Evolutionary Many-Objective Optimisation: Many Once or One Many?

Evan J. Hughes
Dept. Aerospace, Power & Sensors,
Cranfield University, RMCS, Shrivenham,
Swindon, UK. SN6 8LA
ejhughes@iee.org

Abstract- Multi-objective evolutionary algorithms are widely established and well developed for problems with two or three objectives. However, it is known that for many-objective optimisation, where there are typically more than three objectives, the algorithms applying Pareto optimality as a ranking metric may loose their effectiveness [1].

This paper compares three different approaches to generating Pareto surfaces on both multi and many objective problems. The first approach is using an established Pareto ranking method (NSGA II), the second combines multiple single objective optimisations in a single run (MSOPS), and the third uses multiple runs of a single objective optimiser.

The results demonstrate that much can be gained by generating the entire Pareto set in a single run, when compared to repeated single objective optimisations. It is also clear that NSGA II loses its effectiveness as the problem dimensionality increases – it is more effective to use many single objective optimisations than a Pareto-ranking based optimiser on many-objective problems. Ultimately though, "many once or once many" is dependent on algorithm choice, not problem scale.

1 Introduction

In many real engineering problems, more than two objectives need to be optimised simultaneously. It is believed [1] that all multiobjective optimisation algorithms that use Pareto-ranking as a fundamental selection method (such as in NSGA II) will not perform efficiently on many-objective problems with four or more objectives. As the number of objectives increase, more of the population becomes non-dominated and the selective pressure driving the population towards the Pareto set falls rapidly. Eventually all of the selective pressure arises from the operator that spreads the solutions across the non-dominated surface.

The alternative is to use a method that does not rely on Pareto ranking to sort the population. The simplest of these non-Pareto methods is to use a conventional aggregation approach such as weighted min-max (section 2.4) and perform many single objective optimisations, changing the weight vector set a little each time to enable the entire Pareto surface to be sampled.

A natural extension is to attempt to satisfy all the weight vectors simultaneously in a single run of the optimiser. Multiple Single Objective Pareto Sampling (MSOPS) [2] is one method that develops this concept into a practical algorithm.

The hypotheses to be tested in this paper are:

Hypothesis 1 Given a fixed numer of function evaluations, an optimiser that produces an entire Pareto set in one run is better than generating the Pareto set through many single objective optimisations using an aggregation function.

Hypothesis 2 Optimisers that use Pareto ranking based methods to sort the population will be very effective for small numbers of objectives, but not perform as effectively for many objectives when compared to optimisers based on non-Pareto ranking methods.

The reasoning behind hypothesis 1 is that in the first few generations of the optimisation process, one large population is feeding the bulk of the future Pareto set, re-using many of the early objective calculations. Whereas in an optimiser that is generating a single Pareto point at a time, the effective population size per point is restricted. Although often indicated that repeated single objective optimistions are not as effective as a single multiple objective optimisation [3], no evidence has been found that extends to many-objective problems where more than two objectives exist.

The second hypothesis will give independent experimental confirmation of Purshouse's theoretical results [1] when comparing NSGA II against MSOPS. The experiments will also test the performance of NSGA II against performing multiple runs of a single objective optimiser.

Section 2 details the three optimisation algorithms compared in the paper. Section 3 details the design of the experiments to test the two hypotheses. Section 4 presents the results of the experiments and section 5 concludes.

2 Algorithms For Comparison

2.1 NSGA II

The Non-dominated Sorting Genetic Algorithm II (NSGA II) [4] is proving to be a robust optimisation algorithm for a wide range of multi-objective problems. The NSGA II Pareto ranking algorithm is an elitist system and maintains an external archive of the Pareto solutions.

Fundamentally, the non-dominated sorting used segregates the population into 'layers' by first finding the non-dominated solutions in the population, and labels these points as the first front. These points are removed and the non-dominated solutions in the remaining population are then identified and removed. The process continues until the entire population has been classified into layers.

The algorithm updates the current archive by identifying all the non-dominated solutions in the union of the old archive and current population. The layers are taken in turn until the maximum size of the archive is reached (often the population size). The last layer to be added is truncated if

necessary by employing a crowding distance operator and selecting the solutions with the smallest crowding distance for removal, leading to the most diverse set. The crowding distance is the average of the distances in objective space between the point under consideration and the nearest objective values above and below (calculated independently for each objective axis).

New solutions are created by selecting parents from the archive set. The classic NSGA II algorithm uses a tournament selection system based on crowding distance to break ties. In this paper, the same offspring generation process as detailed in section 2 is used to allow the fairest comparison to be drawn between the three different approaches for developing a Pareto set.

For NSGA II, a population of 100 for 100 generations was used with an archive size of 100. The configuration leads to 10,000 objective calculations per run of the optimiser.

2.2 MSOPS

Multiple Single Objective Pareto Sampling (MSOPS) [2] has been used as the second optimisation process to decide which are 'good' points for the next generation. MSOPS performs multiple single-objective aggregation-based optimisations in a single run. Each of the aggregated optimisations is directed by a vector of weights, or target vector. The key advantage is that the algorithm does not rely on Pareto ranking to provide selective pressure. As the target vectors are decided *a-priori*, MSOPS provides an active probing of the Pareto set, rather than passive discovery.

The operation of MSOPS is to generate a set of T target vectors, and evaluate the performance of every individual in the population, of size P, for every target vector, based on a conventional aggregation method. As aggregation methods (eg. weighted min-max, ϵ-constraint, goal attainment etc.) are very simple to process, the calculation of each of the performance metrics is fast.

Thus each of the P members of the population has a set of T scores that indicate how well the population member satisfied the range of target conditions. The scores are held in a score matrix, S, which has dimensions $P \times T$. Each *column* of the matrix S corresponds to one target vector (each column containing P entries) and is ranked, with the best performing population member on the corresponding target vector being given a rank of 1, and the worst a rank of P. The rank values are stored in a matrix R. Each *row* of the rank matrix R may now be sorted, with the ranks for each population member placed in ascending order. The R matrix now holds in the first column the highest rank achieved for each population member across the set of target vectors. The second column will hold the second highest rank achieved etc. Thus the matrix R may be used to rank the population, with the most fit being the solution that achieved the most scores which were ranked 1 etc.

The flexibility of the approach is such that the target vectors can be arbitrary, either generated using some structure, or generated at random within certain limits. As the ranking method employed is based on the number of target vectors that are satisfied the best, a solution at the edge of the objective space will often satisfy vectors that cannot be attained. The focus of the optimisation is naturally drawn to interesting regions of surface such as the boundary of the optimisation surface and discontinuities. As a consequence though, the efficiency of the algorithm is reduced in relation to the number of unobtainable target vectors.

In the trials, only weighted min-max (section 2.4) is used, along with 50 target vectors spread uniformly across the search space (2D / 4D / 6D). Each run used a population of 100 for 100 generations with 50 target vectors, leading to a total of 10,000 objective calculations to generate a Pareto set.

2.3 Repeated Single Objective

In the Repeated Single Objective (RSO) approach, a conventional single objective EA is used, based on the weighted min-max (section 2.4) aggregation function and each weight vector in turn, but with a correspondingly smaller population size and number of generations.

Thus for the 50 weight vector set (the same vector sets are used for MSOPS and RSO to make the results more comparable), 50 separate optimisations are performed. Each single optimisation-run used a population of 10 for 20 generations, one run for each of the 50 weight vectors, leading to a total of 10,000 objective calculations to generate a Pareto set, thus all three methods use exactly the same number of objective calculations to create the Pareto surface.

2.4 Weighted Min-Max

The weighted min-max score of k objectives is calculated using (1),where w_i is the weight of the i$^{\text{th}}$ objective, O_i.

$$s = \max_{i=1}^{k}(w_i O_i) \tag{1}$$

Weighted min-max is able to generate points on both convex and concave Pareto sets. If the optimisation process converges to a solution that exactly 'matches' the weight vector, then $w_1 O_1 = w_2 O_2 = \ldots$, allowing the convergence of the solution with respect to the weights to be assessed. The weight vector corresponds to a point on the Pareto set in the true direction given by the vector $V = [1/w_1,\ 1/w_2,\ \ldots]$. Thus the angle between the vectors V and O indicate whether the solution lies where it was expected or not. If the vector V lies within a discontinuity of the Pareto set, or is outside of the entire objective space, then the angle between the two vectors will be significant. By observing the distribution of the final angular errors across the total weight set, the limits of the objective space and discontinuities within the Pareto set can be identified. This active probing of regions of interest is only available in RSO and MSOPS, compared to NSGA II where the placement of Pareto points is stochastic.

2.5 Core Algorithm Structure

For the results presented in the paper, a method based on Differential Evolution has been used to create the offspring

population for the next generation.

Differential Evolution [5] is an evolutionary technique that uses reproduction that is related to the current spatial distribution of the population. The algorithm generates new chromosomes by adding the weighted difference between two chromosomes to a third chromosome. At each generation, for each member of the parent population, a new chromosome is generated. Elements of this new chromosome are then crossed with the parent chromosome to generate the child chromosome. The child chromosome is evaluated using the objective function. The size and direction of the difference between any pair of chromosomes is determined by the overall spread of the current population. Thus the DE algorithm self adapts to the fitness landscape, reducing the size of the mutations automatically as the search converges. In this way, no separate probability distribution has to be used for mutation which makes the scheme completely self-organising.

The trial chromosome $\vec{P}_t$ may be described as in (2).

$$\vec{P}_t = F(\vec{P}_a - \vec{P}_b) + \vec{P}_c \tag{2}$$

Where chromosomes $\vec{P}_a$, $\vec{P}_b$ & $\vec{P}_c$ are chosen from the population without replacement and F is a scaling factor.

The crossover process is controlled by a crossover parameter C. The crossover region begins at a randomly chosen parameter in the chromosome and then a segment of length L genes is copied from $\vec{P}_t$ to the parent chromosome to create the child chromosome. If the segment is longer than the remaining length of the chromosome, the segment is wrapped to the beginning of the chromosome. The length L is chosen probabilistically and is given by (3).

$$P(L \geq v) = (C)^{v-1}, v > 0 \tag{3}$$

In general, the scaling parameter F and the crossover parameter C lie in the range $[0.5, 1]$. Small values of F mean that the population spread reduces faster and this is more likely to result in the algorithm converging quickly at a local minima. In this paper values of 0.7 for both F and C have been used.

Constraints have been applied using a simple priority method. The constrained and unconstrained solutions are separated and ranked separately, with the best performing solutions in each set being given the rank of 1. The rank value of the worst unconstrained result is then added to the rank values of the population members that are constrained. Thus solutions that violate constraints will always appear worse than the unconstrained solutions. The ranking values are used only for the truncation of the population. The new population is generated in all the three methods by using each individual in turn as a base vector, and then combining with three other vectors chosen at random without replacement. This is not the same approach used conventionally in NSGA II, but allows the three ranking methods to be compared equally. The genes are all real-valued and the chromosomes are the same dimensionality as the number of objectives.

3 Experimental Design

3.1 Introduction

Three experiments have been performed on 2, 4 and 6 objective test problems. Each experiment follows the same format, but for a different number of dimensions in the objective function and chromosome.

For each test problem, 100 independent trials are performed where each optimiser produces a non-dominated set for analysis. A compound Pareto set is then generated based on all 300 non-dominated surfaces (3 algorithms, 100 Pareto sets each). This Pareto set is used as a reference set for the hypervolume comparisons.

For each of the 300 non-dominated surfaces, the hypervolume is calculated to indicate performance. The Wilcoxon rank test is then applied to the three pair-wise combinations of the algorithms, leading to 200 hypervolume results being used to generate each of the test statistics from the rank test (100 from algorithm A, 100 from algorithm B etc.).

The three methods only differ in the means of assessing which solutions are better than others, with the recombination and all other parameters either fixed or directly comparable. Thus the algorithms may not be optimally tuned for solving the three test problems and although each algorithm may be able to achieve better results with tuning (such as selecting a more targeted weight vector set), the comparisons made in this paper are fair.

3.2 Hypervolume Metric

Fundamentally, the hypervolume metric [6] assesses the total area that lies between a chosen reference point that acts as a corner to a hypercube, and the Pareto surface which intersects the hypercube.

For minimisation, the reference point is placed in such a way as to be at least weakly dominated by every member of the set to be investigated. Thus for these experiments, the reference point is chosen as the maximum value observed in any Pareto set reported from any of the three methods across all 100 repeat trials (each test function will attract a different reference point due to the changes in dimensionality). The better the Pareto set, the larger the hypervolume indicator will be.

The hypervolume metric intrinsically accounts for not just the relative proximity to the true Pareto set, but also the diversity of solutions obtained across the set, especially at the edges. It is also desirable for the optimiser to produce an even spread of solutions across the front. The evenness of spread is not assessed by the hypervolume metric, but it is known that NSGA II often achieves well spread results, and the spread of MSOPS and RSO is determined entirely by the choice of target vectors. Therefore just the single metric of hypervolume will be used to assess algorithm performance.

3.3 Wilcoxon Rank Test

Each of the experiments will generate three sets of 100 trial Pareto sets. The hypervolume metric is calculated for each

of the 300 trial Pareto sets, and the hypervolume results can be then compared using a non-parametric statistical measure.

The Wilcoxon rank test (also referred to as Mann-Whitney test) is a non-parametric comparison, and as we do not know what the likely distribution of results will be, is appropriate. The test operates by creating a sorted list of the total 200 hypervolume metric results from the two methods that are being compared currently. The individual rank values achieved by the metric results of each of the two methods are summed and the difference between the rank sums observed and related to a test statistic Z.

If one population of results is consistently higher (therefore better), that population will have more solutions in the top of the ranking and will therefore achieve a lower rank sum total. If the populations of results are drawn from the same distribution, the rank sums will be more equal.

Assuming the null hypothesis that the distribution of the hypervolume metric results are the same for two different methods, we can reject the null hypothesis at a significance level of $\alpha = 0.05$.

The Z statistic is compared against the standard normal distribution probability density function in order to derive significance probability. A level of $\alpha = 0.05$ corresponds to a Z value of 2.81 for a two-tailed test. If the magnitude of the Z value is greater than 2.81, the null hypothesis that the distributions of the hypervolume results are the same can be rejected. The sign of the Z value indicates which of the sets of data are superior.

3.4 Two Objective Problem

To test the behaviour of the three optimisers in two dimensions, a concave and discontinuous Pareto surface has been used [7]. The gene values all lie in the range [0,1] and the objectives are to be minimised.

$$
\begin{aligned}
&F1: \\
&O_1 = x \\
&O_2 = y \\
&0 \geq -(x)^2 - (y)^2 + 1 + 0.1\cos\left(16\arctan\left(\frac{x}{y}\right)\right) \\
&0.5 \geq (x-0.5)^2 + (y-0.5)^2 \\
&0 \leq x, y \leq 1
\end{aligned} \tag{4}
$$

The function has been scaled so the decision space lies within the range $0 \leq x, y \leq 1$.

3.5 Four and Six Objective Problems

For the four and six objective problems, a continuous concave Pareto set was used that has a low density of solutions at the Pareto front. The gene values all lie in the range [0,1] and the objectives are to be minimised.

$$F2\&F3:$$

Table 1: Table of statistical results from the Wilcoxon rank test for 2 objectives

Comparison	Z	Summary
MSOPS vs. RSO	12.22	MSOPS $>$ RSO
RSO vs. NSGA II	-12.22	RSO $<$ NSGA II
MSOPS vs. NSGA II	-8.04	MSOPS $<$ NSGA II

$$
\begin{aligned}
O_i &= 1 - x_i^2 \\
0.5 &< \sum_{i=1}^{N} O_i.^2
\end{aligned} \tag{5}
$$

In equation 5, any N objectives can be accounted for.

4 Results

4.1 Two Objective Problem

Figure 1 shows a plot of the sorted hypervolume metric results from the first experiment. The results show clearly that NSGA II outperforms MSOPS which outperforms RSO.

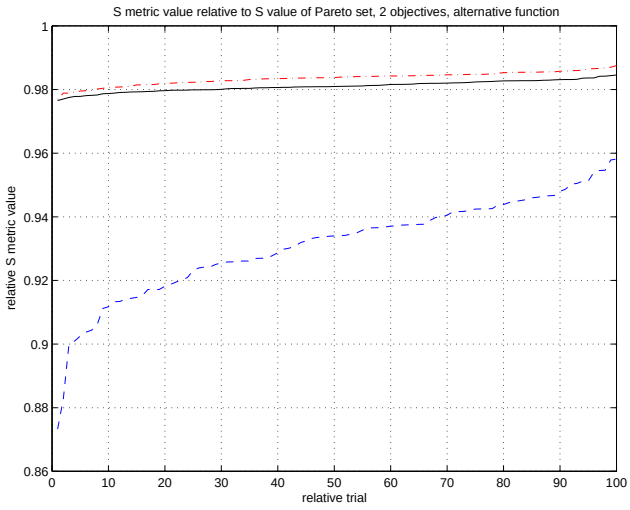

Figure 1: Comparison of sorted hypervolume results for 2 objectives, MSOPS (solid), RSO (dashed) and NSGA II (dot-dash), relative to the hypervolume of the combined Pareto set (larger values better). The results show clearly that NSGA II outperforms MSOPS which outperforms RSO

Table 1 shows the Z test statistic from the Wilcoxon rank test. The tests were performed based on the order in column 1 of the table. For example, the first results show MSOPS vs. RSO and have a positive Z value greater than 2.81, demonstrating that the null hypothesis can be rejected and that MSOPS performs better than RSO. This result is summarised in column 3.

The statistical analysis rejects the null hypothesis in all cases. Although the differences in performance appear large, as figures 2 and 3 show, the difference observable in the Pareto sets is not so distinct, demonstrating the requirement of quantative rather than subjective comparison.

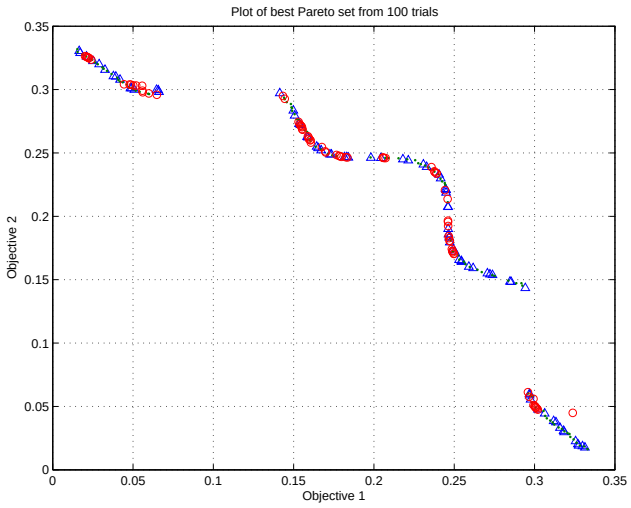

Figure 2: Plot of worst objective fronts for each of the optimisation methods. NSGA II is '.', MSOPS is '△' and RSO is '○'

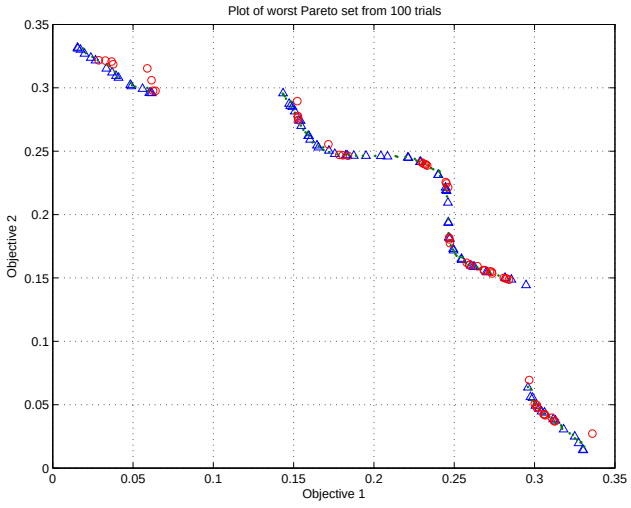

Figure 3: Plot of best objective fronts for each of the optimisation methods. NSGA II is '.', MSOPS is '△' and RSO is '○'

Table 2: Table of statistical results from the Wilcoxon rank test for 4 objectives

Comparison	Z	Summary
MSOPS vs. RSO	12.22	MSOPS $>$ RSO
RSO vs. NSGA II	9.97	RSO $>$ NSGA II
MSOPS vs. NSGA II	12.22	MSOPS $>$ NSGA II

Table 3: Table of statistical results from the Wilcoxon rank test for 6 objectives

Comparison	Z	Summary
MSOPS vs. RSO	12.20	MSOPS $>$ RSO
RSO vs. NSGA II	12.22	RSO $>$ NSGA II
MSOPS vs. NSGA II	12.22	MSOPS $>$ NSGA II

4.2 Four Objective Problem

Figure 4 shows a plot of the sorted hypervolume metric results from the second experiment. The results show clearly that MSOPS outperforms RSO which outperforms NSGA II.

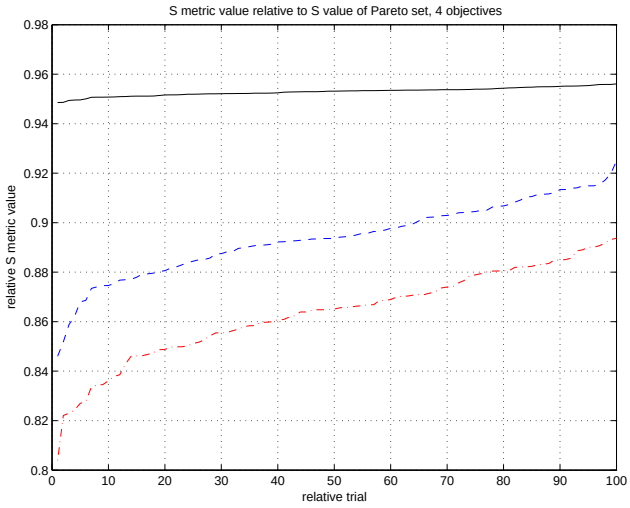

Figure 4: Comparison of sorted hypervolume results for 4 objectives, MSOPS (solid), RSO (dashed) and NSGA II (dot-dash), relative to the hypervolume of the combined Pareto set (larger values better). The results show clearly that MSOPS outperforms RSO which outperforms NSGA II

Table 2 shows the Z test statistic from the Wilcoxon rank test. The statistical analysis rejects the null hypothesis in all cases.

4.3 Six Objective Problem

Figure 5 shows a plot of the sorted hypervolume metric results from the third experiment. The results show clearly that MSOPS outperforms RSO which outperforms NSGA II.

Table 3 shows the Z test statistic from the Wilcoxon rank test. The statistical analysis rejects the null hypothesis in all cases.

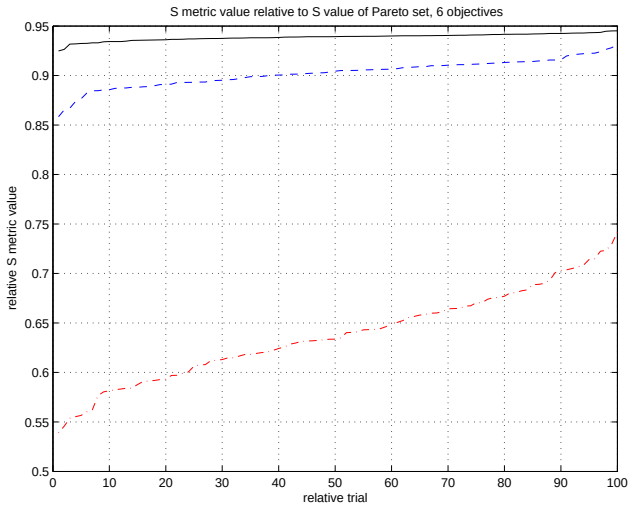

Figure 5: Comparison of sorted hypervolume results for 6 objectives, MSOPS (solid), RSO (dashed) and NSGA II (dot-dash), relative to the hypervolume of the combined Pareto set (larger values better). The results show clearly that MSOPS outperforms RSO which outperforms NSGA II

4.4 Discussion

It is clear from the results in the tables where the repeated single objective optimisation and MSOPS are superior to NSGA-II that Purshouse's [1] analysis was correct and that all methods that rely on Pareto ranking are only suitable for multi-objective problems and not many-objective problems. The experiments also demonstrate conclusively that running many single objective optimisations is superior to Pareto ranking based methods when many-objective problems are being considered, i.e. NSGA II is only better than MSOPS and RSO for two objectives, otherwise the non-Pareto ranking methods are better. Hypothesis 2 is demonstrated with statistical significance on the objective functions used in this paper.

The experiments have also demonstrated that running many single objective optimisations in one run is superior to running individual single objective optimisations over many runs, i.e. MSOPS is consistently better than RSO. Thus hypothesis 1 is demonstrated for the objectives used in this paper.

5 Conclusions

The experiments have shown with a very high statistical significance that the two hypothses can be both accepted:

Hypothesis 1 Given a fixed numer of function evaluations, an optimiser that produces an entire Pareto set in one run is better than generating the Pareto set through many single objective optimisations using an aggregation function.

Hypothesis 2 Optimisers that use Pareto ranking based methods to sort the population will be very effective for small numbers of objectives, but not perform as effectively for many-objectives when compared to optimisers based on non-Pareto ranking methods.

Although only a limited range of test functions have been presented, as hypothesis 1 has been shown to be true in these situations, under the hypervolume metric, the hypothesis is likely to be true in many others functions and metrics. Still the results demonstrate that much can be gained by generating the entire Pareto set in a single run, when compared to repeated single objective optimisations. It is also clear that NSGA II loses its effectiveness as the problem dimensionality increases – it is more effective to use many single objective optimisations than a Pareto-ranking based optimiser on many-objective problems. Ultimately though, "many once or once many" is dependent on algorithm choice not problem scale as the performance of MSOPS vs. RSO and NSGA II vs. RSO have shown.

Bibliography

[1] Robin C. Purshouse. Evolutionary many-objective optimisation: An exploratory analysis. In *The 2003 Congress on Evolutionary Computation (CEC 2003)*, volume 3, pages 2066–2073, Canberra, Australia, 8–12 December 2003. IEEE.

[2] Evan J. Hughes. Multiple single objective pareto sampling. In *Congress on Evolutionary Computation 2003*, pages 2678–2684, Canberra, Australia, 8-12 December 2003. IEEE.

[3] Carlos A. Coello Coello. An updated survey of ga-based multiobjective optimization techniques. In *ACM Computing Surveys*, volume 32, pages 109–143, June 2000.

[4] K. Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Marc Schoenauer, K. Deb, G. Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – PPSN VI*, pages 849–858, Berlin, 2000. Springer.

[5] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI Berkeley, 1995. http://http.ICSI.Berkeley.edu/storn/code.html.

[6] E. Zitzler. *Evolutionary algorithms for Multiobjective Optimisation: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[7] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino. Ga-based decision support system for multicriteria optimization. In *Conference on Systems, Man and Cybernetics: Intelligent Systems for the 21st Century*, volume 2, pages 1556–1561. IEEE, 22-25 October 1995.