An Empirical Comparison of Several Recent Multi-objective Evolutionary Algorithms

Thomas White, Shan He

An Empirical Comparison of Several Recent Multi-Objective Evolutionary Algorithms

Thomas White[1] and Shan He[1,2⋆]

[1] School of Computer Science
[2] Center for Systems Biology, School of Biological Sciences, The University of Birmingham, Birmingham, B15 2TT, UK

Abstract. Many real-world problems can be formulated as multi-objective optimisation problems, in which many potentially conflicting objectives need to be optimized simultaneously. Multi-objective optimisation algorithms based on Evolutionary Algorithms (EAs) such as Genetic Algorithms (GAs) have been proven to be superior to other traditional algorithms such as goal programming. In the past years, several novel Multi-Objective Evolutionary Algorithms (MOEAs) have been proposed. Rather than based on traditional GAs, these algorithms extended other EAs including novel EAs such as Scatter Search and Particle Swarm Optimiser to handle multi-objective problems. However, to the best of our knowledge, there is no fair and systematic comparison of these novel MOEAs. This paper, for the first time, presents the results of an exhaustive performance comparison of an assortment of 5 new and popular algorithms on the DTLZ benchmark functions using a set of well-known performance measures. We also propose a novel performance measure called unique hypervolume, which measures the volume of objective space dominated only by one or more solutions, with respect to a set of solutions. Based on our results, we obtain some important observations on how to choose an appropriate MOA according to the preferences of the user.

Keywords: Multi-objective optimisation, Evolutionary Algorithms, Comparison, Genetic Algorithms

1 Introduction

Many real-world optimisation problems involve multiple objectives, which are generally incommensurable and often conflicting. These so-called multi-objective optimisation problems are notoriously difficult to solve. In recent years, Multi-Objective Evolutionary Algorithms (MOEAs) have been attracting more attention due to their superior performance over traditional multi-objective optimisation algorithms, in terms of effectiveness and robustness. This general trend is reflected by an exponential increase of MOEA applications to many real-world problems. The most popular MOEAs are algorithms based on Genetic Algorithms (GAs), one notable example being NSGA-II [1]. Recently, several novel MOEAs have been proposed by extending traditional EAs

⋆ Correspondence and requests for materials should be addressed to Dr. S. He (email: s.he@cs.bham.ac.uk)

such as Simulated Annealing (SA) or novel EAs such as Scatter Search (SS) and Particle Swarm Optimiser (PSO) to handle multi-objective problems. These novel MOEAs were tested on different set of benchmark functions with varying number of function evaluations. There are some comparison studies that exist for these algorithms, such as [2], in which a set of six representative state-of-the-art multi-objective PSO algorithms were compared. However, to the best of our knowledge, there exists no such fair and systematic comparison of the novel MOEAs this work concerns.

In this paper, we tested several representative novel MOEAs due to their recency and reported success. They include AMOSA [3], a multi-objective SA algorithm; OMOPSO [4] and SMPSO, multi-objective PSO algorithms; and AbYSS [5], a multi-objective SS algorithm. For brevity, we roughly outline each algorithm in section 2.1, and refer the reader to the original papers for a more in-depth description. We also select NSGA-II [1] as our baseline algorithm for comparison.

Apart from a fair and comprehensive comparison of these four novel MOEAs, another main contribution of this paper is the proposal of a novel performance measure, unique hypervolume. This measures the volume of objective space dominated only by one solution, with respect to a set of solutions. We demonstrate the interesting insights this statistic can provide in the interpretation of algorithm performance results.

The remainder of this paper is organized as follows. Section 2 describes the MOEAs, the performance measures and the proposed unique hypervolume. In Section 3, we describe our experiments and discuss the the results. Section 4 concludes the paper.

2 Methods

2.1 Novel MOEAs

AMOSA [3] is a multi-objective adaptation of the original SA algorithm. In the single objective case, simulated annealing performs iterative perturbations upon a solution, accepting the change if it is beneficial and rejecting negative changes with a certain probability, where the probability is exponentially reduced as the algorithm progresses. Many significant changes are necessary to adapt this approach to multi-objective problems. The most important change made by this algorithm is the addition of an archive of non-dominated solutions, and the various probabilities regarding acceptance and rejection rely on the domination status of the new solution with respect to the archive.

OMOPSO [4] and SMPSO [6] are both multi-objective extensions of PSO, an algorithm inspired by the collective behaviour of social animals, like the flocking of birds. The PSO algorithm maintains a population, or so-called swarm, of particles, each a potential solution of the given problem. These particles move in the search space according to some simple rules, relative to each other, to converge upon a better solution. The multi-objective PSO algorithms OMOPSO and SMPSO both select leaders in the particles that are non-dominated with respect to the swarm. In OMOPSO, crowding distance [1] was used to filter out leader solutions. Two mutation operators are proposed to accelerate the convergence of the OMOPSO algorithm. In SMPSO, a constant σ is assigned to each particle of the swarm and of an external archive then select a particle as the leader of the particles with the closest σ value of the external archive.

AbYSS [5] is a MOA based on the well-known SS algorithm. SS algorithm differs from other EAs by using a small population, or so-called the reference set, in which the individuals are combined to construct new solutions systematically. AbYSS extended the original single-objective SS algorithm by incorporating concepts from the multi-objective field, such as Pareto dominance, density estimation, and an external archive to store the non-dominated solutions.

2.2 Performance measures

Measuring and comparing the performance of multi-objective algorithms remains an unsolved problem in the field. The prevailing opinion in the literature is that performance should be measured not only according to how closely the resulting set of solutions converges on the global optimal Pareto front, but also how spread the solutions are across the breadth of the front. An excellent survey of performance measures can be found in [7]. In these paper, we adopt a large set of popular performance measures to evaluate the four novel MOEAs:

- **Epsilon** from [8], calculates the minimum size translation in objective values for one set of solutions to cover another reference set. In the case of these experiments, the reference set used was the known globally optimal Pareto front values.
- **The hypervolume indicator**, originally described in [9], defines the size of multi-dimensional space that a set of solutions cover, without counting the same area twice.
- **Spread**, as defined in [1], which calculates how uniformly spread across the objective space the solutions are within their set.
- **The cover function** from [10], which calculates the proportion of one solution set that is dominated by another set. Thus, if C(X, Y) = 0.9, 90% of solutions in Y are dominated by solutions in X. Note that the converse C(Y, X) must also be considered separately; while most of X could be dominated by a few members Y, most of Y could also be dominated by a few members of X.
- **The Purity function** described in [11] unifies the solutions returned by a set of algorithms on a particular problem, and returns the proportion of each that remain nondominated in the set.
- **The unique hypervolume measure**, as described in detail below.

Unique hypervolume (UH) UH is a property that can be calculated for one solution, with respect to another solution or a set of solutions. Essentially, it quantifies the volume of objective space dominated only by that solution, and not by any other. This idea is represented visually in Fig. 1.

UH has the following advantageous properties:

- It rewards domination proportional to the amount one solution dominates another.
- It implicitly punishes clustering of solutions.
- It rewards distinctly original and innovative solutions.
- It rewards diversity only when this diversity is objectively beneficial, in the context of all other solutions found.

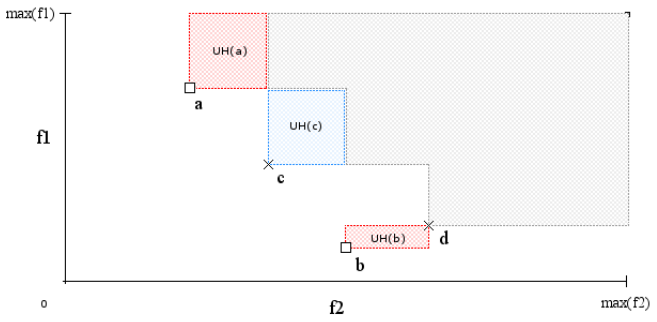

Fig. 1. A diagram showing the UH of 4 points on a two-dimensional minimisation problem. Points **a** and **b** belong to one set of solutions, denoted by a square and red lines, whereas **c** and **d** belong to a second set, denoted by a cross and blue lines. In this diagram, each point dominates the area directly above and to the right of it. The area (or in higher dimensional problems, hypervolume) it alone dominates is its UH. Note that point **b** dominates point **d**; therefore point **d** has no UH, but its presence restricts the UH of point **b**.

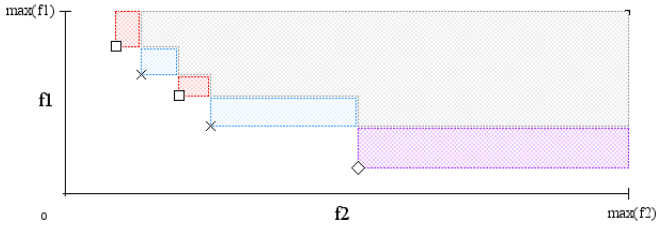

Fig. 2. In this diagram, we demonstrate one of the many situations where the results of UH contradict those of original hypervolume. Here we see three sets of results, colour coded as red, blue and purple. The red and blue sets undoubtedly contain more hypervolume than the purple set. However, the lone purple solution is noteworthy in that it covers a more original combination of objective space, making it a rarer and thus more valuable trade-off point.

UH bears conceptual similarities to the D metric [12]; a measure we have not actually observed in use. The key differences between the two lie in the behaviour in the case of one solution dominating another, and cases where solutions of the same set are neighbours. In Fig. 3 we demonstrate the differences by illustrating the area calculated by each measure on two sets of Pareto fronts. It can be seen from this diagram that UH is a more difficult metric to score well in. They both share the properties of rewarding useful and diverse solutions; however, UH punishes solutions within a set that are not diverse in comparison to each other. Therefore, we can learn more from this new metric because it also reflects the spread of solutions along the Pareto front.

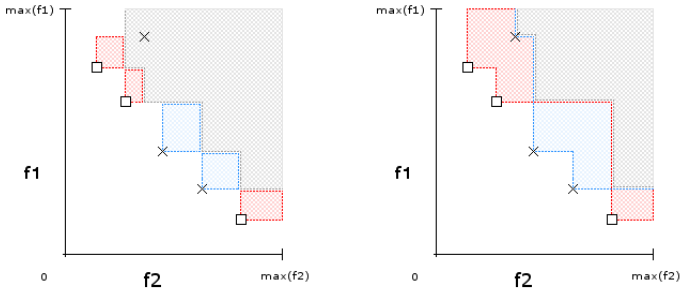

Fig. 3. A demonstration of the difference between the UH (left) and D metric (right), on two identical sets of Pareto fronts.

The UH measure concerns the same property of objective space as the original hypervolume measure, which is notoriously difficult to calculate in an efficient manner. However, our UH measure is comparatively simple to calculate, and does not require a calculation of hypervolume a priori. A simple and efficient method for calculating the UH of a solution is given in Fig. 4.

```
Definitions:
let S denote the set of solutions
let O denote the set of objectives
let side[|S|][|O|] be a two dimensional array
rank(s, o) : rank of solution s according to o
next(s, o) : next ranked solution according to objective o
score(s, o) : objective value of solution s in objective o
max(o) : maximum objective value of objective o
for each objective o in O
    sort the set S according to o
      for each solution s in S
          if ( rank( s, o ) == |S| )
              side[s][o] = max(o) - score(s, o)
        else
            side[s][o] = score(next(s, o), o) - score(s, o)
unique_hypervolume(s) = product( side[s] )
```

Fig. 4. Pseudocode for the calculation of UH.

In prose, we simply sort the set of solutions according to each objective, and record the distance from each solution to the next worst in that objective. If the solution is the worst in the set in an objective, we record the distance to the worst possible objective score. The product of these values gives us the volume of a hyperrectangle in objective space, which only that solution has dominated. It should be noted that this method

requires knowledge of each maximum objective value, assuming each objective is to be minimised. If this data is not available, an alternative is to substitute the worst value found in each objective.

This measure can be used in many ways, such as comparing sets of solutions found by two MOEAs, or finding the most novel solution within a set. In this paper, we can accumulate all the solutions found by every algorithm on a problem into one set, and total the UH of solutions found by each algorithm with respect to that set.

3 Experiments and results

The experiments were performed using a customised version of the jMetal framework [13]. This software package already included the algorithms NSGA-II, OMOPSO, SMPSO and AbYSS, to which we added our implementation of AMOSA. jMetal also has a built in set of quality indicators to score the results of a single run of an algorithm on a problem, aggregating those scores to calculate the average for each algorithm, for comparison in that manner. This is very useful, however it does not facilitate comparison measures that require access to the result solutions themselves. For the purposes of this study, and to enable an implementation of UH, the software was modified accordingly.

Each algorithm was executed on each of the 7 problems from the Deb, Thiele, Laumanns and Zitzler (DTLZ) problem family [14]. We used the default configurations regarding the number of variables and objectives for each problem within jMetal, along with the default function evaluation budget of 25,000 for those problems. The algorithms are evaluated according to the final set of solutions returned at the end. To ensure reliable and statistically significant results, each such experiment was repeated 30 times.

We provide the empirical results of our experiments in the following tables. Note that the best performance in each measure on each problem is shaded a dark grey, and the second best is a lighter grey.

Table 1. Epsilon. Mean and standard deviation.

	AMOSA	NSGA2	OMOPSO	SMPSO	AbYSS
DTLZ1	$1.50e-01_{5.6e-02}$	$1.48e-01_{1.2e-01}$	$4.69e+01_{1.4e+01}$	$5.81e-02_{6.4e-03}$	$1.20e-01_{1.2e-01}$
DTLZ2	$1.70e-01_{3.4e-02}$	$1.25e-01_{1.8e-02}$	$6.07e-01_{1.0e-01}$	$1.39e-01_{1.2e-02}$	$1.29e-01_{1.6e-02}$
DTLZ3	$8.83e-01_{7.6e-01}$	$5.04e+00_{2.4e+00}$	$4.34e+02_{9.0e+01}$	$3.22e-01_{2.3e-01}$	$5.22e+00_{3.0e+00}$
DTLZ4	$4.39e-01_{1.7e-01}$	$1.13e-01_{1.7e-02}$	$1.11e+00_{2.2e-01}$	$1.25e-01_{2.9e-02}$	$1.64e-01_{1.6e-01}$
DTLZ5	$4.69e-02_{1.4e-02}$	$1.07e-02_{1.9e-03}$	$4.35e-01_{9.4e-02}$	$4.72e-03_{4.0e-04}$	$5.10e-03_{7.3e-04}$
DTLZ6	$1.00e-01_{6.7e-02}$	$8.47e-01_{7.3e-02}$	$5.83e+00_{1.1e+00}$	$2.67e-02_{1.2e-01}$	$6.55e-01_{2.0e-01}$
DTLZ7	$1.52e+00_{3.1e-01}$	$1.78e-01_{2.1e-01}$	$1.03e+01_{2.1e+00}$	$1.74e-01_{3.9e-02}$	$1.65e+00_{7.3e-01}$

From the results, it is difficult to select the best method that achieved the best performance across all the 6 performance measures. However, as a baseline algorithm proposed in 2002, NSGA-II performed moderately well on the 7 problems in terms all the measures. Of the 5 MOEAs, SMPSO appeared to be the best algorithm in terms of the distance to the Pareto-optimal front measured by Epsilon: it yields 5 of the best values

Table 2. Hypervolume. Mean and standard deviation.

	AMOSA	NSGA2	OMOPSO	SMPSO	AbYSS
DTLZ1	$6.32e-01_{1.1e-01}$	$5.07e-01_{3.2e-01}$	$0.00e+00_{0.0e+00}$	$7.37e-01_{5.0e-03}$	$6.49e-01_{2.1e-01}$
DTLZ2	$3.45e-01_{1.4e-02}$	$3.74e-01_{5.6e-03}$	$1.89e-02_{6.5e-02}$	$3.51e-01_{4.9e-03}$	$3.82e-01_{6.3e-03}$
DTLZ3	$3.40e-02_{3.6e-02}$	$1.86e-03_{1.0e-02}$	$0.00e+00_{0.0e+00}$	$2.88e-01_{8.4e-02}$	$0.00e+00_{0.0e+00}$
DTLZ4	$7.48e-02_{5.4e-02}$	$3.75e-01_{4.6e-03}$	$1.18e-02_{6.4e-02}$	$3.57e-01_{1.3e-02}$	$3.67e-01_{5.3e-02}$
DTLZ5	$8.08e-02_{6.3e-03}$	$9.29e-02_{1.7e-04}$	$3.39e-03_{1.7e-02}$	$9.37e-02_{9.5e-05}$	$9.40e-02_{4.0e-05}$
DTLZ6	$7.51e-02_{9.6e-03}$	$0.00e+00_{0.0e+00}$	$3.16e-03_{1.7e-02}$	$9.17e-02_{1.7e-02}$	$4.89e-05_{2.6e-04}$
DTLZ7	$2.46e-01_{2.1e-02}$	$2.80e-01_{6.0e-03}$	$8.57e-03_{4.6e-02}$	$2.74e-01_{5.9e-03}$	$2.44e-01_{2.5e-02}$

Table 3. Spread. Mean and standard deviation.

	AMOSA	NSGA2	OMOPSO	SMPSO	AbYSS
DTLZ1	$1.20e+00_{1.7e-01}$	$8.47e-01_{1.9e-01}$	$6.09e-01_{8.7e-02}$	$6.78e-01_{5.0e-02}$	$9.51e-01_{1.1e-01}$
DTLZ2	$8.34e-01_{8.6e-02}$	$6.98e-01_{5.0e-02}$	$5.58e-01_{5.6e-02}$	$6.15e-01_{3.1e-02}$	$7.71e-01_{5.4e-02}$
DTLZ3	$1.30e+00_{1.6e-01}$	$1.02e+00_{1.3e-01}$	$6.09e-01_{5.6e-02}$	$8.74e-01_{2.9e-01}$	$9.38e-01_{9.7e-02}$
DTLZ4	$1.73e+00_{2.2e-01}$	$6.72e-01_{3.9e-02}$	$1.08e+00_{1.4e-01}$	$6.76e-01_{6.5e-02}$	$6.57e-01_{1.1e-01}$
DTLZ5	$1.04e+00_{7.8e-02}$	$4.40e-01_{4.0e-02}$	$5.79e-01_{8.9e-02}$	$1.72e-01_{5.9e-02}$	$1.35e-01_{1.8e-02}$
DTLZ6	$1.50e+00_{8.5e-02}$	$8.40e-01_{5.5e-02}$	$5.93e-01_{9.8e-02}$	$1.49e-01_{1.1e-01}$	$8.99e-01_{6.7e-02}$
DTLZ7	$1.32e+00_{6.3e-02}$	$7.40e-01_{4.3e-02}$	$6.65e-01_{6.3e-02}$	$6.99e-01_{3.8e-02}$	$7.03e-01_{5.4e-02}$

Table 4. Coverage. Relating this table to the conventional C(X,Y) notation, the algorithm named in the row heading represents X, and the algorithm named in the column heading replaces Y. For example, C(AMOSA, NSGA2) = 0.460 on DTLZ1.

		AMOSA	NSGA2	OMOPSO	SMPSO	AbYSS
DTLZ1	AMOSA	–	0.460	1.0	0.293	0.252
	NSGA2	0.698	–	1.0	0.515	0.317
	OMOPSO	0.003	0.120	–	0.0	0.0
	SMPSO	0.804	0.58	1.0	–	0.347
	AbYSS	0.889	0.691	1.0	0.663	–
DTLZ2	AMOSA	–	0.196	0.965	0.530	0.025
	NSGA2	0.417	–	0.968	0.651	0.017
	OMOPSO	0.102	0.040	–	0.125	0.008
	SMPSO	0.373	0.142	0.954	–	0.02
	AbYSS	0.538	0.480	0.976	0.809	–
DTLZ3	AMOSA	–	0.969	1.0	0.128	1.0
	NSGA2	0.533	–	1.0	0.1232	0.996
	OMOPSO	0.0	0.0	–	0.001	0.0
	SMPSO	1.0	1.0	1.0	–	1.0
	AbYSS	0.243	0.951	1.0	0.121	–
DTLZ4	AMOSA	–	0.037	0.831	0.043	0.020
	NSGA2	0.037	–	0.936	0.454	0.017
	OMOPSO	0.070	0.024	–	0.052	0.003
	SMPSO	0.100	0.220	0.930	–	0.029
	AbYSS	0.129	0.492	0.968	0.682	–
DTLZ5	AMOSA	–	0.196	0.916	0.168	0.015
	NSGA2	0.765	–	0.951	0.439	0.047
	OMOPSO	0.327	0.062	–	0.022	0.004
	SMPSO	0.780	0.388	0.951	–	0.075
	AbYSS	0.887	0.601	0.986	0.706	–
DTLZ6	AMOSA	–	1.0	0.957	0.109	1.0
	NSGA2	0.007	–	0.953	0.018	0.225
	OMOPSO	0.021	1.0	–	0.044	1.0
	SMPSO	0.021	1.0	0.959	–	1.0
	AbYSS	0.017	1.0	0.953	0.027	–
DTLZ7	AMOSA	–	0.262	0.911	0.296	0.125
	NSGA2	0.191	–	0.975	0.566	0.047
	OMOPSO	0.032	0.046	–	0.056	0.016
	SMPSO	0.276	0.425	0.972	–	0.078
	AbYSS	0.309	0.562	0.970	0.641	–

Table 5. Purity. We also report how many solutions each algorithm returned in total, and how many of those remained non-dominated with respect to solutions of other algorithms (thus (X/Y) means it returned Y solutions, of which X were not dominated by any other.)

	AMOSA	NSGA2	OMOPSO	SMPSO	AbYSS
DTLZ1	0.027 (81/2979)	0.271 (813/3000)	0.0 (0/720)	0.258 (775/3000)	0.546 (1638/2997)
DTLZ2	0.196 (590/3000)	0.466 (1399/3000)	0.012 (13/1074)	0.124 (374/3000)	0.924 (2773/3000)
DTLZ3	0.0 (0/2960)	0.0 (0/2334)	0.0 (0/686)	0.468 (1156/2467)	0.0 (0/2802)
DTLZ4	0.082 (197/2385)	0.477 (1433/3000)	0.019 (11/565)	0.248 (745/2999)	0.929 (2788/3000)
DTLZ5	0.083 (251/2998)	0.346 (1039/3000)	0.006 (5/806)	0.222 (668/3000)	0.851 (2552/2998)
DTLZ6	0.979 (2937/3000)	0.0 (0/3000)	0.039 (85/2138)	0.810 (2416/2982)	0.0 (0/2953)
DTLZ7	0.533 (1599/3000)	0.317 (953/3000)	0.009 (6/613)	0.230 (690/2999)	0.738 (2214/2999)

Table 6. Total UH.

	AMOSA	NSGA2	OMOPSO	SMPSO	AbYSS
DTLZ1	$2.05e-12$	$7.79e-11$	0.0	$7.91e-11$	$1.14e-10$
DTLZ2	$1.25e-10$	$9.13e-10$	$4.13e-12$	$3.01e-10$	$1.69e-09$
DTLZ3	0.0	0.0	0.0	$2.11e-40$	0.0
DTLZ4	0.0	$2.62e-09$	$7.49e-12$	$8.98e-10$	$4.40e-09$
DTLZ5	$2.51e-11$	$4.17e-10$	$2.53e-12$	$2.63e-10$	$1.06e-09$
DTLZ6	$2.59e-09$	0.0	$3.97e-10$	$7.31e-09$	0.0
DTLZ7	$7.33e-11$	$1.59e-09$	$7.45e-12$	$1.66e-09$	$1.89e-09$

and 1 second best on of the 7 problems. We can see that SMPSO obtained 3 of the best values and 2 second best in the Hypervolume measure, and it also obtained 1 of the best values and 5 second best values in the Spread measure.

The AbYSS algorithm scored the highest purity rating on 5 of the 7 test problems. The coverage results in these cases confirm that it is rare for the solutions of another algorithm to cover more than 2% of its solutions. Curiously, on the remaining 2 problems, all of its solutions were fully dominated by others. Despite having so many non-dominated solutions, this did not result in relatively high amounts of UH; information which is open to interpretation. Our opinion is that its frequently poor Spread results are suggestive of tightly clustered solutions on the Pareto front, which UH strongly punishes. In support of this, we can say that on problem 4, when it has the best spread and second best hypervolume, it has the best UH.

Interestingly, OMOPSO performed the poorest across all 7 problems in terms of Epsilon, despite regularly scoring the best according to the Spread metric. The purity results for OMOPSO reveal it is consistently yielding less solutions than the other algorithms. Our observation of the inferior performance of OMOPSO compared with other MOEAs is consistent with the results in [2], where the authors found that while OMOPSO outperformed other multi-objective PSO algorithms, second only to SMPSO.

The performance of AMOSA is also not satisfactory. In terms of Epsilon and Hypervolume, it only obtained 2 second best values. It is consistently outperformed by the other MOEAs in terms of both Spread and UH.

Problem 2 produced an example of the interesting information UH can provide. The best hypervolume score was awarded to AbYSS with $3.82e-01$, whereas NSGA-II was a very close second with $3.74e-1$. However, NSGA-II found nearly double the amount of UH as AbYSS, a surprising difference given the initial figures. This result also coincides with NSGA-II performing superior to AbYSS on problem 2 in both Spread and Epsilon.

Perhaps the most remarkable set of results emerged on problem 3. Some solutions of SMPSO dominated the solutions found by every other algorithm. The purity measure shows that 1156 of its 2467 total solutions were preserved on the non-dominated front, showing that this was a consistently superior performance, not a single fluke result. The total UH of SMPSO in problem 3 being incredibly low indicates that those 1156 solutions were closely clustered together.

Choosing the best performer depends upon the preferences of the user. On the one hand, an indicator that reflects poorly on SMPSO is purity; in this it is often outperformed by AbYSS, NSGA-II and AMOSA. Looking at the coverage results, we can see that it is common for AbYSS or NSGA-II to dominate over 50% of its solutions. Given that it frequently came best or second best in terms of Epsilon, Hypervolume and Spread, we must assume that the few remaining non-dominated solutions are responsible for the significant scores. Thus, if the priority is to find a smaller set of solutions closest to the global optimal Pareto front, SMPSO would be the correct algorithm to select. However, if finding a broader set of trade-off solutions is necessary, AbYSS may be preferred due to its impressive performance in purity.

4 Conclusion

In this paper, we conducted a fair and systematic comparison of four representative novel MOEAs proposed in recent years, adopting the most popular MOEA, NSGA-II as the baseline algorithm for comparison. We employed a range of well-known performance measures to evaluate the performance of the 5 MOEAs on the DTLZ problem family. We also proposed a novel performance metric, called unique hypervolume (UH), which can effectively quantify the volume of objective space dominated uniquely by the solution of an algorithm against those of other algorithms. Based on our results, we observed that for finding a smaller set of good solutions, SMPSO would be the best choice. If we are more interested in discovering the trade-off region in multi-objective problems, AbYSS is preferred. In terms of future work, due to the interesting insight into the results gained through the measurement of total unique hypervolume, especially its ability to quantify the novelty of solution within a set, we intend to explore this idea further. More specifically, we will be using the measure to observe performance of multi-objective optimisation algorithms on design problems, where innovation is particularly valued.

Acknowledgment

Mr Thomas White and Dr Shan He are supported by EPSRC (EP/J01446X/1).

References

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

2. Durillo, J.J., García-Nieto, J., Nebro, A.J., Coello, C.A., Luna, F., Alba, E.: Multi-objective particle swarm optimizers: An experimental comparison. In: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization. EMO '09, Berlin, Heidelberg, Springer-Verlag (2009) 495–509
3. Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K.: A simulated annealing-based multiobjective optimization algorithm: Amosa. Evolutionary Computation, IEEE Transactions on **12**(3) (2008) 269–283
4. Reyes, M., Coello Coello, C.: Improving pso-based multi-objective optimization using crowding, mutation and ϵ-dominance. In Coello, C., Hernández, A., Zitler, E., eds.: Third International Conference on Evolutionary MultiCriterion Optimization, EMO 2005. Volume 3410 of LNCS., Springer (2005) 509–519
5. Nebro, A.J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J.J., Beham, A.: AbYSS: Adapting Scatter Search to Multiobjective Optimization. IEEE Transactions on Evolutionary Computation **12**(4) (August 2008)
6. Nebro, A., Durillo, J., García-Nieto, J., Coello Coello, C., Luna, F., Alba, E.: Smpso: A new pso-based metaheuristic for multi-objective optimization. In: 2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009), IEEE Press (2009) 66–73
7. Okabe, T., Jin, Y., Sendhoff, B.: A critical survey of performance indices for multi-objective optimisation. In: Evolutionary Computation, 2003. CEC'03. The 2003 Congress on. Volume 2., IEEE (2003) 878–885
8. Fonseca, C., Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. In: Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Volume 216. (2005)
9. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. Evolutionary Computation, IEEE Transactions on **3**(4) (1999) 257–271
10. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation **8**(2) (2000) 173–195
11. Bandyopadhyay, S., Pal, S., Aruna, B.: Multiobjective gas, quantitative indices, and pattern classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **34**(5) (2004) 2088–2099
12. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications. Shaker (1999)
13. Durillo, J.J., Nebro, A.J.: jMetal: A java framework for multi-objective optimization. Advances in Engineering Software **42**(10) (2011) 760–771
14. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA). (2002) 825–830