Performance Scaling of Multi-objective Evolutionary Algorithms

V. Khare[1], X. Yao[1] and K. Deb[2] *

[1] School of Computer Science
The University of Birmingham
Edgbaston, Birmingham B15 2QW, UK
{msc39vxk, X.Yao}@cs.bham.ac.uk
[2] Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur 208016, INDIA
deb@iitk.ac.in

KanGAL Report No. 2002009

Abstract. MOEAs are getting immense popularity in the recent past, mainly because of their ability to find a wide spread of Pareto-optimal solutions in a single simulation run. Various evolutionary approaches to multi-objective optimization have been proposed since 1985. Some of fairly recent ones are NSGA-II [9], SPEA2 [19], PESA [1] (which are included in this study) and others. They all have been mainly applied to two to three objectives. In order to establish their superiority over classical methods and demonstrate their abilities for convergence and maintenance of diversity, they need to be tested on higher number of objectives. In this study, these state-of-the-art MOEAs have been investigated for their scalability with respect to the number of objectives (2 to 8). They have also been compared on the basis of -(1) Their ability to converge to pareto front, (2) Diversity of obtained non-dominated solutions and (3) Their running time. Four scalable test problems [9] are used for the comparative study.

1 Introduction

Evolutionary algorithms are often praised for their ability to search multiple solutions in parallel and to handle complicated tasks such as discontinuities, multi-modality and noisy function evaluations. Their population based approach, which enables them to find multiple optimal solutions in one single run, is especially useful in the conetxt of multi-objective optimization which involves the task of finding more than one optimal solution. Though there exists a number of efficient algorithms, most of the work that has been done in evolutionary multi-objective optimization is restricted to 2 and 3 objectives. The main motivation

* This work is partly supported by The Europian Commission through its grant AS1/B7-301/97/0126-08 and School of Computer Science, The University of Birmingham.

behind this work is to investigate how these algorithms behave when tested on higher dimensional problems.

Scalability of some recent algorithms (section 3) is to be investigated using an experimental study (section 5). This study would involve experiments with chosen algorithms (section 3) on four scalable (in terms of objectives) test problems (section 2) for 2 to 8 objectives. For comparison purposes, three performance metrics are to be used (section 4). Results are presented and discussed in section 6.

2 Test Problems

Most earlier studies on MOEAs introduced test problems which were either simple or not scalable [16, 14, 11]. Few others were too complicated to visualize the exact shape and location of the resulting Pareto-optimal (PO) front. In this study four scalable test problems, namely DLTZ1, DLTZ2, DlTZ3 and DLTZ6 [9], are chosen on the basis of the following properties- (1) They are easy to construct (Bottom-up approach [9]), (2) They can be scaled to any number of decision variables and objectives, (3) Exact shape and location of the resulting PO front for these problems are known and (4) Difficulties in both converging to the true PO front and maintaining a widely distributed set of solutions can be controlled. In these test problems, the total number of variables are $n = M+k-1$. Where M is the number of objectives and k can be set by the user giving him the freedom to scale to any number of variables. All these test problem require a functional $g(\mathbf{x}_M)$ to be set, where the cardinality $|\mathbf{x}_M| = k$. Choices of $g(\mathbf{x}_M)$ and k were made according to the suggestions in [9].

For DLTZ1, DLTZ2 and DLTZ3, the PO solutions corresponds to $\mathbf{x}_M = \{0.5, 0.5, \ldots, 0.5\}^T$ and the objective function values corresponding to the PO lie on $\sum_{m=1}^{M} f_m = 0.5$ for DLTZ1 and on $\sum_{m=1}^{M} f_i^2 = 1$ for DLTZ2 and DLTZ3. The g function in DLTZ3 introduces $(3^k - 1)$ local PO fronts and one global PO front. All local PO fronts are parallel to the global PO front and an MOEA can get stuck to any of these local PO fronts, before converging to the global PO front (at $g^* = 0$). On the other hand, DLTZ6 tests an MOEA's ability to converge to a curve. In this case there is only one independent variable describing the PO front.

3 Algorithms Used

Earlier MOEAs (MOGA [10], NSGA [17] and NPGA [12]) were critisized for their dependence on sharing parameter [8] and lack of elitism [15, 18]. Different algorithms that over come these shortcomings have been proposed. Few such algorithms are - PAES [13], SPEA [21] and NSGA-II [8]. In these algorithms, elitism maintains the knowledge acquired during the algorithm execution by conserving the individuals with best fitness in the population or in an auxiliary population. For the maintenance of spread of solutions grid based techniques

(PAES), clustering (SPEA) or crowding (NSGA-II) were used. Further improvements to these algorithms have also been proposed.

NSGA-II (Non-dominated Sorting Genetic Algorithm-II) with controlled elitism [5] limits the maximum number of individuals belonging to each front by a geometric decreasing function (governed by the reduction rate r) to introduce more diversity into the population. NSGA-II, as reported in [8], outperformed PAES in preserving the spread of non-dominated front on five 2-objective test problems (listed in [8]). PESA (Pareto Enveloped-based Selection Algorithm) [1], an improvement of PAES, uses the hyper-cubes grid division not only for crowding as in PAES, but also for selection process. PESA was compared with PAES and SPEA on six test functions $\mathcal{T}_1$ to $\mathcal{T}_6$ [2] (each of which is a 2-objective problem defined on m parameters) and was reported to outperform SPEA and PAES on these test functions.

SPEA2 (Strength Pareto Evolutionary Algorithm 2) [19] was proposed as an improvement of SPEA and incorporated a revised fitness assignment strategy, a density estimation technique and an enhanced archive truncation method. Performance of SPEA2 was compared with SPEA, NSGA-II and PESA on some combinatorial and continuous problems. Similar performances of NSGA-II and SPEA2 were reported along with the fast convergence properties of PESA on these problems. All the test problems used here (except for the 3 and 4-objective *knapsack problem* and 3 and 4-objective *KP-750-m problem* [21]) were 2-objective problems.

In this study we have taken three of these recent MOEAs (PESA, SPEA2 and NSGA-II) and compared their performances on specially designed scalable test problems (section 2) over 2 to 8 objectives. For a detailed description of these algorithm we refer the readers to the original papers. Since the test problems that we are dealing with have a continuous space, real encoding should be preferred to avoid problems related to hamming cliffs and to achieve arbitrary precision in the optimal solution. For this reason, in all the algorithms real-coded parameters were used and crossover (*Simulated Binary Crossover* or SBX [7]) and mutation (*polynomial mutation* [4]) operators were applied directly to real parameter values.

4 Performance Metrics

Zitzler [20] showed that for an $M-$objective optimization problem, at least M performance metrics are needed to compare two or more sets of solutions. Use of any number less than M would result in inaccurate judgement because of dimensionality reduction. Deb [6] suggested that we can overcome this dimensionality problem using a *functionally independent* set of variables, which would of course make it theoretically inaccurate but practically feasible. He also suggested two new *running* performance metrics - one for measuring the convergence to the reference set and other for measuring the diversity in population members at every generation of an MOEA run. In this study these two metrics (with slight variation) have been used in addition to a third one, which simply measures the

Table 1. Parameter settings for calculating the diversity metric

Reference plane	$M-$th objective function $f_M = 0$
Number of grids (G_i)	Population size
Target (or reference) set of points (P^*)	One assumed solution in each grid

running time of the MOEA. All three metrics, which were applied to only the final non-dominated set obtained by an MOEA to evaluate its performance, have been discussed in detail below.

The following metric represents the distance between the set of converged non-dominated solutions and the global PO front; hence lower values of convergence metric represent good convergence ability. Let P^* be the reference or target set of points on the PO front and let $\mathcal{F}$ be the final non-dominated set obtained by an MOEA. Then from each point i in $\mathcal{F}$ the smallest normalized Euclidean distance to P^* will be:

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^{M} \left(\frac{f_k(i) - f_k(j)}{f_k^{max} - f_k^{min}} \right)^2} \tag{1}$$

Here, f_k^{max} and f_k^{min} are the maximum and minimum function values of k-th objective function in P^*. However in this study, no target points were chosen because equations of PO fronts were known for all the four test problems. The orthogonal distance of a point A in the non-dominated set from the PO front was calculated directly from the equation of PO front. E.g. in DLTZ2 or DLTZ3

$$d_i = \|\boldsymbol{r}_A\| - 1 \tag{2}$$

Once these distances are known the convergence metric can be obtained by averaging the normalized distance for all points in $\mathcal{F}$.

Diversity metric is a number in the range $[0, 1]$, where 1 corresponds to best possible diversity and a 0 corresponds to worst possible diversity. In calculating the diversity metric, the obtained non-dominated points are projected on a hyper-plane, thereby losing a dimension of the points. The plane is divided into a number of small grids (or $M-1$ dimensional boxes). Depending on each grid contains a non-dominated point or not, a diversity metric is defined. If all grids are represented with at least one point, the best possible (with respect to the chosen number of grids) diversity measure is achieved. If some grids are not represented by a non-dominated point, the diversity is poor. Various parameters required to calculate this metric and their chosen values are given in table 1.

Calculating the diversity metric Following steps are involved in calculating the diversity metric:

Table 2. Lookup table for calculating diversity metric

h(...j-1 ...)	h(...j ...)	h(...j+1 ...)	m(h(...j ...))
0	0	0	0.00
0	0	1	0.50
1	0	0	0.50
0	1	1	0.67
1	1	0	0.67
0	1	0	0.75
1	1	1	1.00

1. For each grid indexed by $(i, j, \ldots)$ calculate following two arrays:

$$H(i,j,\ldots) = \begin{cases} 1, \text{ if the grid has a representative point in} P^*; \\ 0, \text{ otherwise.} \end{cases}$$
$$= 1, \text{ for the chosen reference set} P^* \quad (3)$$
$$h(i,j,\ldots) = \begin{cases} 1, \text{ if H(i, j, } \ldots \text{) } = 1 \text{ and the grid has a representative point in} \mathcal{F}; \\ 0, \text{ otherwise.} \end{cases}$$
$$= \begin{cases} 1, \text{ if the grid has a representative point in } \mathcal{F} \text{ for the chosen} P^*; \\ 0, \text{ otherwise.} \end{cases} \quad (4)$$

2. Assign a value $m(H(i, j, \ldots))$ to each grid depending on its and its neighbor's $h()$. Similarly, calculate $m(H(i, j, \ldots))$ using $H()$ for reference points.
3. Calculate the diversity metric (of the population P of non-dominated solutions produced by an MOEA) by averaging the individual $m()$ values for $h()$ with respect to that for $H()$:

$$D(P) = \frac{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(h(i,j,\ldots))}{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(H(i,j,\ldots))} \quad (5)$$

Value function $m()$ for the grid was calculated by using its $h()$ and two neighboring $h()$ dimension-wise. With a set of three consecutive binary $h()$ values, there are a total of 8 possibilities. Suggested [6] values of $m()$ were used in this study (Table 2). Two or more dimensional hyper-planes are handled by calculating the above metric dimension-wise.

Figure 1 shows a sample calculation of diversity metric in case of 2-objective DLTZ2 or 2-objective DLTZ3 problem. Here circles represent the reference or target points, in every partition there is one such point and hence the number of partions is same as population size. Boxes represent the set of non-dominated points given by an MOEA. The $f_2 = 0$ is used as the reference plane here and the complete range on f_1 values are divided into $G_1 = 10$ grids. This complete range depends on the PO front the algorithm has converged to and the resulting

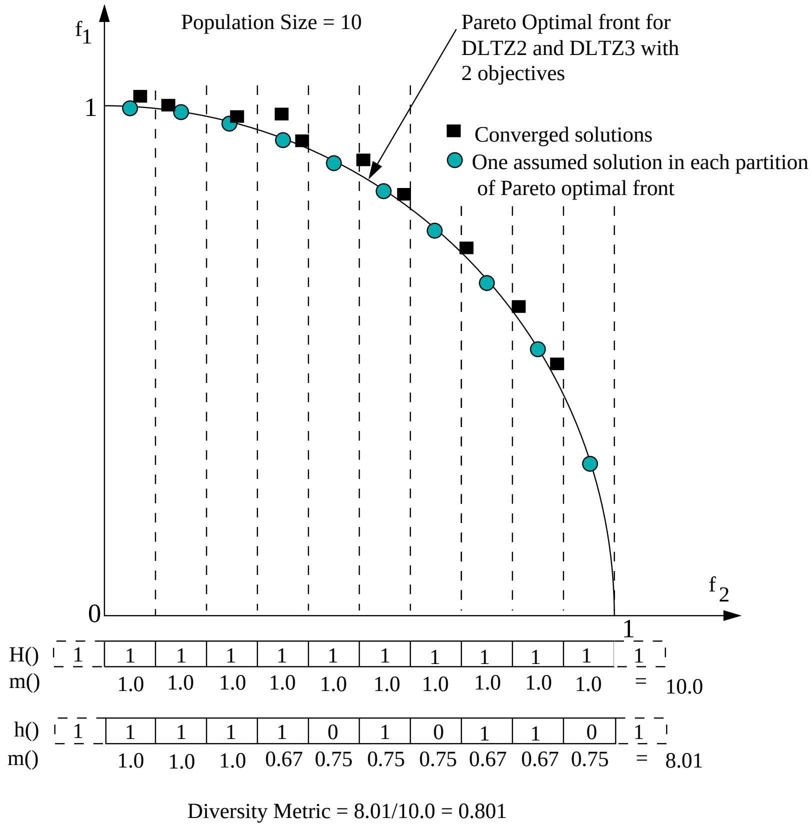

Fig. 1. Calculating the diversity metric

divesrsity metric will also be different. For boundary grids, an imaginary neighboring grid with a $h()$ and $H()$ value of one is always assumed. In figure 1, these grids are shown with dashed lines. Even if we have more than one points in a grid, $h()$ still remains as one. Based on moving window containing three consecutive grids, the $m()$ values are computed in the figure. To avoid the boundary effects of using the imaginary grid, the metric value is normalized as follows:

$$\overline{D}(P) = \frac{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(h(i,j,\ldots)) - \sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(\mathbf{0})}{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(H(i,j,\ldots)) - \sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(\mathbf{0})} \tag{6}$$

where $\mathbf{0}$ is a zero-valued array. Though in our study $H(i,j,\ldots)$ is always equal to 1 (each grid contains one reference point), $H(i,j,\ldots) \neq 0$ consideration in computing the $\overline{D}(P)$ term and the boundary grid adjustment was suggested to allow a generic way to handle disconnected PO fronts in the original work [6].

Ideally we want our algorithm to give us non-dominated solutions on the global PO front and if we divide our objective space corresponding to this global PO front into grids equal to the population size, then one point in each grid would be the best possible diversity ($\overline{D}(P) = 1$). We will call this diversity metric (obtained by splitting the global PO region into grids) *diversity metric1*. But if the algorithm isn't able to converge to the global PO front then the above metric will not be able to measure the diversity of non-dominated solutions produced by the MOEA. In such cases we should calculate our diversity metric based on the actual converged front instead of global PO front. We will call this diversity metric (obtained by splitting the converged PO region into grids) *diversity metric2*.

To compare two algorithms (for diversity) the use of diversity metric2 should be preferred because even if one of them has converged to true PO front and the other hasn't, the diversity metric2 of the former will be almost equal to its diversity metric1 value, which would not be the case with latter.

The third metric used in the study is simply the running time of an algorithm (in seconds) for the particular settings. It has been included in the performance metric set to evaluate how an MOEA scales in terms of time with increase in number of objectives. A linear or polynomial increase in running time is acceptable but an exponential increase is undesirable.

5 Experimental Study

To make the comparisons fair the number of function evaluations were kept constant for all three algorithms on a particular setting. Before the actual experimentation some tuning of the parameters involved was required. Finding values for which an algorithm works best is, in itself, an optimization problem and if we are judging the performance on the three metrics it becomes a Muti-objective Optimization Problem (MOOP) for each of the parameter involved. A very simplistic approach was adopted to tune (instead of optimizing) these parameters. Experiments with only 2-objectives were used and the purpose of this tuning was to find a set of values for which an MOEA performs well. Table 3 gives the tuned values used for all the experimentation. Parameter tuning was carried out on problems DLTZ2 and DLTZ3 and only the convergence and diversity metrics were used to evaluate the performance of the algorithm, running time was not considered.

Population size plays a crucial rule in the performance of an MOEA. As the number of objective functions (M) increases, more and more solutions tend to lie in the first non-dominated front. Most MOEAs assign the similar fitness to all solutions in the first non-dominated front. So as the number of objective functions increase, there is no (very little) selection advantage to any of these solutions. In absence of any selection pressure for better solutions, the task of recombination and mutation operators to find better solutions may be difficult in general. It has been shown empirically [3, pages 404–405] that for a particular M, the proportion of non-dominated solutions decreases with population size.

Table 3. Tuned parameter values

Parameter	PESA	SPEA 2	NSGA-II
Crossover probability p_c	0.8	0.7	0.7
Distribution index (DI) for SBX η_c	15	15	15
Mutation probability p_m (if $n = \#$ of variables)	1/n	1/n	1/n
DI for polynomial mutation η_m	15	15	20
Ratio of internal population size to archive size	1:1	1:1	1:1
# of grids per dimension (PESA)	10	-	-

Table 4. Population scheme - for Internal or main population (which is same as external population as the result of parameter tuning)

M	Population Size	Maximum proportion of non-dominated solutions
2	20	0.2
3	50	0.22
4	100	0.28
5	150	0.36
6	250	0.45
7	400	0.52
8	600	0.60
9	850	0.68
10	1150	$\sim$0.75

If we require a population with a user-specified maximum proportion of non-dominated solutions, then these empirical results can be used to estimate what would be a reasonable population size. This requirement on population size increases exponentially with M. Ideally to investigate the scaling of an MOEA we should present it with a population having equal proportion of non-dominated solutions, for all M, to start with, but this is practically impossible because of exponential increase in population size. The population scheme used in this study is given in table 4. This scheme is quadratic with $R^2 = 0.9916$.

Number of generations used for different problems and different number of objectives (M) are listed in table 5. More number of function evaluations were used for DLTZ3 and DLTZ6 because they can introduce more difficulties to a MOEA in converging to PO front and in finding a diverse set of solutions. DLTZ3 tests the ability of an MOEA by introducing local PO fronts and DLTZ6 tests them for their ability to converge to a curve.

Number of generations from 6-objectives onwards was doubled because none of the algorithms was able to converge to the global PO front in these many generations. Converging to a PO front means having a convergence metric less than a threshold (say ϵ). Any appropriate value of ϵ can be chosen. But here, instead of choosing some such threshold, the algorithms were compared (for

Table 5. # of generations for different problems and # of objectives (M)

	# of generations	
	For $M = 2, 3$ and 4	**For $M = 6$ and 8**
DLTZ1 & DLTZ2	300	600
DLTZ3 & DLTZ6	500	1000

convergence) solely on the basis of the convergence metric that they can achieve in given number of function evaluations.

6 Results

Tables 6, 7 and 8 give the convergence metric, diversity metric(1 & 2) and running times respectively for all three algorithms over 2 to 8 objectives. Presented values are averaged over 30 runs for 2, 3 and 4 objectives and 10 runs for 6 and 8 objectives. As an illustration, results for problem DLTZ3 are plotted in figure 2. Following is the discussion over the results obtained. Few other points, which have been observed during this extensive comparative study, are also discussed.

1. ***Scalability*** Each algorithm scale differently in terms of the performance metrics chosen.
   - PESA scale very well in terms of convergence but poorly in terms of diversity maintenance and running time.
   - SPEA2 scales well in terms of diversity maintenance but suffers in converging to the global PO front and in running time.
   - NSGA-II scales well in terms of running time and diversity maintenance but suffers in converging to the global PO front.
2. ***Convergence to PO front*** Ability to converge to the PO front was found best in PESA, though it cannot produce a very diverse solution on the converge front.
   SPEA2 has better convergence than NSGA-II for small number of objectives but for higher number of objectives both of them have comparable performances, which is inferior to PESA. Both SPEA2 and NSGA-II had difficulties in dealing with local PO fronts especially for higher number of objectives.
3. ***Diversity in obtained solutions*** Even in terms of diversity of solutions in the converged front SPEA2 and NSGA-II have similar performances, which is much better than PESA.
4. ***Running Time*** NSGA-II was the fastest of all three algorithms, primarily because it doesn't involve expensive calculations related to clustering (as in SPEA2) or grid based calculations (as in PESA). Exponential increase in running time for PESA makes it impractical for higher objectives.
5. ***Grid size in PESA*** Grid size in PESA is a crucial factor. If we choose very fine grids we can hope to get a good performance in terms of diversity but that would make the algorithm even more expensive.

Table 6. Results of Convergence Metric (averaged over 30 runs for 2, 3 and 4 objectives and 10 runs for 6 and 8 objectives)

	Mean			Standard Deviation		
Objectives	**PESA**	**SPEA2**	**NSGA-II**	**PESA**	**SPEA2**	**NSGA-II**
Convergence Metric for DLTZ1						
2	2.86948	3.08825	2.27666	5.93164	5.35433	5.43593
3	0.04419	0.04843	0.38360	0.12320	0.05331	0.50094
4	0.02317	0.29925	3.10281	0.09059	0.66360	4.08272
6	0.00117	5.99951	120.19162	0.00089	7.67166	101.19802
8	0.00407	498.27151	465.30155	0.00015	13.38934	14.79745
Convergence Metric for DLTZ2						
2	0.00008	0.00026	0.00180	0.00019	0.00029	0.00082
3	0.00035	0.00663	0.01003	0.00013	0.00224	0.00234
4	0.00170	0.03369	0.04529	0.00039	0.00846	0.01373
6	0.00301	2.00216	1.67564	0.00040	0.07843	0.10533
8	0.00689	2.35258	2.30766	0.00109	0.03523	0.04242
Convergence Metric for DLTZ3						
2	22.52023	16.87313	21.32032	22.90480	16.72477	11.15397
3	1.80296	2.39884	5.65577	5.78546	4.72212	6.26729
4	1.16736	4.00596	66.94049	3.50522	4.00594	39.06815
6	0.15035	217.95360	1273.30601	0.12692	76.62720	64.21416
8	7.23062	1929.94832	1753.41364	2.25611	11.09337	62.63447
Convergence Metric for DLTZ6						
2	0.79397	0.77622	0.63697	0.32237	0.23794	0.29986
3	0.20528	0.29271	0.24515	0.21199	0.23631	0.22849
4	3.60430	5.07137	6.32619	0.38084	0.22360	0.36229
6	5.30454	10.53682	9.48750	0.31227	0.00819	0.27429
8	6.32247	10.62932	10.27306	0.10668	0.04585	0.05803

6. ***Swapping offspring in SBX*** Swapping the offspring in SBX helped improving the performance of all three algorithms. This is because the way SBX has been formulated - either offspring1 always gets a variable value greater than offspring2 or it always gets a smaller value. PESA always takes one offspring generated each time and rejects the other one, hence improved performance on PESA can be explained but the improvement in performance of SPEA2 and NSGA-II is something that is to be investigated.

7 Conclusion

All of the work that has been done in MOEAs is mostly limited to 2 and 3 objectives. In this paper scalability issues related to three of the state-of-the-art algorithms (PESA, SPEA2 and NSGA-II) were explored. These algorithms were tested for their scalability with respect to number of objectives (2 to 8). These

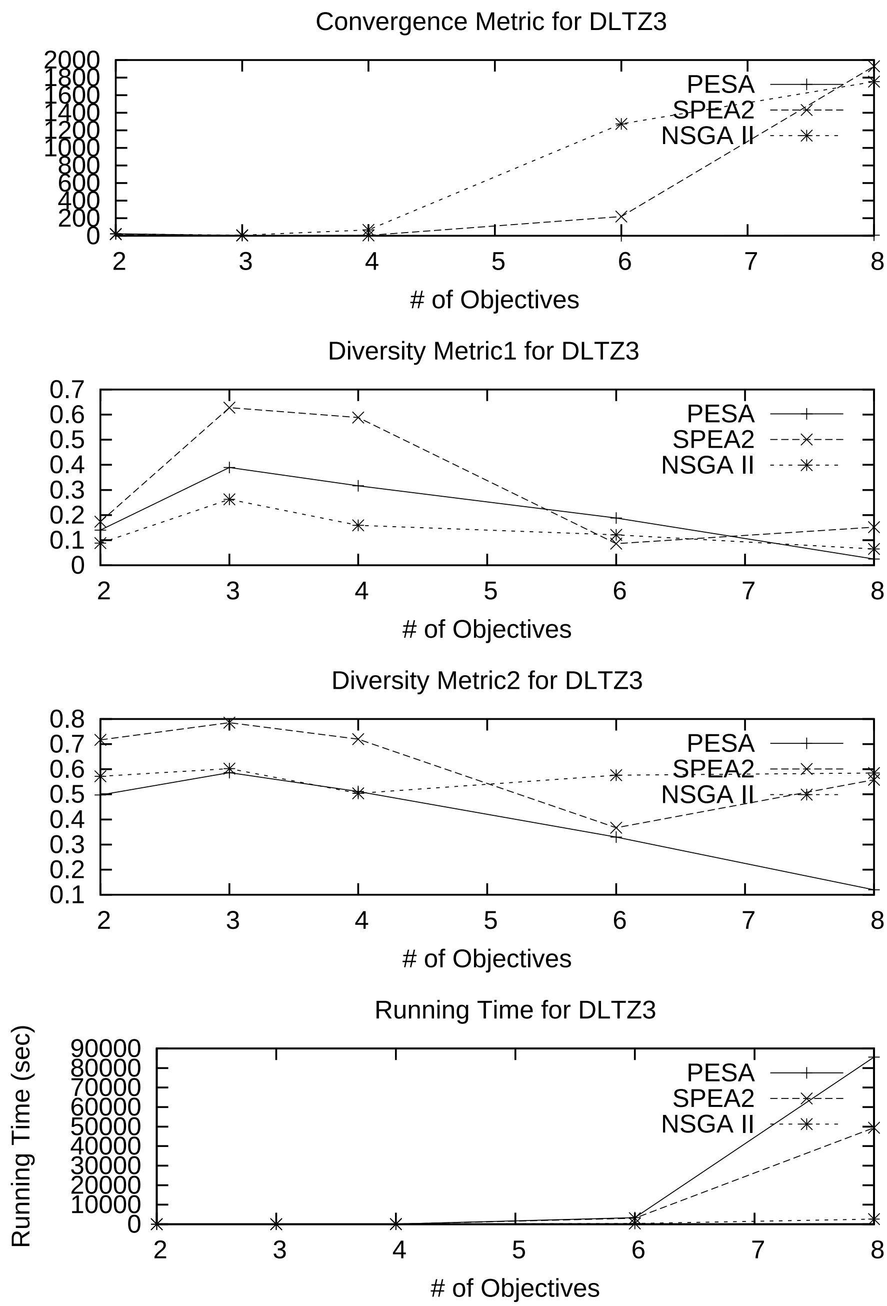

Fig. 2. Results for DLTZ3 (500 generations for 2, 3 and 4 objectives and 1000 generations for 6 and 8 objectives)

Table 7. Results of Diversity Metrics 1 & 2 (averaged over 30 runs for 2, 3 and 4 objectives and 10 runs for 6 and 8 objectives)

	Mean			Standard Deviation		
Objectives	**PESA**	**SPEA2**	**NSGA-II**	**PESA**	**SPEA2**	**NSGA-II**
Diversity Metric1 for DLTZ1						
2	0.25093	0.55656	0.44784	0.14059	0.34527	0.23329
3	0.42116	0.63186	0.57752	0.07563	0.04565	0.15660
4	0.37605	0.54176	0.38676	0.07125	0.02007	0.11091
6	0.33643	0.35645	0.19343	0.04046	0.15384	0.04365
8	0.25245	0.26107	0.19744	0.00764	0.01021	0.01238
Diversity Metric2 for DLTZ1						
2	0.50019	0.86516	0.72559	0.17016	0.11721	0.13887
3	0.52274	0.78292	0.76969	0.10693	0.06054	0.09305
4	0.48240	0.67836	0.58683	0.09175	0.04362	0.09545
6	0.39297	0.48162	0.38293	0.05970	0.11778	0.01930
8	0.29631	0.35056	0.37425	0.02745	0.02485	0.01172
Diversity Metric1 for DLTZ2						
2	0.57396	0.81867	0.75177	0.09135	0.01766	0.03891
3	0.57163	0.67260	0.74996	0.04344	0.03255	0.02064
4	0.52708	0.62136	0.71360	0.03692	0.01773	0.01881
6	0.47099	0.29675	0.48248	0.02660	0.01939	0.01613
8	0.43230	0.30944	0.52913	0.04908	0.00628	0.01019
Diversity Metric2 for DLTZ2						
2	0.58509	0.82186	0.75863	0.09700	0.01992	0.04302
3	0.57993	0.71680	0.81107	0.04528	0.03629	0.02452
4	0.57993	0.71680	0.81107	0.04528	0.03629	0.02452
6	0.52335	0.64928	0.73253	0.03503	0.01423	0.01139
8	0.57082	0.64134	0.75665	0.02060	0.00327	0.01022
Diversity Metric1 for DLTZ3						
2	0.14023	0.17339	0.08846	0.14497	0.19218	0.06994
3	0.38965	0.62793	0.26244	0.13220	0.14088	0.19989
4	0.31659	0.58861	0.15869	0.09393	0.02637	0.03146
6	0.18812	0.08588	0.12068	0.06554	0.01392	0.01290
8	0.02463	0.15186	0.06456	0.00247	0.01223	0.01244
Diversity Metric2 for DLTZ3						
2	0.49708	0.71668	0.57169	0.14339	0.15166	0.15533
3	0.58655	0.78540	0.60255	0.09515	0.11143	0.18633
4	0.51138	0.72007	0.50374	0.07063	0.05674	0.04957
6	0.32959	0.36687	0.57644	0.06817	0.02543	0.01418
8	0.11972	0.55771	0.58456	0.01179	0.00873	0.01278
Diversity Metric1 for DLTZ6						
2	0.20191	0.44404	0.40846	0.14198	0.25537	0.19784
3	0.41962	0.64655	0.66157	0.06423	0.14591	0.13731
4	0.22558	0.22917	0.12825	0.02790	0.01438	0.01845
6	0.27631	0.07593	0.07544	0.02356	0.00710	0.01760
8	0.27801	0.05583	0.04766	0.00488	0.00991	0.00391
Diversity Metric2 for DLTZ6						
2	0.55795	0.85028	0.78386	0.12064	0.06017	0.08507
3	0.56629	0.83748	0.81710	0.14311	0.03616	0.06102
4	0.48907	0.47386	0.59749	0.03777	0.02237	0.02803
6	0.48725	0.60226	0.65365	0.02286	0.01100	0.01746
8	0.44885	0.55396	0.62897	0.03112	0.02598	0.00585

Table 8. Results of Running Time (averaged over 30 runs for 2, 3 and 4 objectives and 10 runs for 6 and 8 objectives)

	Mean			Standard Deviation		
Objectives	**PESA**	**SPEA2**	**NSGA-II**	**PESA**	**SPEA2**	**NSGA-II**
Running Time for DLTZ1						
2	2.53	3.70	2.90	0.51	0.47	0.55
3	9.13	17.38	4.53	1.56	0.49	0.80
4	70.53	91.45	17.00	5.53	1.04	0.39
6	4229.2	1839	213.9	184.11	251.08	4.53
8	95465.67	34735.33	1552.72	4879.89	2121.46	51.05
Running Time for DLTZ2						
2	4.60	4.07	3.27	0.50	0.26	0.59
3	19.825	20.425	4.925	3.69	0.59	0.83
4	193.975	107.45	18.7	17.41	1.28	2.42
6	8082.8	2408	235.9	127.77	315.43	30.78
8	334641.5	33722	1570.8	4321.10	1283.36	61.32
Running Time for DLTZ3						
2	5.567	6.633	5.733	0.74	0.49	0.79
3	11.833	29.333	7.972	1.50	0.63	1.13
4	80.8	131.925	29.75	8.91	1.38	2.84
6	3402.4	3103.1	384.4	189.63	357.13	44.93
8	85559.4	49382.2	2605.5	2826.43	514.14	114.42
Running Time for DLTZ6						
2	3.77	4.23	3.20	0.51	0.44	0.56
3	22.23	32.47	7.73	4.38	0.78	0.45
4	295.15	178.18	31.00	22.07	7.23	0.39
6	13528.2	5640.1	386.9	167.72	681.69	2.47
8	262802.9	53809.8	2741.1	16819.82	5404.04	176.50

algorithms were tested for their performance, on four scalable test problems, namely - DLTZ1, DLTZ2, DlTZ3 and DLTZ6. According to our study, it is clear that conclusions drawn from experimental comparisons on 2 or 3 objectives cannot be generalised to a higher number of objectives. More work needs to be done to fully understand the behaviour of MOEAs on different number of objectives.

To compare two or more sets of non-dominated solutions of an $M-$objective problem require at least M performance metrics, otherwise this would result in an inaccurate judgement caused by reduction in dimensionality [20]. However, having M performance metrics would make the comparison practically infeasible. In this study three performance metrics were used, in terms of which the scalability of these algorithms were assessed. First metric measures closeness of obtained non-dominated solution to the global PO front. The second metric indicates the diversity of solutions in the obtained non-dominated set. Running time was also included as one of the metric to evaluate how an MOEA scales in terms of time with increase in number of objectives.

As the result of the study on four test problems, PESA was found to be best in terms of converging to the PO front, but it lacks good diversity maintenance. Also the algorithm was found to be slow because of expensive grid based calculations. Exponential increase in running time makes the algorithm infeasible for higher number of objectives. SPEA2 and NSGA-II performed equally well on convergence and diversity maintenance. Their convergence level was inferior to that of PESA but diversity maintenance was better. NSGA-II was found to be much faster than SPEA2 because of the expensive clustering of solutions. Running times for NSGA-II were found to be an order of magnitude less than that of SPEA2 for higher objectives.

Comparing these algorithms with some classical multi-objective optimizers on the same test problems could lead to interesting results.

References

1. D. W. Corne, J. D. Knowles, and M. J. Oates. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
2. K. Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Technical Report CI-49/98, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, 1998.
3. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.
4. K. Deb and M. Goyal. A Combined Genetic Adaptive Search (geneAS) for Engineering Design. *Computer Science and Informatics*, 26(4):30–45, 1996.
5. K. Deb and T. Goyal. Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence. KanGAL report 200004, Indian Institute of Technology, Kanpur, India, 2000.
6. K. Deb and S. Jain. Running Performance Metrics for Evolutionary Multi-objective Optimization. Technical Report 2002004, KanGAL, Indian Institute of Technology, Kanpur 208016, India, 2002.
7. K. Deb and A. Kumar. Real-coded Genetic Algorithms with Simulated Binary Crossover: Studies on Multimodal and Multiobjective Problems. *Complex Systems*, 9:431–454, 1995.
8. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
9. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
10. C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423,

San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.
11. C. M. Fonseca and P. J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, Spring 1995.
12. J. Horn and N. Nafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.
13. J. D. Knowles and D. W. Corne. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. In *1999 Congress on Evolutionary Computation*, pages 98–105, Washington, D.C., July 1999. IEEE Service Center.
14. F. Kursawe. A Variant of Evolution Strategies for Vector Optimization. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197, Berlin, Germany, oct 1991. Springer-Verlag.
15. G. T. Parks and I. Miller. Selective Breeding in a Multiobjective Genetic Algorithm. In A. E. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving From Nature — PPSN V*, pages 250–259, Amsterdam, Holland, 1998. Springer-Verlag.
16. J. D. Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.* PhD thesis, Vanderbilt University, 1984.
17. N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.
18. E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.
19. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.
20. E. Zitzler, M. Laumanns, L. Thiele, C. M. Fonseca, and V. Grunert da Fonseca. Why Quality Assessment of Multiobjective Optimizers Is Difficult. In W. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. Potter, A. Schultz, J. Miller, E. Burke, and N. Jonoska, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pages 666–673, San Francisco, California, July 2002. Morgan Kaufmann Publishers.
21. E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.