MSOPS-II: A general-purpose Many-Objective optimiser

Evan J. Hughes, *Member, IEEE, IET*

***Abstract*— Existing evolutionary methods capable of true Many-Objective optimisation have been limited in their application: for example either initial search directions need to be specified *a-priori*, or the use of hypervolume limits the search in practice to less than 10 objective dimensions.**

This paper describes two extensions to the Multiple Single Objective Pareto Sampling (MSOPS) algorithm. The first provides automatic target vector generation, removing the requirement for initial *a-priori* designer intervention; and secondly redefines the fitness assignment method to simplify analysis and allow more comprehensive constraint handling.

The significant enhancements allow the new MSOPS-II ranking process to be used as part of a general-purpose multi/many objective optimisation algorithm, requiring minimal initial configuration.

I. Introduction

Recent research has demonstrated that many-objective optimisation, where 4 or more objectives are being considered, is not trivial [1], [2], [3]. Evolutionary optimisation algorithms have been developed that are capable of generating a useful approximation of the Pareto surface when many objectives are considered, however until now they have had limitations: for example the first true many-objective evolutionary optimiser, MSOPS [4], required a set of target vectors that define the search direction to be specified *a-priori*; SMS-MOEA [5] is based on the hypervolume metric and consequently processing is only practical at present for less than 10 objectives due to the time-consuming calculation of the hypervolume. This paper describes two extensions to the original MSOPS algorithm that enable it to be used as a general-purpose objective ranking method for 2+ objectives.

The first extension is the re-definition of the objective aggregation process. Originally a double ranking approach was employed which had a tendency to focus heavily on the edges of the Pareto front regions. The new aggregation method can be described in a compact analytic form and can be processed more quickly. The new aggregation method provides a more consistent search focus during the optimisation run, although the difference between final non-dominated fronts generated by the two methods is statistically insignificant. Multiple assessment metrics, such as weighted min-max and VADS can still be applied simultaneously. One key advantage of the new aggregation method is to provide a numerical fitness assignment to each member of the population, allowing a range of ranking methods to be employed. In particular, this paper discusses how the process of stochastic ranking [6] can be employed for controlled constraint handling.

Evan J. Hughes is with the Department of Aerospace, Power and Sensors, Cranfield University, DCMT, Shrivenham, UK. SN4 8AE (phone: +44(0)1793 785255; fax: +44(0)1793 785902; email: ejhughes@theiet.org).

The second extension is the ability to generate the target vectors for the search on-the-fly, removing the need to specify any search directions *a-priori*. Thus when an unknown problem is first tackled, the automatic target vector generation system can be used to explore the many-objective space, calculating approximations of the objective scalings and utopia point. The information on the objective structure can then be used to target specific regions of the objective space using an MSOPS *a-priori* target vector specification process and therefore aid the decision maker in generating informed selection of the final solutions. Additionally, a full archive of the best solution set identified by each of the metrics is maintained by the algorithm in order to aid final decision making, and also for use within the reproduction methods to allow a true elitist behaviour.

Section II describes the original MSOPS algorithm and describes the new fitness aggregation method. Section III details the automatic target vector generation process. Section IV provides a comparison and discussion of the new MSOPS-II algorithm against the original MSOPS system. Finally section V concludes.

II. Multiple Single Objective Pareto Sampling

Multiple Single Objective Pareto Sampling (MSOPS) [4] is a technique that allows multiple single objective optimisation searches to be run in parallel and therefore exploit a larger effective working population. Each of the aggregated optimisations is directed by its own vector of weights, or target vector. Thus the algorithm uses a matrix of target vectors to search in parallel. It is also possible to combine searches in different directions, with different reference points, searches using different aggregation functions, all within a single optimisation run. The key advantage is the algorithm does not rely on Pareto ranking to provide selective pressure. When 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 *a-priori* target vectors, T, and evaluate the performance of every individual in the population, 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 members of the population set P has a set of scores, one for each member of T, 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 $||P|| \times ||T||$, where $||\cdot||$ indicates set cardinality. Figure 1 demonstrates the process for a single

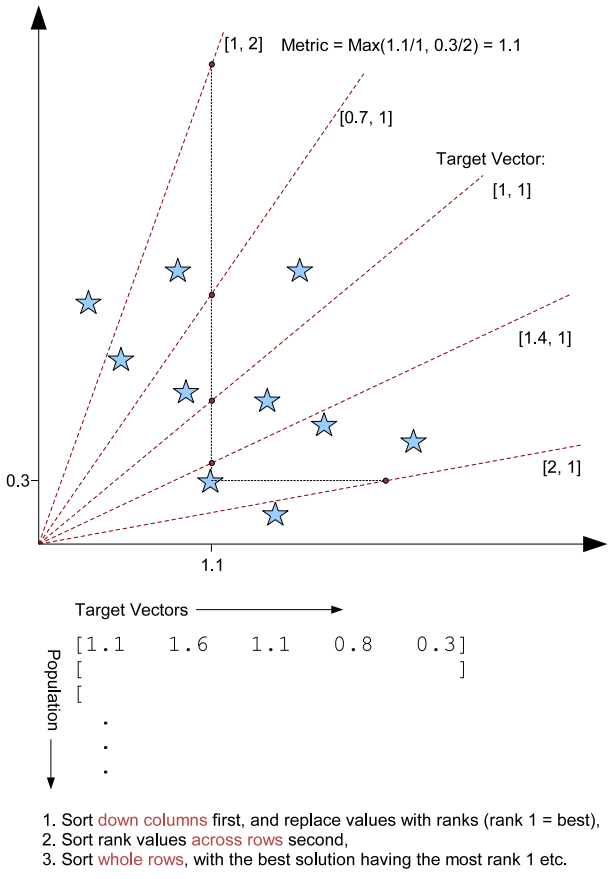

Fig. 1. Process of generating MSOPS ranking. Details are for one population member at [1.1, 0.3], forming one row of matrix S for the weighted min-max aggregation method with 5 target vectors. Note: The target vectors do not need to be normalised to unit length as they are sorted in independent columns.

population member using weighted min-max as the aggregation function. Each *column* of the matrix S corresponds to one target vector (across the population P) and is now 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.

In this original ranking process, if a solution was at the edge of the Pareto/Objective front and there were target vectors beyond the region as demonstrated in figure 2, the population member right at the edge would be the best on all of the unobtainable target vectors. Thus there is significant selective pressure towards solutions at the edges of the front (extreme edges, or edges of discontinuities), but could cause a clustering of the population in these regions. More fundamentally, the use of an *a-priori* target vector generation process required much decision maker input but could suffer heavily from a loss of performance if many target vectors did not transect the region of feasible objective space. These target vectors serve to focus the search at the edges of the region, but otherwise go unused and can curtail the efficiency of the algorithm. By generating target vectors automatically, it is this loss of efficiency that is corrected.

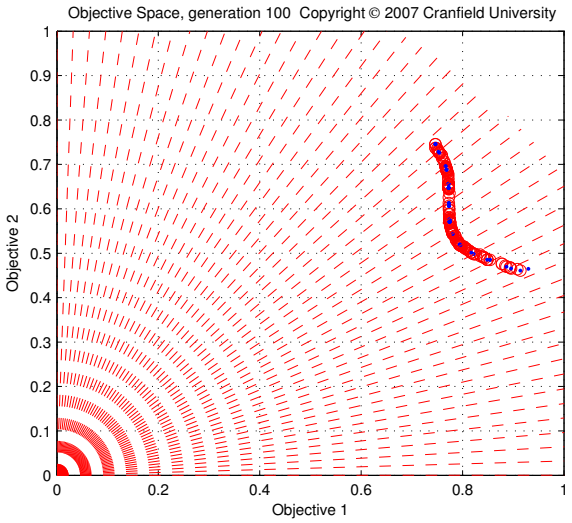

Fig. 2. Original *a-priori* target vector generation method showing scenario with many target vectors not intersecting the feasible objective surface. The dashed lines show the direction of the target vectors, circles represent the archive of non-dominated points, the dots are the locations of the population members on the final generation.

Additionally, as the ranking was based on a lexicographical ordering process, it was not trivial to incorporate methods such as stochastic ranking to handle constraints. In the original MSOPS algorithm, the constrained solutions (infeasible) were removed from the population and the valid solutions (feasible) ranked based on the above MSOPS ranking method. The constraints were aggregated into a single value (typically a Euclidian norm or weighted min-max process) and then the constrained solutions ranked in order of degree of constraint violation, and rank penalised to ensure infeasible solutions are always ranked lowest. Additionally, to help to improve the discovery of the edges of the objective region, the population members that had an extreme objective value as part of their evaluation were removed from the ranking process and given the highest possible rank values to guarantee that they would be propagated into subsequent generations. Therefore the final rank was generated as (in priority order):

1) The valid population members that have extreme objective values in each dimension,
2) the valid population members ranked based on the target vector distribution,
3) the constrained solutions ranked by degree of constraint violation.

This ranking process functions well, but has a tendency

to over-penalise constrained solutions early in the search process.

A. MSOPS-II fitness assignment process

The fitness assignment process has been re-evaluated and simplified. The original MSOPS process required sorting down the columns of matrix S to create matrix R, then R is sorted across the rows, and then finally the rank-order is generated by a sorting of rows. Thus three independent sort operations were used. The first operation was creation of the matrix S as order $\mathcal{O}(pv)$ then the first sort is of order $\mathcal{O}(pv\log(v))$, where p is the number of elements in the population being sorted (usually the aggregation of the parent and offspring populations), and v is the number of target vectors. The second sort is $\mathcal{O}(vp\log(p))$ and the final sort is $\mathcal{O}(p\log(p))$. In practice, p will often be greater than v and therefore the process is dominated by $\mathcal{O}(vp\log(p))$.

We can define a many-objective problem as:

Find $\vec{X}$ that minimises $\vec{F}(\vec{X}) = [F_1(\vec{X}), \ldots, F_{N_f}(\vec{X})]$

subject to:

$$\vec{g}(\vec{X}) \leq 0 := [g_q(\vec{X}) \leq 0,\ \forall q \in [1, N_g]] \quad (1)$$

$$\vec{h}(\vec{X}) = 0 := [h_w(\vec{X}) = 0,\ \forall w \in [1, N_h]] \quad (2)$$

where $F()$ is one of N_f objective functions, $g()$ is one of N_g inequality constraints and $h()$ is one of N_h equality constraints. In the remainder of this paper (and also the accompanying software) it is assumed that all equality constraints have been converted to inequality constraints by defining $g(\vec{X}) = h(\vec{X}) - \epsilon$, where ϵ is a small tolerance value. A set $C \subseteq P := \{\exists g_q(\vec{X}_j) \not\leq 0 : \forall q \in [1, N_g], j \in [1, ||P||]\}$ can be defined which contains all solutions which violate at least one constraint, leading to the set $K \subseteq P$ as the set of valid solutions $K := P \setminus C$.

For the new approach, the aggregate fitness, r_i, of the i^{th} member of P is calculated using equation 3, where $f_n(\vec{F}_i, \vec{V}_n, \vec{Z}_n)$ is the aggregation function n with target vector $\vec{V}_n$ and reference point $\vec{Z}_n$ for objective vector $\vec{F}_i$.

$$r_{i \in P} = \min_{\forall n \in T} \left(\frac{f_n(\vec{F}_i, \vec{V}_n, \vec{Z}_n)}{\min_{\forall j \in K \neq i}(f_n(\vec{F}_j, \vec{V}_n, \vec{Z}_n))} \right) \quad (3)$$

Thus the fitness for a population member is the best metric value obtained, after scaling the results of each target vector, relative to the best performing valid solution on the target vector (not including the population member currently being evaluated). Equation 3 is calculated by:

1) Calculate metrics for each population member on each target vector to create matrix S,
2) for each column (target vector), find the minimum and second smallest metric value (from valid population members only),
3) scale each column by the minimum value found, except for the row which gave the minimum value: use the second lowest to scale this result,
4) for each row (population member), find the minimum scaled value to represent fitness, resulting in a column vector as the final aggregate fitness,
5) sort column vector to rank population.

The new fitness aggregation method still requires matrix S to be generated in $\mathcal{O}(pv)$ time. Finding the minimum and second lowest can be performed in almost linear time, hence the time complexity is approximately $\mathcal{O}(p)$. The second minimum process is linear in time and has a complexity $\mathcal{O}(v)$. The final sort of the population is $\mathcal{O}(p\log(p))$. The new process is therefore dominated by $\mathcal{O}(pv)$. In comparison to the original MSOPS ranking scheme, the time complexity was dominated by $\mathcal{O}(vp\log(p))$.

Two key aggregation methods have been employed within the MSOPS process previously: weighted min-max and Vector-Angle Distance Scaling (VADS). Empirical studies have shown that running the MSOPS algorithm with both Weighted Min-Max and VADS will provide superior optimisation performance than VADS or weighted min-max alone. As each column of the matrix S that holds the raw aggregated performance is normalised by a best-performing solution in the column, a compound S matrix may be formed by extending the rows by concatenating the results from the different metrics used within the system. Thus the ability to use multiple metrics simultaneously is preserved. The weighed min-max and VADS metrics are detailed as:

1) Weighted Min-Max: The weighted min-max score of N_f objectives is calculated using (4), where w_i is the weight of the i^{th} objective, F_i, and Z_i is the i^{th} component of a utopia reference point.

$$f = \max_{i=1}^{N_f}(w_i(F_i - Z_i)) \quad (4)$$

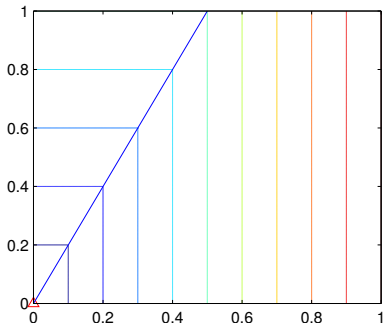

Fig. 3. Iso-fitness contours for the Weighted Min-Max aggregation method. Weights are $w_1 = 2$ and $w_2 = 1$, reference is $\vec{Z} = [0\ 0]$

Figure 3 shows the iso-fitness contours for the weighted min-max aggregation function with a reference point at the origin. An iso-fitness contour shows lines of constant fitness after the fitness assignment process. Thus the relative merit of different objective vectors locations can be established. Points which lie on the same iso-fitness contour will result in exactly the same fitness value after the fitness assignment process.

Weighted min-max iso-fitness lines form 'corners' and the method 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 $w_1F_1 = w_2F_2 = \ldots$, 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 $\vec{V} = [1/w_1,\ 1/w_2,\ \ldots]$.

Weighted Min-Max is sometimes also referred to as a Weighted Tchebychev Norm (spelling of Tchebychev may vary) and is a variant of the L^p norm method with $p = \infty$. The Weighted Min-Max used in this paper can alternatively be considered as a weighted L^∞ metric.

2) Vector Angle Distance Scaling (VADS): Vector Angle Distance Scaling (VADS) is a new metric first introduced in [4]. The metric is designed specifically for identifying the *Objective Front*, rather than just the Pareto front. The Objective front is the entire leading-edge of the feasible objective space region. The Pareto front is therefore a subset of the objective front. If the objective front is identified, then areas where 'gaps' appear in the Pareto set can be analysed: if there are objective front solutions that lie within the gap, then the break in the Pareto front is a discontinuity due to a very deep or reentrant concavity. If there are no objective front solutions in the region, then it is likely that the feasible objective region is comprised of disconnected sub-regions. In bi-objective problems, it is not difficult to identify regions of discontinuity in the Pareto front alone. However, even with 3 objectives, a discontinuity may present as a 'hole' and is not simple to identify without knowing the shape of the objective front too.

The VADS score is the magnitude of the vector of objectives ($|\vec{F}|$), divided by the cosine of the angle between the vector of objectives and a target vector, where the resulting angle cosine is then raised to a high power. Thus an objective vector that forms a point lying on the target vector is assigned a fitness which is the distance along the target vector. As the objective vector strays from the target vector, the fitness is increased rapidly with increasing offset angle.

The cosine of the angle can be calculated conveniently by a dot product operation. The score equation for N_f objectives is calculated using (5), where $\vec{V}$ is the N_f-dimensional unit-length target vector which describes the point on the objective front to search for, $\vec{F}$ is the N_f-dimensional objective vector, $\vec{Z}$ is a utopia reference point, $|\cdot|$ indicates vector magnitude and b is a constant factor for scaling the cosine result (typically $b = 100$).

$$f = \frac{|\vec{F} - \vec{Z}|}{\left(\vec{V} \cdot \frac{\vec{F}-\vec{Z}}{|\vec{F}-\vec{Z}|}\right)^b} \tag{5}$$

Low values for b may lead to difficulty in identifying very sharp concavities in the objective front. The dot product of the vector $\vec{V}$ with the shifted objective vector $\vec{F} - \vec{Z}$ must remain positive for the basic VADS metric to function correctly, and consequently objective offset $\vec{Z}$ may be adjusted automatically for proper operation.

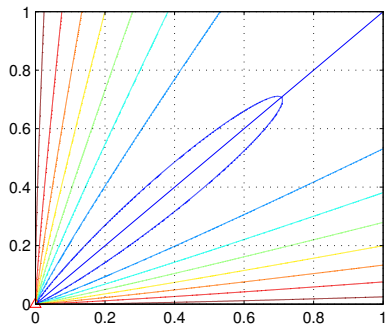

Fig. 4. Iso-fitness contours for the Vector-Angle Distance Scaling aggregation method. Weights are $\vec{V} = [1, 1]$, $b = 100$, reference is $\vec{Z} = [0\ 0]$. Logarithm of fitness plotted for clarity.

Figure 4 shows the iso-fitness contour for a weight vector of $\vec{V} = [1, 1]$, reference of $\vec{Z} = [0\ 0]$ and shaping parameter $b = 100$. The 'tear-drop' shaped iso-fitness contour is made thinner by increasing b, allowing sharper concavities to be probed. With very high values of b, care must be taken to prevent numerical instability. In the figure, the logarithm has been taken to reduce the dynamic range of the metric values experienced in the optimisation process. The use of logarithms allows (5) to be re-formulated as shown in (6) and reduces the impact of numerical imprecision.

$$f = \exp\left((b+1)\log(|\vec{F}|) - b\log\left(\vec{V} \cdot (\vec{F} - \vec{Z})\right)\right) \tag{6}$$

As the weight vector is changed, the iso-fitness contour follows the vector, rather than being aligned to the objective axes as in the weighted min-max function. The final solution identified by an optimiser using the VADS metric should have the objective vector $\vec{F}$ lying parallel to the target vector $\vec{V}$. Thus the angle between the two vectors can be used to assess final convergence. As VADS is tolerant of 'folds' in the objective surface that cause discontinuities in the Pareto front, angular errors between $\vec{V}$ and $\vec{F}$ indicate non-obtainable sections in the objective region.

III. Automatic Target Vector Generation

An ideal set of target vectors would provide even coverage of the feasible objective surface, given by the set $\mathcal{Z} \subset \mathbb{R}^{N_f}$. If we consider a utopia point that is defined as the minimum achieved objective value in each objective dimension, then one definition of an even coverage is a set of target vectors where for each vector, the angle to the nearest neighbour vector, when projected from the utopia point, is maximised. This definition using angles between vectors is satisfactory with even high-dimensional objective spaces. If Euclidean distances were considered, then they would need to be distances as measured across the great-arc of the hypersphere in order to achieve a uniform distribution. As the dot product of two unit-length vectors provides the cosine of the angle between them, angle is preferred as it is very simple to calculate.

Once the extreme limits of the feasible objective region are identified, giving limits to the set $\mathcal{Z}$, a uniform spread of

N vectors could be generated by the minimisation problem:

Find the set of vectors $\mathcal{Q} = \{\vec{V}_1, \ldots, \vec{V}_N\}$ such that

$$z = \max_{i=1}^{N} \max_{j=1, j\neq i}^{N} (\vec{V}_i \cdot \vec{V}_j) \qquad (7)$$

is minimised, subject to $\mathcal{Q} \subset \mathcal{Z}$

The inner maximum operator finds the nearest neighbour in angle (dot product provides cosine of angle and $\cos(0) = 1$, therefore maximising finds smallest angle), the outer maximisation finds the smallest of the nearest neighbour angles in order to give the metric z. The set of vectors Q is then optimised in order to minimise the metric z, i.e. maximise the worst-case angle to a nearest neighbour.

Unfortunately (7) is not often simple to optimise. However if we form an archive, the set $\mathcal{A} \subset \mathbb{R}^{N_f}$, of all the best solutions found to date, an approximation can be formed by 'boot-strapping' the problem by setting $\mathcal{Z} = \mathcal{A}$, so that the archive of the population is used to supply a finite set of vectors to search over. Unfortunately even finding an optimum set out of this restricted search space is not quick to process.

A further approximation is to use the current population to provide the source for the new target vectors. The procedure is sub-optimal, but effective in practice due to the semi-random nature of the EA re-visiting areas regularly. The method is to take the current target vector set and augment it with each member of the population in turn (with the objectives offset by the reference point, scaled and then normalised to create a valid search direction vector). Once augmented by a population member, all the angles to the nearest neighbours are calculated, and the solutions with the nearest neighbours are identified. By definition, there will always be at least two solutions with the same nearest neighbour as pairwise comparisons have been made, so the second nearest neighbour etc. must be considered in turn for the pair until the most crowded is identified and can be removed, restoring the target vector set back to its original size. The next population member is then used to augment the new set and the process repeated. The process requires $\mathcal{O}(p(v+1)^2)$ calculations per generation, but I hypothesise that the complexity could be reduced through the correct choice of data structure and re-use of prior pairwise calculations.

By only harvesting target vectors from the current populations, there is minimal opportunity for target vectors outside of the Pareto region to be generated, improving algorithm efficiency. However, the exploration of the edges of the objective region is not as strong when compared to a set of vectors generated *a-priori*. Figure 5 shows the final target vector set from a typical two-objective optimisation run. It is quite apparent that the spread of target vectors is sufficiently uniform and covers the objective surface well.

Importantly, the maximum range of the objective values that are encountered should be tracked in order to scale the objectives sensibly as the dot-product between vectors is influenced by objective scaling. There may be some extreme

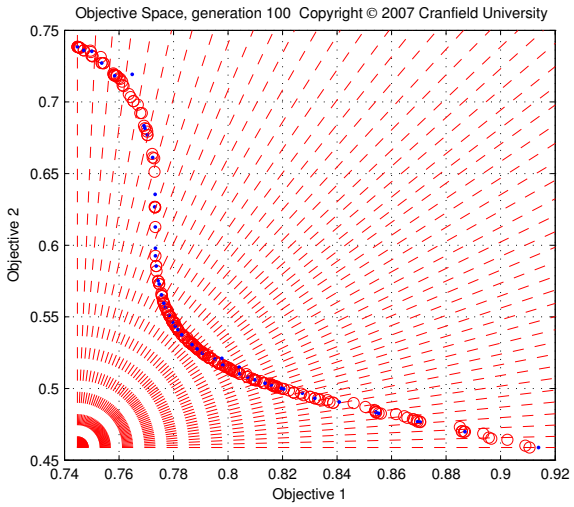

Fig. 5. Example final set of 50 target vectors generated automatically showing sufficiently uniform spreading in angle across the objective surface. Population of 50 for 100 generations. Automatically generated utopia point = [0.75 0.46], automatic objective scaling = [0.16 0.28]

functions (for example if objective values can go to infinity) that may still require manual scaling. Additionally, the minimum objective values in each dimension ever observed in a valid solution should also be recorded in order to generate a utopia point for a reference for the target vectors. In the early generations of the algorithm, the objective scaling and utopia point are updated often, however towards the end of the optimisation process they are more stable. The early instability will reduce the efficiency of the algorithm slightly, but the effect appears to be minor in practice.

IV. Experimental Design and Results

A. Hypotheses

Four hypotheses must be tested:

1) The difference in behaviour between the original ranking method and the new analytic fitness assignment process is insignificant;
2) With fixed vector sets, performance is reduced when many target vectors do not intercept the feasible objective region;
3) The automatic target vector generation method is superior to a fixed target vector set that has many target vectors which do not intercept the feasible objective space;
4) Choosing one parent from an elite archive as part of the crossover operator is superior to crossover just within the current population.

B. Evolutionary Algorithm Design

For demonstration purposes, a variant of a real-valued Evolutionary Programme that uses adaptive mutation and restricted crossover from an archive has been used. The MSOPS-II process does not specify the offspring generation/ selection method as all operators should be designed to suit the chromosome structure/ problem being optimised.

The offspring generation process performs the following steps for each member of the population to generate one offspring:

1) A tournament selection between $N_T = 5$ members of the archive to select a solution which is closest to the current population member. Euclidian distance is calculated, with the tournament being performed either in objective or decision space with 50% probability.
2) Intermediate crossover between the chosen archive solution and the current population member is used to create a proposed offspring: $X'_i = \alpha_i X_i + (1-\alpha_i) X_{Ai} : \forall i$, where X'_i is a gene of the proposed offspring, X_i is a gene of the current population member, X_{Ai} is a gene of the archive member and α_i is a uniform random number in the range [-0.25, 1.25]. Uniform crossover between the current population member and the proposed offspring is performed using a probability of swapping genes of 50% per gene loci.
3) The standard deviation of the mutation of the population member is mutated by $\sigma = 0.977\sigma \exp(U/2)$, where U is a uniform random number in the range [0,1].
4) Gaussian mutation is added to each gene with a probability of 90%, where the mutation standard deviation is given for each gene that is mutated by $X''_i = X'_i + N(0, \sigma^2)$.
5) The genes in the chromosome are tested against the upper/lower bound of the gene. Any genes that are found to violate a bound are set to a random value chosen uniformly in the range between the gene limit and the gene value of the parent solution (from the population not the archive). This mechanism is important for functions where the Pareto set lies close to the gene limits in decision space: simply cropping the gene value at the limit does not work as well.

For the experiments where an archive is not used, where an archive set is mentioned above, it is replaced by the current population. A population of 50 for 100 generations and 50 target vectors were used in all trials. The tests using the original MSOPS ranking method set all constrained solutions to be inferior in the ranking to the valid solutions. For the analytic fitness assignment method, a variant of stochastic ranking [6] with $p_f = 0.4$ was used. It should also be noted that the crossover and mutation are quite simple and may not perform well on rotated functions.

The classic stochastic ranking process performs a comparison on objectives if both are feasible, or if a uniform random number is less than the factor p_f; and compares on constraint violation otherwise. It was found that as the Pareto front is defined by a constraint boundary, constrained solutions that were between the Pareto front and the origin were often forced to move **away** from the Pareto front, rather than towards it as the objective values improved when moved towards the origin. Thus a modification was made so that if it was decided to test on objective fitness, but one or both of the solutions were constrained, the MSOPS assigned fitness used for ranking was also tested. As the MSOPS fitness is normalised only by valid solutions, any fitness assignment less than unity for a constrained solution must be forward (in a minimisation sense) of the current non-dominated front and could be at risk of being beyond the Pareto front. In these cases where the assigned fitness was less than unity for one or both of the constrained solutions, then the comparison was performed in constraint space, rather than the intended objective space. When the test was performed in constraint space, the constraint aggregation method was to take the Euclidean norm (L^2 norm) of the constraint violation vector.

C. Test Objective Functions

For simplicity, the hypotheses were evaluated on two-objective and 5 objective functions that are both constrained highly. The 2D function is based on the Tanaka [7] objective function, but with two additional constraints to further compress the feasible region. The objective function is defined in equation 8 and is demonstrated graphically through a random sample of the total objective space (objective space and decision space are the same) in figure 6. The 5D function in (9) is formed by constraint boundaries of a hypersphere and a 15 degree cone.

$$\begin{aligned} F_1 &= x, \quad F_2 = y \\ 0 &\geq -(x)^2 - (y)^2 + 1 + 0.1\cos\left(16\arctan\left(\frac{x}{y}\right)\right) \\ 0.5 &\geq (x-0.5)^2 + (y-0.5)^2 \\ 0 &\leq x - y, \quad 0 \geq x - 2y, \quad 0 \leq x, y \leq \pi \end{aligned} \tag{8}$$

$$\begin{aligned} F_i &= x_i, \quad 0 \leq x_i \leq 1 \\ q &= \sum_{i=1}^{5} x_i^2, \quad q \geq 1, \quad \cos(15^\circ) \leq \sqrt{\frac{5}{q}} \sum_{i=1}^{5} x_i \end{aligned} \tag{9}$$

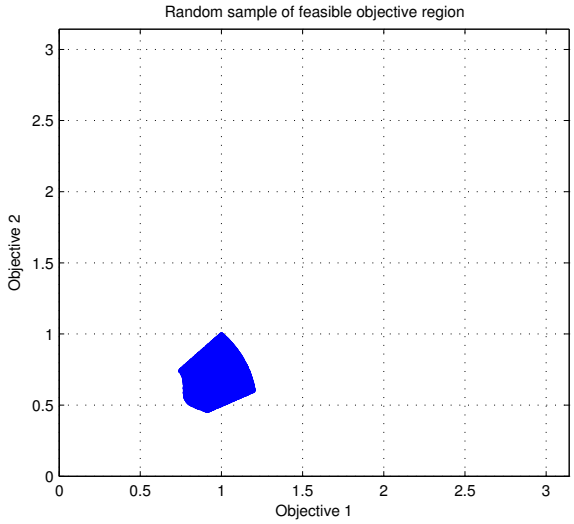

Fig. 6. Random sample of feasible objective region showing highly constrained nature of 2D test function

D. Experimental Design

To enable statistical testing of the hypotheses, 100 repeated trials of each algorithm configuration were performed. In total 9 algorithm variants were evaluated:

1) Original MSOPS ranking, full set of target vectors (+);
2) New MSOPS ranking, full set of target vectors (○);
3) Original ranking, automatic target vectors (△);
4) New ranking, automatic target vectors (♢);
5) Original ranking, restricted set of target vectors (□);
6) New ranking, restricted set of target vectors (⋆);
7) Random search of 5000 evaluations (equivalent pop of 50, 100 generations) (·)
8) Original ranking, full set of target vectors but no Archive in crossover;
9) Original ranking, restricted set of target vectors but no Archive in crossover.

The full set of vectors was spread over the total possible objective region and suffers from the problem of many target vectors not playing a significant role in the rank assignment process (as shown in figure 2 for 2D problem). The restricted set is targeted to just cover the feasible region of the objective space (as shown in figure 7). Both vector sets use the origin as the reference location. The experiments with automatic target vector generation also generate the vector reference point automatically (as shown in figure 5).

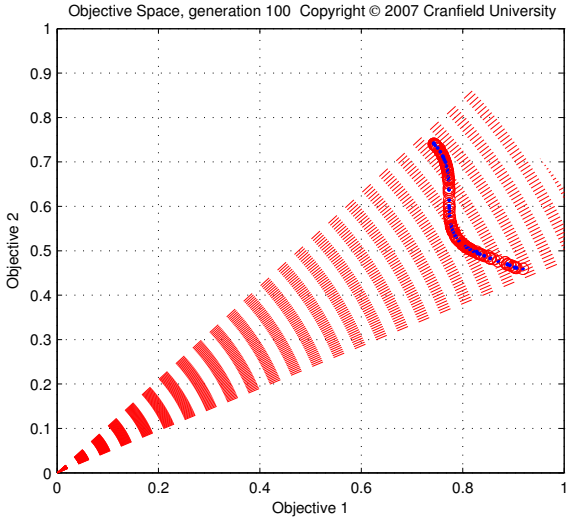

Fig. 7. 50 Target vectors confined in a fan between vectors [2.1 1] and [1 1.05]. The dashed lines show the direction of the target vectors, circles represent the archive of non-dominated points, the dots are the locations of the population members on the final generation.

E. Results and Discussion

1) Hypothesis 1: The hypothesis "The difference in behaviour between the original ranking method and the new analytic fitness assignment process is insignificant" was tested using the first 6 of the experimental algorithms to test the hypothesis under the different possible algorithms conditions. Figure 8 shows the results of hypervolume analysis (larger is better) for the 2D problem, where the hypervolume of the 100 trials have been sorted and plotted as a cumulative distribution. It is clear that most of the distributions are skewed and therefore the mean and variance of the results will be unreliable with only 100 evaluations. The median performance (which corresponds to the 50^{th} point on the horizontal axis) is the best general indicator.

A Wilcoxon rank-sum test was performed between all the pairs and the null hypothesis that the distributions are the same could not be accepted at a two-tailed test level of $\alpha = 0.01$. A Kruskall-Wallis test of the two triplets also indicates that the null hypothesis that all three are the same cannot be accepted in either case. However, the location of the reference point is critical to the generation of the plots and the order of algorithm performance changes as the reference is moved away from the origin. It was decided that the hypervolume measure performance is unreliable in this instance and is not a good general indicator. Investigations indicate that the problem lies with the fixed target vector experiments being very good at finding the edges of the feasible region and therefore gaining a high hypervolume score, but not necessarily being as good in general at approaching the true Pareto front.

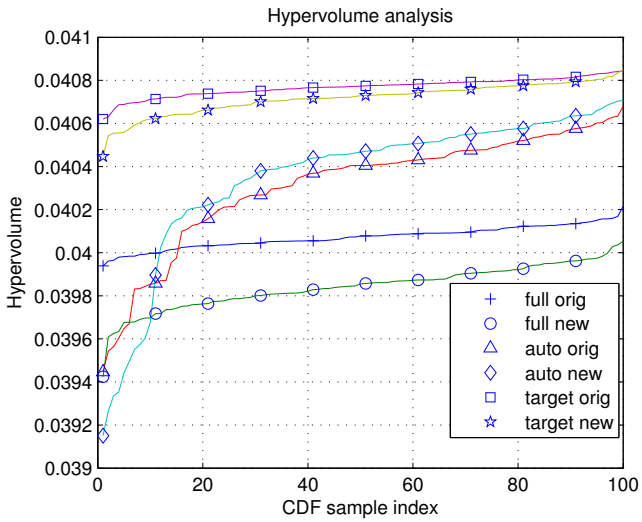

Fig. 8. CDF Hypervolume results for first 6 experimental algorithms. Reference point is set as nadir of all 600 non-dominated sets and lies at [0.9242 0.7554]. Larger hypervolume is superior

An alternative metric was sought that would provide reliable results. The median Multi-Objective Equivalent Random Search (MOERS) [8] was chosen as, like hypervolume, it can capture the effects of spread, distribution and distance to Pareto-front. In practice, the results have been repeatable and are therefore considered more reliable. Figure 9 shows the results of the median MOERS analysis for 200 test vectors chosen at random from the non-dominated surface of a 10^8 point random search of the 2D objective function.

The MOERS results are shown on a logarithmic scale, relative to the true 5000 point search used by all of the algorithms. As confirmation of correct operation of the metric, the random search results appear on the graph at a

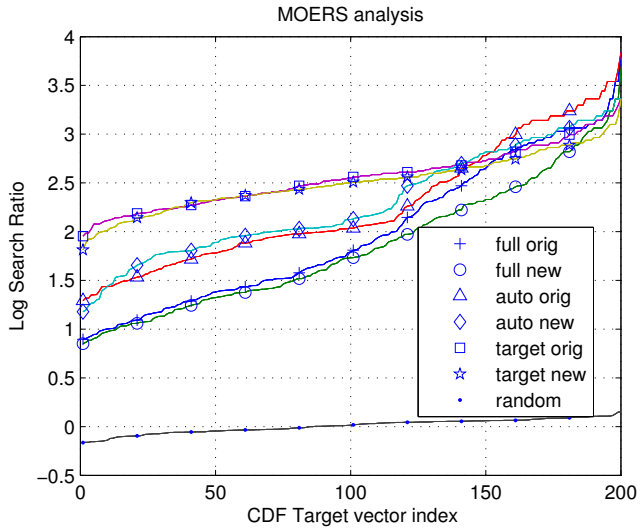

Fig. 9. CDF median MOERS results for first 7 experimental algorithms. Graph shows Logarithm of the equivalent search ratio, i.e. referenced directly to the median random search performance.

unity performance ratio, therefore being plotted around zero on the logarithmic scale. All of the other algorithms tested show a median performance of over 30 times better than the random search.

A pairwise ranksum analysis of the median MOERS results for the 2 objective problem show that the Null Hypothesis that the results are from the same distribution cannot be rejected for: 1:2 $\rho = 0.0867$; 3:4 $\rho = 0.3531$; 5:6 $\rho = 0.1849$, where ρ is the Wilcoxon test statistic. All other pairs have $\rho \ll \alpha = 0.01$ and therefore we reject the null hypothesis that the distribution are the same. Thus from the median MOERS results, hypothesis 1 that the old and new ranking methods have similar performance cannot be rejected. For the 5 objective problem, similar results were obtained.

2) Hypothesis 2: The hypothesis "With fixed vector sets, performance is reduced when many target vectors do not intercept the feasible objective region" is confirmed by the Wilcoxon rank-sum test between pairs 1:5 and 2:6 returning $\rho \ll 0.01$, showing that under conditions where less than one quarter of the target vectors pass through the feasible region (as in figure 2), the null hypothesis of the distributions being the same as for when the target vectors are matched closely to the objective region (figure 7), must be rejected. Again a similar result was observed with the 5 objective problem. The MOERS results confirm that the set of vectors designed to target the objective region yields superior performance.

3) Hypothesis 3: The hypothesis "Automatic target vector generation is superior to a fixed target vector set that has many target vectors which do not intercept the feasible objective space" is confirmed by the Wilcoxon rank-sum test between pairs 1:3 and 2:4 rejecting the null hypothesis at $\rho < 0.01$ and the median MOERS results for the automatic target vector generation process being superior. However, it must also be noted that the test between pairs 3:5 and 4:6 also rejects the null hypothesis and the automatic vector generation results are inferior to the fully targeted vector set. Thus we can conclude that although automatic target vector generation is useful for exploring unknown problems, if the target vectors **can** be generated *a-priori*, then they should. For the 5 objective problem, the performance of the automatic target generation could not be distinguished statistically from the targeted set (targeted formed a 17 degree cone encompassing the actual 15 degree cone of the constrained region).

4) Hypothesis 4: The hypothesis "Choosing one parent from an elite archive as part of the crossover operator is superior to crossover just within the current population" was tested by comparing algorithm pairs 1:8 and 5:9. In both cases the null hypothesis that the performance is the same could not be rejected with values of $\rho = 0.60$ and $\rho = 0.72$ respectively. Again similar results were observed for the 5 objective problem. Therefore the use of the archive over choosing a second parent from the working population made no significant difference.

V. Conclusions

This paper has presented significant modifications to the established MSOPS optimiser that allow a general-purpose many-objective optimisation algorithm to be developed. The functionality of the algorithm modifications have been tested for statistical significance, allowing an informed algorithm design to suit the degree of *a-priori* knowledge of the optimisation problem.

Prototype algorithm software that will re-produce all of the results in this paper is available for academic use at [9].

References

[1] R. C. Purshouse, "Evolutionary many-objective optimisation: An exploratory analysis," in *The 2003 Congress on Evolutionary Computation (CEC 2003)*, vol. 3. Canberra, Australia: IEEE, 8–12 December 2003, pp. 2066–2073.

[2] E. J. Hughes, "Evolutionary many-objective optimisation: Many once or one many?" in *IEEE Congress on Evolutionary Computation, 2005*, vol. 1. IEEE, Sep. 2005, pp. 222–227.

[3] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, Aggregation-, and Indicator-based methods in many-objective optimization," in *Evolutionary Multi-Criterion Optimization, EMO 2007*. Matsushima, Japan: Springer LNCS 4403, 2007, pp. 742–756.

[4] E. J. Hughes, "Multiple single objective pareto sampling," in *Congress on Evolutionary Computation 2003*. Canberra, Australia: IEEE, Dec. 2003, pp. 2678–2684.

[5] B. Naujoks, N. Beume, and M. Emmerich, "An EMO algorithm using the hypervolume measure as selection criterion." in *Evolutionary Multi-Criterion Optimization, EMO 2005*. Guanajuato, Mexico: Springer LNCS 3410, Mar. 2005, pp. 62–76.

[6] T. P. R. adn Xin Yao, "Stochastic ranking for constrained evolutionary optimisation," in *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3. IEEE, Sep. 2000, pp. 284–294.

[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*, vol. 2. IEEE, 22-25 October 1995, pp. 1556–1561.

[8] E. J. Hughes, "Multi-objective equivalent random search," in *Parallel Problem Solving from Nature - PPSN IX*. Reykjavik, Iceland: Springer LNCS 4913/2006, Sep. 2006, pp. 463–472.

[9] [Online]. Available: http://code.evanhughes.org