Fitness Assignment methods for Many-Objective Problems

Evan J Hughes[1]

Dept. Aerospace, Power and Sensors,
Cranfield University,
DCMT, Shrivenham, Swindon, UK.
ejhughes@theiet.org

1 Introduction

1.1 Background

All optimisation algorithms, whether of conventional design, or based on evolutionary methods, rely on being able to perform a direct comparison between two competing solutions. In order to derive a selective pressure (or gradient) towards an optima, the comparison should yield that either solution A is superior to B, or vice versa. For progress towards the global optima, then the comparison must also report that the true superior solution is indeed superior. If the solutions are equivalent, then there is no information as to which may be genotypically closer to the optimum and no progress towards the optima can be made unless a superior solution to either A or B exists elsewhere, or can be generated somehow.

With single objective problems, the assignment of a degree of *fitness* that is used to compare two solutions, is often straightforward. Complexities are introduced however when constraints are also considered. With more than one objective, it is likely that there no longer exists a single solution, but rather the best objective values are described by the Pareto front. The concept of non-domination applies and infers that two solutions lying on the Pareto front are therefore equivalent *until some additional external preferences are applied*. In order to derive a gradient or selective pressure however, the optimisation algorithm will still require a single-dimensional fitness assignment method that allows solution A to be compared directly with solution B, even though the algorithm may be maintaining an entire Pareto front in a single run.

Pareto ranking methods alone, as described in chapter **??**, will create a selective bias towards solutions on the Pareto front, but will not necessarily produce solutions that are spread across the front or at the edges. Additional elements in the fitness assignment are required to aid the Pareto ranking in order to create a diverse solution set.

Various forms of Pareto ranking and sharing/clustering have been exploited in recent years to develop a large number of multi-objective optimisation algorithms which can solve bi-objective optimisation problems effectively and reliably, for example, NSGA-II, SPEA-II etc. However, it is known that many of the methods which are efficient on bi-objective problems do not scale well to problems with large numbers of objectives (4+ typically cause issues) [8, 6].

As the number of objectives increases, typically the proportion of non-dominated solutions within a search population increases [4]. The result is that if all of the solutions are non-dominated, then all of the solutions will have the same Pareto rank, and the search towards the Pareto front reaches a plateau. In practice, the selective pressure is low even in the early phases of the search when dominated solutions exist, as only few Pareto ranks are needed to classify the population. The secondary elements of the fitness assignment function now dominate. These secondary elements are often sharing or clustering methods and serve to distribute the solutions across the non-dominated front. Thus with many-objectives, the initial optimisation progression is weakly towards the Pareto front, then in the later stages of the optimisation, the solutions are just spread out evenly. As the dimensionality increases, the spreading actions dominate rapidly giving non-dominated solutions distributed evenly, but not near the true Pareto front.

Real engineering problems are often characterised by many objectives, many constraints, or both. Often problems have constraints where information on the degree of constraint is available, and the constraints can be converted to objectives (for example chapter **??**). The constraint conversion however increases the dimensionality of the objective space (primarily in the early phases of the search until the constraints are satisfied).

The problem is how to design a many-objective fitness assignment method that will allow an optimisation algorithm to produce non-dominated fronts that are both well-spread and are also a good approximation of the true Pareto front. Currently, there are few algorithms that are designed specifically to tackle many-objective problems.

1.2 Many-Objective Fitness Assignment Methods

Two alternative approaches have been employed to date to derive useful fitness assignment processes for many objective problems: either augment the Pareto ranking concept with functions that can aid the progression towards the Pareto front; or to use approaches that do not use Pareto ranking. A general observation is that methods based on Pareto ranking still perform well on bi-objective problems, but may have computational performance issues when scaling to very large numbers of objectives; whereas many non-Pareto ranking methods scale well computationally, but are not necessarily so efficient at approximating the Pareto front with low numbers of objectives. Why do these different methods behave so differently?

Fundamentally, the fitness assignment process maps the multi/many objective space into a single dimension to allow the solutions to be ranked. There are some idealised optimisation behaviours which we would like to promote through the fitness assignment process:

1. A solution that is dominated should not be assigned a fitness superior to its dominating solution;
2. If two solutions are non-dominated, the solution with the closest neighbours should be inferior;
3. If two solutions are non-dominated, the solution closest to the true Pareto front should be superior;
4. Constrained solutions should be inferior to feasible solutions.

These 4 idealised optimisation behaviours all assume that the objective functions (and constraint satisfaction) themselves improve (whether in a minimisation or maximisation sense) as the true solution quality improves. It must be remembered that it is possible for the objective functions only to be good indicators of true solution performance in specific regions of the search space, leading to multi-modalities and therefore 'local' Pareto optimal fronts. However the fitness assignment process is by definition applied *after* the objectives have been defined and therefore must assume that the objective functions do indeed provide a true reflection of the solution quality. The ability of any optimisation algorithm to escape local optima is a property of how the new trial solutions are generated, not the fitness assignment process: generally in evolutionary methods mutation-based techniques are employed to help search for global optima.

Whenever a high dimensional space is mapped to a lower dimension, information has to be discarded and a compromise is often drawn, leading to a non-ideal fitness assignment and therefore potentially inappropriate ranking of the solutions. There are also examples where the idealised behaviour may not always produce the best performing algorithm: the handling of constraints may well be improved by compromising on item 4 [1]. Constraint handling techniques are considered further in chapter **??**.

Many approaches to multi-objective fitness assignment exploit Pareto ranking methods, which treat item 1 above as the dominant requirement, with item 2 as a secondary ranking element. Interestingly, many of these methods (for example NSGA, NSGA-II, MOGA, SPEA, SPEA-II etc.) have no mechanism for addressing item 3 directly. Instead the algorithms rely on the concept that in bi-objective spaces, driving away from dominated solutions (item 1) is a very good approximation of item 3 (moving towards the Pareto front). However for many-objective spaces where the majority of solutions are non-dominated, the approximation breaks as item 1 becomes ineffective. Thus we can now see the mechanism which describes the variation in behaviour between the Pareto ranking and non-Pareto methods: whether they address item 3 indirectly through Pareto concepts, or directly through other means.

Unfortunately, item 3 above is difficult to convert to a fitness metric, as if we knew the Pareto front from the design of the fitness assignment process, we would have solved a large part of the problem, with only the corresponding decision-space region to be identified! All of the methods that function well for many-objective problems have item 3 as the primary fitness assignment mechanism, with item 2 and sometimes also item 1 as secondary processes.

1.3 Fitness assignment visualisation

When designing and evaluating alternative fitness assignment mechanisms, it is useful to be able to visualise the behaviour the assignment process in the objective space. A simple mechanism is to create a sample set of solution points in objective space, and then evaluate the fitness that would be given to each solution. One of the solution points can then be moved across the entire objective region, usually by gridding the space to some convenient resolution, and then graphing the variation of fitness of each solution point (including a graph for the point that is moving) as this single point is moved. The result for a bi-objective problem is a set of 3D surfaces that describes how the fitness of the moving and fixed points vary as the one objective point moves. If we draw contour lines of constant fitness on these surfaces, then we can create maps of *iso-fitness contours* that can be used to visualise the behaviour of the fitness assignment method.

Although the iso-fitness contour concept can be extended to many objectives, high-dimensional visualisation becomes an issue. However, in conjunction with the 4 idealised optimisation behaviours described in section 1.2, the key characteristics of the fitness assignment process can often be determined from visualisation in two dimensions, and the expected behaviour in many dimensions predicted accurately. For simple aggregation functions, where the fitness of a point is independent of the location of other objective vectors, the iso-fitness contours alone suffice and a detailed example is provided in section 3.1.

For more complex ranking methods, we need to visualise how the rank-order would be influenced by the geometry of the points. For these methods (such as NSGA etc.) it is more useful to consider a contour of *relative iso-fitness*. A map of relative fitness is calculated by subtracting the fitness map of a fixed point, from the fitness map of the moving point (assuming the assigned fitness value is to be minimised in the ranking process). For example, figure 1 shows the locations of 5 fixed points. As a 6^{th} point is moved through the objective region, the fitness of the point of interest (at [0.7, 0.85] in the figure) is calculated and subtracted from the fitness that the 6^{th} moving point would have at the current location of the 6^{th} point on the graph. The result is that points which lie on a zero-valued contour are directly equivalent to the fixed point being studied, when considered for ranking. All regions which dominate the fixed point of interest should have a value less than zero, and all regions that are dominated should have values greater than zero. Thus item 1 in section 1.2 would be satisfied. To satisfy item 2, non-dominated solutions that are more crowded than the point of interest should have a positive relative fitness, and non-dominated solutions that are less crowded should have a negative relative fitness. To satisfy item 3, the local gradient of the fitness contour should always be to improve all objectives, i.e. falling towards the origin of the graph (in a minimisation sense), or as a worst-case leave an objective unchanged. Any regions where the gradient is towards degrading an objective value may allow a solution nearer to the true Pareto front to appear inferior to one further away. Figure 1 demonstrates these regions of relative fitness and the direction of the gradients on the contour where points are directly equivalent to the point of interest at location [0.7, 0.85]. Section 2.2 provides a detailed example of the iso-fitness contour visualisation process in action.

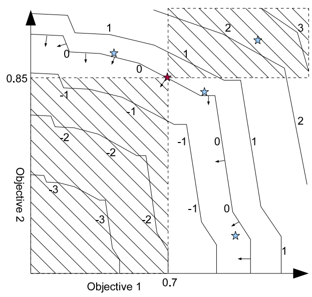

Fig. 1. Example relative iso-fitness map showing an idealised relative fitness behaviour in the dominated, dominating and non-dominated regions of the fixed-point at [0.7, 0.85] being examined.

The use of iso-fitness contours to visualise many-objective fitness assignment methods provides a simple mechanism to analyse *typical* fitness behaviours. The process can be automated in high-dimensions where the dominated and dominating regions can be assessed for correct relative performance, and the local gradients calculated in the non-dominated region and tested for any reverse-gradient conditions. However, although automation can identify fitness assignment methods that are unlikely to work well in many-objective spaces, the ability of the fitness assignment methods to create well-spread solution sets is difficult to ascertain. In addition, the shape of the iso-fitness contours is often conditioned on the distribution of the trial points in the objective region. Thus for accurate automated analysis, a Monte-Carlo process is advised where many different example sets of objective vectors are tested in order to explore the potential for adverse fitness behaviours.

1.4 Chapter Structure

This chapter discusses the behaviour of existing fitness assignment methods designed for bi-objective problems, and then methods that can function with many-objectives. Enhancements to Pareto ranking are discussed in section 2, and non-Pareto methods are discussed in section 3. Section 3.3 discusses how some of the fitness methods may be used to aid the visualisation of the Pareto front with many objectives, and section 4 concludes.

2 Pareto Ranking Extensions

2.1 Introduction

It is known that as the dimensionality of the objective space increases, then the proportion of solutions which are non-dominated in the initial random population tends to increase rapidly [4].

Generally, with many-objective problems and therefore very few dominated individuals, the selective pressure on the remaining population is very low as non-dominated individuals are considered equivalent. The spreading mechanisms dominate the selection process and the solutions are spread, rather than progressing towards the Pareto front [8]. The problem is now: how can Pareto ranking methods be extended to restore the selective pressure towards the Pareto front (i.e. item 3 in section 1.2)? Realistically, we would rather have a set of points close to the Pareto front but poorly spread, rather than a well-spread set of solutions that are far from the true optimal surface.

The Non-Dominated Sorting process alone can only separate a population into individual rank layers. Alternative strategies, such as used in SPEA and MOGA, count levels of domination and provide similar layered structures.

2.2 Non-Dominated Sorting Genetic Algorithm

The following example is based on the classic NSGA algorithm (see chapter **??**) that consists of a non-dominated sorting step, followed by sharing within the sorted layers. The weakness in the original sharing method was that *a-priori* knowledge was often needed in order to set the share radius. For demonstration purposes here, a large fixed share radius of $\sigma = 0.4$ has been used.

Figure 2 shows the relative fitness surface for point [0.7, 0.85] in a 6 point set, 5 of which are in fixed locations and the 6^{th} is moved through the objective region in order to generate the fitness surface. The values on the graph are the difference in fitness value between the moving and fixed point at [0.7, 0.85]. Figure 3 shows the corresponding relative iso-fitness contour map. Thus if the moving point was at location [0.4, 0.63], then the figures show that the fitness

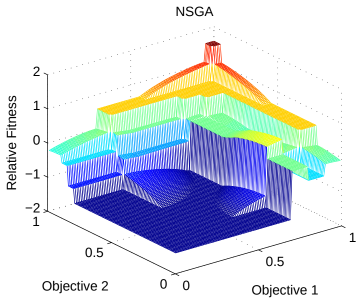

Fig. 2. 3D relative fitness surface for point [0.7, 0.85] and the NSGA method with one moving and 5 fixed objective vectors. Spreading factor is $\sigma = 0.4$

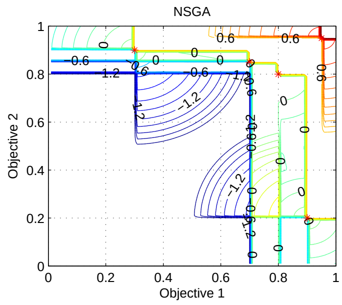

Fig. 3. Relative iso-fitness contours for point [0.7, 0.85] and the NSGA method with one moving and 5 fixed objective vectors. Spreading factor is $\sigma = 0.4$

of the moving point would be less by a value of 1.2 and therefore more likely to be selected after ranking.

The key feature of figure 3 is that although the non-dominated front is visible clearly (as a line connecting all the non-dominated points), many of the contours traverse *across* the objective space, rather than focussing towards the origin (which would be an ideal solution). If we consider point [0.2, 0.8] for example, the relative fitness is very low, demonstrating that the point is highly attractive. If we now consider point [0.75, 0.5], the point is non-dominated with respect to the test point at [0.7, 0.85], however the local gradient is focussed towards improving objective 1, but *degrading* objective 2! Therefore the requirement of item 3 in section 1.2 is compromised as we are moving away from the Pareto front. The problem with the classic NSGA algorithm is that as the dimensionality increases, regions with adverse fitness gradient structures become more common and the optimisation process is compromised.

With the contour alignment in the non-dominated region approximately normal to the non-dominated front, the selective focus is to spreading the solutions, rather than driving towards the Pareto front. With many objectives, the problems are exacerbated.

2.3 Non-Dominated Sorting Genetic Algorithm II

Figures 4 and 5 show the relative fitness surface and relative iso-fitness contours for the NSGA-II fitness assignment process with point [0.7, 0.85] as a fixed reference. NSGA-II uses crowding distance rather than fitness sharing (the use of $-\infty$ at the edges has been modified to provide a consistent and representative fitness landscape). It is clear that the fitness gradient is changed significantly over the original NSGA algorithm. The crowding operator is calculated based on the location of the neighbour solutions to the point (in the same non-dominated rank), and the result is that the fitness in the local region remains constant as long as the local neighbour solutions are the same. The fitness surface is now dominated by plateaus, rather than continuous gradients. The optimisation performance is improved over NSGA as there are fewer regions where the gradient is away from the Pareto surface, however they still exist and point [0.75, 0.5] is a good example. Unfortunately, within the plateaus, there is no selective pressure to either converge towards the Pareto surface, or to spread evenly, however a plateau is preferable to a reverse-gradient in the non-dominated region. In a practical algorithm, a moderate or large population size would be desirable in order to reduce the scale of each plateau region (i.e. smaller distances to neighbours). As the NSGA-II algorithm maintains the elite solutions within the working population, in practice the population sizes are often sufficiently large to make the ranking process perform well in low-dimensions. With many-objective problems however, still having areas in the non-dominated region of reverse-gradient degrades the algorithm performance.

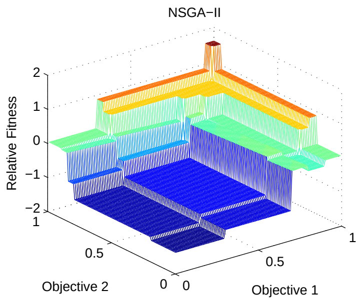

Fig. 4. 3D relative fitness surface for point [0.7, 0.85] and the NSGA-II method with one moving and 5 fixed objective vectors.

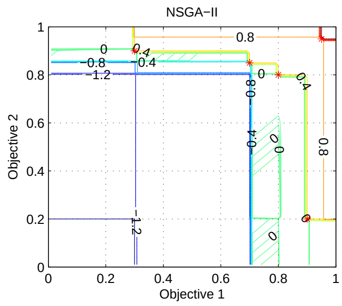

Fig. 5. Relative iso-fitness contours for point [0.7, 0.85] and the NSGA-II method with one moving and 5 fixed objective vectors.

2.4 Multi-Objective Genetic Algorithm

The Multi-Objective Genetic Algorithm (MOGA) [4] counts the number of solutions that a point is dominated by, and then uses a sharing mechanism to spread the solutions. The fitness landscape that is obtained is very similar to NSGA, however not all implementations of the algorithm confine the sharing mechanism to individual rank layers, of the same domination count, as in the NSGA algorithm. Figures 6 and 7 show the relative fitness surface generated from the MOGA algorithm *when the sharing was not confined to rank layers*.

In the figures, a share distance of $\sigma = 0.1$ has been used. Issues with reverse gradient in the non-dominated region are visible clearly in many re-

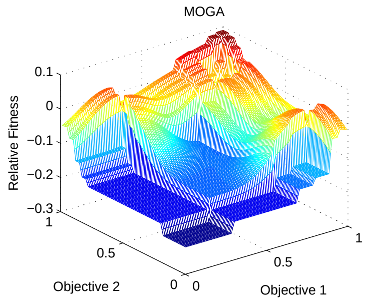

Fig. 6. 3D relative fitness surface for point [0.7, 0.85] and the MOGA method with one moving and 5 fixed objective vectors. Spreading factor is $\sigma = 0.1$

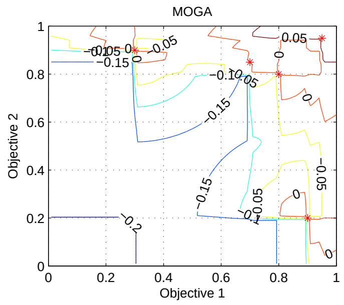

Fig. 7. Relative iso-fitness contours for point [0.7, 0.85] and the MOGA method with one moving and 5 fixed objective vectors. Spreading factor is $\sigma = 0.1$

gions (point [0.85, 0.3] for example). The point [0.85 0.85] is interesting as the gradient is converging towards it, acting as a local attractor. This is a direct result of using global sharing, rather than restricting sharing to within rank/domination layers, and is caused by empty regions within the dominated space being emphasised by the sharing process. Additionally, the point [0.85 0.85] shows that item 1 in section 1.2 is compromised as there are solutions in the dominated region that are superior to the test-point at [0.7, 0.85]. If rank/domination layer sharing is applied, then the fitness surface is very similar to the surface obtained with NSGA. The MOGA algorithm does not scale well to many-objective problems either.

2.5 Hyper-volume Selection

Fundamentally, the hypervolume metric [10] assesses the total volume that lies between a chosen reference point that acts as a corner to a hypercube, and the non-dominated front, described by a set of points, which intersects the hypercube. The closer the non-dominated front is to the Pareto front, then the larger the hypervolume.

A simple way to use the hypervolume metric to augment the basic Pareto ranking process is to first use Non-dominated sorting to establish the front that a particular point belongs to, and then calculate the hypervolume of the set of points which includes all points on that front and all points that are worse. The point of interest is then removed from the set and the hypervolume re-calculated, allowing a change in hypervolume, ΔS, to be established. The change ΔS is then normalised by the maximum possible hypervolume to give an indicator of *local* worth $\Delta S/V_{max}$. The fitness is then the Pareto rank layer index number minus $\Delta S/V_{max}$. As long as $\Delta S/V_{max} < 1$ then the Pareto ranking structure will be preserved, with the local shaping of the fitness surface being provide by the hypervolume metric.

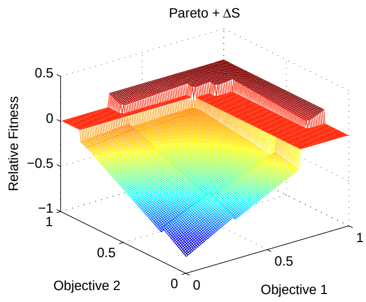

Fig. 8. 3D relative fitness surface for point [0.7, 0.85] and the hybrid Non-Dominated Sorting/ Hypervolume method with one moving and 5 existing objective vectors.

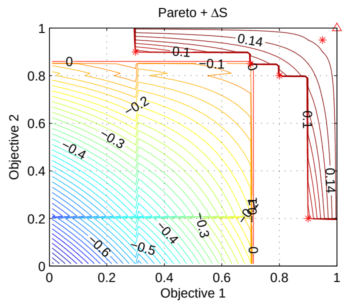

Fig. 9. Relative iso-fitness contours for point [0.7, 0.85] and the hybrid Non-Dominated Sorting/ Hypervolume method with one moving and 5 existing objective vectors.

For minimisation, the hypervolume reference point R is placed in such a way as to be at least weakly dominated by every member of the set to be investigated. Often, the reference point is chosen as the maximum value observed in all objectives so far.

Figures 8 and 9 show the resultant relative iso-fitness contours for a reference point at $\mathbf{R} = [1, 1]$. It is clear to see that the fitness gradient is always well-defined in the dominated/ dominating regions, however in the non-dominated region, there is a significant plateau before the non-dominated front, which can reduce overall algorithm efficiency. There are no regions of reverse relative fitness gradient however.

The hypervolume metric produces a weak 'spreading' effect, however it is clear that points are being driven slowly towards [0.3, 0.9] by a gradient that is primarily independent of solution locations: although it may cause solutions to crowd locally in the area, the focus will aid the discovery of better extreme solutions. The main drive is towards the Pareto front and the distribution of solutions is a weak secondary process with this simple implementation. Many of the practical implementations of the hypervolume metric [7, 3] use extra processes to improve the distribution of the solutions further. The lack of any non-dominated regions with reversed-gradient characteristics makes the method suitable for many-objective optimisation, however the processing complexity of the hypervolume often limits its applicability.

2.6 Indicator-Based Evolutionary Algorithm

The *Indicator-Based Evolutionary Algorithm* (IBEA) process [11] does not use Pareto ranking directly, but instead uses *indicator functions* that allow the fitness of a solution in a population to be determined. The indicators however, are designed to preserve Pareto rank specifically.

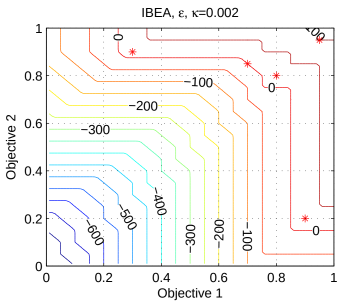

Fig. 10. Relative iso-fitness contours for point [0.7, 0.85] and the IBEA method with one moving and 5 existing objective vectors. Scaling factor $\kappa = 0.002$

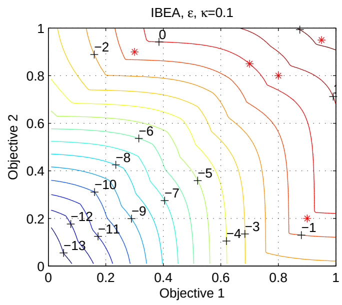

Fig. 11. Relative iso-fitness contours for point [0.7, 0.85] and the IBEA method with one moving and 5 existing objective vectors. Scaling factor $\kappa = 0.1$

Figure 10 shows the iso-fitness contours for the IBEA ϵ indicator in equation 1, with a shape parameter of $\kappa = 0.002$. In (1) O_{iA} is the value of the i^{th} objective of population member A, k is the number of objectives and P is a set of objective vector points that describe the population. For visualisation purposes, the difference in the logarithm of the fitness has been plotted.

$$
\begin{aligned}
I(A,B) &= \min_{\epsilon}\{O_{iA} - \epsilon \leq O_{iB} \text{ for } i \in \{1,\ldots,k\}\} \\
f &= \sum_{\mathbf{O}_A \in P \neq \mathbf{O}_B} -exp(-I(A,B)/\kappa)
\end{aligned} \tag{1}
$$

It is clear to see that the relative iso-fitness contours of the ϵ indicator are structured with only a weak relationship to the shape of the non-dominated surface. The relative fitness values in the dominated and dominating regions are correct. The fitness gradient is well structured with no reverse gradients, and in regions beyond the edges of the non-dominated front, will promote good edge exploration. The fitness method does lack any intrinsic directionality to aid a uniform distribution of solutions across the Pareto front however. The lack of gradient for forming uniform distributions in the population is evidenced in figure 10 by solutions which are well-spread (e.g. [0.9, 0.2]) not being promoted as superior to the point under test at [0.7, 0.85].

Figure 11 shows how an alternative choice of the scaling factor κ can change the structure of the fitness surface. With the larger value of $\kappa = 0.1$, the algorithm will not perform so well on highly concave Pareto fronts as the iso-contours do not form sharp 'corners' and will penalise Pareto solutions within a deep concavity. In figure 11, the points [0.3, 0.9] and [0.9, 0.2] both lie on a fitness contour that is superior to the test point [0.7, 0.85], which is in turn superior to point [0.8, 0.8]. All of the points are non-dominated, but the modified fitness function is promoting solutions which are isolated over those which are crowded and will achieve a superior spread of solutions than with the smaller value for κ.

The configuration is likely to give good performance on convex and mildly concave Pareto fronts. The fitness contours also demonstrate that although a strict Pareto relationship is maintained when comparing two solutions in isolation, the act of combining the results for a population of solutions can produce iso-fitness contours that do not follow the non-dominated front, yet not compromise any of the ideal requirements itemised in section 1.2. The IBEA method will work well on many-objective functions, however an external diversity mechanism, such as clustering of an archive, is recommended to achieve a controlled spread of solutions.

2.7 Summary of Pareto Methods

If the Pareto rank is enforced in the fitness process, with inferior ranks guaranteed to have a worse fitness than true non-dominated solutions, it is possible to create optimisers that perform very well in bi-objective problems. However, for many-objective problems, the gradient of the fitness within the non-dominated regions must also always focus towards improving all objectives to some degree.

The IBEA and hypervolume methods have both demonstrated advantageous fitness gradient structures, but at the expense of limited (if any) solution distribution characteristics intrinsic within the fitness assignment. The desire to move away from existing solutions, yet not degrade performance on any objectives are conflicting in many circumstances. Realistically, the degree of solution spreading that can be generated by the fitness assignment function alone is limited, but an external archive can be used to help impose a uniform spread of solutions across the Pareto front, for example, through clustering etc.

3 Non-Pareto Ranking Methods

An alternative many-objective fitness assignment process 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 sum (section 3.1) 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) [5] is one method that develops this concept into a practical algorithm.

Many early multi-objective EAs do not use Pareto ranking methods, such as VEGA (see chapter ??) and Weighted Average Ranking [2]. Many of these early approaches have some merit in many-objective spaces and are worthy of investigation.

3.1 Repeated Single Objective

In the Repeated Single Objective (RSO) [6] approach, a conventional single objective EA is used, based on an aggregation function, and repeat runs are performed for different target search directions, allowing a Pareto front to be constructed from a sequence of spot solutions. To allow direct comparison with true multi-objective EAs, each run of RSO uses a correspondingly smaller population size and number of generations to keep the total number of evaluations to identify a Pareto front consistent.

The RSO method is very simple but does require an *a-priori* specification of the directions to search, in order to populate the Pareto front, which can be difficult with previously unseen problems. The RSO method is known to be effective in high-dimensional many-objective optimisation problems [6]. The performance and applicability of RSO to different objective structures is determined primarily by the choice of the aggregation functions used to identify optimal solutions.

A key benefit of pre-specification of search directions is that designer preferences can be incorporated very easily and the search focussed to only regions of interest. Additionally, Pareto front analysis may be performed as described in section 3.3.

Aggregation Functions

Aggregation functions have been used for many years in classical gradient-based optimisation. Generally, a single aggregation function will yield a single Pareto point, however, all of the aggregation functions described here are effective with both low and high-dimensional objective spaces. In practice, once the structure of the Pareto surface has been approximated, a decision has to be made about the particular Pareto point to choose. An aggregation function can then be used in a single-objective optimisation framework to identify a single near-Pareto solution.

The following common aggregation functions have been plotted with their fitness functions arranged for objective minimisation. The fitness contours have been drawn in the same context as the population-based methods, with the 5 example population points plotted to allow direct comparison. It should be remembered however that the following aggregation methods are conditioned only by the weight vector and control parameters etc., and not by the location of the other population members. Thus the iso-fitness contour plots are contours of true fitness values, but should still allow dominated individuals to be inferior to dominating solutions.

Weighted Sum

The weighted sum score of k objectives is calculated using (2),where w_i is part of the weight vector $\mathbf{W} = [w_1,\ w_2,\ \ldots]$ and is the weight of the i$^{\text{th}}$ objective O_i.

$$f = \sum_{i=1}^{k} (w_i O_i) \tag{2}$$

Weighted sum will not introduce discontinuities into the gradient of the aggregated function but is able to generate points **only on convex** Pareto fronts. The location of the point on the Pareto front is highly dependent on the shape of the front itself, however the search 'direction vector' may be described as $\mathbf{V} = \mathbf{W}$.

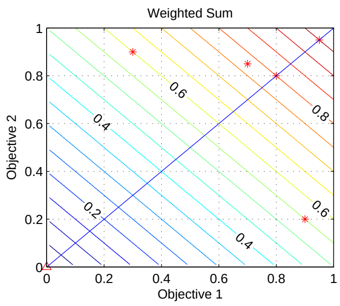

Fig. 12. Iso-fitness contours for the Weighted Sum aggregation method. Weights are $\mathbf{W} = [1, 1]$

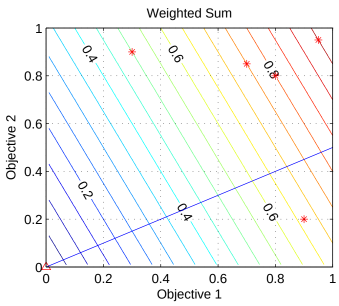

Fig. 13. Iso-fitness contours for the Weighted Sum aggregation method. Weights are $\mathbf{W} = [2, 1]$

Figures 12 and 13 show the *iso-fitness contours* for the weighted sum aggregation function with two different weighting vectors. The diagonal line radiating from the origin shows the search direction vector $\mathbf{V}$. The iso-fitness contours form hyperplanes normal to the search direction $\mathbf{V}$. The weighted sum method will return Pareto points where the normal to the Pareto front is parallel to $\mathbf{V}$, and therefore normal to the iso-fitness contours. Thus the aggregation method is unsuitable if points within a concave region of a Pareto front are to be identified. The primary benefit of the weighted sum is that if the gradients of the individual objective functions are continuous, then the gradient of the resulting fitness value, f, will also be continuous. No additional constraints are created, allowing the weighted sum to be used as an aggregation method with *all* optimisation algorithms.

Goal Attainment

The goal attainment score of k objectives is calculated by transforming all of the objectives into objective-space constraints using (3),where w_i is the weight of the i$^{\text{th}}$ objective O_i, and Z_i is the i$^{\text{th}}$ dimension of an idealised reference point $\mathbf{Z}$.

$$\begin{aligned}&\text{Minimise } \gamma \text{ subject to:}\\ &O_i - w_i\gamma \leq Z_i \ \ \forall i \in [1, k]\end{aligned} \tag{3}$$

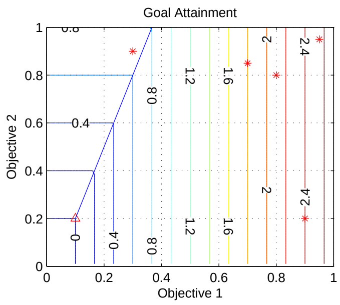

Fig. 14. Iso-fitness contours for the Goal Attainment aggregation method. Reference point is $\mathbf{Z} = [0.1, 0.2]$, weights are $\mathbf{W} = [1, 3]$

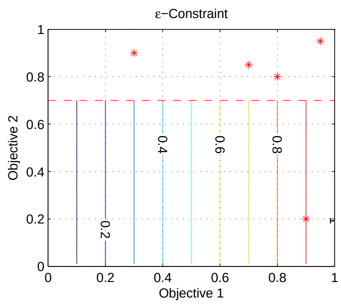

Fig. 15. Iso-fitness contours for the ϵ-constraint aggregation method. Objective 1 is being minimised, while constraining objective 2 to ≤ 0.7

The control parameter γ is reduced until the constrained region consists of a single feasible point. This point lies on the Pareto front. If the final value of γ is negative, the reference point $\mathbf{Z}$ has been dominated. Figure 14 shows the iso-fitness contours for the goal attainment aggregation function. As the iso-fitness contours always form a 'corner' which has its sides aligned parallel to the objective axes, goal attainment 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 $C = (O_1 - Z_1)/w_1 = (O_2 - Z_2)/w_2 = \ldots$, where C is a constant, 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 $\mathbf{V} = [w_1,\ w_2,\ \ldots]$ (after offsetting by the reference point $\mathbf{Z}$). Thus the angle between the vectors $\mathbf{V}$ and $\mathbf{O} - \mathbf{Z}$ indicate whether the solution lies where it was expected or not. If the vector $\mathbf{V}$ 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 can only be performed if the weight vectors are defined prior to the optimisation run. Section 3.3 provides examples of the process.

Unfortunately goal attainment relies heavily on the optimisation algorithm being able to implement constraints efficiently. Constraint handling in evolutionary processes is possible, but not often efficient.

ϵ-constraint

The ϵ-constraint metric converts all but one of the objectives into objective space constraints. The optimisation process operates on the one remaining objective and the Pareto front point chosen is usually a point that best satisfies the objective, and is just within the feasible region defined by the constraints on the remaining objectives. As the constraint locations are moved, other Pareto points may be identified. Equation (4) describes the process, where C_i is the constraint location of the i^{th} objective O_i.

$$\begin{aligned} &\text{Minimise } f = O_j \text{ subject to:} \\ &O_i \leq C_i \ \ \forall i \neq j \in [1, k] \end{aligned} \tag{4}$$

Figure 15 shows the iso-fitness contours for the ϵ-constraint aggregation function. Like goal attainment, the iso-fitness contours always form a 'corner' which has its sides aligned parallel to the objective axes, thus ϵ-constraint is able to generate points on both convex and concave Pareto sets. If the objective to be minimised is chosen carefully, the gradient of the optimisation surface can be very favourable, for example if the first three objectives are highly multi-modal, but the fourth is unimodal, it makes sense to constrain the first three and optimise the fourth.

Weighted Min-Max

The weighted min-max score of k objectives is calculated using (5),where w_i is the weight of the i^{th} objective, O_i.

$$f = \max_{i=1}^{k}(w_i O_i) \tag{5}$$

Figures 16 and 17 show the iso-fitness contours for the weighted min-max aggregation function with two different weighting vectors. Like goal attainment, 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 $w_1 O_1 = w_2 O_2 = \ldots$, 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 $\mathbf{V} = [1/w_1,\ 1/w_2,\ \ldots]$.

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 can be considered as a weighted L^∞ metric but with a reference point at the origin.

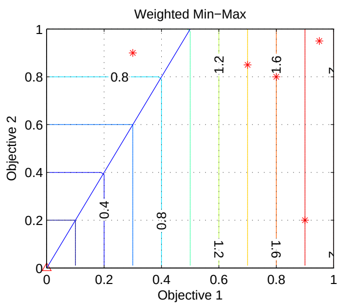

Fig. 16. Iso-fitness contours for the Weighted Min-Max aggregation method. Weights are $\mathbf{W} = [2\;1]$

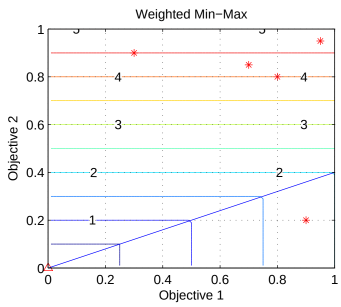

Fig. 17. Iso-fitness contours for the Weighted Min-Max aggregation method. Weights are $\mathbf{W} = [2\;5]$

L^p Norm

The L^p Norm score of k objectives is calculated by using (6),where O_i is the i$^{\text{th}}$ objective of the vector $\mathbf{O}$, Z_i is the i$^{\text{th}}$ dimension of an idealised reference point $\mathbf{Z}$, W_i is a weighting component and p is a scalar factor that determines the shape of the iso-fitness contours. For the classic L^p norm methods, unity weighting factors $W_i = 1$ are usually assumed.

$$f = \left[\sum_{i=1}^{k} W_i |O_i - Z_i|^p \right]^{\frac{1}{p}} \tag{6}$$

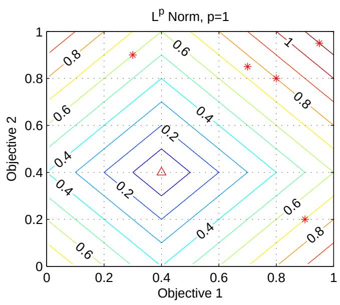

Fig. 18. Iso-fitness contours for the L^p Norm aggregation method. Reference point is $\mathbf{Z} = [0.4, 0.4]$, shape is $p = 1$

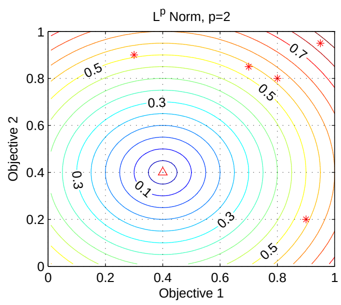

Fig. 19. Iso-fitness contours for the L^p Norm aggregation method. Reference point is $\mathbf{Z} = [0.4, 0.4]$, shape is $p = 2$

Figures 18, 19 and 20 show the iso-fitness contours for the L^p norm aggregation function with three different shape parameters. The classic L^p norm

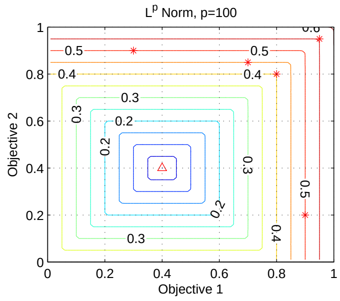

Fig. 20. Iso-fitness contours for the L^p Norm aggregation method. Reference point is $\mathbf{Z} = [0.4, 0.4]$, shape is $p = 100$

does not use a weight vector, rather the Pareto point that is closest to the reference point $\mathbf{Z}$ will give the lowest aggregated value. Thus as $\mathbf{Z}$ is moved, points on the Pareto front can be generated. The reference point $\mathbf{Z}$ must dominate the nearest points on the Pareto front, otherwise the optimiser will simply converge to the point $\mathbf{Z}$ if it lies within the objective region.

Figure 18 uses a shape of $p = 1$ and is therefore the L^1 norm, or *Manhattan* distance. The iso-fitness contours form lines that have similar properties to the weighted sum, however if the reference point is placed within a concave region of the Pareto front, points within the concavity can be found, although the reference must be placed very close to the Pareto front in order to identify regions of sharp concavities.

Figure 19 uses a shape of $p = 2$ and is therefore the L^2 norm, or *Euclidean* distance. The iso-fitness contours form circular contour lines, allowing shallow concave regions to be identified easily. Sharp concavities will still require the reference point to be placed very close to the Pareto front. With low values of p, the gradient of the objective functions and therefore the gradient of the aggregated fitness is maintained.

Figure 20 uses a shape of $p = 100$ and is therefore the L^{100} norm. The iso-fitness contours approximate corners now, similar to the corners displayed by the Weighted Min-Max method, allowing even quite sharp concave regions to be identified easily. With these high values of p, the gradient of the aggregated function can be subject to numerical errors and appear discontinuous. A metric with $p = \infty$ is the Tchebychev Norm, and the infinite power is approximated by a max() operation.

Vector Angle Distance Scaling (VADS)

Vector Angle Distance Scaling (VADS) is a new metric first introduced in [5]. 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. All of the metrics so far that are capable of identifying the Pareto surface in the presence of concavities use an iso-fitness contour that forms a 'corner'. To identify regions of the objective front, a mechanism is needed to form the iso-fitness contours into acute angles in order to allow deep probing into highly concave regions. The use of these metrics for surface analysis is discussed further in section 3.3.

The VADS score is the magnitude of the vector of objectives ($|\mathbf{O}|$), 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 k objectives is calculated using (7), where $\mathbf{V}$ is the k-dimensional unit-length target vector which describes the point on the objective front to search for, $\mathbf{O}$ is the k-dimensional objective vector, $|\cdot|$ indicates vector magnitude and q is a constant factor for scaling the cosine result (typically $q = 100$). The vector $\mathbf{V}$ may also be described in terms of the weight vector used in the other metrics as the normalisation $\mathbf{V} = \mathbf{W}/|\mathbf{W}|$.

$$f = \frac{|\mathbf{O}|}{\left(\mathbf{V} \cdot \frac{\mathbf{O}}{|\mathbf{O}|}\right)^q} \tag{7}$$

Low values for q may lead to difficulty in identifying very sharp concavities in the objective front. The dot product of the vector $\mathbf{V}$ with the objective vector $\mathbf{O}$ must remain positive for the basic VADS metric to function correctly, and consequently objective offsets may be necessary for proper operation.

Figure 21 shows the iso-fitness contour for a weight vector of $\mathbf{W} = [1, 1]$ and shaping parameter $q = 100$. The 'tear-drop' shaped iso-fitness contour is made thinner by increasing q, allowing sharper concavities to be probed. With very high values of q, 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 (7) to be re-formulated as shown in (8) and reduces the impact of numerical imprecision.

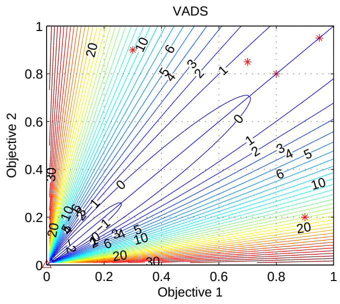

Fig. 21. Iso-fitness contours for the Vector-Angle Distance Scaling aggregation method. Weights are $\mathbf{W} = [1, 1]$, $q = 100$. Logarithm of fitness plotted for clarity.

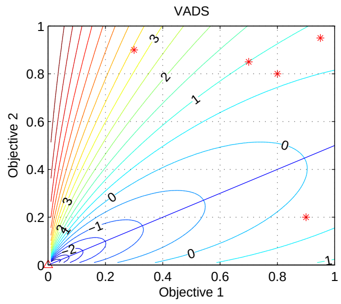

Fig. 22. Iso-fitness contours for the Vector-Angle Distance Scaling aggregation method. Weights are $\mathbf{W} = [2, 1]$, $q = 10$. Logarithm of fitness plotted for clarity.

$$f = \exp((q + 1)\log(|\mathbf{O}|) - q \log(\mathbf{V} \cdot \mathbf{O})) \tag{8}$$

Figure 22 shows the iso-fitness contour for a weight vector of $\mathbf{W} = [2, 1]$ and shaping parameter $q = 10$. With the lower value for q, the 'tear-drop' shaped iso-fitness contour is fatter and therefore less able to probe deep folds in the objective surface. It is also clear that as the weight vector is changed, the iso-fitness contour follows the vector, rather than being aligned to the objective axes.

The final solution identified by an optimiser using the VADS metric should have the objective vector $\mathbf{O}$ lying parallel to the target vector $\mathbf{V}$. 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 $\mathbf{V}$ and $\mathbf{O}$ indicate non-obtainable sections in the objective region.

3.2 Multiple Single Objective Pareto Sampling

Multiple Single Objective Pareto Sampling (MSOPS) [5] 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 vectors. Thus the algorithm uses a matrix of target vectors to search in parallel. It is also possible to combine searches in different directions and with different reference points, and searches using different aggregation functions all within a single optimisation run. The key advantage is that the algorithm does not rely on Pareto ranking to provide selective pressure. As the target vectors are generally 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 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. Each *column* of the matrix S corresponds to one target vector (across the population P). The aggregate fitness, f_i, of the i^{th} member of P is calculated using equation 9, where $f_n(\mathbf{O}_i, \mathbf{V}_n, \mathbf{Z}_n)$ is the aggregation function n with target vector $\mathbf{V}_n$ and reference point $\mathbf{Z}_n$ for objective vector $\mathbf{O}_i$ (which is the i^{th} member of P).

$$f_{i \in P} = \min_{\forall n \in T} \left(\frac{f_n(\mathbf{O}_i, \mathbf{V}_n, \mathbf{Z}_n)}{\min_{\forall j \neq i \in P}(f_n(\mathbf{O}_j, \mathbf{V}_n, \mathbf{Z}_n))} \right) \tag{9}$$

The flexibility of the approach is such that the target vectors can be arbitrary, either generated to give full coverage of the objective space if no *a-priori* domain knowledge exists, or with some structure to target key elements of the search volume. As the fitness combination method employed is based on the set of fixed target vectors, the target vector set determines the final spread of the obtained solutions. As a consequence though, the efficiency of the algorithm is reduced in relation to the number of unobtainable target vectors that do not pass through the feasible objective region.

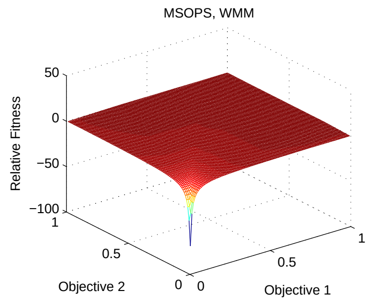

Fig. 23. 3D relative fitness surface for point [0.7, 0.85] and the MSOPS method with one moving and 5 existing objective vectors and using 10 Weighted Min-Max target vectors.

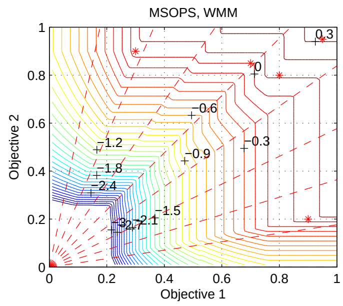

Fig. 24. Relative iso-fitness contours for point [0.7, 0.85] and the MSOPS method with one moving and 5 existing objective vectors and using 10 Weighted Min-Max target vectors.

Figures 23 and 24 show the relative fitness surface and contours for the MSOPS algorithm using 10 target vectors, referenced at the origin, and the

Weighted-Min Max aggregation function. It is clear in figure 24 that the relative fitness gradient in the non-dominated region is never counter to the objective directions and so will provide rapid progress towards the Pareto front; however the iso-fitness contours are also not aligned to the true non-dominated surface of the other population members, rather they are aligned to a combination of the target vectors and non-dominated surface. For example point [0.8, 0.8] is non-dominated and yet does not lie on the same fitness contour as the other non-dominated solutions as it is far from a target vector. In contrast point [0.9, 0.2] is closer to a target vector line and so has a better fitness (however is still inferior to the test point [0.7, 0.85]). The result is that the final population will cluster around the points where the target vectors cut the Pareto front (or the nearest feasible point in a weighted min-max sense).

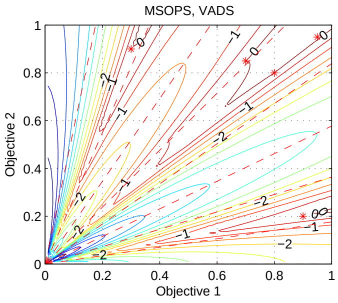

Fig. 25. Relative iso-fitness contours for point [0.7, 0.85] and the MSOPS method with one moving and 5 existing objective vectors and using 10 VADS target vectors (plotted as logarithm of fitness). Shape parameter is $q = 100$

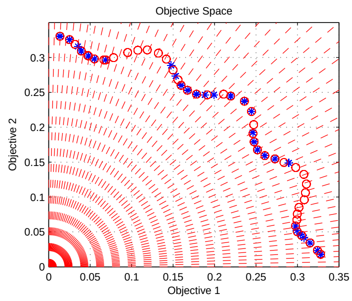

Fig. 26. Plot of objective/decision space for function (10). Circles are VADS solutions, stars are Weighted Min-Max solutions.

Figure 25 shows the MSOPS relative iso-fitness contours when using the VADS Aggregation metric. The VADS contours in figure 25 are very complicated and it is clear that the directions of the target vectors are a dominating factor in the description of the fitness surface. The VADS metric is designed for identifying the *Objective* front profile, rather than just the Pareto front. Thus highly concave and re-entrant surfaces may be probed with this metric. Unfortunately, the relative fitness gradient in the non-dominated regions are often not ideal and optimisation performance is compromised in both bi-objective and many-objective problems.

Empirical studies have shown that running the MSOPS algorithm with both Weighted Min-Max and VADS Aggregation will provide superior optimisation performance than VADS alone. When both metrics are combined, the weighted min-max process dominates initially and minimises the effects of reverse gradients in the non-dominated regions. As the algorithm converges, the VADS metric can help to provide a more balanced search in difficult regions such as extreme convexities.

3.3 Pareto Front Analysis

To demonstrate the use of RSO or MSOPS for analysis of the objective and Pareto front, the Tanaka two-objective test function has been studied [9]:

$$\begin{aligned} O_1 &= x \\ O_2 &= y \\ 0 &\geq -(x)^2 - (y)^2 + 1 + 0.1\cos\left(16\arctan\left(\frac{x}{y}\right)\right) \\ 0.5 &\geq (x-0.5)^2 + (y-0.5)^2 \\ 0 &\leq x, y \leq 1 \end{aligned} \tag{10}$$

Figure 26 shows the result of MSOPS using combined Weighted Min-Max and VADS, applied to (10) with 51 target vectors (shown as dashed) and a population of 50 (run for 100 generations). The stars indicate the best set of solutions found with the weighted min-max and the circles are the best VADS solutions. The 51 weight vectors were generated *a-priori* using the origin as a reference point and designed to cover the objective space with equal angles between neighbouring vectors.

It is clear that points on the boundary of the objective front have been identified. The 'leading edge' of the objective space is identified by VADS, while Weighted Min-Max finds the Pareto front. The use of two aggregation functions is very useful for analysing the behaviour of the objectives, rather than just the Pareto front. The area around [0.1, 0.3] is a discontinuity in the Pareto front and as such has only been identified in the VADS search. The corresponding plots of angular errors between each target vector and the 'best performing' objective vector for VADS and min-max respectively are shown in figures 27 & 28, sorted according to the weights with V_1 (the first element of the target vectors) increasing. It is clear that many of the target vectors were satisfied with an error less than 2° to their nearest objective vector for VADS; but there are areas with high errors for weighted min-max, indicating that some target vectors could not be obtained exactly. These errors correspond to the limits of the Pareto set in the VADS plot (first, and last two target vectors in figure 27) and also to the discontinuities in the function in the weighted min-max plot (around vectors 15 and 37 in figure 28).

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. 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

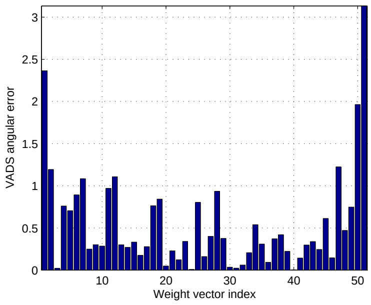

Fig. 27. Plot of angular errors (in degrees) of target vectors to their best performing objective vector using the VADS metric and function (10)

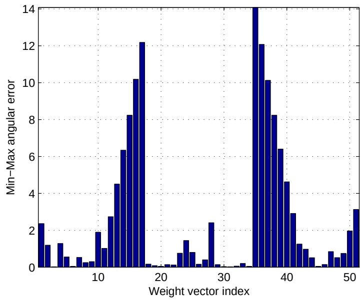

Fig. 28. Plot of angular errors (in degrees) of target vectors to their best performing objective vector using the weighted min-max metric and function (10)

simple though with RSO or MSOPS 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 and incorporation of designer preferences and interactive decision-making. Strategies may be used to yield extra information about the Pareto front such as generating a set of target vectors that lie on a plane, allowing 'slices' through the Pareto front to be visualised to test for continuity.

3.4 Summary of Non-Pareto Methods

Both RSO and MSOPS are capable of generating Objective and Pareto fronts in low and high-dimensional objective spaces. However, the MSOPS process is more efficient in practice and is recommended if multiple target vectors are to be considered. The RSO algorithm is best when a single final optimal solution is to be generated. Interestingly, although the IBEA method has been described as using Pareto concepts, the relationship to MSOPS is very strong and it could be argued that MSOPS is an indicator-based algorithm that has not been restricted to identifying the Pareto front alone.

As both RSO and MSOPS utilise aggregation functions, the wide variety of functions available allow a comprehensive analysis of the objective and Pareto surface to be performed.

4 Conclusions and Recommendations

Most optimisation algorithms to date have focussed on bi-objective problems and many, unfortunately, do not extend well to many-objective problems with 4+ objectives. This chapter has shown that the properties of the fitness assignment process can be visualised and analysed to assess the suitability of a method for many-objective optimisation. Some of the fitness assignment methods may also be used to aid analysis and visualisation of the objective and Pareto fronts.

It is unlikely that a simple fitness assignment function will provide both selective pressure towards the Pareto front, while also providing effective drive towards a set of well-spread solutions. It is more likely that an influence/mechanism external to the fitness assignment process may be needed to ensure that a satisfactory distribution of solutions is obtained, such as clustering or automatic target vector generation.

References

1. Thomas Phillip Runarsson adn Xin Yao. Stochastic ranking for constrained evolutionary optimisation. In *IEEE Transactions on Evolutionary Computation*, volume 4, pages 284–294. IEEE, September 2000.
2. P. J. Bentley and J. P. Wakefield. An analysis of multiobjective optimization within genetic algorithms. Technical Report ENGPJB96, University of Huddersfield, UK, 1996. Online: http://citeseer.ist.psu.edu/62443.html.
3. L. Bradstreet, L. Barone, and L. While. Maximising hypervolume for selection in multi-objective evolutionary algorithms. In *IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1744 – 1751, Vancouver, Canada, July 2006. IEEE.
4. Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001. ISBN 0-471-87339-X.
5. Evan J. Hughes. Multiple single objective pareto sampling. In *Congress on Evolutionary Computation 2003*, pages 2678–2684, Canberra, Australia, 8-12 December 2003. IEEE.
6. Evan J. Hughes. Evolutionary many-objective optimisation: Many once or one many? In *IEEE Congress on Evolutionary Computation, 2005*, volume 1, pages 222–227, 2005.
7. B. Naujoks, N. Beume, and M. Emmerich. Multi-objective optimisation using S-metric selection: application to three-dimensional solution spaces. In *IEEE Congress on Evolutionary Computation, CEC 2005*, volume 2, pages 1282 – 1289, Edinburgh, UK, September 2005. IEEE.
8. 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.
9. 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.
10. E. Zitzler. *Evolutionary algorithms for Multiobjective Optimisation: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.
11. Eckart Zitzler and Simon Knzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature - PPSN VIII*, volume LNCS 3242/2004, pages 832–842, Birmingham, UK, 2004. Springer.

Index