Enhancing Diversity for Average Ranking Method in Evolutionary Many-Objective Optimization

Miqing Li, Jinhua Zheng, Ke Li, Qizhao Yuan, and Ruimin Shen

Institute of Information Engineering, Xiangtan University,
411105 Hunan, China
limit1008@126.com, jhzheng@xtu.edu.cn, JerryI00@yahoo.com.cn,
yuanyuan_xtu@163.com, srmxp@126.com

Abstract. The average ranking (AR) method has been shown highly effective to provide sufficient selection pressure searching towards Pareto optimal set in many-objective optimization. However, as lack of diversity maintenance mechanism, the obtained final set may only concentrate in a subregion of Pareto front. In this paper, we propose a diversity maintenance strategy for AR to balance convergence and diversity during evolution process. We employ grid to define an adaptive neighborhood for each individual, whose size varies with the number of objectives. Moreover, a layering selection scheme integrates it and AR to pick out well-converged individuals and prohibit or postpone the archive of adjacent individuals. From an extensive comparative study with original AR and two other diversity maintenance methods, the proposed method shows a good balance among convergence, uniformity and spread.

Keywords: Multiobjective optimization, Many-objective optimization, Average ranking, Diversity maintenance.

1 Introduction

During the recent past, evolutionary multiobjective optimization (EMO) algorithms have been receiving an extensive interest, mainly because of their potential to find a well-distributed approximation of Pareto optimal set. Nonetheless, most of them merely focus on the problems with two or three objectives, in spite of the fact that the problems with more than three objectives widely exists in real-world application [1], which is generally termed many-objective problems. One of the main reasons for this occurrence is that the proportion of nondominated solutions in a population rises rapidly with the increasing of the number of objectives [2]. The Pareto dominance relation based algorithms, such as NSGA-II [3] and SPEA2 [4], would fail to provide enough selection pressure to distinguish these solutions for searching towards the Pareto front.

Very recently, some non-Pareto-based techniques have been proposed specially for solving many-objective problems; such as, k-optimality, preference order ranking, favour relation, contraction-expansion, and so forth [2,5,6]. These

R. Schaefer et al. (Eds.): PPSN XI, Part I, LNCS 6238, pp. 647–656, 2010.

methods commonly employ some other optimality relations replacing or enhancing Pareto dominance relation to increase the selection pressure, and they seem to perform well in terms of converging close to the optimum. However, as lack of effective diversity maintenance mechanism, the final sets obtained by these relations are usually just a subset of the Pareto optimal set [6]. In fact, as for many-objective problems, it is not a trivial job to provide sufficient selection pressure towards the Pareto front and at the same time maintain a good distribution of solutions. The conflict between the requirements of convergence and diversity is gradually aggravated with the growing of the number of objectives, due to the fact that the size of feasible objective space for a certain problem increasing with the dimensionality of the optimization problem [7].

Average ranking (AR) proposed by Bentley and Wakefield [8] is regarded as an alternative to rank individuals in multiobjective population, though the authors were not particularly concerned with many objective problems. In recent years, the AR method has been found to perform successfully in searching towards the Pareto front in many-objective optimization [5,6]. However, similar to the aforementioned non-Pareto-based methods, it often converges into a subset of Pareto front because of the lack of diversity maintenance mechanism [6]. In this paper, we incorporate a diversity maintenance strategy into AR to cover this shortage. We define a grid-based adaptive neighborhood for each individual in the population to preserve the suitable spacing among them. Moreover, a layering selection method utilizing it and AR is designed to pick out well-converged individuals and prohibit or postpone the archive of neighboring individuals.

The remainder is structured as follows. Section 2 describes the AR method and shows its properties. Section 3 is devoted to detail our diversity maintenance strategy. Section 4 provides a comparison of the proposed method versus the original AR and other high dimension optimization techniques. Finally, in Section 5 the results are summarized and directions for future line are pointed out.

2 Average Ranking

The average ranking method compares all individuals on each objective and ranks them independently. For a specific solution, a rank for each objective is assigned based on the level of its objective value among all solutions in the population. Thus each solution has M ranks (where M is the number of objectives), and the final rank is obtained by summing them. Table 1 illustrates the AR method with a simple example considering 4-objective solutions.

Clearly, AR is capable of distinguishing the nondominated solutions according to their ranks on different objectives. Additionally, it is also computationally simple and range-independent since all objective values are compared separately. Corne and Knowles have reported that AR outperforms some more complicated ranking strategies in terms of *cover metric* [5]. However, as lack of diversity maintenance mechanism, the population may converge into a subregion of the Pareto front. For instance, in Table 1 if the size of the population is 3, the winner

Table 1. An example of the AR method

solution	(f_1, f_2, f_3, f_4)	rank1	rank2	rank3	rank4	AR
A	(1, 1, 6, 5)	1	1	4	2	8
B	(1, 3, 5, 6)	1	3	2	3	9
C	(2, 2, 5, 6)	3	2	2	3	10
D	(6, 5, 1, 7)	4	4	1	5	14
E	(7, 5, 7, 1)	5	4	5	1	15

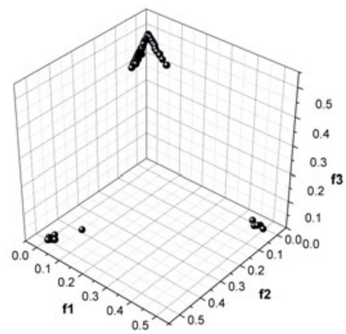


Fig. 1. Final solutions obtained by AR on DTLZ1

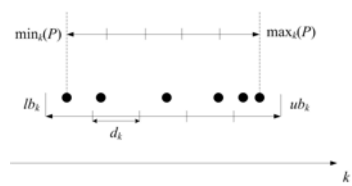


Fig. 2. Setting of grid in the kth objective

will be **A**, **B** and **C** according to the AR values of them. Unfortunately, though they seem to perform better in terms of convergence, they concentrate in a tiny region against the whole objective space, and have a high likelihood of evolving towards a local region of the optimal front. Figure 1 gives the final solutions set obtained by AR on 3-objective DTLZ1 problem.

3 The Proposed Diversity Maintenance Method

Grid technique has been widely used in the field of evolutionary multiobjective optimization. Many grid-based EMO algorithms have been proven to perform well in maintaining diversity when problems have two or three objectives. Here, we expand its potential to many-objective problems. First we fix a grid environment, where the population dwells.

3.1 Grid Setting

Borrowing from AGA [9], the grid is determined by the distribution of the current population. Fig. 2 illustrates the setting of grid in the kth objective.

First the minimum and maximum values of the objective k among the individuals in a population P are found and thus denoted as $\min_k(P)$ and $\max_k(P)$, respectively. Afterward, the lower and upper boundaries of grid in the kth objective are determined by them:

$$lb_k = \min_k(P) - (\max_k(P) - \min_k(P))/(2\times div) \quad (1)$$

$$ub_k = \max_k(P) + (\max_k(P) - \min_k(P))/(2\times div) \quad (2)$$

where div is a constant parameter, the number of divisions of the objective space in each dimension, set by the user (e.g., in Fig. 2 $div = 5$). Accordingly, the original M-dimensional objective space will be divided into div^M hyperboxes. Thus, the hyperbox width in the kth objective, d_k, could be formed as

$$d_k = (ub_k - lb_k)/div \quad (3)$$

Therefore, according to lb_k and d_k, the grid coordinate of any individual in the kth objective is determined as

$$G_k(\mathbf{A}) = \lfloor (F_k(\mathbf{A}) - lb_k)/d_k \rfloor \quad (4)$$

where $G_k(\mathbf{A})$ is the grid coordinate of individual $\mathbf{A}$ in the kth objective, $F_k(\mathbf{A})$ is the actual objective value in the kth objective. For instance, in Fig. 2, the grid coordinates of all individuals (from left to right) in the kth objective are 0, 1, 2, 3, 4 and 4, respectively.

3.2 Adaptive Neighborhood

Many existing grid-based EMO algorithms encounter difficulties in their scalability to many-objective optimization. One of the main reasons is that their density estimation mechanisms, which only consider the crowding of unit hyperbox in grid, may be invalid in high dimensional space. As the increase of objectives, the number of hyperboxes in grid will grow exponentially [5] (the number of hyperboxes in a k-objective problem is r^k, where r is the divisions in each dimension).

In this paper, we present an adaptive neighborhood based density estimation strategy to address this issue. The neighborhood of individuals here is composed of several hyperboxes around it, and the size of it will vary with the number of objectives. Specifically, for individual A, the neighborhood of it is defined as:

$$N(\mathbf{A}) = \left| \{X : \sum_{k=1}^{M} |G_k(\mathbf{A}) - G_k(X)| < M\} \right| \quad (5)$$

where $|\cdot|$ denotes the cardinality of a set, $G_k(\mathbf{A})$ implies the grid coordinate of individual $\mathbf{A}$ in the kth objective, $G_k(\boldsymbol{X})$ stands for the coordinate of hyperbox $\boldsymbol{X}$ in the kth objective, and M is the number of objectives. It is clear to note that the range of neighborhood of individual is determined by variable M. As M becomes larger, the number of hyperboxes in the neighborhood of individuals will increase steadily. This seems to be consistent with the total number of hyperboxes in grid environment. In the following, we describe the diversity maintenance method using the neighborhood.

3.3 Layering Selection

In this section, we introduce a layering selection approach integrating AR and adaptive neighborhood to determine the survival of individuals. The individual

Algorithm 1. *Layering Selection (P)*

Require: P(candidate set), Q(archive set), N(archive size), CP(current layer solutions set), NP(next layer solution set)

```
1: Q ⟵ null, NP ⟵ null, CP ⟵ P        /* Initialize sets Q, NP, and CP*/
2: while |Q| < N do
3:   if CP = null then
4:     CP ← NP
5:     NP ← null
6:   end if
7:   q ← FindoutBest(CP) /* Find out the individual with the best AR value in CP*/
8:   Q ← Q ⋃ {q}                      /* Put the best individual into archive set*/
9:   CP ← CP \ {q}                    /* Remove the best individual from CP*/
10:  for all p ∈ CP do
11:    if G(p) ∈ N(q) then
12:      NP ← NP ⋃ {p}
         /* Add the individual who is the neighbor of q into NP and delete it from CP*/
13:      CP ← CP \ {p}
14:    end if
15:  end for
16: end while
17: return Q
```

with best AR value in current layer is selected to be archived firstly, and the neighbors of it (i.e., the individuals located in its neighborhood) will be demoted to next layer, no matter how good their ranks are. Algorithm 1 gives a detailed procedure of this approach.

The essential purpose of the algorithm is to prohibit or postpone the entry of adjacent individuals. Function *FindoutBest* (line 7) is designed to find out the best individual according to AR in current layer. The lines 10-15 of the algorithm is implemented a punishment to the neighbors of the best individual by relegating them to next layer. If the current layer is null, the next layer is activated to continue the above selection procedure (lines 3-6). Fig. 3 illustrates the algorithm with a simple example on 2-objective optimization problem. Initially, the current layer set contains individuals **A-H**. **G** is picked out firstly into the archive since it has the best AR value (7). Correspondingly, the neighbors (**B**, **C**, **D**, **E**, **F**) of **G** are degraded into the next layer (shown in Fig. 3(b)). Repeat this procedure until the current layer is null (shown in Fig. 3(e)). At this time, the next layer will be activated and turn into the current layer. So the best individual (**D**) in new current layer is selected, and the neighbors (**B**, **C**, **E**, **F**) of it enter the new next layer correspondingly (shown in Fig. 3(f)). Finally, the individuals in the archive are **A**, **D**, **G**, **H**, and **J**, when the vacancies are filled up.

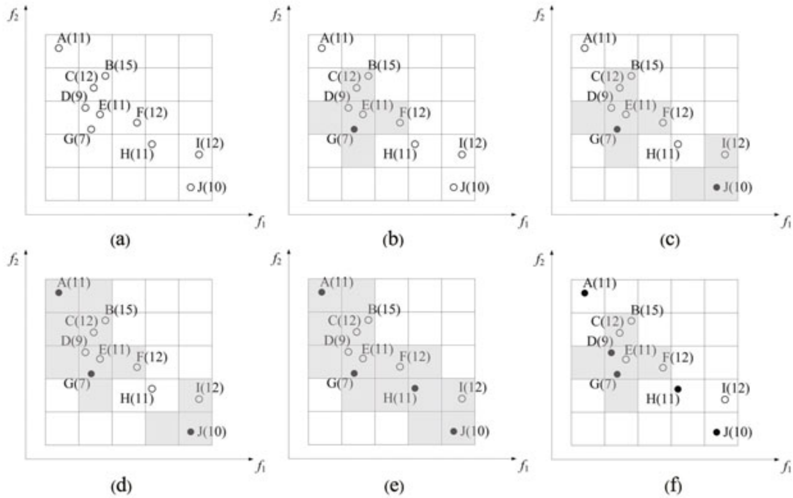


Fig. 3. An illustration of layering selection algorithm. Where archive size is set to 5. The value in the brackets corresponds to AR of individuals. Hollow points stand for the candidate individuals for archive and black points stand for the individuals that have selected into the archive set. Shadow area indicates the neighborhood of the selected individuals in current layer (i.e., the candidate individuals located in this area have been demoted to next layer)

4 Experimental Setup and Results

In this section, two diversity maintenance methods, improved Crowding Distance [10] and DMO [7], as well as Original AR algorithm are introduced to validate the proposed method. The improved crowding distance method assigns a zero distance (instead of an infinity distance) to extreme solutions in order to advance the convergence of algorithm [10]. DMO employs a diversity management operator to control or promote the diversity requirement. If the diversity indicator is smaller than 1 according to normal *maximum spread* test, the diversity promotion mechanism (i.e., crowding distance) is activated, conversely deactivated. For a fair comparison,the two methods are incorporated into AR. We note them as AR+CD$'$ and AR+DMO, respectively. The original AR algorithm, similar to [5], is implemented to select individuals according to AR for variation yet renew the archive in a random way. Anyway, the four algorithms (AR, AR+CD$'$, AR+DMO and the proposed method) adopt identical fitness strategy (AR) in selection for variation stage and yet adopt distinct diversity maintenance schemes (random selection, improved Crowding Distance, DMO and grid-based technique) in selection for survival stage. In addition, these algorithms are embeded into NSGA-II template for a fair comparison. Parent population combines with current population for generating the best half offspring. The last allowed nondominated front is considered by the above schemes instead of crowding distance. In the following, several performance metrics and test problem used in comparison are introduced in brief.

4.1 Performance Metrics and Test Problem

Usually, there are three goals that EMO algorithms can be identified and measured in performance [11]: (i) the distance of the resulting solutions to Pareto front (PF) should be minimized; (ii) a uniform distribution of the solutions found is desirable and (iii) the extent of the solutions should be maximized. In this paper, three performance metrics (CM [12], DM [12] and MS [13]), which directly evaluate each of the above goals, are considered.

The convergence metric CM calculates the average distance of the obtained solutions set away from the Pareto front. Similar to the studies in [10], the distance to the Pareto front is determined analytically without using a reference set. The uniformity metric DM measures the homogenization for a set of points. In DM, the obtained nondominated points are projected on a hyperplane, which is divided into a number of boxes. Depending on each box contains a point or not, the DM value is defined. DM takes the value between zero and one (one is the ideal result), and the larger value it achieves, the better is the uniformity. The detailed description of DM can be referred in [12]. The spread metric MS is an improved version of *Maximum Spread* considering the distribution of the Pareto front [13]. The original MS, which measures the length of the diagonal of the hypercube formed by the extreme objective values in a given set, may be influenced heavily by convergence of algorithm. The improved MS is devised to introduce the extreme values of the Pareto front for ease this effect. It also takes the value between zero and one and a higher value will tell about a larger extent of the obtained nondominated set.

To benchmark the performance of the four algorithms, the scalable function DTLZ2 [14] is invoked.The number of objectives used in this experiment is 3, 4, 6, 8, 10, 12, and 15. The total number of decision variables of the function is l=M+n−1. Where M is the number of objectives and n can be set by user to specify the distance to PF. According to [14], n=10 is used in DTLZ2.

4.2 Comparative Experiment

All compared algorithms are given real-valued decision variables. A crossover probability p_c=1.0 and a mutation probability p_m=1/l (where l is the number of decision variables) are used. The operators for crossover and mutation are simulated binary crossover (SBX) and polynomial mutation with the both distribution indexes 20. We run each algorithm independently 100 times. In each run a population of 100 individuals during 300 generations is predefined. For the proposed method, the parameter div setting for different number of objectives is shown in Table 2.

Tables 3, 4 and 5 give the convergence, uniformity and spread comparison respectively for all four algorithms over 3, 4, 6, 8, 10, 12, and 15 objectives. The values in the tables correspond to mean and standard deviation. In order to give a visual comparison, Figure 4 plots the distribution of the final solution set for

Table 2. The *div* setting of the proposed method

Objective number	3	4	6	8	10	12	15
Division	20	18	15	14	13	12	11

Table 3. CM comparison of the four EMOAs

Obj.	AR	AR + CD′	AR+DMO	P roposed Method
3	$0.002630_{(0.001219)}$	$0.007697_{(0.000969)}$	$0.003127_{(0.001131)}$	$0.002122_{(0.000512)}$
4	$0.000265_{(0.000368)}$	$0.019888_{(0.002790)}$	$0.009224_{(0.003546)}$	$0.005162_{(0.001205)}$
6	$0.000197_{(0.000161)}$	$0.061420_{(0.010694)}$	$0.076881_{(0.017319)}$	$0.011763_{(0.002084)}$
8	$0.000314_{(0.000427)}$	$0.464933_{(0.093250)}$	$0.166829_{(0.024957)}$	$0.166829_{(0.024957)}$
10	$0.000414_{(0.000522)}$	$1.175100_{(0.145026)}$	$0.258695_{(0.034931)}$	$0.031825_{(0.008503)}$
12	$0.000634_{(0.000454)}$	$1.489080_{(0.120641)}$	$0.335911_{(0.049553)}$	$0.065004_{(0.028429)}$
15	$0.001182_{(0.000610)}$	$1.686800_{(0.101894)}$	$0.507604_{(0.062442)}$	$0.144395_{(0.039438)}$

Table 4. DM comparison of the four EMOAs

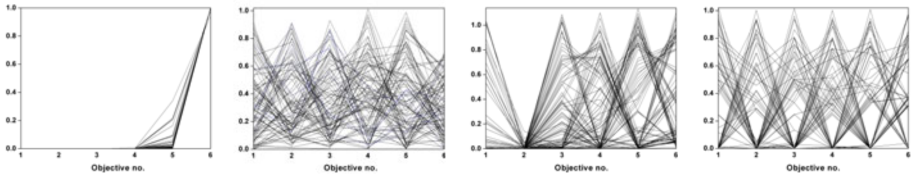
Obj.	AR	AR + CD′	AR+DMO	P roposed Method
3	$0.285620_{(0.061040)}$	$0.765828_{(0.033490)}$	$0.307042_{(0.092739)}$	$0.875861_{(0.038620)}$
4	$0.024635_{(0.006037)}$	$0.750573_{(0.056049)}$	$0.272139_{(0.073947)}$	$0.830936_{(0.043014)}$
6	$0.013498_{(0.004680)}$	$0.728653_{(0.062698)}$	$0.291955_{(0.053626)}$	$0.769867_{(0.036861)}$
8	$0.007968_{(0.002830)}$	$0.577113_{(0.095491)}$	$0.232654_{(0.035723)}$	$0.687197_{(0.015683)}$
10	$0.003084_{(0.001091)}$	$0.143857_{(0.032180)}$	$0.179114_{(0.015738)}$	$0.554510_{(0.018500)}$
12	$0.003356_{(0.000000)}$	$0.097137_{(0.019159)}$	$0.107763_{(0.011902)}$	$0.430962_{(0.031788)}$
15	$0.000625_{(0.000000)}$	$0.045647_{(0.023319)}$	$0.075254_{(0.024951)}$	$0.344784_{(0.040216)}$

Table 5. MS comparison of the four EMOAs

Obj.	AR	AR + CD′	AR+DMO	P roposed Method
3	$0.954843_{(0.101269)}$	$0.976454_{(0.015941)}$	$0.999668_{(0.001187)}$	$0.999982_{(0.000050)}$
4	$0.136769_{(0.045679)}$	$0.945741_{(0.018671)}$	$0.982307_{(0.031182)}$	$1.000000_{(0.000000)}$
6	$0.111958_{(0.035708)}$	$0.944685_{(0.019716)}$	$0.857206_{(0.048480)}$	$0.999716_{(0.000808)}$
8	$0.098869_{(0.032621)}$	$0.997528_{(0.005891)}$	$0.726130_{(0.024045)}$	$0.998269_{(0.005425)}$
10	$0.078272_{(0.034608)}$	$0.999999_{(0.000004)}$	$0.664174_{(0.030018)}$	$0.991779_{(0.017571)}$
12	$0.063128_{(0.022658)}$	$0.999996_{(1.177e-06)}$	$0.614995_{(0.033558)}$	$0.978742_{(0.031660)}$
15	$0.064900_{(0.019471)}$	$1.000000_{(1.583e-07)}$	$0.545415_{(0.029611)}$	$0.965983_{(0.039171)}$

four algorithms by parallel coordinates on the problem with 6 objectives. Inferred from the CM value in Table 3, the proposed method could in general reach the Pareto front of the problem with all considered number of objectives. The other two diversity maintenance methods perform well for 3, 4, and 6 objectives, but encounter difficulty in case of more objectives. Note that original AR obtains the better CM values than the proposed method in all case except the number of objectives is equal to 3. However, from Tables 4 and 5, this result is achieved at the cost of the loss of diversity. In most case, the final solution set obtained by AR locates in a microscopic part of the Pareto front.

Concerning uniformity assessment metric DM in Table 4, the proposed method achieves the best values on the problem with all considered number of objectives. AR+CD′ performs better than AR+DMO on the problem with 3, 4, 6, and 8 objectives, but slightly worse in case of higher dimension. The original AR algorithm obtains the worst results for all objectives. This is due to the final solutions



(a) AR (b) AR + CD′ (c) AR+DMO (d) Proposed method

Fig. 4. Distribution of final solution set by parallel coordinates on six-objective DTLZ2

set of it concentrated practically into a point, rather than distributed over the Pareto front.

Table 5 shows the spread comparison results from the MS metric. Clearly, the proposed method reaches the boundary of the whole Pareto front for all considered number of objectives. AR achieves a fairly good value on 3-objective problem, but fail in case of more objectives. Similarly, AR+DMO only performs well for the problem with 3 and 4 objectives. AR+CD′ can obtain a passable value for 3, 4 and 6 objectives problem. The proximity of the four algorithms to the boundary of Pareto front on 6-objective DTLZ2 could be seen in Figure 4. Additionally, it is interesting to note that the MS values obtained by AR+CD′ have a sudden raise when the number of objective reaches 8. This occurrence could be attributed to the reason that AR+CD′ fails to approximate to the Pareto front on 8 or more objectives problem. The maximum value in each objective obtained by it would exceed one, thereby producing a misleading result with respect to MS.

In summation, from the comparative studied above, we can conclude that the proposed method produces a good balance with regard to convergence, uniformity, and spread in the specific settings of grid parameter. Due to space limitations, we do not show results of it with different settings. Actually, in some preliminary trials, we found that grid division has more effect upon uniformity than convergence and spread, especially in lower dimension space (e.g., objective = 3, 4, or 6).

5 Conclusions

This paper has presented a diversity maintenance strategy for original AR algorithm to balance its convergence and diversity in evolutionary many-objective optimization. The proposed method has defined a grid-based adaptive neighborhood varied with the number of objectives to preserve a suitable spacing among individuals. Moreover, a layering approach integrating it and AR has been introduced to select the promising individuals for archive store. Simulation experiments have been studied by providing a detailed comparison with other three algorithms (AR, AR + CD′, and AR+DMO). The results reveal that the proposed algorithm has been successful in finding a near-optimal, uniformly distributed and well-extended solution set. Possible avenues of future work include the investigation of grid parameter, more many-objective test problems, and the incorporation of layer information to automatically tune the division setting according to the number of objectives.

References

1. Purshouse, R.C., Fleming, P.J.: Conflict, Harmony and Independence: Relationships in Evolutionary Multi-criterion Optimisation. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 16–30. Springer, Heidelberg (2003)
2. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: 2008 Congress on Evolutionary Computation (CEC 2008), pp. 2424–2431. IEEE Service Center, Hong Kong (2008)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6(2), 182–197 (2002)
4. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Evolutionary Methods for Design, Optimisation and Control, CIMNE, Barcelona, Spain, pp. 95–100 (2002)
5. Corne, D., Knowles, J.: Techniques for highly multiobjective optimization: Some non-dominated points are better than others. In: Proc. of 2007 Genetic and Evolutionary Computation Conference (GECCO 2007), London, pp. 773–780 (July 2007)
6. Jaimes, A.L., Quintero, L.V.S., Coello Coello, A.C.: Ranking Methods in Many-objective Evolutionary Algorithms. In: Chiong, R. (ed.) Nature-Inspired Algorithms for Optimisation, pp. 413–434. Springer, Berlin (2009)
7. Adra, S.F., Fleming, P.J.: A Diversity Management Operator for Evolutionary Many-Objective Optimisation. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 81–94. Springer, Heidelberg (2009)
8. Bentley, P.J., Wakefield, J.P.: Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms. In: Soft Computing in Engineering Design and Manufacturing, Part 5, London, pp. 231–240 (June 1997)
9. Knowles, J., Corne, D.: Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. IEEE Trans. Evol. Comput. 7(2), 100–116 (2003)
10. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 742–756. Springer, Heidelberg (2007)
11. Li, M., Zheng, J.: Spread Assessment for Evolutionary Multi-Objective Optimization. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 216–230. Springer, Heidelberg (2009)
12. Deb, K., Jain, S.: Running Performance Metrics for Evolutionary Multi-Objective Optimization. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2002), pp. 13–20 (2002)
13. Goh, C.K., Tan, K.C.: An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization. IEEE Trans. Evol. Comput. 11(3), 354–381 (2007)
14. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. in Evolutionary Multiobjective Optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Theoretical Advances and Applications, pp. 105–145. Springer, Berlin (2005)