Multiple Single Objective Pareto Sampling

Evan J. Hughes
Department of Aerospace, Power and Sensors,
Cranfield University,
Royal Military College of Science,
Shrivenham, Swindon, England. SN6 8LA
ejhughes@iee.org

Abstract- This paper details a new non-Pareto evolutionary multi-objective algorithm, Multiple Single Objective Pareto Sampling (MSOPS), that performs a parallel search of multiple conventional target vector based optimisations, e.g. weighted min-max.

The method can be used to generate the Pareto set and analyse problems with large numbers of objectives. The method allows bounds and discontinuities of the Pareto set to be identified and the shape of the surface to be analysed, despite not being able to visualise the surface easily.

A new combination metric is also introduced that allows the shape of the objective surface that gives rise to discontinuities in the Pareto surface to be analysed easily.

1 Introduction

Many methods have been developed that exploit the population of an evolutionary algorithm in order to generate an entire Pareto set of a multi-objective problem during a single run.

The methods can be split broadly into non-Pareto and Pareto based methods. The non-Pareto based methods (VEGA, etc. (Deb, 2001)) generate a Pareto set implicitly, without making a direct comparison to check domination/non-domination with other members of the population. Pareto based methods (MOGA, NSGA, SPEA, etc. (Deb, 2001) and bounding methods (Schutze *et al.*, 2003)) rely on ranking the population based on a direct measure of Pareto dominance within the population.

Traditional optimisation approaches often use aggregation or target vector approaches (Deb, 2001), but only allow a single solution to be found. Successful methods (Jin *et al.*, 2001) have exploited the aggregation and target vector approaches, changing the weight or target vector of the search during evolution in order to develop a full Pareto set during the run.

Unfortunately, most of the methods (especially Pareto based approaches) have difficulty in generating Pareto sets, or even analysing small regions of the Pareto set, of problems that have large numbers of objectives. In Pareto based methods, the bulk of the population becomes non-dominated very rapidly with large numbers of objectives and the selective pressure is reduced, leading to stagnation of the algorithm.

Many practical Pareto sets are also discontinuous and may have large 'gaps'. The general shape of 'interesting regions' on the Pareto front are also difficult to visualise and analyse with high numbers of objectives.

This paper presents a new non-Pareto based method that is capable of generating efficiently Pareto sets for problems with many objectives. The method allows all traditional aggregation and target vector approaches to be incorporated into an evolutionary algorithm which generates a Pareto front in a single run.

2 Method

2.1 Population Ranking

The concept is to generate a set of T target vectors (or goals, aggregation weights etc., depending on the methods to be applied), and evaluate the performance of every individual in the population, of size P, for every target vector, based on the chosen aggregation/target vector method(s). As target vector (eg. weighted min-max, ϵ-constraint, goal attainment etc.) and aggregation (eg. simple weighted sum, L^p norm etc.) methods 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 (each column containing P entries) 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.

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.

0-7803-7804-0 /03/$17.00 © 2003 IEEE

2.2 Target Vector Methods

The approach has been applied to a range of aggregation and target vector methods including weighted sum, ϵ-constraint, goal attainment and L^p norm. Some of the methods (eg. ϵ-constraint & goal attainment) rely on turning some or all of the objectives into constraints. The target vector then becomes a vector describing the locations of the constratints, but an EA that is capable of handling constraints effectively is required to maximise the performance of the approaches. The following two approaches have found to be satisfactory for many multi-objective problems.

2.2.1 Weighted Min-Max

The weighted min-max score of k objectives is calculated using (1),where w_i is the weight of the i^{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 $w_1 O_1 = w_2 O2 = \ldots$, 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.

2.2.2 Vector Angle Distance Scaling (VADS)

This paper introduces a new metric, Vector Angle Distance Scaling (VADS). The VADS score is the magnitude of the vector of objectives ($|O|$), divided by the cosine of the angle between the vector of objectives and a target vector, which has been raised to a high power. The cosine of the angle can be calculated conveniently by a dot product operation. The score equation for k objectives is calculated using (2), where $\hat{V}$ is the k-dimensional unit-length target vector which describes the point on the Pareto front to search for, O is the k-dimensional objective vector and q is a constant factor (typically 100).

$$s = \frac{|O|}{\left(\hat{V} \cdot \frac{O}{|O|}\right)^q} \tag{2}$$

VADS is able to generate points on both convex and concave Pareto sets, and has an interesting property that it does not just identify the Pareto front, but rather the *objective front*, i.e. the leading edge of the objective space. Thus highly concave regions that may appear as discontinuities in the Pareto front can be identified. Low values for q may lead to difficulty in identifying very sharp concavities in the objective front.

The final solution should have the objective vector O lying parallel to the target vector 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 V and O indicate unobtainable regions or dislocations in the objective surface.

2.2.3 Dual Optimisation

If the same set of target vectors is used to generate two R matrices, one based on scores calculated with the weighted min-max, and the other with VADS, the rows of the two matrices can be concatenated (giving a matrix of size $P \times 2T$) and sorted, allowing each population member to be assessed by each weight, and for both scoring methods. Thus the focus will be more biased towards the Pareto set as often a solution will score highly for both methods, but the leading edge of the objective surface will also be retained.

2.3 Evolutionary Algorithm

This paper describes a new method for ranking a population, essentially based on a set of single-objective tests; thus the method may be applied to a wide range of EA approaches. The method has been tested with traditional genetic algorithms, evolutionary programmes and evolutionary strategies and works well. For the results presented in the paper, a method based on Differential Evolution has been used.

Differential Evolution (Storn and Price, 1995) 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 (3).

$$\vec{P}_t = F(\vec{P}_a - \vec{P}_b) + \vec{P}_c \tag{3}$$

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 (4).

$$P(L \geq v) = (C)^{v-1}, v > 0 \quad (4)$$

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 priority is applied in the initial stage where the score matrix S is ranked to generate the matrix R. 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 population size should ideally be larger than the number of target vectors used. Typically the population should be one and a half or twice the size as the target vector set, but as with many EA approaches, the larger the population is, the better the initial search is and the less likely the algorithm is to be deceived by local optima. The processing speed of the algorithm is a trade between the population size and also the number of target vectors. In general the processing time of the algorithm is approximately $O(MTP\log(P))$ if $P > T$, or $O(MPT\log(T))$ if $P < T$ where M is the number of objectives, T is the number of target vectors and P is the population size. The behaviour is mainly influenced by the ranking of the population members for each target vector using an $O(P\log(P))$ sort algorithm.

If the population is smaller than the number of target vectors, the population will naturally distribute itself among the vector set. If the best value obtained for each target vector is recorded, even with very small population sizes, a very reasonable approximation of the Pareto set and objective front can be obtained.

As a consequence of generating the rank matrix R, it is simple to implement niche formation by applying restricted breeding, based on proximity in objective space. In the Differential Evolution selection phase, three vectors must be chosen from the population to determine the new trial chromosome. Each row of the matrix R holds a set of target vector rankings for the population member corresponding to the row. The target vector that the corresponding objective values are closest to can therefore be identified. The rankings in this column can then be used to identify other population members that also perform well on this target vector. By definition, they must be close in objective space to the current population member of interest and can therefore be used to form a restricted breeding pool. For all the examples in this paper, the closest half of the population was used to form basic restricted breeding.

3 Example Application

To demonstrate the behaviour of the ranking method and approach to analysis of the Pareto surface, two multi-objective test functions will be studied (Van Veldhuizen and Lamont, 1998).

$$\begin{aligned} F1: & \\ f_1 &= x \\ f_2 &= y \\ 0 &\geq -(x)^2 - (y)^2 + 1 + 0.1\cos\left(16\arctan\left(\frac{x}{y}\right)\right) \\ 0.5 &\geq (x-0.5)^2 + (y-0.5)^2 \\ 0 &\leq x, y \leq 1 \end{aligned} \quad (5)$$

$$\begin{aligned} F2: & \\ p &= 6x - 3 \\ q &= 6y - 3 \\ f_1 &= 0.5(p^2+q^2) + \sin(p^2+q^2) \\ f_2 &= \frac{(3p-2q+4)^2}{8} + \frac{(p-q+1)^2}{27} + 15 \\ f_3 &= \frac{1}{p^2+q^2+1} - 1.1\exp(-p^2-q^2) + 0.1 \\ 0 &\leq x, y \leq 1 \end{aligned} \quad (6)$$

The functions are (Van Veldhuizen and Lamont, 1998) Tanaka and Viennet(3) respectively and have been scaled so in the decision space $0 \leq x, y \leq 1$, and also offset to make all the objective values greater than zero.

3.1 Two Objective Example

Figures 1 shows the result of the combined weighted min-max and VADS search applied to (5) with 21 target vectors (shown as dashed) and a population of 30 for 50 generations. The dots represent the distribution of the final population, the stars indicate the best set of solutions found with the weighted min-max and the circles are the best VADS solutions.

It is clear that points on the boundary of the objective surface have been identified. The 'leading edge' of the objective space is identified by VADS, while the min-max finds the Pareto set. This feature is very useful for analysing the behaviour of the objectives, rather than just the Pareto set. The area around [0.1 0.3] is a discontinuity in the Pareto set and as such has only been identified in the VADS search. The corresponding vectors of angular errors for VADS and min-max respectively are shown in figures 2 & 3, sorted according to the weights with v_1 (the first element of the target vectors) increasing. It is clear that many of the search vectors were satisfied with an error less than 2° away from their target vector for VADS; but there are areas with high errors for weighted min-max, indicating that some vectors could not be satisfied. These errors correspond to the limits of the Pareto set in the VADS plot (first two and last target

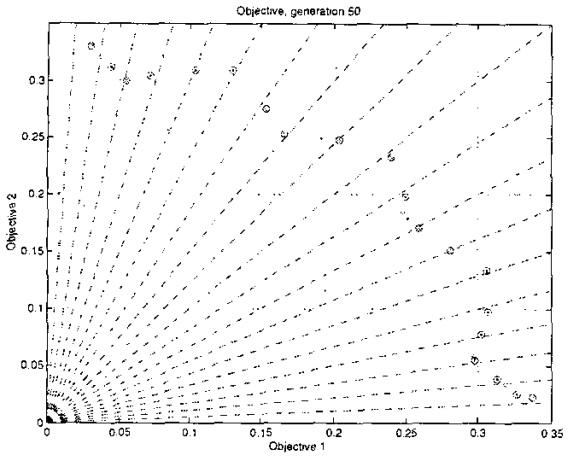

Figure 1: Plot of objective/decision space for function (5)

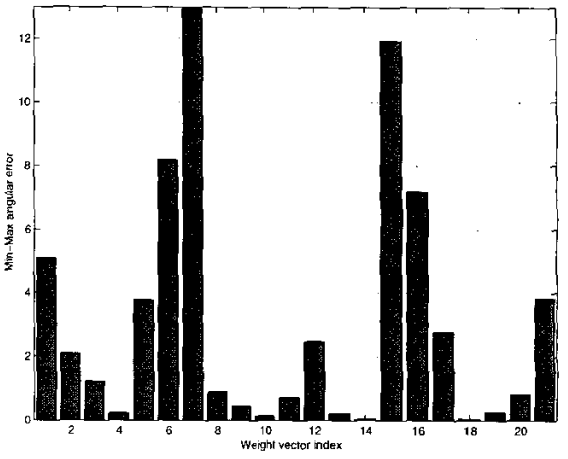

Figure 3: Plot of angular errors to weight vectors for weighted min-max and function (5)

vectors) and also to the discontinuities in the function in the weighted min-max plot (around vectors 6 and 16).

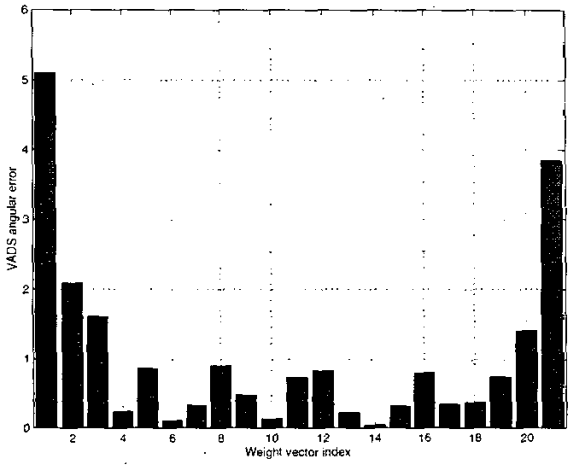

Figure 2: Plot of angular errors to weight vectors for VADS and function (5)

This example demonstrates that because we know *a-priori* the regions of the Pareto set that are being investigated, based on the set of target vectors, we can quantify how close the optimisation result came. It is also clear that as it is the number and quality of target vectors that are satisfied that determines the score of a population member, the number of objectives has little effect on the performance of the algorithm, unlike in Pareto based methods. With large numbers of objectives though, large numbers of target vectors may be required if a detailed search is to be performed across the entire objective space in one pass. It is simple though with the MSOPS method to target a range of smaller areas with each run. The areas can be of varying size and diverse in each run if necessary, providing extreme flexibility in the optimisation process. Strategies may be used to yield extra information about the Pareto surface such as generating a set of target vectors that lie on a plane, allowing 'slices' through the Pareto set to be visualised to test for continuity. If the sum of the objectives found along the slice is plotted, it is simple to analyse the degree of convexity in interesting regions of functions with large numbers of objectives that cannot be visualised directly. Figure 4 shows a plot of the sum of the closest objectives to each target vector for F1. The rise in the middle of the plot indicates that the function is in general concave.

3.2 Three Objective Example

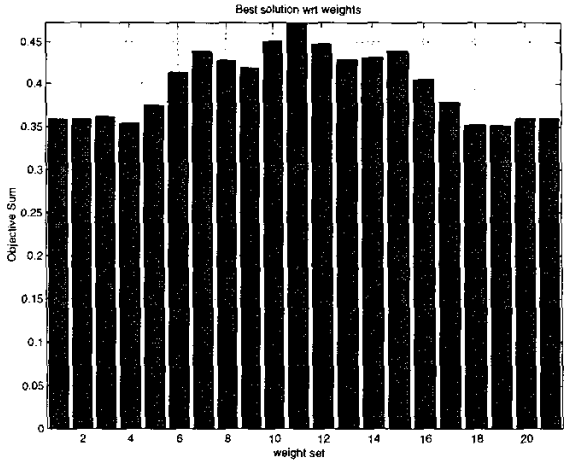

Figure 4: Plot of sum of objectives for function (5)

In the two objective case, it is simple to visualise the objective surface. With three objectives, visualisation is more difficult and it becomes almost impossible with four or more.

Figure 5 shows the decision space result of applying both VADS and weighted min-max simultaneously to F2 with

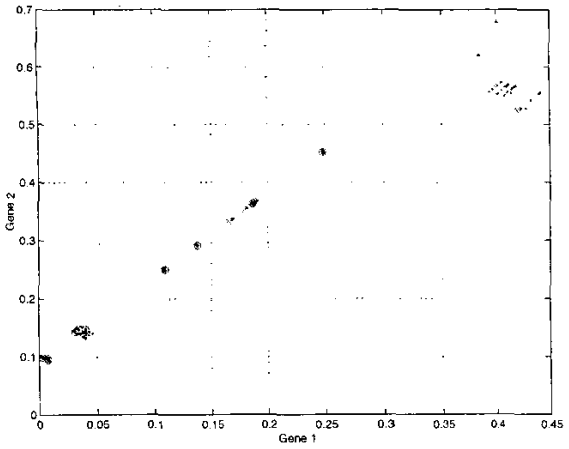

Figure 5: Decision space for 3 objective function (6)

101 randomly generated target vectors, a population of 150 and 50 generations.

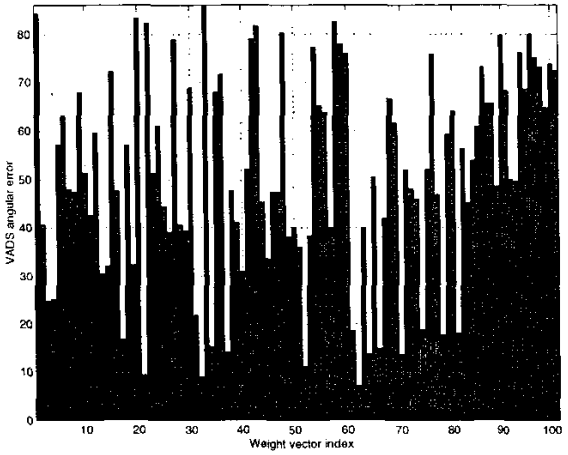

Figure 6: Plot of angular errors to weight vectors for VADS and function (6), sorted by v_1

Figures 6 & 7 show the angular error for VADS and weighted min-max, sorted based on v_1. It is clear that most areas could not be satisfied accurately by either metric and therefore correspond to regions outside of the objective space, while those that are covered by VADS only are likely to be discontinuities, or very flat areas in the Pareto surface. There may also be occasions where the weighted min-max only finds a solution. These areas correspond to regions of very high concavity in the Pareto set.

Figures 8 & 9 show the same VADS data as in figure 6, but sorted with respect to v_2 and v_3 respectively. There is clearly a strong correlation between the angular errors and v_2 and v_3. As we are only interested in the region where the objective surface exists, we can reduce the range of each of the elements of the target vector, based on the VADS data. We can see from figure 8 that there is no need to examine the area covered by the vectors below vector 65. Vector 65 in

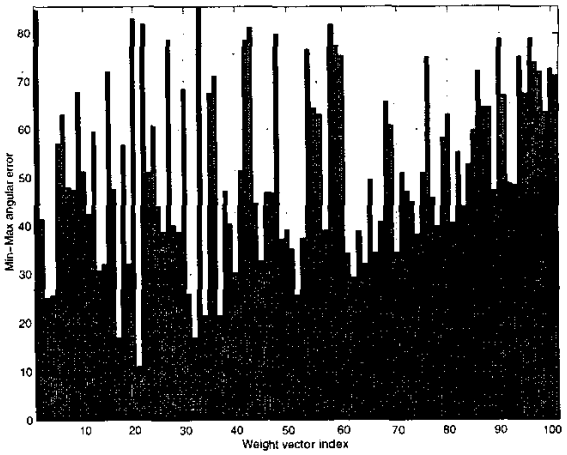

Figure 7: Plot of angular errors to weight vectors for weighted min-max and function (6), sorted by v_1

the plot has a v_2 component of 0.66, and therefore we only need to generate target vectors with a range of [0.66,1] for the elements of v_2. The other elements of the target vector may be constrained accordingly, refining the search to the feasible areas of the objective space.

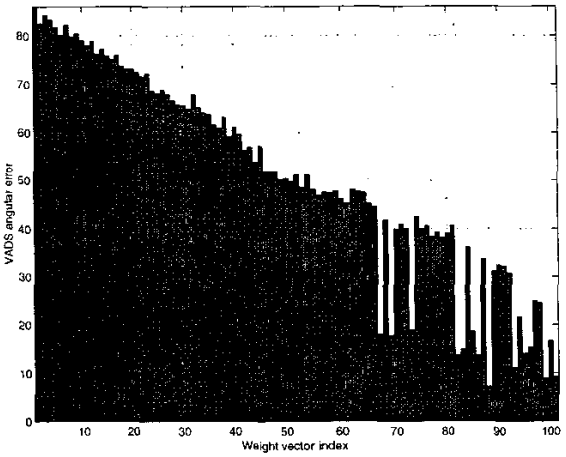

Figure 8: Plot of angular errors to weight vectors for VADS and function (6), sorted by v_2

After two refinement steps, the VADS error, sorted by v_3 is shown in figure 10, and the resulting decision space in figure 11. The unity length target vectors are constrained to the regions [0,0.49], [0.87 1] and [0 0.058] for v_1 to v_3 respectively. It is clear that v_3 operates in only a very small region and has a dominating effect on the valid region of the objective surface. Although Euclidean distance is used in the generation of the set of results, the ability to detect the edges of the objective surface and refine the search, even with many objectives, allows the relative scaling and offset of the different objectives to be analysed. In this example, the minimum of the three objectives over the final set of weight vectors is [0.003 15.0 0.0002], indicating an offset

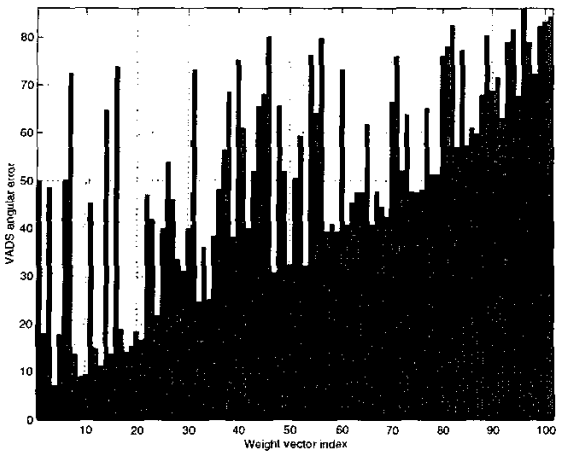

Figure 9: Plot of angular errors to weight vectors for VADS and function (6), sorted by v_3

on the second objective, and the range of the objectives is [8.1 1.9 0.3], indicating a large range of relative scaling. Figure 12 shows the sum of the three objectives, sorted by v_1. The figure shows a clear relationship between v_1 and the shape of the surface. With such extreme scaling of the objectives, it is difficult to infer the general shape of the objective surface, but the depression in vectors $1-30$ indicates that the region is generally convex, and the steady increase with v_1 increasing indicates the objectives are strongly correlated to the behaviour of objective 1.

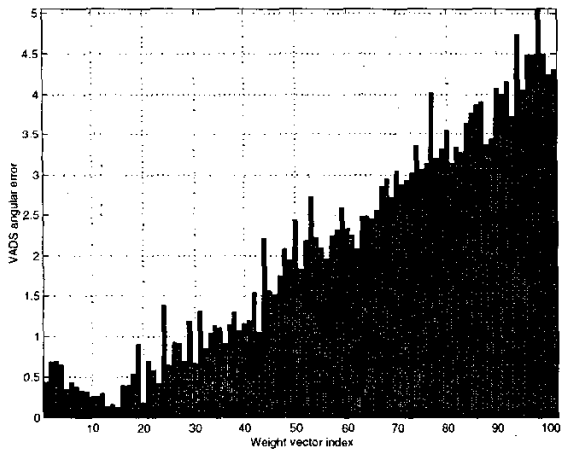

Figure 10: Plot of angular errors to weight vectors for VADS and function (6), sorted by v_3

The final decision space shows that the Pareto set of the function is discontinuous in decision space, and forms a sharp line of points. There is also a region that is detected by weighted min-max only, indicating a very sharp concavity. The weighted min-max points scattered amongst the VADS points on the long straight section of the result also indicates that the Pareto set may lie almost along an axis of the objective space. The final VADS plot of figure 10 also indicates that the spread of errors is quite high, except for a small region, suggesting that the vectors were not quite matched exactly and that the Pareto surface is very narrow, probably forming a line.

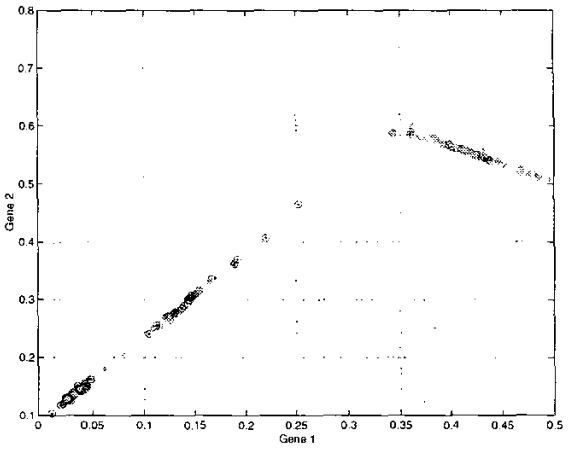

Figure 11: Final Decision space for 3 objective function (6)

Figure 13 shows a plot of the points forming the final population of the algorithm. It is clear that the objective surface has a large convex region for low values of objective 1 and does indeed form a line.

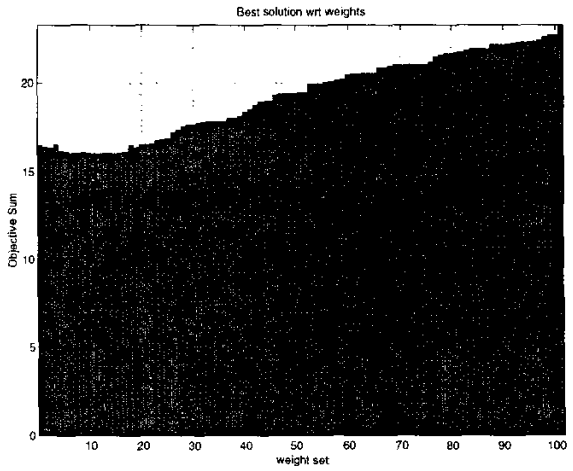

Figure 12: Plot of sum of objectives for function (6), sorted by v_1

4 Conclusions

The results of optimising the function with three objectives demonstrates that much information can be obtained about the objective surface, without having to visualise the results of the optimisation in multiple dimensions simultaneously. The Multiple Single Objective Pareto Sampling method, combined with the VADS and weighted min-max metrics,

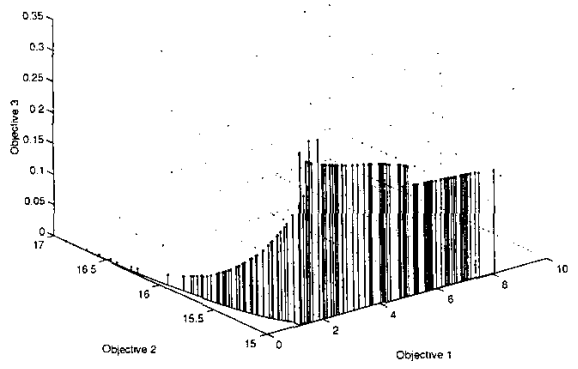

Figure 13: Plot of objectives from final generation for function (6)

has demonstrated that complex objective spaces can be analysed easily and efficiently using multiple target vector approaches. It is clear that it is very simple to focus the search into areas of the objective space that appear to be interesting.

When compared to Pareto methods where the population becomes entirely non-dominated very rapidly as the number of objectives increases, the structure of MSOPS means that the number of target vectors determines the ability to find the Pareto surface. Trials have been performed with over 1000 target vectors, and yet the EA is still able to converge well. As the number of objectives increases, more target vectors will be required to give full coverage of the potential objective space with reasonable spacing between vectors. For very large numbers of objectives, the algorithm allows the search to be focused on small areas of the potential objective space at a time.

Bibliography

Deb, Kalyanmoy (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

Jin, Yaochu, Tatsuya Okabe and Bernhard Sendhoff (2001). Adapting weighted aggregation for multiobjective evolutionary strategies. In: *Evolutionary Multi-Criterion Optimization, EMO 2001*. Springer LNCS 1993. pp. 96–110.

Schutze, Oliver, Sanaz Mostaghim, Michael Dellnitz and Jurgen Teich (2003). Covering pareto sets by multilevel evolutionary subdivision techniques. In: *Evolutionary Multi-Criterion Optimization, EMO 2003*. Springer LNCS 2632. pp. 118–132.

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

Van Veldhuizen, David A. and Gary B. Lamont (1998). Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03. Air Force Institute of Technology.