

# Convergence Versus Diversity in Multiobjective Optimization

Shouyong Jiang<sup>(✉)</sup> and Shengxiang Yang

Centre for Computational Intelligence (CCI),  
School of Computer Science and Informatics, De Montfort University,  
The Gateway, Leicester LE1 9BH, UK  
shouyong.jiang@email.dmu.ac.uk, syang@dmu.ac.uk

**Abstract.** Convergence and diversity are two main goals in multiobjective optimization. In literature, most existing multiobjective optimization evolutionary algorithms (MOEAs) adopt a convergence-first-and-diversity-second environmental selection which prefers nondominated solutions to dominated ones, as is the case with the popular nondominated sorting based selection method. While convergence-first sorting has continuously shown effectiveness for handling a variety of problems, it faces challenges to maintain well population diversity due to the overemphasis of convergence. In this paper, we propose a general diversity-first sorting method for multiobjective optimization. Based on the method, a new MOEA, called DBEA, is then introduced. DBEA is compared with the recently-developed nondominated sorting genetic algorithm III (NSGA-III) on different problems. Experimental studies show that the diversity-first method has great potential for diversity maintenance and is very competitive for many-objective optimization.

## 1 Introduction

Multiobjective optimization problems (MOPs) widely exist in real-world applications, such as scheduling [11] and design [12]. MOPs often have several conflicting objectives for which any improvement in one objective inevitably aggravates another. Due to multiobjectivity, there is no single optimal solution. Instead, the optima of MOPs is a set of trade-off solutions, known as Pareto-optimal set (POS). Correspondingly, the image of the POS in the objective space is called Pareto-optimal front (POF).

Multiobjective optimization evolutionary algorithms (MOEAs) are a class of important methods for solving MOPs. MOEAs employ a population of candidate individuals and optimize them in an evolutionary manner. As a result, a set of solutions can be obtained in a single run. Besides, MOEAs do not necessarily require any knowledge and information of the MOPs to be optimized, i.e., continuousness or differentiability. All these features make MOEAs very suitable for solving MOPs. So far, a large number of MOEAs [2, 9, 10] have been proposed in the evolutionary computation community.

In the design of MOEAs, two goals should be considered: (1) minimizing the gap between candidate solutions and the true POS (convergence) and (2)

maximizing the distribution of candidate solutions (diversity). However, these two goals are generally assumed to be conflicting [13]. In practice, most existing MOEAs achieve convergence by prior Pareto-based sorting of the evolving population and diversity by the additional calculation of individuals' density information. The well-known nondominated sorting genetic algorithm II (NSGA-II) [2] and strength pareto evolutionary algorithm 2 (SPEA2) [10] are representative examples of this method. Such a method actually performs environmental selection in a convergence-first-and-diversity-second manner. That is, nondominated individuals [2] are preferable to dominated ones although dominated individuals may contribute considerably to population diversity. While this method works well in two- and three-objective optimization problems, it has encountered great difficulties in many-objective optimization where problems have four or more objectives [6]. This is mainly because a large portion of the population becomes nondominated as the number of objectives increases. In this case, the convergence-first selection will consider only nondominated individuals and leave little room for diversity selection. If all nondominated individuals are themselves not diversified, it would lead to a detrimental diversity loss due to the convergence-first selection.

Inspired by the assumption that dominated individuals can contribute to population diversity, this paper proposes a new diversity-first sorting approach with the aid of a set of diverse reference directions. The approach sorts the population into different fronts, each front representing a level of diversity and convergence. Then, a diversity-first sorting based evolutionary algorithm (DBEA) is introduced. Empirical studies and algorithm comparisons demonstrate the promise of DBEA for multi- and many-objective optimization.

The remainder of the paper is organized as follows. Section 2 reviews the classic nondominated sorting method. Section 3 presents a new diversity-first sorting method, followed by our detailed implementation of DBEA in Sect. 4. Experimental design and comparison results are presented in Sect. 5. Finally, Sect. 6 concludes the paper.

## 2 Classic Sorting Methods

Most existing MOEAs are convergence-first based methods, such as NSGA-II [2] and SPEA2 [10]. Convergence-first based methods prefer convergence to diversity. They sort population depending mainly on individuals' convergence<sup>1</sup>. One of the most important sorting methods is the nondominated sorting used in NSGA-II. In the following, we will briefly describe how nondominated sorting works, followed by some discussions on its advantages and disadvantages.

### 2.1 Nondominated Sorting

In every generation, when the parent population (P) and offspring population (Q) are combined to form a union population (R) of size  $2N$ , environmental

<sup>1</sup> Note that, although some algorithms like SPEA2 sort individuals by exploiting both convergence and diversity, convergence is priorly considered and emphasised.

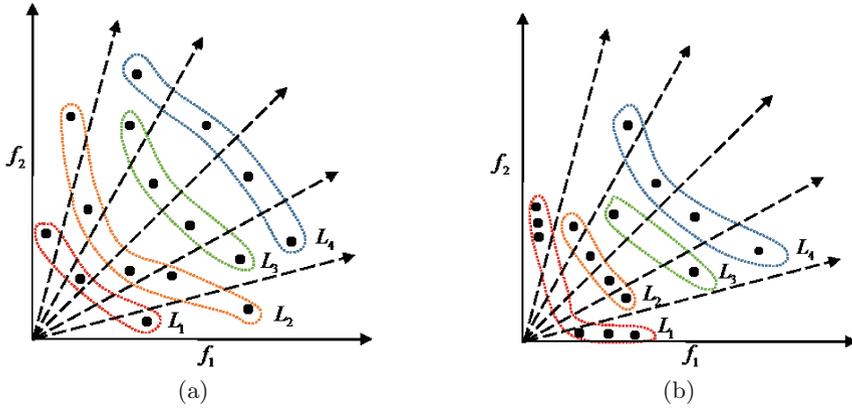


Fig. 1. Nondominated sorting.

selection should be carried out on  $R$  to construct a new parent population of size  $N$  for the next generation. The nondominated sorting strategy can be used for selection and works as follows. First, each individual is compared with all other individuals in  $R$ , and all nondominated solutions of  $R$  are identified and assigned to front  $L_1$ . Then, individuals in  $L_1$  are removed from  $R$ , and the remaining individuals in  $R$  are compared with each other to determine the nondominated set, which are assigned to front  $L_2$ . The procedure is repeated until no individuals are left in  $R$ , i.e., all individuals have been assigned to a front.

Figure 1(a) gives a graphical illustration of the nondominated sorting. The main idea behind the nondominated sorting is to classify the entire combined population into different nondominated fronts according to individuals' convergence. After the non-dominated sorting, the new population can be constructed by selecting solutions of different non-dominated fronts, one at a time. The selection starts with individuals of the first front  $L_1$  and continues with those of the second front  $L_2$ , followed by the rest of the fronts and so on. Since only  $N$  slots are allowable in the new population, not all fronts can be considered. When the last allowed front (e.g.,  $L_l$ ) is being considered, there may exist more individuals in  $L_l$  than the remaining slots in the new population. In this situation, niche-preservation strategies, such as crowding distance [2], the-farthest-the-first method [1], and nearest neighbour technique [10], are desirable for selecting the remaining number of individuals from front  $L_l$  in order to maintain diversity.

## 2.2 Advantages and Disadvantages

**Advantages.** The nondominated-sorting based selection favours convergence so that individuals in better fronts will be priorly preserved. The selection is helpful for a fast convergence speed.

**Disadvantages.** The nondominated-sorting based selection may undermine population diversity if well-converged individuals are not diversified. Figure 1(b) presents an example where diversity loss occurs if only six nondominated individuals are allowed to be preserved. The loss of diversity in the example could further incur evolutionary stagnation where overcrowded boundary regions are overexploited and intermediate regions are left unexplored.

### 3 Proposed Sorting Method

On the basis of discussions on convergence-first MOEAs, we propose and analyse a diversity-first sorting method in the following subsections.

#### 3.1 Diversity-First Sorting

The proposed diversity-first sorting method works as follows. First, the objective space is partitioned into a number of subspaces with the aid of a reference direction set  $W$ . Reference directions in  $W$  are required to be uniformly distributed. Then, each individual (whose objective values need to be normalized beforehand) in the combined population is associated with a subspace. This can be done by identifying the nearest reference direction to the considered individual. In each subspace, individuals are assigned a fitness value that can reflect its convergence level. Potential fitness assignment approaches for this purpose can be scalarizing functions used in MOEA/D [9], strength fitness in SPEA2 [10], or nondominated ranks in NSGA-II [2], whichever is the easiest for users to implement. An individual with the best fitness from each subspace is assigned to front  $L_1$ . After that, the individual in  $L_1$  are removed from the subspaces, and another with the best fitness from each subspace is assigned to front  $L_2$ . If multiple solutions have the same fitness, a random one is considered. This procedure continues until each individual in each subspace has been assigned to a front. Note that, in case that a subspace is empty, this subspace is skipped.

Figure 2(a) illustrates the outcome of diversity-first sorting, where population distribution is identical to that of Fig. 1(a). After the sorting, the new population can be constructed by selecting solutions of different fronts, one at a time. Similar to the nondominated sorting, not all fronts can be considered due to the limited number of slots in the new population. If the last allowed front (e.g.,  $L_l$ ) has more individuals than the remaining slots, random selection on  $L_l$  can be performed to fill up the new population. Note that, it is advisable to use techniques that are helpful for convergence to select individuals from  $L_l$ . For example, fitness assignment can be performed on  $L_l$ , and individuals with relatively good convergence are priorly selected.

#### 3.2 Advantages and Disadvantages

**Advantages.** As can be seen from Fig. 2(b) (where the population distribution is the same as that of Fig. 1(b)), the diversity-first sorting enhances local diversity

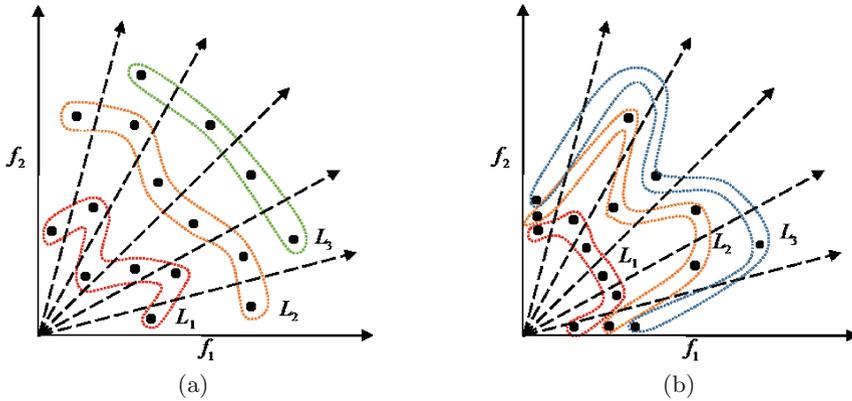


Fig. 2. Diversity-first sorting.

in each subspace. As a result, population diversity can be well maintained during the evolution. Besides, population convergence is also properly considered in the course of sorting. The sorting method can provide a good coverage and spread of approximation.

**Disadvantages.** Since the diversity-first sorting employs a reference direction set for population partition, the resulting population distribution depends largely on the uniformity of the reference direction set. The potential drawbacks remain unknown, and a future work will be devoted to these aspects.

### 4 Diversity-First Based Evolutionary Algorithm (DBEA)

In this section, we present an MOEA based on the proposed diversity-first sorting, called DBEA for short. The framework of DBEA is described in Algorithm 1. Several key components of DBEA are explained as follows.

Reference direction set  $W$ :  $W$  can be constructed on a unit simplex using Das and Dennis’s systematic approach [4] if the number of objectives is small. Otherwise,  $W$  is constructed by two-layered approach mentioned in [6].

Objective normalization: similar to NSGA-III [6], DBEA identifies all extreme points and then use them to construct a hyperplane. The intercepts of objective axes and the hyperplane can be computed. DBEA uses these intercepts and the utopia point to normalize the objective values of individuals.

Population partition: for each population individual, DBEA computes the acute angle between its normalized objective vector and each reference direction. The reference direction having the minimum acute angle is considered the right subspace that the individual should reside in.

**Algorithm 1.** Framework of DBEA

---

```

1: Input:  $N$  (population size)
2: Output: approximated Pareto-optimal set
3: Generate a diverse reference direction set  $W$ ;
4: Create an initial parent population  $P$ ;
5: while stopping criterion not met do
6:   Apply genetic operators on  $P$  to generate offspring population  $\bar{P}$ ;
7:    $Q := P \cup \bar{P}$ ; /*parent and offspring are combined*/
8:   Normalize objectives of members in  $Q$  and partition  $Q$  into different subspaces;
9:    $(L_1, L_2, \dots) := \text{diversity-first-sort}(Q)$ ; /*diversity-first sorting is triggered*/
10:   $(S_1, S_2, \dots) := \text{nondominated-sort}(L_i)$ ; /* $L_i$  is the last front to be included*/
11:  Select continuously individuals from  $(S_1, S_2, \dots)$  until  $P$  is filled up;
12: end while

```

---

Diversity-first Sorting: to facilitate sorting, individuals should be distinguishable in terms of convergence. In this paper, DBEA simply applies nondominated sorting in each subspace, leading to each individual having a rank. Individuals with better rank values are priorly selected. In case there is a tie between individuals, those having the smallest perpendicular distance to the associated reference direction are preferred.

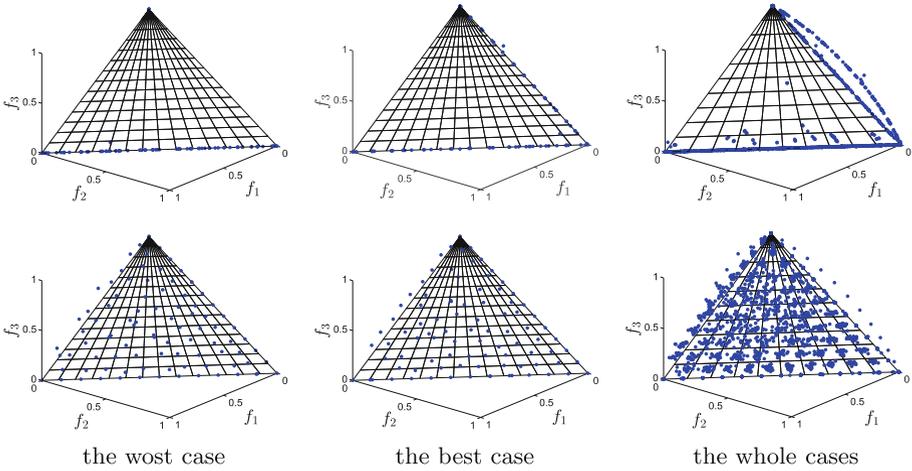
Selection on the last front to be included: DBEA performs the nondominated sorting on the last allowed front  $L_i$ , resulting in a series of subfronts  $\{S_1, S_2, \dots\}$ . Then, DBEA selects individuals on these subfronts, starting from the first subfront  $S_1$ . If the last subfront to be included has more individuals than the remaining slots, individuals are randomly copied to the new population.

## 5 Experimental Study

To make a proper and fair comparison, algorithms to be considered should have similar framework other than different methodologies. In our experiments, we would like to compare DBEA with the recently-developed NSGA-III algorithm [6]. DBEA and NSGA-III have very similar framework but differ mainly in distinct sorting methods. For both algorithms, the simulated binary crossover (SBX) [3] and polynomial mutation [5] are used as variation operators. As suggested in [6], the crossover probability is  $p_c = 1.0$  and its distribution index is  $\eta_c = 30$ . The mutation probability is  $p_m = 1/n$  and its distribution  $\eta_m = 20$ . In the following subsections, DBEA and NSGA-III will be first tested on a hard problem that challenges algorithms' diversity performance. After that, these two algorithms will be compared on many-objective optimization.

### 5.1 Results on a Hard Three-Objective Problem

Liu *et al.* [8] introduced several hard-to-converge problems with considerably deceptive properties and strong variable linkages. As a testing example, we choose the three-objective MOP6 to distinguish the difference between diversity-first sorting and convergence-first sorting. MOP6 places deceptive attractors on



**Fig. 3.** Approximated POFs for MOP6 over 20 runs. Top: NSGA-III; bottom: DBEA.

the boundary of the objective space. If population diversity is not properly maintained, the search will get trapped into local optima. As a consequence, not all the POF regions can be found. Both algorithms use the systematic design method [4] to generate 91 reference directions or points. Correspondingly, the population size in DBEA and NSGA-III was set to 92, which, as suggested in NSGA-III, is the smallest multiple of four higher than reference points. The maximum number of generations was set to 5000, which is much higher than normal settings due to the hardness of this problem.

Figure 3 shows the worst-case, best-case, and whole MOP6 approximations of NSGA-III and DBEA over 20 independent runs. It can be clearly observed from the figure that, in all runs, NSGA-III prefers some boundary solutions and misses a large part of the POF of MOP6. In contrast, DBEA is always capable of obtaining a set of diversified solutions, although a few boundary solutions do not converge perfectly. The poor performance of NSGA-III is mainly caused by its convergence-first based selection. In NSGA-III, environmental selection is based on individuals' convergence level (nondominated sorting). That is, individuals on better sorting fronts have priority to be selected first. If the selected individuals all reside in a local search space (the boundary region in the case of MOP6), the evolution will experience a dramatic diversity loss, resulting in NSGA-III not being able to diversify the solution set any more. Thus, NSGA-III fails in this situation. On the contrary, the diversity-first based selection seems to be a wise option, as it can maintain population diversity at a high level. Therefore, DBEA shows better performance than NSGA-III on the considered test problem.

### 5.2 Results on WFG Problems

The previous subsection has demonstrated the superiority of DBEA over NSGA-III in the three-objective case in terms of diversity. One may wonder

**Table 1.** Best, mean, and worst IGD values of compared algorithms over 20 runs

Problem	M=3		M=5		M=8		
	NSGA-III	DBEA	NSGA-III	DBEA	NSGA-III	DBEA	
WFG4	<b>0.00946</b>	0.01068	<b>0.28054</b>	0.28103	0.71357	<b>0.71239</b>	
	<b>0.01300</b>	0.01313	– 0.28338	<b>0.28310</b>	≈ 0.72304	0.72473	≈
	<b>0.01714</b>	0.01661	0.28749	<b>0.28557</b>	<b>0.73107</b>	0.74052	
WFG5	0.09445	<b>0.09386</b>	<b>0.34631</b>	0.34813	0.75591	<b>0.75563</b>	
	0.09683	<b>0.09483</b>	† 0.35063	<b>0.34879</b>	† 0.75870	<b>0.75820</b>	†
	0.09917	<b>0.09559</b>	0.35283	<b>0.35015</b>	0.76268	<b>0.76214</b>	
WFG6	<b>0.07036</b>	0.07828	0.34445	<b>0.33282</b>	<b>0.74635</b>	0.74721	
	<b>0.12819</b>	0.13388	– <b>0.37310</b>	0.37950	≈ 0.79313	<b>0.78546</b>	≈
	<b>0.16738</b>	0.19958	<b>0.40561</b>	0.41077	0.83568	<b>0.82690</b>	
WFG7	0.00993	<b>0.00920</b>	<b>0.27209</b>	0.27646	0.70625	<b>0.70584</b>	
	<b>0.01232</b>	0.01322	– 0.27733	<b>0.27720</b>	≈ 0.72276	<b>0.72168</b>	†
	<b>0.01553</b>	0.01681	0.27974	<b>0.27943</b>	0.74034	<b>0.73439</b>	
WFG8	0.23728	<b>0.23639</b>	0.58692	<b>0.57160</b>	1.45400	<b>1.27610</b>	
	0.25205	<b>0.24869</b>	† 0.61987	<b>0.60271</b>	† 1.57690	<b>1.57910</b>	†
	0.26632	<b>0.26161</b>	0.62657	<b>0.61874</b>	1.73940	<b>1.72546</b>	
WFG9	0.07266	<b>0.05796</b>	0.39788	<b>0.38946</b>	<b>0.91319</b>	0.92767	
	0.11684	<b>0.11420</b>	† 0.42125	<b>0.41719</b>	† 1.00700	<b>0.99933</b>	†
	0.36338	<b>0.36184</b>	0.48428	<b>0.45785</b>	1.19820	<b>1.12270</b>	

whether DBEA can perform well in higher-dimensional cases. To this end, we test DBEA and NSGA-III on several WFG [7] test problems having three to eight objectives. Both algorithms use the same population size by setting identical reference directions or points with the two-layered method [6]. That is, 92, 210, and 156 for 3, 5, and 8 objectives, respectively. The maximum number of generations was 500, 1000, and 1500 for 3, 5, and 8 objectives, respectively. Each algorithm was executed 20 independent runs.

In order to quantify the performance of algorithms, we employ the reference point based inverted generational distance (IGD) suggested by [6] and hypervolume (HV) [14] as our performance metrics. The reference vector for the computation of HV was set as the nadir point of the true POF plus one. All reported HV values come from the normalization of originally computed HV values.

Tables 1 and 2 present the IGD and HV values of two compared algorithms, respectively, where the best results are highlighted in bold face. The Wilcoxon rank-sum test at a 0.05 significance level was employed to compare the statistical significance of difference between two algorithms. “†”, “≈”, and “–” in the tables denote DBEA is better than, equivalent to, and worse than NSGA-III, respectively. It is easy to see that, the performance of DBEA improves as the number of objectives increases. For three objectives, NSGA-III performs better than DBEA on WFG4, WFG6 and WFG7 whereas DBEA wins on the other

**Table 2.** Best, mean, and worst HV values of compared algorithms over 20 runs

Problem	M=3		M=5		M=8		
	NSGA-III	DBEA	NSGA-III	DBEA	NSGA-III	DBEA	
WFG4	<b>0.85281</b>	0.85230	<b>0.94849</b>	0.94844	0.98060	<b>0.98074</b>	
	<b>0.85191</b>	0.85191	0.94808	<b>0.94809</b>	≈ 0.97992	<b>0.97995</b>	†
	<b>0.85145</b>	0.85151	0.94758	<b>0.94764</b>	<b>0.97908</b>	0.97836	
WFG5	0.82652	<b>0.82655</b>	0.91367	<b>0.91393</b>	0.93682	<b>0.93701</b>	
	0.82595	<b>0.82639</b>	† 0.91307	<b>0.91379</b>	† 0.93637	<b>0.93682</b>	†
	0.82541	<b>0.82614</b>	0.91244	<b>0.91321</b>	0.93527	<b>0.93642</b>	
WFG6	<b>0.83362</b>	0.83117	0.91607	<b>0.92151</b>	0.94384	<b>0.94624</b>	
	<b>0.81705</b>	0.81540	– <b>0.90336</b>	0.90120	≈ 0.92088	<b>0.92489</b>	†
	<b>0.80628</b>	0.79718	<b>0.89009</b>	0.88789	0.89975	<b>0.90262</b>	
WFG7	<b>0.85218</b>	0.85217	<b>0.94875</b>	0.94869	0.98069	<b>0.98071</b>	
	<b>0.85180</b>	0.85169	– 0.94838	<b>0.94841</b>	† 0.98030	<b>0.98042</b>	†
	<b>0.85120</b>	0.85120	0.94799	<b>0.94801</b>	0.97975	<b>0.97988</b>	
WFG8	0.80278	<b>0.80351</b>	0.88415	<b>0.88548</b>	0.91022	<b>0.92898</b>	
	0.80088	<b>0.80167</b>	† 0.88249	<b>0.88358</b>	† 0.89509	<b>0.90005</b>	†
	0.79854	<b>0.79838</b>	0.87989	<b>0.88171</b>	0.88229	<b>0.88473</b>	
WFG9	0.83951	<b>0.84112</b>	0.92595	<b>0.92771</b>	<b>0.94890</b>	0.94712	
	0.83030	<b>0.83188</b>	† 0.92242	<b>0.92445</b>	† <b>0.93507</b>	0.93103	≈
	0.74526	<b>0.74771</b>	0.91616	<b>0.92146</b>	0.81770	<b>0.81948</b>	

problems. For five and eight objectives, DBEA generally obtains better results than NSGA-III in terms of IGD and HV. Since NSGA-III is a leading method for many-objective optimization, such observation implies DBEA can perform well in the case of many objectives. Thus, diversity-first based selection is effective and applicable to many-objective optimization.

## 6 Conclusions

While convergence-first based MOEAs have been increasingly reported to be effective in solving a variety of MOPs, they may come across difficulties in maintaining population diversity, resulting in a poor approximation of the POF. For this reason, this paper has suggested a new diversity-first sorting method to overcome the difficulty of convergence-first sorting. The advantages and disadvantages of convergence-first and diversity-first sorting methods have been briefly discussed. Afterwards, a new algorithm based on the proposed sorting method, i.e., DBEA, has been suggested.

The proposed DBEA has been examined and compared with the recently-developed NSGA-III algorithm on several test problems with different optimization difficulties. Experimental results have shown that DBEA has great

advantages in maintaining diversity for problems where NSGA-III fails. Furthermore, DBEA has also great potential for many-objective optimization, as indicated by its outperformance over NSGA-III in many cases.

Inspired by these encouraging performance, we would like to extend the current work to other classes of MOEAs, such as indicator-based selection methods and decomposition-based MOEAs in the future. Also, the convergence part of DBEA needs to be investigated. Different fitness assignment techniques will be integrated into the diversity-first sorting, and their suitability and effectiveness will be investigated.

**Acknowledgments.** This work was funded by the Engineering and Physical Sciences Research Council (EPSRC) of U.K. under Grant EP/K001310/1.

## References

1. Chen, B., Zeng, W., Lin, Y., Zhang, D.: A new local search-based multiobjective optimization algorithm. *IEEE Trans. Evol. Comput.* **19**(1), 50–73 (2015)
2. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
3. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. *Complex Syst.* **9**(4), 115–148 (1995)
4. Das, I., Dennis, J.: Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* **8**(3), 631–657 (1998)
5. Deb, K., Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design. *Comput. Sci. Inf.* **26**(4), 30–45 (1996)
6. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**(4), 577–601 (2014)
7. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **10**(2), 477–506 (2006)
8. Liu, H., Gu, F., Zhang, Q.: Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Trans. Evol. Comput.* **18**(3), 450–455 (2014)
9. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)
10. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm. Technical report, ETH Zürich, Zürich, Switzerland (2001)
11. Salinas, S., Li, M., Li, P.: Multi-objective optimal energy consumption scheduling in smart grids. *IEEE Trans. Smart Grid* **4**(1), 341–348 (2013)
12. Fesanghary, M., Asadi, S., Geem, Z.W.: Design of low-emission and energy-efficient residential building using a multi-objective optimization algorithm. *Build. Environ.* **49**, 245–250 (2012)
13. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) *PPSN 2004*. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
14. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.* **3**(4), 257–271 (1999)