A Decomposition Based Evolutionary Algorithm for Many Objective Optimization with Systematic Sampling and Adaptive Epsilon Control

Md. Asafuddoula, Tapabrata Ray, and Ruhul Sarker

School of Engineering and Information Technology
University of New South Wales, Canberra, Australia
Md.Asaf@student.adfa.edu.au

Abstract. Decomposition based evolutionary approaches such as MOEA/D and its variants have been quite successful in solving various classes of two and three objective optimization problems. While there have been some attempts to modify the dominance based approaches such as NSGA-II and SPEA2 to deal with many-objective optimization, there are few attempts to extend the capability of decomposition based approaches. The performance of a decomposition based approach is dependent on (a) the mechanism of reference points generation i.e. one which needs to be scalable and computationally efficient (b) the method to simultaneously deal with conflicting requirements of convergence and diversity and finally (c) the means to use the information of neighboring subproblems efficiently. In this paper, we introduce a decomposition based evolutionary algorithm, wherein the reference points are generated via systematic sampling and an adaptive epsilon scheme is used to manage the balance between convergence and diversity. To deal with constraints efficiently, an adaptive epsilon formulation is adopted. The performance of the algorithm is highlighted using standard benchmark problems i.e. DTLZ1 and DTLZ2 for 3, 5, 8, 10 and 15 objectives, the car side impact problem, the water resource management problem and the constrained ten-objective general aviation aircraft (GAA) design problem. The study clearly highlights that the proposed algorithm is better or at par with recent reference direction based approaches.

Keywords: many-objective optimization, generation of reference points, adaptive epsilon comparison, constraint-handling.

1 Introduction

Many objective optimization typically refers to problems with the number of objectives greater than four [1]. There is significant amount of literature discussing the challenges involved in solving them and interested readers may refer to [1] for further details. The commonly used dominance based methods for multi-objective optimization, such as NSGA-II, SPEA2 etc. are known to be inefficient for many-objective optimization as non-dominance does not provide adequate selection pressure to drive the population towards convergence. There has been a number of attempts to modify the underlying selection pressure through the use of substitute distance measures [2][3], average rank

R.C. Purshouse et al. (Eds.): EMO 2013, LNCS 7811, pp. 413–427, 2013.

domination [4], fuzzy dominance [5], ε-dominance [6][7], adaptive ε-ranking [8] etc. without great success. In all the above approaches, while the diversity and the convergence of the population improved during the course of evolution, there is no guarantee that the final non-dominated set spans the entire Pareto surface uniformly.

There are also radically different approaches to deal with many objective optimization, such as attempts to identify the reduced set of objectives [9] or corners of the Pareto front [10] and subsequently solving the problem using these reduced set of objectives. Other attempts include interactive use of decision makers preferences [11], use of reference points [12][13] or solution of the problem as a hypervolume maximization problem [14]. While some progress has been made along these lines, the limiting factors include the inability to obtain solutions close to Pareto set for an accurate identification of redundant objectives, decision making burden associated with preference elicitation and the computational complexity of hypervolume computation.

Decomposition based evolutionary algorithms are yet another class of algorithms originally introduced as MOEA/D [15], wherein the multiobjective optimization problem is decomposed into a series of scalar optimization problems. In a decomposition based approach, one need to generate uniformly distributed reference directions and adopt a method of scalarization. In the context of many objective optimization, the first issue relates to the design of a computationally efficient scheme to generate W uniform reference directions for a M objective optimization problem, where M is typically more than four and W is of the same order as the population size. The second issue relates to scalarization, which essentially assigns the *fittest* individual to each reference direction. The notion of *fittest* is essentially derived using a tradeoff between convergence and diversity measured with respect to any given reference direction. One of the early attempts to generate uniformly distributed reference directions appear in the works of Hughes [13] . The method was not computationally efficient for problems with more than six objectives and often resulted in a large number of reference directions that in turn required a huge population size. More recently, computationally efficient and scalable sampling schemes have been used in the context of many-objective optimization. A systematic sampling [16] scheme has been used in M-NSGA-II [17] while an uniform sampling scheme has been used within MOEA/D [18] to deal with many objective optimization problems.

The second issue related to scalarization has been addressed via two fundamental means i.e. through a systematic association and niche preservation mechanism as in M-NSGA-II [17] or through the use of a penalty function(i.e. an aggregation of the projected distance along a reference direction and the perpendicular distance from a point to a given reference direction) within the framework of MOEA/D. The performance of the penalty function based approach is dependent on the penalty parameter, while the association and the niche preservation process require a careful implementation to address a number of possibilities.

In this paper, we introduce a decomposition based evolutionary algorithm for many-objective optimization. The reference directions are generated using systematic sampling, wherein the points are systematically generated on a hyperplane with unit intercepts in each objective axis. The process of reference point generation is the same as adopted in M-NSGA-II [17]. The fine balance between convergence and diversity along a reference direction is managed using an adaptive epsilon model eliminating the need for the penalty parameter. While M-NSGA-II [17] is a generational model, our proposed algorithm is a steady state form. Furthermore, to deal with constraints, an adaptive epsilon level based scheme is introduced which has been demonstrated to be more effective over *feasibility first* schemes in the context of constrained optimization [19].

The details of the proposed algorithm are presented in Section 2. The performance of the proposed algorithm on benchmark problems (DTLZ1 and DTLZ2 for 3, 5, 8, 10 and 15 objectives) is presented and compared with MOEA/D-PBI and M-NSGA-II in Section 3. In addition to the above set of mathematical benchmarks, the performance of the algorithm is also compared using a number of engineering design problems (car side impact, water resource management and the constrained ten-objective general aviation aircraft (GAA) design). The final section summarizes the contributions and future directions for further improvement.

2 Proposed Algorithm

A many-objective optimization problem can be defined as follows:

$$\begin{aligned} min.\ & [f_1(x), f_2(x), f_3(x), ...f_M(x)], x \in \Omega \\ S.t. \quad & g_j(x) \leq 0, j = 1, 2, ...p \\ & h_k(x) = 0, k = 1, 2, ...q \end{aligned} \tag{1}$$

where $f_1(x), f_2(x), f_3(x), ...f_M(x)$ are the M objective functions, p is the number of inequalities and q is the number of equalities.

The pseudocode of the algorithm is presented below and the subsequent components are discussed in the following subsections.

Algorithm 1. DBEA-Eps

Input: Gen_{max} maximum number of generations, W the number of reference points

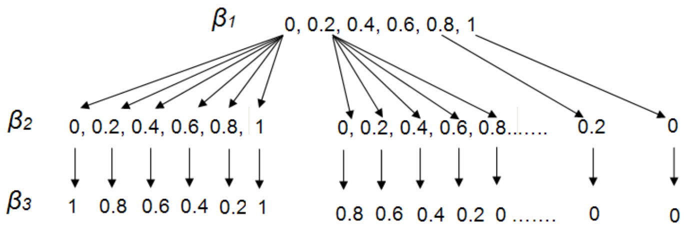
Generate the reference points and assign their neighborhood
Initialize the population P; $|P|$ = W
Evaluate the initial population and compute the ideal point $\bar{z}_j = (f_1^{min}, f_2^{min}, ..., f_M^{min})$ and intercepts a_i's for $i = 1$ to M
Scale the individuals of the population
while $(gen \leq Gen_{max})$ **do**
  for i=1:W **do**
    Assign the base parent as P_i
    I=Select a mating partner for (P_i)
    Create a child via recombination as C_i
    Evaluate C_i and compute the distances ($d1$ and $d2$) using all reference directions
    Replace the parent P_k with C_i using *single-first encounter*, where k denotes the index of the first parent satisfying the condition of replacement
    Update the ideal point ($\bar{z}$), the intercepts and re-scale the population
  end for
end while

The algorithm consists of four major components i.e. (a) generation of reference directions and assignment of neighborhood (b) computation of distances along and perpendicular to each reference direction (c) method of recombination using information from neighboring subproblems and finally (d) adaptive epsilon comparison to manage the balance between convergence and diversity.

2.1 Generation of Reference Points and Assignment of Neighborhood

A structured set of reference points (β) is generated spanning a hyperplane with unit intercepts in each objective axis using the algorithm outlined in [16]. The approach generates W points on the hyperplane with a uniform spacing of $\delta = 1/p$ for any number of objectives M. The process of generation of the reference points is illustrated for a 3-objective optimization problem i.e. (M=3) and with an assumed spacing of $\delta = 0.2$ i.e ($p = 5$) in Figure 1. The process results in the generation of 21 reference points.

$$W = {}^{(M+p-1)}C_p \tag{2}$$



(a)

β_1	β_2	$(1-\beta_1-\beta_2)$
0	0	1.0000
0	0.2000	0.8000
0	0.4000	0.6000
0	0.6000	0.4000
0	0.8000	0.2000
0	1.0000	0
0.2000	0	0.8000
0.2000	0.2000	0.6000
0.2000	0.4000	0.4000
0.2000	0.6000	0.2000
0.2000	0.8000	0
0.4000	0	0.6000
0.4000	0.2000	0.4000
0.4000	0.4000	0.2000
0.4000	0.6000	0
0.6000	0	0.4000
0.6000	0.2000	0.2000
0.6000	0.4000	0
0.8000	0	0.2000
0.8000	0.2000	0
1.0000	0	0

(b)

Fig. 1. (a) the reference points are generated computing βs recursively (b) the table shows the combination of all βs in each column

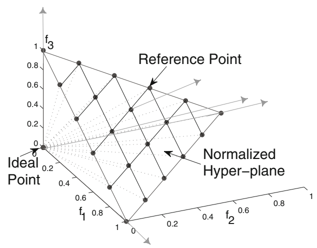


Fig. 2. A set of reference points in a normalized hyper-plane for number of objectives, $M = 3$ and $p = 5$

The distribution of the reference points are presented in Figure 2. The reference directions are formed by constructing a straight line from the origin to each of these reference points. The population size of the algorithm is set to the number of reference points. For every reference point, its neighborhood consists of T closest reference points computed based on a Euclidean distance amongst them. The initial population consists of W individuals generated randomly within the variable bounds. Such solutions are thereafter assigned randomly to a reference direction during the phase of initialization.

2.2 Computation of Distances along and Perpendicular to Each Reference Direction

Since in a generic many objective optimization problem, the objectives may assume negative values or values in varying orders of magnitude, it is important to scale them appropriately. The ideal point of a population is denoted by $\bar{z}_j = (f_1^{min}, f_2^{min}, \ldots.., f_M^{min})$ and the extreme point is denoted by $z_j{}^e = (f_1^{max}, f_2^{max}, \ldots.., f_M^{max})$. A hyperplane is created using the solutions that have led to the coordinates of the extreme point. The intercepts of the hyperplane along the objective axes are denoted by $a_1, a_2, \ldots., a_M$. The generic equation of a plane through these points can be represented using the following equation

$$Af_1 + Bf_2 + \ldots.. + Cf_M = 1 \tag{3}$$

where, A, B,....,C are the unit normal of the plane. The intercepts of the plane with the axis are given by $a_1 = 1/A$, $a_2 = 1/B$,....., and $a_M = 1/C$.

In the event, the number of such solutions are less than M or any of the a_i's are negative, a_i's are set to f_i^{max}. Every solution in the population is subsequently scaled as follows:

$$f_j^{'}(x) = \frac{f_j(x) - \bar{z}_j}{a_j - \bar{z}_j}, \forall j = 1, 2, \ldots M \tag{4}$$

For any given reference direction, the performance of a solution can be judged using two measures d_1 and d_2 as depicted in Equation 8. The first measure d_1 is the Euclidean distance between origin and the foot of the normal drawn from the solution to the reference direction, while the second measure d_2 is the length of the normal. Mathematically, d_1 and d_2 are computed as follows:

$$d1 = w^T f_j^{'}(x) \tag{5}$$

$$d2 = ||f_j^{'}(x) - w^T f_j^{'}(x) w|| \tag{6}$$

where w is a unit vector along any given reference direction. It is clear that a value of $d_2 = 0$ ensures the solutions are perfectly aligned along the required reference direction ensuring perfect diversity, while a smaller value of d_1 indicates superior convergence. These two measures are subsequently used to control diversity and convergence of the algorithm via an adaptive epsilon scheme.

2.3 Mating Partner Selection

The information and similarity of neighboring subproblems are exploited via the process of parter selection. The mating partner for P_i (where i is the index of the current individual in a population) is selected using of the following rules i.e. rule 1: select a parent from the neighborhood with a probability of τ and rule 2: select a random parent from the population with a probability of $(1-\tau)$.

2.4 Method of Recombination

In the recombination process, two child solutions are generated using simulated binary crossover (SBX) operator [20] and polynomial mutation. The first child is considered as an individual attempting to replace any parent in the population.

2.5 Adaptive Epsilon Comparison to Manage the Balance between Convergence and Diversity

Since every solution is assigned to a reference direction, the average deviation ε_{CD} for the population of solutions is computed using Equation 7, where d_{2i} denotes the d_2 measure of the i^{th} individual in the population.

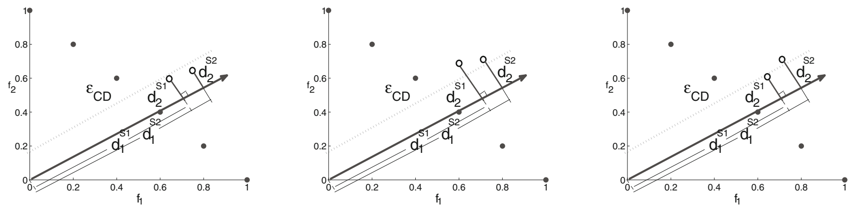
$$\varepsilon_{CD} = \frac{\sum_{i=1}^{W} d_{2i}}{W} \tag{7}$$

whenever a child solution is created, its d_2 measure is computed along all reference directions and the child solution replaces a single parent based on the following rule Equation 8.

$$(d_1,d_2) <_{\varepsilon_{CD}} (d_1,d_2) \Leftrightarrow \begin{cases} d_1 < d_2, \text{ if } & d_2, d_2 < \varepsilon_{CD} \\ d_1 < d_2, \text{ if } & d_2 = d_2 \\ d_2 < d_2, \text{ otherwise} \end{cases} \tag{8}$$

It is also worth noting that this process is a *single-first encounter* replacement scheme whereby, the child solution can only replace a single parent and the first encountered parent meeting the condition is replaced. Whenever a replacement is successful, a check is performed to identify if there is a need to re-compute the ideal point or the intercepts. The population needs to be re-scaled in the event the ideal point or the intercepts have changed.

The possible epsilon comparison scenarios are presented in Fig. 7 as Case 1, Case 2 and Case 3. Let us assume the parent solution is denoted by $(s1)$ and the child solution is denoted by $(s2)$). Case 1: Both the solutions have their d_2 values less than ε_{CD}. One with the smaller d_1 is selected i.e.$(s1)$. Case 2: Both the d_2 values are more than ε_{CD}. One with the lower d_2 value is selected i.e.$(s2)$. Case 3: One solution has its d_2 value more than ε_{CD} and the other has its d_2 value less than ε_{CD}. One with the smaller d_2 value is selected i.e. $(s1)$.



(a) (b) (c)

Fig. 3. (a) Case 1 (b) Case 2 (c) Case 3

2.6 Constraint Handling

The constraint handling approach used in this work is based on epsilon level comparison and has been reported earlier in [21]. The feasibility ratio (FR) of a population refers to the ratio of the number of feasible solutions in the population to the number of solutions (W). The allowable violation is calculated as follows:

$$CV = \sum_{i=1}^{p} max(g_i, 0) + \sum_{i=1}^{q} max(|h_i - \varepsilon|, 0) \quad (9)$$

$$CV_{mean} = \frac{1}{W} \sum_{j=1}^{W} (CV_j) \quad (10)$$

$$\text{Allowable violation}(\varepsilon_{CV}) = CV_{mean} * FR \quad (11)$$

An epsilon level comparison using this allowable violation measure is used to compare two solutions. If two solutions have their constraint violation value less than this epsilon level, the solutions are compared based on their objective values i.e. via d_1 and d_2 measures. Such a constraint handling scheme has been demonstrated to be more efficient than *feasibility first* schemes.

3 Experimental Results

In this section, we present the results of proposed decomposition based evolutionary algorithm (DBEA-Eps) and compare its performance with M-NSGA-II and MOEA/D-PBI [22] for DTLZ1 and DTLZ2 problems with 3,5,8,10 and 15 objectives.

The population sizes used in this study are the same as those adopted in [22]. In our proposed algorithm, the probability of crossover is set to 1 and the probability of mutation is set to $p_m = 1/D$, where D is the dimensionality of the problem. The distribution index of crossover is set to η_c=30 and the distribution index of mutation is set to η_m=20 as in [22]. The probability of selecting parent from its neighborhood (τ) is set to 0.9 and the neighborhood size is set to 20.

To assess the performance, we have selected IGD [23][15] as a performance metric. The IGD metric in our simulation results is calculated by normalizing the approximated set with the theoretical ideal and nadir points for the DTLZ problems.

3.1 Performance on Unconstrained DTLZ Problems

In this comparison, we have reported the best, median and worst IGD results obtained using 20 independent runs for DTLZ1 and DTLZ2. The results are compared against M-NSGA-II and MOEA/D-PBI in Table 3. In Fig 4 and Fig 5, the final Pareto front is shown for three-objective problems of DTLZ1 and DTLZ2.

Table 1. IGD statistics for problems DTLZ1 and DTLZ2 using 20 independent runs

Test Problem	Obj.	MaxGen	Strategy	Best	Median	Worst
DTLZ1	3	400	DBEA-Eps	**8.771e-5**	9.521e-3	5.854e-1
			M-NSGA-II	4.880e-4	1.308e-3	4.880e-3
			MOEA/D-PBI	4.095e-4	**1.495e-3**	**4.743e-3**
DTLZ1	5	600	DBEA-Eps	**1.771e-5**	**2.183e-4**	3.782e-1
			M-NSGA-II	5.116e-4	9.799e-4	1.979e-3
			MOEA/D-PBI	3.179e-4	6.372e-4	**1.635e-3**
DTLZ1	8	750	DBEA-Eps	**4.387e-5**	**3.581e-4**	**1.981e-3**
			M-NSGA-II	2.044e-3	3.979e-3	8.721e-3
			MOEA/D-PBI	3.914e-3	6.106e-3	8.537e-3
DTLZ1	10	1000	DBEA-Eps	**7.691e-4**	**1.504e-3**	**2.700e-3**
			M-NSGA-II	2.215e-3	3.462e-3	6.869e-3
			MOEA/D-PBI	3.872e-3	5.073e-3	6.130e-3
DTLZ1	15	1500	DBEA-Eps	**1.696e-3**	**2.606e-3**	**2.686e-3**
			M-NSGA-II	2.649e-3	5.063e-3	1.123e-2
			MOEA/D-PBI	1.236e-2	1.431e-2	1.692e-2
DTLZ2	3	250	DBEA-Eps	2.040e-2	4.138e-2	6.417e-2
			M-NSGA-II	1.262e-3	1.357e-3	2.114e-3
			MOEA/D-PBI	**5.432e-4**	**6.406e-4**	**8.006e-4**
DTLZ2	5	350	DBEA-Eps	**1.199e-3**	3.024e-3	2.272e-2
			M-NSGA-II	4.254e-3	4.982e-3	5.862e-3
			MOEA/D-PBI	1.219e-3	**1.437e-3**	**1.727e-3**
DTLZ2	8	500	DBEA-Eps	**1.172e-3**	**2.899e-3**	6.915e-3
			M-NSGA-II	1.371e-2	1.571e-2	1.811e-2
			MOEA/D-PBI	3.097e-3	3.763e-3	**5.198e-3**
DTLZ2	10	750	DBEA-Eps	3.656e-3	3.657e-3	3.657e-3
			M-NSGA-II	1.350e-2	1.528e-2	1.697e-2
			MOEA/D-PBI	**2.474e-3**	**2.778e-3**	**3.235e-3**
DTLZ2	15	1000	DBEA-Eps	**5.160e-3**	**5.960e-3**	**5.960e-3**
			M-NSGA-II	1.360e-2	1.726e-3	2.114e-2
			MOEA/D-PBI	5.254e-3	6.005e-3	9.409e-3

One can observe that our algorithm obtained the best IGD values in 8 instances out of 10. In terms of the median performance, our algorithm was the best in 6 instances thereby indicating competitive performance with recently proposed forms.

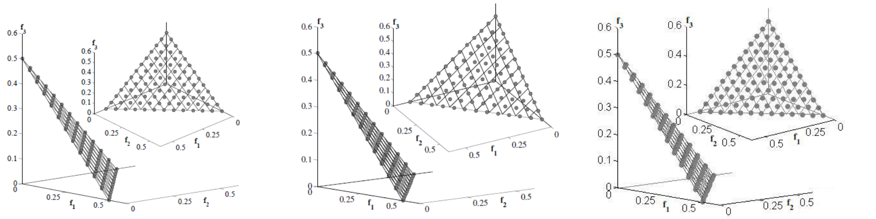


Fig. 4. Obtained solutions by (a) M-NSGA-II (b) MOEA/D-PBI (c) DBEA-Eps for DTLZ1

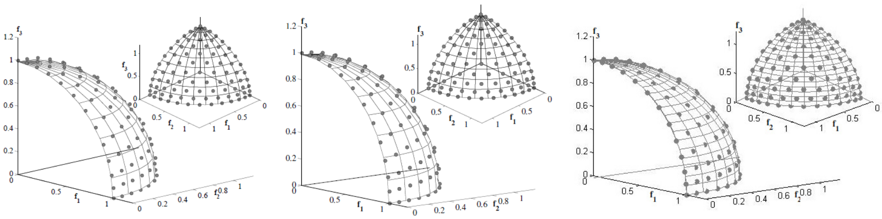
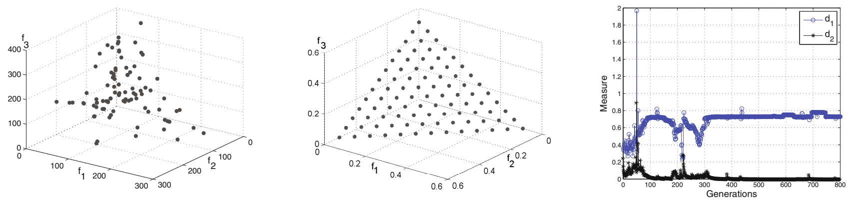


Fig. 5. Obtained solutions by (a) M-NSGA-II (b) MOEA/D-PBI (c) DBEA-Eps for DTLZ2

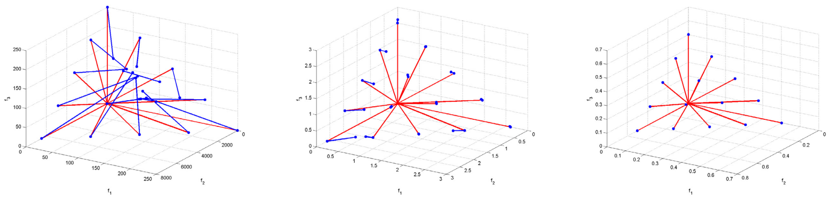
In order to observe the process of evolution, we computed the average performance of the population i.e. average of the d_1 and d_2 values for the individuals for DTLZ1 (3 objectives). One can observe from Figure 6, that the average d_2 converges to near zero (i.e. near perfect alignment to the reference directions) while the average d_1 measure stabilizes at around 0.5 indicating convergence to the Pareto front.



(a) (b) (c)

Fig. 6. (a) the initial population of DTLZ1 test problem for number of objectives 3 (b) the final Pareto-front of DTLZ1 test problem for number of objectives 3 (c) the convergence of distance measure over the generations

The association mechanism (i.e. solutions to each reference direction) for a 3-objective DTLZ1 problem is presented in Figure 7. The figure shows the associations in generation 1, 200 and 400 using 15 reference points. One can observe that although initially the association is random, the solutions automatically get associated to the closest reference directions during the course of evolution via the pressure induced by $d2$. This alleviates the need of an extensive niching and association operation as encountered in M-NSGA-II [17].



(a) (b) (c)

Fig. 7. (a) the initial population of DTLZ1 test problem for number of objectives 3 with 15 reference points (b) at generation 200 (c) at final generation 400

4 Constrained Engineering Design Problems

Since the performance of the proposed algorithm was competitive on unconstrained test problems, we investigated its performance on three constrained engineering design optimization problems i.e. the three-objective car-side-impact problem [24] with ten inequality constraints, five-objective water resource management problem [25] with seven inequality constraints and finally the ten objective general aviation aircraft (GAA) design problem [7] having a single inequality constraint.

4.1 Car Side Impact Problem

The problem aims to minimize the weight of a car, the pubic force experienced by a passenger and the average velocity of the V-Pillar responsible for bearing the impact load subject to the constraints involving limiting values of abdomen load, pubic force, velocity of V-Pillar, rib deflection etc [24].

The problem is solved using DBEA-Eps and MOEA/D-PBI. The algorithms are run for 500 generations and the final non-dominated front is shown in Fig 8.It is important to note that the results of MOEA/D-PBI is derived without scaling which could be a reason among others for poor performance.

4.2 Water Resource Management Problem

This is a five objective problem having seven constraints taken from the literature [25]. The parallel coordinate plot generate using our proposed algorithm (DBEA-Eps) is presented in Fig 9. The best IGD value across 20 runs is $3.29e-2$ and the IGD is computed using the reference set of 2429 solutions [26]. A population of 210 solutions has been used and evolved over 1000 generations.

In Fig 10, a scatter plot-matrix is presented. The results from the DBEA-Eps are shown in the top-right plots vis-a-vis the known reference set of 2429 solutions (shown in bottom-left plots).

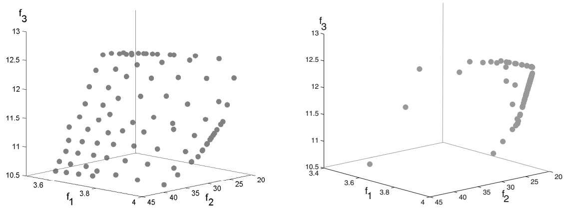


Fig. 8. Solutions obtained using (a) DBEA-Eps (b) MOEA/D-PBI on three-objective car side impact problem

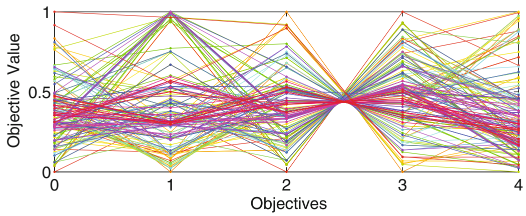


Fig. 9. Solutions obtained using DBEA-Eps on five-objective water problem

4.3 General Aviation Aircraft (GAA) Design Problem

This problem was first introduced by Simpson et al. [27] and has been recently solved using an evolutionary algorithm [7]. The problem involves 9 design variables i.e. cruise speed, aspect ratio, sweep angle, propeller diameter, wing loading, engine activity factor, seat width, tail length/ diameter ratio and taper ratio and the aim is to minimize the takeoff noise, empty weight, direct operating cost, ride roughness, fuel weight, purchase price, product family dissimilarity and maximize the flight range, lift/ drag ratio and cruise speed. Previous studies encountered difficulties in obtaining feasible solutions due to tight constraints [27].

In this example, we have used 100 reference points and the population was allowed to evolve over 5000 generations. A reference set of 412 non-dominated solutions obtained from ε-MOEA and Borg-MOEA is used to compute the IGD metric. The results of the proposed algorithm are compared with four other algorithms i.e. ε-MOEA, Borg-MOEA, MOEA/D and ε-NSGA-II [7]. We have also computed the hypervolume using the ideal point of (i.e.[73.251, 1881.5, 59.114, 1.7977, 359.92, 41879, -2580.2, -16.823, -204.02, 0.26847]) and the extreme point of (i.e.[74.036, 2011.5, 79.993, 2, 483.13, 44590, -2000, -14.408, -189.3, 1.9844]) obtained from the reference set. The performance of the algorithms are compared using the hypervolume in Table 2 and IGD in Table 3. One can observe that the proposed algorithm performs marginally better than others for this problem.

Figure 11 shows the parallel coordinate plot. The figure clearly shows that DBEA-Eps is able to find a widely distributed set of nondominated points for 10-objective GAA design problem.

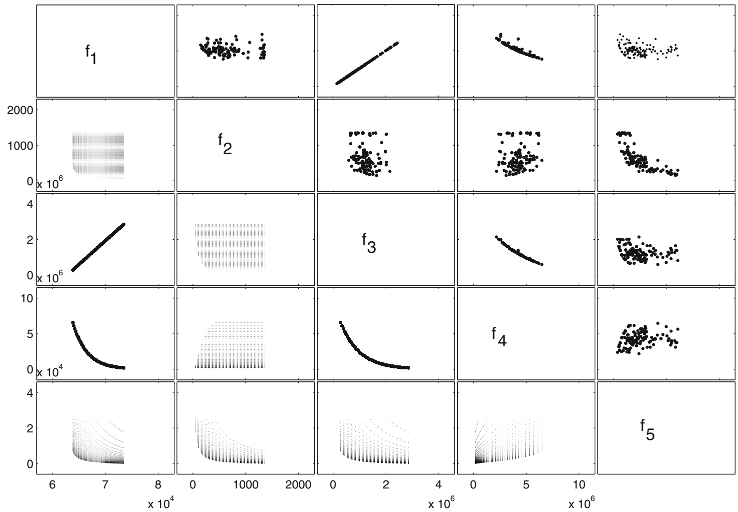


Fig. 10. A scatter plotmatrix showing DBEA-Eps (top-right plots) vis-a-vis the known reference set of 2429 solutions (bottom-left plots)

Table 2. Performance metric value of product family design problem using 50 independent runs

Algorithm	Function Evaluation	Hypervolume			
		Best	Mean	Worst	Std
DBEA-Eps	50,000	0.02899	0.01715	0.00689	0.04561
ε-MOEA		0.02032	0.01032	0.00259	0.04125
Borg-MOEA		0.02245	0.01013	0.00424	0.02327
MOEA/D		0.00092	0.00087	0.00045	0.00145
ε-NSGA-II		0.01636	0.01005	0.00236	0.05232

Table 3. Performance metric value of product family design problem using 50 independent runs

Algorithm	Function Evaluation	IGD			
		Best	Mean	Worst	Std
DBEA-Eps	50,000	0.62070	0.80123	0.82430	0.09210
ε-MOEA		0.98312	0.99123	0.99678	0.10312
Borg-MOEA		0.98211	0.99113	0.99337	0.02321
MOEA/D		0.99117	0.99587	0.99723	0.02145
ε-NSGA-II		0.98571	0.98872	0.99131	0.72123

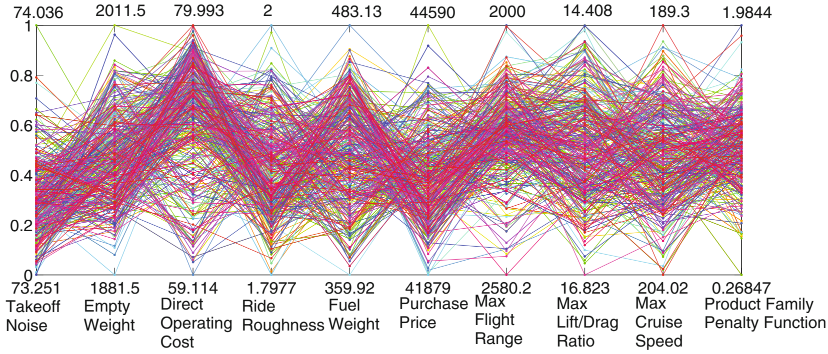


Fig. 11. Parallel coordinate plot of the approximation of Pareto set produced by DBEA-Eps is in a different color trace

5 Conclusion

In this paper, a decomposition based evolutionary algorithm with adaptive epsilon comparison is introduced to solve unconstrained and constrained many objective optimization problems. The approach utilizes reference directions to guide the search, wherein the reference directions are generated using a systematic sampling scheme as introduced by Das and Dennis [16]. The algorithm is designed using a steady state form. In an attempt to alleviate the problems associated with scalarization(commonly encountered in the context of reference direction based methods), the balance between diversity and convergence is maintained using an adaptive epsilon comparison. Such a process also eliminates the need for a detailed association and niching operation as employed in M-NSGA-II. In order to deal with constraints, an epsilon level comparison is used which is known to be more effective than methods employing *feasibility first* principles. The performance of the algorithm is presented using DTLZ1 and DTLZ2 problems with objectives ranging from 3 to 15. Furthermore, three constrained engineering design optimization problems with three to seven constraints (car side impact, water resource management and a general aviation aircraft design problem) have been solved to illustrate the performance of the proposed algorithm. The preliminary results indicate that the proposed algorithm is able to deal with unconstrained and constrained many-objective optimization problems better or at par with existing state of the art algorithms such as M-NSGA-II and MOEA/D-PBI.

Acknowledgement. The second author would like to acknowledge the support of Future Fellowship offered by the Australian Research Council.

References

1. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: Proc. IEEE World Congress Computational Intelligence, pp. 2419–2426 (2008)

2. Köppen, M., Yoshida, K.: Substitute Distance Assignments in NSGA-II for Handling Many-objective Optimization Problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 727–741. Springer, Heidelberg (2007)
3. Singh, H.K., Isaacs, A., Ray, T., Smith, W.: A Study on the Performance of Substitute Distance Based Approaches for Evolutionary Many Objective Optimization. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 401–410. Springer, Heidelberg (2008)
4. Kachroudi, S., Grossard, M.: Average rank domination relation for NSGA-II and SMPSO algorithms for many-objective optimization. In: Proc. Second World Congress Nature and Biologically Inspired Computing (NaBIC), pp. 19–24 (2010)
5. Wang, G., Wu, J.: A new fuzzy dominance GA applied to solve many-objective optimization problem. In: Proc. Second Int. Conf. Innovative Computing, Information and Control (ICICIC 2007), pp. 617–621 (2007)
6. Zou, X., Chen, Y., Liu, M., Kang, L.: A new evolutionary algorithm for solving many-objective optimization problems. IEEE Transactions on Systems, Man, and Cybernetics-Part B 38(5), 1402–1412 (2008)
7. Hadka, D., Reed, P.M., Simpson, T.W.: Diagnostic assessment of the Borg MOEA for many-objective product family design problems. In: IEEE World Congress on Computational Intelligence, Brisbane, Australia, pp. 10–15 (June 2012)
8. Aguirre, H., Tanaka, K.: Adaptive ε-ranking on MNK-landscapes. In: Proc. IEEE Symposium Computational Intelligence in Miulti-Criteria Decision-Making (MCDM 2009), pp. 104–111 (2009)
9. Saxena, D.K., Deb, K.: Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective. In: IEEE Congress on Evolutionary Computation (2008)
10. Singh, H.K., Isaacs, A., Ray, T.: A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. IEEE Transactions on Evolutionary Computation 15(4), 539–556 (2011)
11. Deb, K., Sinha, A., Korhonen, P.J., Wallenius, J.: An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. IEEE Transactions on Evolutionary Computation 14, 723–739 (2010)
12. 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)
13. Hughes, E.J.: MSOPS-II: A general-purpose many-objective optimiser. In: Proc. IEEE Congress Evolutionary Computation, pp. 3944–3951 (2007)
14. Bader, J., Zitzler, E.: Hype: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation 19, 45–76 (2011)
15. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)
16. Das, I., Dennis, J.E.: Normal-bounday intersection: A new method for generating Pareto optimal points in multicriteria optimization problems. SIAM J. Optim. 8(3), 631–657 (1998)
17. Deb, K., Jain, H.: Handling many-objective problems using an improved NSGA-II procedure. In: IEEE World Congress on Computational Intelligence, Brisbane, Australia, pp. 10–15 (June 2012)
18. Tan, Y.Y., Jiao, Y.C., Lib, H., Wang, X.K.: MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. Computers & Operations Research (January 2012)

19. Takahama, T., Sakai, S.: Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations. IEEE Transactions on Evolutionary Computation 9(5), 437–451 (2005)
20. Deb, K., Agarwal, R.B.: Simulated binary crossover for continuous search space. Complex Systems 9(2), 115–148 (1995)
21. Asafuddoula, M., Ray, T., Sarker, R., Alam, K.: An adaptive constraint handling approach embedded MOEA/D. In: IEEE World Congress on Computational Intelligence, June 10-15, pp. 1–8 (2012)
22. Deb, K., Jain, H.: An improved NSGA-II procedure for many-objective optimization, Part I: Solving problems with box constraints. KanGAL Report No. 2012009 (June 2012)
23. Lili, Z., Wenhua, Z.: Research on performance measures of multi-objective optimization evolutionary algorithms. In: Proc. 3rd Int. Conf. Intelligent System and Knowledge Engineering (ISKE), vol. 1, pp. 502–507 (2008)
24. Deb, K., Jain, H.: An improved NSGA-II procedure for many-objective optimization, Part II: Handling constraints and extending to an adaptive approach. KanGAL Report No. 2012010 (June 2012)
25. Ray, T., Tai, K., Seow, K.C.: An evolutionary algorithm for multiobjective optimization. Engineering Optimization 33(3), 399–424 (2001)
26. Durillo, J., Nebro, A.: jMetal: a java framework for multi-objective optimization. Advances in Engineering Software 42, 760–771 (2011)
27. Simpson, T.W., Chen, W., Allen, J.K., Mistree, F.: Conceptual design of a family of products through the use of the robust concept exploration method. In: 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, vol. 2, pp. 1535–1545 (1996)