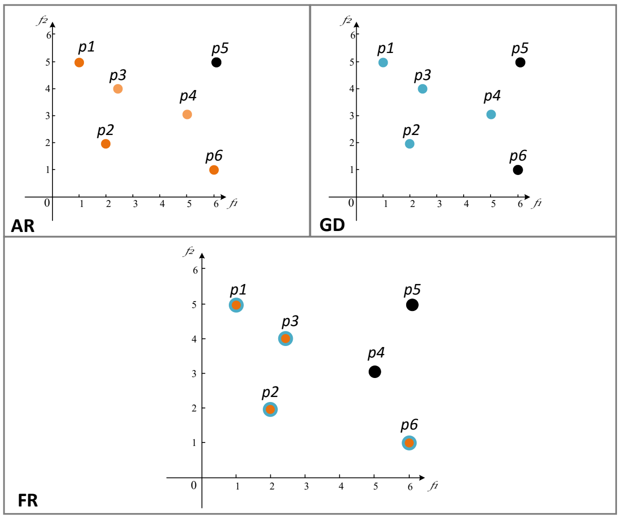

Fig. 1. A graphic illustration on AR, GD and FR

3.2. Optimal guides selection and archive maintenance

In mono PSO, the swarm pursues the orientation of *pBest* and *gBest* and can converge rapidly. More importantly, the *gBest* and *pBest* can be identified easily and explicitly because of the sole objective. However, optimal guides are not so simple to be determined in the general MOP, due to the fact that the Pareto-based dominance cannot obtain a specific relation facing multimodal or weak efficient solutions, but the optimal guides are crucial for the convergence of algorithm. At this point some of the random selection such as roulette selection strategy will reduce the efficiency and effectiveness of the algorithm.

Notice that in section 3.1, we have gained the ranking information that the individual among the whole population, so we can get the merits of individual, which can clearly portray the relation between individuals.

On the other hand, the diversity of obtained solutions is also a primary concern of an algorithm [21][22],which can offer more reference information to the decision-maker. Since the optimal guides is the manipulator of the swarm, their direction will determine the way of the algorithm directly.

In order to give greater redundancy to optimal guides, we use the sum of Chebychev distance [23] of particle to the rest of the items in objective space as a measurement of particle's density to obtain the distribution of the particles in the solution space. Let M be the number of objective. Then the Chebychev distance of any two individuals is shown in the Eq. (9).

$$d(\vec{p}, \vec{p^*}) = \max_{1 \le j \le M} \left\{ \left| p_j^* - p_j \right| \right\} \tag{9}$$

For particle X_i, the density information is stated as follows:

$$D(X_i) = \sum_{\substack{j=1 \\ i \neq j}}^{pop} d(X_i, X_j) \tag{10}$$

where, $D(X_i)$ is the chebychev distance between X_i and X_j. The greater $D(X_i)$ is, the fewer the particles are around it with a better distribution. After obtaining the distribution of the particles, we combined the merits of the individuals to gain the comprehensive ranking which is shown in Eq. (11).

$$CR(X_i) = \frac{FR(X_i)}{D(X_i)} \tag{11}$$

By Eq. (11), we get the properties of the particles that contain the dominance and distribution. Consequently, in this paper, we take the first 50% of CR value into the external archive directly. In each iteration, after acquiring the CR value, the external file set is updated, which discards the complex external archive maintenance method. In optimal guides, as the overall ranking information of the swarm has been obtained including the merits and the distribution of the particles, we take the following approach for the selection of optimal guides.
***gBest* :** According to the CR value, the particle with the minimum CR value is taken as the *gBest* of the swarm;

***pBest* :** Compare the current CR value of the individual with the historical value. Take the smaller one as the *pBest* to participate in the iteration. We call this algorithm MOPSO-CR

3.3. Algorithm Procedure

In light of the framework of PSO and ranking operations, the MOPSO-CR process is described as follows.

BEGIN
1.Initialize: pop. Size N, objective number M,
2.For each particle i,
  Generate the position and velocity of each particle randomly,
  Evaluate
End for
3.Update external archive
4.For t=1 to max generations (max.gen)
  For each particle i in swarm,
    Calculate the FR (Eq. (8)) D (Eq. (9)) CR (Eq. (10))
    Obtain the *gBest* and *pBest*;
    Update the velocity, position of each particle (Eq. (2) , Eq. (3))
    Push the particles on top 50% of CR values into external archive
  End For
  For each particle j in External archive
    Calculate the FR (Eq. (8)) D (Eq. (9)) CR (Eq. (10)) of each particle in external archive
    Merge the swarm
  End for
  Update external archive
  $t \leftarrow t + 1$
End For
END

3.4. Computational complexity analysis

By analyzing the procedure in section 3.3, we can see that the main complexity of the algorithm lies in the calculation of CR value. The complexity of some main scenarios are shown in Table.2.

It can be seen that the complexity of the proposed algorithm is $O(MN^2)$, the same as the NSGA-II.

Table 2. Computational complexity on MOPSO-CR

Scenario	**Time complexity**
Fusion Ranking	$O(MN^2)$
Selection of *pBest*	$O(N)$
Election of *gBest*	$O(N)$
Updating the *velocity* and *position*	$O(MN)$
External archive maintenance	$O(MN)$

Table 3. Adapted parameters for each candidate

NSGA-II	MOPSO-σ	MOPSO-CD	MOPSO-CR
Pc=1.0 Pm=1/n	w=0.4 c1=2 c2=2 Mr=0.03	w=0.4 c1=2 c2=2 Mr=0.03	w=0.4 c1=2 c2=2

4. Performance assessment

4.1. Experimental setup

To test the effectiveness of the algorithm, this section offers the performance comparisons among some representative and specialized algorithms (NSGA-II [24], MOPSO-σ [17], MOPSO-CD [25]) which validate on some standard benchmark functions including KUR [26], ZDT{1, 2} [27], DTLZ{1, 2} [28]. The selected benchmarks cover types of MOPs, with linear, convex, concave, etc., with two or three optimization objectives. The parameters of adopted algorithms and benchmark suites are shown in Table 3 and Appendix A respectively.

4.2. Performance metrics

Generally speaking, the assessment of an algorithm are mainly from the convergence and diversity. Due to this fact, the following two widely used metrics are adopted to perform the proposed approaches:
Generational distance (GD) [29]: measures the difference between the obtained approximation set of non-dominated solutions and the true Pareto front. This metric is defined as Eq. (11):

$$GD \triangleq \frac{1}{n}\sqrt{\sum_{i=1}^{n} d_i^2} \tag{12}$$

where d_i is the Euclidean distance between the non-dominated solution i and its nearest member in Pareto font. The smaller the GD, the better the performance. A value of zero indicates all the found non-dominated solutions lie in the Pareto front.

Error Ratio (ER) [29]: measures the error percentage of obtained non-dominated solutions. It is defined as:

$$ER = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{13}$$

For ER, the smaller, the better. A value of zero reveals that all found non-dominated solutions belong to the true Pareto front.

4.3. Results and discussion

Before proceeding to any inference, it is instructive to visualize the experiment to gain some insight on the features of the obtained results. Fig.2-Fig.6 show the convergence and distribution of the candidates.

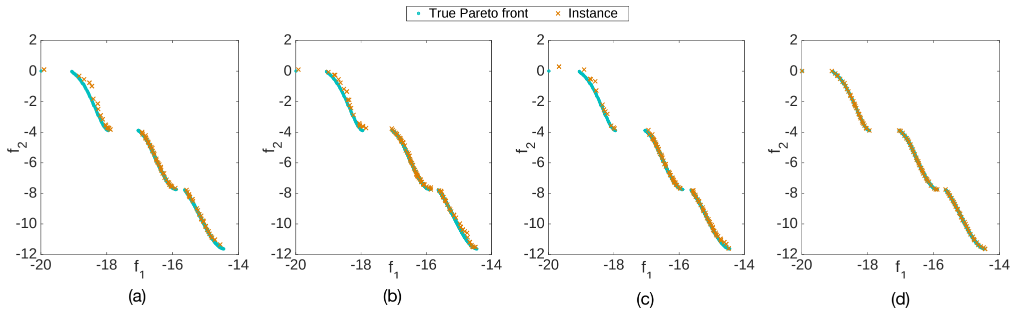

Fig. 2. Non-inferior solutions obtained by candidates on KUR. (a)NSGA-II(b) MOPSO-σ (c) MOPSO-CD (d) MOPSO-CR

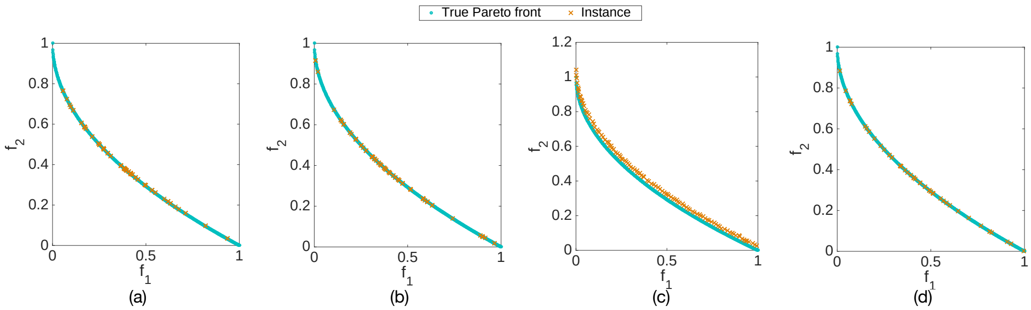

Fig. 3. Non-inferior solutions obtained by candidates on ZDT1. (a)NSGA-II(b) MOPSO-σ (c) MOPSO-CD (d) MOPSO-CR

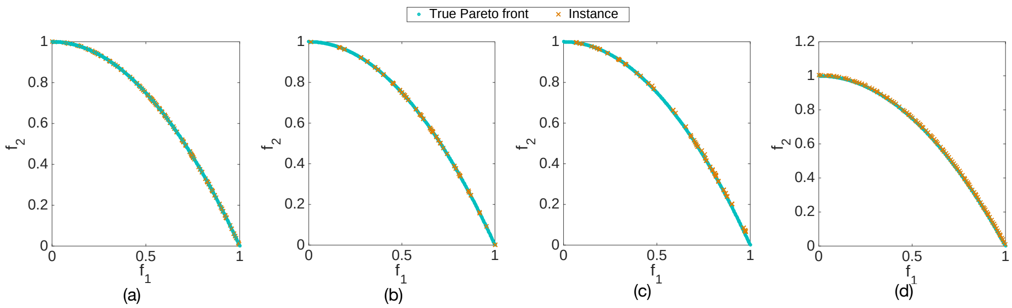

Fig. 4. Non-inferior solutions obtained by candidates on ZDT2. (a)NSGA-II(b) MOPSO-σ (c) MOPSO-CD (d) MOPSO-CR

In bi-objective issues, each algorithm can achieve a proximate and uniform solutions which indicates the candidates can deal with these simple bi-objective problems easily and efficiently. And MOPSO-CR gained the stable performance compared to other algorithms. However, the situation is a little different in tri-objective issues. NSGA-II has a little deviation on DTLZ1, a proportion of solutions are diverged from true pareto front. Like bi-objective, MOPSO-CR gains the decent behaviour.

After visually demonstrating the results, we implement a quantitative analysis on metrics of GD and ER which are shown in Table 4 and Table 5. Thirty independent runs are implemented for each test problem to avoid randomness, and the statistics values include mean value (Mean) and standard deviations (Std.). Bold number is the winner in such corresponding performance metric.

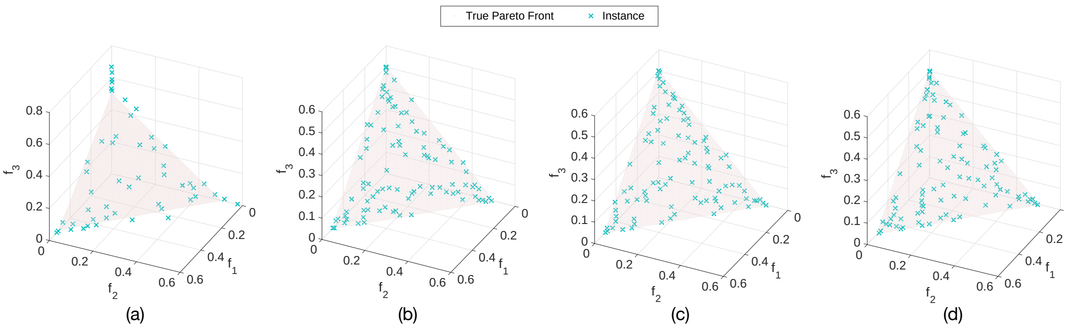

Fig. 5. Non-inferior solutions obtained by candidates on DTLZ1. (a)NSGA-II2 (b) MOPSO-σ (c) MOPSO-CD (d) MOPSO-CR

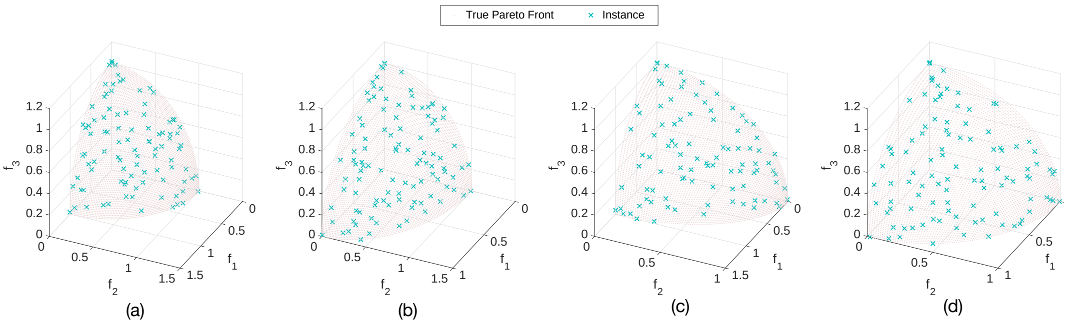

Fig. 6. Non-inferior solutions obtained by candidates on DTLZ2. (a)NSGA-II(b) MOPSO-σ (c) MOPSO-CD (d) MOPSO-CR

Table 4. Comparison results of NSGA-II, MOPSO-σ, MOPSO-CD and MOPSO-CR based on IGD.

Instance	NSGA-II		MOPSO-σ		MOPSO-CD		MOPSO-CR	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
KUR	5.20E-03	4.26E-04	4.90E-03	3.83E-04	4.71E-03	2.91E-04	**3.82E-03**	2.77E-04
ZDT1	4.41E-03	3.25E-04	**3.79E-03**	5.60E-04	5.72E-03	8.26E-04	3.97E-03	5.22E-04
ZDT2	**3.19E-03**	4.34E-04	7.31E-03	5.29E-04	5.44E-03	3.36E-04	3.72E-03	3.09E-04
DTLZ1	6.79E-03	5.37E-04	4.81E-03	4.11E-04	**3.28E-03**	2.19E-04	4.13E-03	3.44E-04
DTLZ2	3.52E-03	4.42E-04	3.19E-03	5.99E-04	3.38E-03	4.77E-04	**2.73E-03**	2.71E-04

Table 5. Comparison results of NSGA-II, MOPSO-σ, MOPSO-CD and MOPSO-CR based on ER

Instance	NSGA-II		MOPSO-σ		MOPSO-CD		MOPSO-CR	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
KUR	1.72E-01	6.77E-02	8.33E-02	2.98E-03	3.42E-02	1.19E-03	**2.20E-03**	1.18E-04
ZDT1	9.70E-02	3.22E-02	**5.63E-02**	1.27E-03	6.39E-01	2.02E-01	8.97E-02	2.93E-02
ZDT2	5.22E-02	3.10E-03	8.94E-02	4.30E-02	3.18E-02	9.10E-03	**8.70E-03**	2.11E-03
DTLZ1	3.27E-01	8.90E-02	3.90E-02	1.45E-02	**8.01E-02**	3.89E-02	1.09E-01	5.21E-03
DTLZ2	9.42E-02	2.27E-02	8.70E-02	4.12E-02	9.02E-02	3.20E-02	**3.92E-03**	1.13E-03

5. Conclusion and future work

For the average ranking is easy to aggregate on the edge of optimal front and Global Detriment is easy to exclude the extreme points, this paper integrates the merits and drawbacks of AR and GD, and proposes a new ranking scheme called fusion ranking which incorporates the average ranking and global detriment. On the basis of ranking and density information which obtained by chebychev distance among particles, we propose a new optimal guides selection strategy. Furthermore, in aspect of external archive updating, we present a new maintenance strategy by utilizing the ranking and density information, which makes the obtained solutions have more diversity and uniformity. The following performance assessment verifies the effectiveness of the proposed method.

Further work remains to be conducted on incorporating the ranking strategy into some other evolutionary algorithms and algorithms with higher dimensions.

Acknowledgements

This work was supported by the National Science & Technology Pillar Program during the Twelfth Five-year Plan Period (2012BAD10B01) and National Natural Science Foundation of China (61379123).

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Neural Networks, 1995. Proceedings., IEEE International Conference on, Vol. 4, IEEE, 1995, pp. 1942–1948.

[2] F. V. Nepomuceno, A. P. Engelbrecht, A self-adaptive heterogeneous pso for real-parameter optimization, in: Evolutionary Computation (CEC), 2013 IEEE Congress on, IEEE, 2013, pp. 361–368.

[3] A. R. Jordehi, J. Jasni, N. A. Wahab, M. Kadir, M. Javadi, Enhanced leader pso (ELPSO): a new algorithm for allocating distributed tcsc's in power systems, International Journal of Electrical Power & Energy Systems 64 (2015) 771–784.

[4] W. H. Lim, N. A. M. Isa, Particle swarm optimization with increasing topology connectivity, Engineering Applications of Artificial Intelligence 27 (2014) 80–102.

[5] C. A. C. Coello, M. R. Sierra, A coevolutionary multi-objective evolutionary algorithm, in: Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, Vol. 1, IEEE, 2003, pp. 482–489.

[6] K. Deb, Multi-objective evolutionary algorithms, in: Springer Handbook of Computational Intelligence, Springer, 2015, pp. 995–1015.

[7] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evolutionary computation 10 (3) (2002) 263–283.

[8] L. S. Batista, F. Campelo, F. G. Guimarães, J. Ramírez, Pareto cone ϵ-dominance: improving convergence and diversity in multiobjective evolutionary algorithms, Lect. Notes Comput Sci.(Springer) 6576 (2011) 76–90.

[9] X. Zou, Y. Chen, M. Liu, L. Kang, A new evolutionary algorithm for solving many-objective optimization problems, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 38 (5) (2008) 1402–1412.

[10] R. T. Marler, J. S. Arora, The weighted sum method for multi-objective optimization: new insights, Structural and multidisciplinary optimization 41 (6) (2010) 853–862.

[11] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, Evolutionary Computation, IEEE Transactions on 11 (6) (2007) 712–731.

[12] M. Asafuddoula, T. Ray, R. Sarker, A decomposition-based evolutionary algorithm for many objective optimization, Evolutionary Computation, IEEE Transactions on.

[13] P. J. Bentley, J. P. Wakefield, Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms, in: Soft computing in engineering design and manufacturing, Springer, 1998, pp. 231–240.

[14] M. Garza-Fabre, G. T. Pulido, C. A. C. Coello, Ranking methods for many-objective optimization, in: MICAI 2009: Advances in Artificial Intelligence, Springer, 2009, pp. 633–645.

[15] C. A. C. Coello, M. S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, Vol. 2, IEEE, 2002, pp. 1051–1056.

[16] G. G. Yen, H. Lu, Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation, Evolutionary Computation, IEEE Transactions on 7 (3) (2003) 253–274.

[17] S. Mostaghim, J. Teich, Strategies for finding good local guides in multi-objective particle swarm optimization (mopso), in: Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE, IEEE, 2003, pp. 26–33.

[18] A. Menchaca-Mendez, C. A. C. Coello, Selection operators based on maximin fitness function for multi-objective evolutionary algorithms, in: Evolutionary Multi-Criterion Optimization, Springer, 2013, pp. 215–229.

[19] R. Balling, S. Wilson, The maxi-min fitness function for multi-objective evolutionary computation: Application to city planning, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), 2001, pp. 1079–1084.

[20] D. Dasgupta, G. Hernandez, A. Romero, D. Garrett, A. Kaushal, J. Simien, On the use of informed initialization and extreme solutions sub-population in multi-objective evolutionary algorithms, in: Computational intelligence in miulti-criteria decision-making, 2009. mcdm'09. ieee symposium on, IEEE, 2009, pp. 58–65.

[21] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Computing Surveys (CSUR) 45 (3) (2013) 35.

[22] S. F. Adra, P. J. Fleming, Diversity management in evolutionary many-objective optimization, Evolutionary Computation, IEEE Transactions on 15 (2) (2011) 183–195.

[23] A. López Jaimes, C. A. Coello Coello, Study of preference relations in many-objective optimization, in: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM, 2009, pp. 611–618.

[24] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, Evolutionary Computation, IEEE Transactions on 6 (2) (2002) 182–197.

[25] C. R. Raquel, P. C. Naval Jr, An effective use of crowding distance in multiobjective particle swarm optimization, in: Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation, ACM, 2005, pp. 257–264.

[26] F. Kursawe, A variant of evolution strategies for vector optimization, in: Parallel Problem Solving from Nature, Springer, 1991, pp. 193–197.

[27] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary computation 8 (2) (2000) 173–195.

[28] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA), Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA), 2002, pp. 825–830.

[29] D. A. Van Veldhuizen, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, Tech. rep., DTIC Document (1999).

Appendix A. Related benchmark functions

Instance	Expression	Dim.	Obj.	Domain	Properties
KUR	$\begin{cases} f_1(x) = \sum_{i=1}^{2} (-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})) \\ f_2(x) = \sum_{i=1}^{3} (\lvert x_i\rvert^{0.8} + 5\sin(x_i)^3) \end{cases}$	3	2	$-5 \le x_i \le 5$ $(i = 1, 2, \cdots m)$	Disconnected PF
ZDT1	$\begin{cases} f_1(x) = x_1, \quad f_2(x) = g(1 - \sqrt{f_1/g}) \\ g(x) = 1 + 9\sum_{i=2}^{m} x_i/(m-1) \end{cases}$	30	2	$0 \le x_i \le 1$	Convex PF
ZDT2	$\begin{cases} f_1(x) = x_1, \quad f_2(x) = g(1 - (f_1/g)^2) \\ g(x) = 1 + 9\sum_{i=2}^{m} x_i/(m-1) \end{cases}$	30	2	$0 \le x_i \le 1$	Non-convex PF
DTLZ1	$\begin{cases} \min f_1(x) = 0.5x_1x_2\cdots x_{m-1}[1 + g(x_m)] \\ \min f_2(x) = 0.5x_1x_2\cdots(1 - x_{m-1})[1 + g(x_m)] \\ \vdots \\ \min f_{m-1}(x) = 0.5x_1(1 - x_2)[1 + g(x_m)] \\ \min f_m(x) = 0.5(1 - x_1)[1 + g(x_m)] \\ g(x_m) = 100[\lvert x_m\rvert + \sum_{x_i \in x_m} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))] \end{cases}$	10	3	$0 \le x_i \le 1$ $(i = 1, 2, \cdots m)$	Linear PF
DTLZ2	$\begin{cases} \min f_1(x) = [1 + g(x_m)]\cos(x_1{}^{\pi}/2)\cdots\cos(x_{m-2}{}^{\pi}/2)\cos(x_{m-1}{}^{\pi}/2) \\ \min f_2(x) = [1 + g(x_m)]\cos(x_1{}^{\pi}/2)\cdots\cos(x_{m-2}{}^{\pi}/2)\sin(x_{m-1}{}^{\pi}/2) \\ \min f_3(x) = [1 + g(x_m)]\cos(x_1{}^{\pi}/2)\cdots\sin(x_{m-2}{}^{\pi}/2) \\ \vdots \\ \min f_m(x) = [1 + g(x_m)]\sin(x_1{}^{\pi}/2) \\ g(x_m) = \sum_{x_i \in x_m} (x_i - 0.5)^2 \end{cases}$	10	3	$0 \le x_i \le 1$ $(i = 1, 2, \cdots m)$	Concave