DIRECTED MULTI-OBJECTIVE OPTIMISATION

M. Brown[a] and R. E. Smith[b]

[a]Control Systems Centre, School of Electrical and Electronic Engineering,
University of Manchester, UK
martin.brown@manchester.ac.uk

[b]The Intelligent Computer Systems Centre, The University of West of England,
Bristol, UK
robert.smith@uwe.ac.uk

ABSTRACT
While evolutionary computing inspired approaches to multi-objective optimization have many advantages over conventional approaches; they generally do not explicitly exploit directional/gradient information. This can be inefficient if the underlying objectives are reasonably smooth, and this may limit the application of such approaches to real-world problems. This paper develops a local framework for such problems by geometrically analyzing the multi-objective concepts of descent, diversity and convergence/optimality. It is shown that locally optimal, multi-objective descent direction can be calculated that maximally reduce all the objectives and a local sub-space also exists that is a basis for diversity updates. Local convergence of a point towards the optimal Pareto set is therefore assured. The concept of a population of points is also considered and it is shown that it can be used to locally increase the diversity of the population while still ensuring convergence and a method to extract the local directional information from the population is also described. The paper describes and introduces the basic theoretical concepts as well as demonstrating how they are used on simple test problems.

Keywords: Multi-objective optimization, Gradient updates, Descent, Diversity, Convergence.

1. INTRODUCTION

Multi-Objective Optimization (MOO) is a challenging problem in many disciplines, from product design to planning (Coello et. al., 2002, Deb and Jain, 2002, Deb, 2000) and Evolutionary Computation (EC) approaches to such problems have had many successes in recent years. In the realm of real-valued, single-objective optimization, recent results with EC algorithms that more explicitly exploit gradient information have shown distinct performance advantages (Deb et. al., 2002). However, as will be shown in this paper, the rationale employed in these EC algorithms must be adjusted for multi-objective EC. This paper provides a local, theoretical framework and some empirical evidence for these adjustments, when the objectives are suitably smooth.

A key aspect of this work is efficiency. Efficiency is defined in terms of minimizing the number of objectives' evaluations that must be performed in order to achieve satisfactory convergence. The inevitable trade-off is that there is an increase in the computation time necessary to design the next population. Typical applications where efficiency is important include situations where the designs/objectives are costly (time or money) to evaluate or ones where parameter adaptation is happening on-line and it is argued that in these cases, an increase in computation time for designing the next population is insignificant compared to the cost of evaluating poor design candidates. Previous hybrid techniques utilizing both gradient and EC methods have involved simple combinations of the two techniques (Lahanas et. al., 2003, Quagliarella and Vicini, 1997), for instance, using the gradient technique to seed the EC population, or using a simple weighted combination of objectives in a gradient improvement of individual population members. The theory described in this paper explicitly uses the directional information available in the EC population to determine the generation of "child" individuals.

This paper describes how evolutionary multi-objective optimization can efficiently utilize approximate, local directional (gradient) information. The local gradients associated with each point in the population can be combined to produce a multi-objective gradient (MOG) where all the objectives are increasing (at least non-decreasing). The MOG indicates whether the design is locally Pareto optimal, or if the design can be improved further by altering the parameters along the descent direction defined by the negative MOG. In addition, an orthogonal direction (in parameter or objective space) can be defined to this descent direction that can improve the diversity of the population. This allows a family of procedures to be defined that try and find the complete Pareto set/front or else span a restricted region, relative to some initial design (Laumanns et. al., 2002). In this paper, the term convergence is interpreted as meaning that a point, or set of points, will move towards the Pareto front. This can occur in either a descent or a diverse direction, as both types of updates can be considered to be converging towards to Pareto front.

The main problem associated with the conventional approach to steepest descent multi-objective optimization is the need to estimate the local gradient for each design. Therefore, viewing the problem from an EC perspective (where a population of designs is maintained) allows directional information to be obtained from neighbouring designs, thus lowering the number of design evaluations that must be performed. This paper presents theory on how this information should be used and describes its interpretation as an intelligent crossover operation. In describing the theory, insight is gained into the structure of the multi-objective problem by analyzing the geometry of the directional cones at different stages of learning. Reasons for the apparently rapid rate of initial convergence (but poor rate of final convergence) in typical multi-objective EC algorithms are also described.

The concepts described in this paper are only directly applicable to differentiable multi-objective design problems. However, a large number of complex shape and formulation optimization problems (Sobieszczanski-Sobieski and Haftka, 1997, Parmee et. al., 2000) are differentiable. Moreover, the insights and theory presented may aid the reasoning used in EC algorithm design on a broader class of problems.

2. GEOMETRY OF MULTI-OBJECTIVE OPTIMIZATION

In this section, the basic MOO problem is introduced. The concept of point-wise descent is discussed and is locally interpreted as finding an update that simultaneously reduces all objectives. This naturally leads onto investigating which local cones in parameter space produce multi-objective descent (dominating), diversity or ascent updates. The geometry of these cones is examined and also compared with multi-objective updates that lie in the negative gradient simplex, formed from the negative gradient vectors associated with the individual objectives. These geometrical concepts underpin the work in Section 3 that shows how the concepts of descent, diversity and convergence can be formulated for updating a single point, which are then extended to a population of points in Section 4.

2.1. Generic Multi-Objective Optimization Problem

Any MOO problem, whether the aim is to adapt a single point or a population of points, is defined by the design parameters, the objectives and the set of constraints. In this paper, the design parameters are represented by an n-dimensional, convex, continuous space $\mathbf{X}$. The objectives are represented by an m-dimensional, differentiable space $\mathbf{F}$. The constraints are specified by two vector functionals $\mathbf{g}()$ and $\mathbf{h}()$, that represent the equality and inequality constraints. This paper is not primarily concerned with the constraint set, although much of the theory can be easily extended to handle standard linear equality and inequality constraints.

The standard formulation for a multi-objective problem is of the form:

$$\begin{aligned} &\min \mathbf{f}(\mathbf{x}) \\ &st \quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \\ &\qquad \mathbf{h}(\mathbf{x}) \le \mathbf{0} \end{aligned} \tag{1}$$

and there are many design problems that can be represented in this way. Jointly minimizing each objective is, in general, not possible as they typically conflict close to optimal solutions. In fact, Pareto optimality is sometimes referred to as "zero sum optimality", as, at optimality, it is not possible to decrease an objective without increasing at least one other. The basic requirement of (1) is to jointly minimize **f**, which imposes a partial ordering on the set of potential solutions.

A design $\mathbf{x}_1$ **dominates** another design $\mathbf{x}_2$ when $\mathbf{f}(\mathbf{x}_1) \le \mathbf{f}(\mathbf{x}_2)$ and the vector comparison is taken element wise. Strictly speaking, there should be at least one objective that is strictly less in value. This can be written as $\mathbf{x}_1 \le_{\mathbf{f}} \mathbf{x}_2$, and if a sequence of points are generated such that $\mathbf{x}_{k+1} \le_{\mathbf{f}} \mathbf{x}_k$, it is said to be a **descending sequence**. When a design is globally Pareto optimal, there does not exist any other point that dominates it, as illustrated by the red curves in **Figure 1**. The set of all such points is the solution to (1).

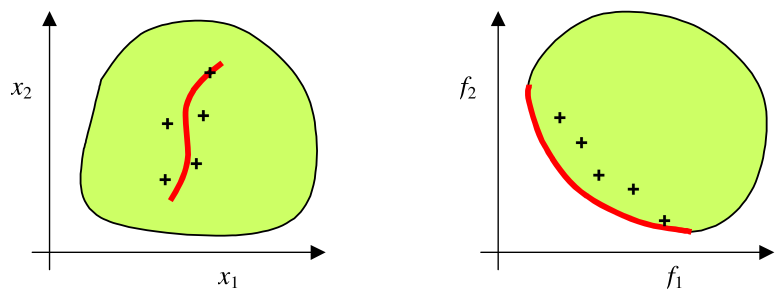

Figure 1. The set of Pareto optimal points (red curves) in parameter space (left) and objective space (right) and a population of approximate solutions.

A point is locally Pareto optimal if the point is not dominated by another point in a local neighbourhood. The set of all such locally Pareto optimal points is a superset of the set of globally Pareto optimal points. This paper is largely concerned with calculating locally Pareto optimal points, and then the globally Pareto optimal points can be found by performing the dominance test on the complete set of locally optimal points.

When **f**() is differentiable, the solution to (1) is, in general, a set of discontinuous curves. If it is required to calculate this, potentially infinite, set of optimal values, typically a population of points is used to approximately sample the curves and the population is iteratively adapted to produce "better solutions". The new population should be a better approximation to the unknown Pareto set/front than the old one, in some sense, and if $\mathbf{X}_k$ represents the (p*n) population of p designs at the k^{th} iteration, then it is required that:

$$\mathbf{X}_{k+1} \le_{\mathbf{f}} \mathbf{X}_k \tag{2}$$

where the definition of the population operator $\le_{\mathbf{f}}$, is left vague at this point and simply represents some measure of convergence.

2.2. Single-Objective Half-Spaces

For a single objective, f, the concept of dominance is simply that the objective value associated with one point should be strictly less than the dominated point. Assuming that the objective is smooth, the contour associated with a particular point therefore divides the space into two regions, one containing the area dominated by that point, and the other containing the area that dominates it, as illustrated in Figure 2. Using a first order, Taylor series approximation, this is represented by contour's tangent at that point, where the tangent divides the space into two half-spaces and is perpendicular to the gradient. Local updates to the point either lie in the negative half-space (reduce the objective), or in the positive half-space (increase the objective), and if an update direction is selected at random, both are equally likely.

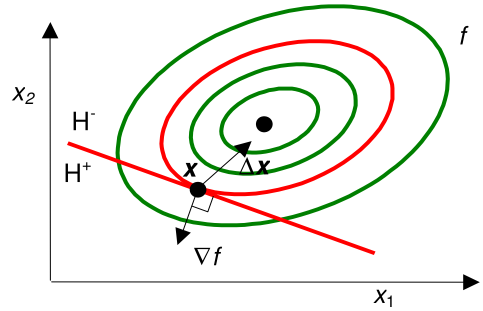

Figure 2. An interpretation of global (red circular contour) and local (red line) dominance for a single objective and a single point x.

2.3. Multi-Objective Half-Spaces

To extend this concept to a multi-objective setting, it is possible to consider the intersection of each objective's half-spaces in parameter space. This partitions the parameter space into, at most, 2^m mutually exclusive, directional cones, one of which represents the simultaneous decrease in all the objectives (**descent** cone) and another represents the simultaneous increase in all the objectives (**ascent** cone). The remaining subset of 2^m-2 cones all decrease at least one objective, while simultaneously increasing another, and may be regarded as **diversity** cones. This is illustrated in Figure 3 for a two parameter, two objective problem, where each cone is labeled by the qualitative change (increase, +, and decrease, -) in the set of objectives and the half-spaces for the j^{th} objective are denoted by $\mathbf{H}_j^-$ and $\mathbf{H}_j^+$. It should be stressed that only one of these cones contains undesirable updates (ascent cone), therefore selecting an update direction is ill-posed in this framework.

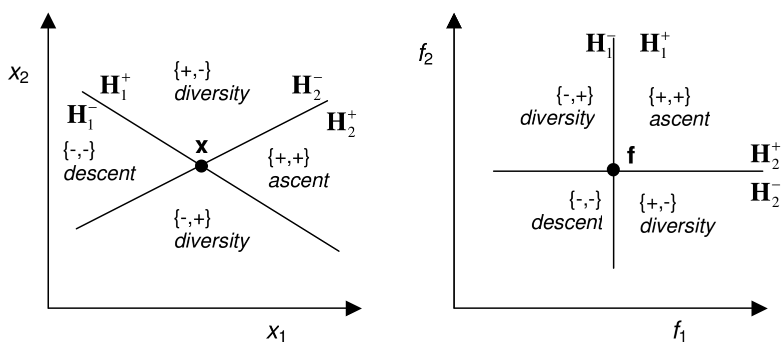

Figure 3. The set of local cones, for a design {x,f} which partition parameter and objective space. Each cone is labelled by the qualitative/sign change in the objectives.

Even though there are 2^m possible cones, only some types of objective changes can be realized at each point. An obvious example of this is when the point is locally Pareto optimal and the descent and ascent cones both collapse to zero.

2.3.1 Descent Cone

For the multi-objective space, any search direction that lies in the negative half-space of all the objectives will simultaneously minimize them, and the search direction will be "aligned" with the negative gradients associated with each objective, as in the single objective case. This is known as the **descent cone** and is illustrated in Figure 3 and Figure 4. By definition, a new local point that lies in this cone will **dominate** the existing point, as all the objectives are reduced.

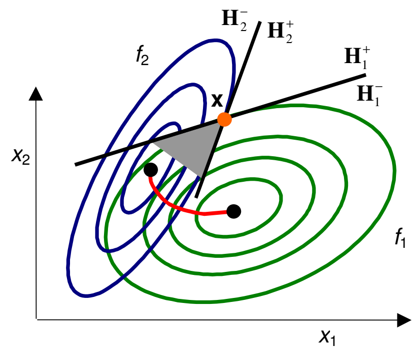

Figure 4. The descent cone (shaded area) for a 2 parameter, 2 objective optimization problem. The optimal Pareto set is shown as the red curve.

It is also useful to consider the size of the descent cone during initial and final stages of optimization process, as illustrated in Figure 5. When a point is far from the local optima, the objectives' gradients are aligned and the descent cone is almost equal to the half-spaces associated with each objective. Therefore, if the search directions are randomly chosen, there is a 50% chance that a search direction will simultaneously reduce all the objectives (as happens with a single objective), although this may be slow if the search direction is nearly orthogonal to the aligned negative gradients. However, when a point is close to the Pareto set, the individual gradients are almost contradictory, the size of the descent cone is extremely narrow and there is a small probability that a random update will lie in this descent cone.

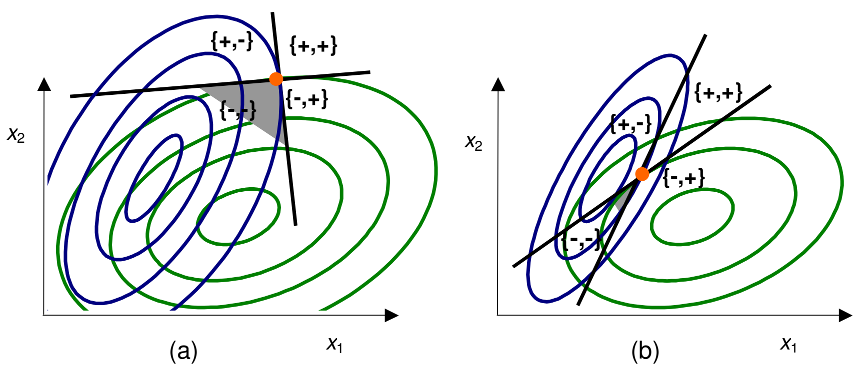

Figure 5. The descent cone (shaded, {-,-}) for a 2 parameter, 2 objective design problem during initial (a) and final (b) stages of convergence. The descent cone shrinks to zero during the final stages of convergence, and the corresponding diversity cones cover nearly all of the parameter space.

2.3.2. Diversity Cones

In some situations it may be preferable to extend the diversity of the population while still converging towards the Pareto set/front. The (non-empty) descent cone is surrounded by at most m cones, where only one of the objectives increases and all of the remaining objectives decrease. This is illustrated in **Figure 2**, where the descent cone is surrounded by 2 diversity cones, where only one of the objective's increases. If $n<m$, or if the objectives are linearly dependent, it may not be possible to increase some of the objectives individually, and this information is contained in the non-singular eigenvectors of the local Jacobian matrix.

2.4. Negative Gradient Simplex

Just as the objectives' tangents locally specify the descent and diversity cones, the gradients that are perpendicular to the tangents can be dynamically combined to form the negative gradient simplex:

$$\mathbf{V} = -\sum_i \lambda_i \mathbf{g}_i \quad st \quad \boldsymbol{\lambda} \geq \mathbf{0} \quad \sum_i \lambda_i = 1 \tag{3}$$

where $\mathbf{g}_i$ denotes the i^{th} objective's gradient. During the initial stages of learning, the individual (negative) gradients are likely to be highly correlated, and any descent direction that lies in this negative gradient simplex will also lie in the descent cone. However, as a point moves closer to the optimal, local Pareto set/front the gradients become increasingly conflicting and only a small percentage of the directions that lie in the negative gradient simplex also correspond to descent directions, as illustrated in Figure 6.

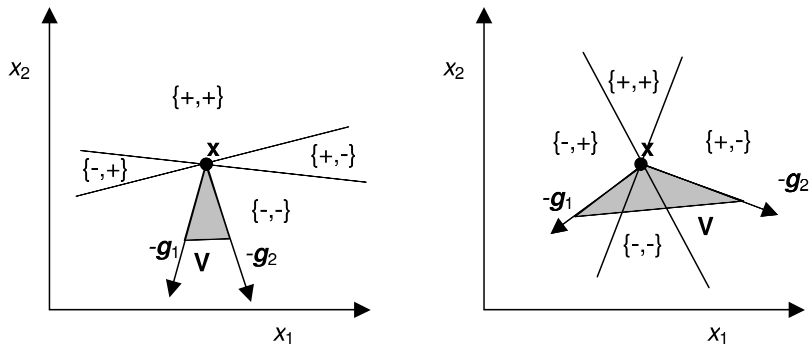

Figure 6. The gradient cone during initial stages (left) and final stages (right) of learning.

During initial learning, the (negative) gradients are aligned and any update in the cone will always correspond to a descent direction that reduces all objectives. During final stages, the gradients are increasing in opposite directions, and some updates would increase at least one objective.

For a sufficiently close starting point, minimising the individual objectives will mean that a point will follow a path to the vertices of the local Pareto set/front and following any other gradient path, that lies in the interior of the gradient cone will converge to some point in the interior of the local Pareto set/front. These may not be dominant updates as they may represent diverse updates. However, any first-order, gradient update should lie in the negative gradient cone. This is illustrated in Figure 7 and further considered in Section 3.1.

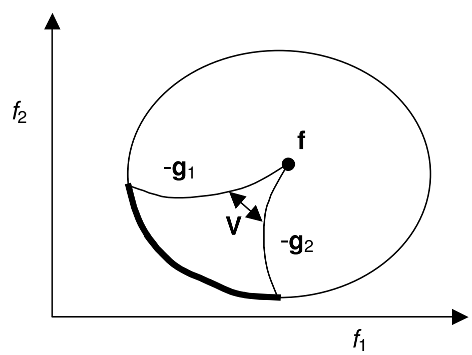

Figure 7. An illustration of how the individual gradients project a point onto the local Pareto front and any update that lies in the negative gradient cone will converge to the local Pareto front.

3. MULTI-OBJECTIVE STEEPEST DESCENT AND DIVERSITY

In this section, the concept of point-wise MOO is discussed, where point-wise optimization refers to iterating a single point towards the optimal, local Pareto set. The problem of calculating a locally optimal multi-objective descent direction is discussed and it is shown how to perform the calculation in either parameter (primal) or objective (dual) space. This generalizes to considering updates that move in diverse directions and a discussion about how to measure local optimality and convergence, building on the geometrical concepts introduced in Section 3.

3.1. Multi-Objective Descent

When the aim is to simultaneously improve a set of objectives, a natural question to ask is how to calculate an update that lies in the corresponding descent cone, as described in Section 2.3. This is an ill-posed problem as there are many such updates. The calculation can be made unique (Fliege, 2000) by requiring that the individual objectives' reductions are maximal by minimizing the largest change across the set of objectives:

$$
\begin{aligned}
&(\alpha^*, \mathbf{s}^*) = \arg\min_{\alpha,\mathbf{s}} \alpha + \tfrac{1}{2}\|\mathbf{s}\|_2^2 \\
&\text{st} \qquad \mathbf{J}^T\mathbf{s} \le \mathbf{1}\alpha
\end{aligned}
\tag{4}
$$

where **s** is the parameter update direction, α^* represents the smallest reduction in the objectives and **J** is the local Jacobian matrix defined by:

$$
\mathbf{J} = \left[\frac{\partial f_j}{\partial x_i}\right]
\tag{5}
$$

This is the primal form of the Quadratic Programming (QP) problem in (n+1) dimensions with m linear inequality constraints. It requires as large a (local) reduction in the objectives as possible for a fixed sized parameter update. It can easily be shown that $\alpha^* \le 0$, simply by considering a zero length update, and when it is strictly negative, this corresponds to a descent update. When α^* is zero, this corresponds to a point being locally Pareto optimal. When all the linear inequality constraints are active, the primal form of the multi-objective optimization problem reduces each objective by the same amount and thus the current point is locally projected towards the Pareto front at 45^o in objective space (assuming that the objectives have been scaled to a common range). Finally, it should be stressed that the descent update calculation is independent of whether the Pareto front is convex or concave.

Using Lagrangian theory, (4) can be represented in its dual form (Fliege, 2000):

$$
\begin{aligned}
&\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \tfrac{1}{2}\|\mathbf{J}\boldsymbol{\lambda}\|_2^2 \\
&st \quad \boldsymbol{\lambda} \ge \mathbf{0} \\
&\textstyle\sum_j \lambda_j = 1
\end{aligned}
\tag{6}
$$

where $\boldsymbol{\lambda}$ is an m-dimensional vector of Lagrange multipliers. . This is now a QP problem in m variables with m linear inequality constraints and 1 equality constraint. When $m < n$, solving the dual QP problem will be more efficient, and $\mathbf{s}^*$ and $\boldsymbol{\lambda}^*$ are related by:

$$
\mathbf{s}^* = -\mathbf{J}\boldsymbol{\lambda}^*
\tag{7}
$$

This (search) direction as termed the (negative) multi-objective gradient (MOG), **d**, or steepest descent direction. It is calculated from a non-negative linear combination of the individual gradients., although the dynamic Lagrange multipliers, $\boldsymbol{\lambda}$, are a function of **x**. Therefore, the descent direction is "aligned" with the individual negative gradients and lies in the negative gradient simplex, see Figure 6. The link with weighted optimization should also be noted, but it should be stressed that this procedure is valid for both convex and concave Pareto fronts, see

Section 5, as the Lagrange multipliers are dynamic. Finally, estimating the local Jacobian at each point is further considered in Section 4.

A final point worth noting, is that soft preference information can be incorporated into the QP problem, by modifying the primal problem as:

$$\begin{aligned}&(\alpha^*, \mathbf{s}^*) = \arg\min_{\alpha,\mathbf{s}} \alpha + \tfrac{1}{2}\|\mathbf{s}\|_2^2 \\ &\text{st} \qquad \mathbf{J}^T\mathbf{s} \le \mathbf{w}\alpha\end{aligned} \tag{8}$$

where $\mathbf{w}$ is an m-dimensional vector that represents the relative importance of each objective. As with the standard QP problem, the calculated update won't necessarily reduce each objective by these relative amounts, but will attempt to. This can also be simply translated into a dual form.

3.2. Multi-Objective Diversity

The (negative) MOG maximally reduces the individual objectives according to the primal MOO problem (4). Geometrically, this is interpreted as choosing the smallest update that lies in the negative gradient simplex, **V**. Therefore, **d** is **perpendicular** to **V**, when all the Lagrange multipliers are non-zero and if one or more Lagrange multipliers are zero, the negative MOG descent direction is perpendicular to **V** in the corresponding sub-space. However, the aim may be to perform a partial trade-off, where one or more objectives increase, while one or more objectives decrease. This is particularly important if a point is being updated, relative to an existing population that forms a local estimate of the Pareto set/front. The aim would then be to drive the population towards the Pareto set/front, while also improving the diversity of the population. Therefore, the concept of multi-objective diversity needs to be developed, relative to the concept of descent.

As illustrated in Figure 7, performing gradient descent on any of the objectives individually will produce an update path, through parameter/objective space, that iterates towards the corresponding minimum. Similarly, there exists a local, dynamic scalarized objective that will be reduced for any update that lies in the negative gradient simplex, **V**. In addition, the negative gradient simplex **V** spans the space of update directions towards the complete, local Pareto set and any update in **V** dominates any other potential update direction that does not lie in **V**. Therefore, in a local, first order sense the negative gradient simplex **V** contains the optimal (non-dominated) set of potential updates that can drive a point to anywhere on the local, optimal Pareto set, whether moving in either a descent or a diverse direction. This is illustrated in Figure 8, where **V** lies completely in the descent cone, but as also illustrated in Figure 6, **V** may also contain diverse update directions, especially when the points are close to the optimal Pareto set.

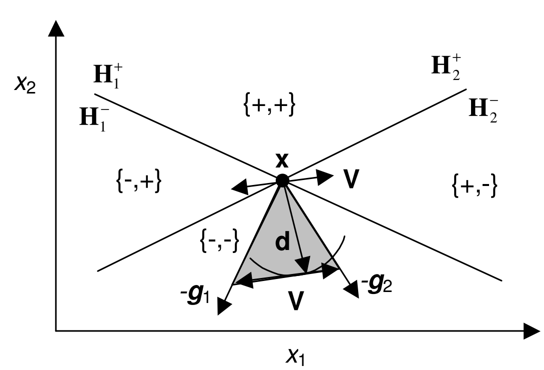

Figure 8. The diversity plane V is simply the simplex formed from the individual gradients. The dominant descent direction, d, by definition, lies in this simplex.

To actually use this locally optimal diversity sub-space in order to calculate an update direction, an update direction should be calculated of the form

$$\mathbf{s} = \mathbf{d} + \mu \mathbf{v} \tag{9}$$

where $\mathbf{v} \in \mathbf{V}$ and $\mathbf{d} \perp \mathbf{v}$. To actually calculate how the orthogonal descent and diversity components should be mixed, it is necessary to consider what the original goal of the MOO problem is, and this will be further developed in Section 4, where the concept of jointly optimizing a set of points will be considered. It should be stressed that the actual parameter update will always lie in the negative gradient simplex, so convergence to the local Pareto set/front is assured.

3.3. Multi-Objective Optimality

This interpretation of MOO descent, described in Section 3.1, is appropriate, as long as the design is not locally Pareto optimal (i.e., as long as there exists a descent cone that will simultaneously reduce all the objectives). To test whether a design is locally Pareto optimal (Fliege, 2000)(Miettinen, 1999) is an important part of any design process and this can be formulated as:

$$\boldsymbol{\lambda} \in \mathrm{N}(\mathbf{J}(\mathbf{x}))$$

for any non-zero vector $\boldsymbol{\lambda} \geq \mathbf{0}$, where $\mathrm{N}(\mathbf{J})$ is the null space of the Jacobian, $\mathbf{J}$. Equivalently:

$$\mathbf{J}(\mathbf{x})\boldsymbol{\lambda} = \mathbf{0} \tag{10}$$

This is a simple test to perform, because it represents the zero solution to both (4) and (6), meaning that there does not exist a descent update. The geometric interpretation of this test in objective space (shown in Figure 9) is that there exists a non-negative combination of the individual gradients that produces an identically zero vector. The descent cone is empty and any changes to the design parameters will affect only the Jacobian's range, $R(\mathbf{J}^T)$, which is orthogonal to $\boldsymbol{\lambda}$. This is the limiting case of the situation described in Section 3.1, during the final stages of convergence, when the gradients become aligned, but in the opposite direction. When the alignment is perfect (local Pareto optimality), any change to the design parameters will increase at least one of the objectives. In fact, for an optimal design, $R(\mathbf{J}^T)$ defines the local tangent to the Pareto front and thus defines a local basis for the space that must be locally sampled in order to generate the local Pareto set/front. It is worthwhile noting that the diversity sub-space still exists and that $\mathbf{V}$ should coincide with $R(\mathbf{J}^T)$.

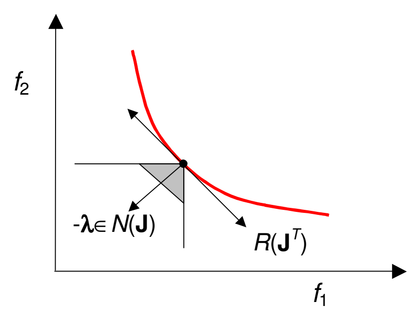

Figure 9. The geometrical interpretation of the Null space and Range of the Jacobian matrix when a design is Pareto optimal. $\boldsymbol{\lambda}$ specifies the local normal to the Pareto front.

4. POPULATION-BASED MUTLI-OBJECTIVE OPTIMIZATION

The multi-objective concepts described in Section 3 were explicitly addressing issues associated with updating a single point towards the local, optimal Pareto set. However, they are also relevant for EC or population-based optimization algorithms where the aim is to evolve a set of points towards the optimal Pareto set, so that they approximately sample it. The work described in this section considers how to calculate diverse updates for each population member. It also discusses how neighbouring points in the population can be used to estimate/approximate the local Jacobian. This idea is fundamental to the concept of efficient population-based MOO, as the aim is to construct a batch of designs that will provide information about how to maximally improve the next batch of designs.

4.1. Population-based Diversity

The basic aim of a population-based, MOO algorithm is to evolve the population so that at each step, the population moves closer to the Pareto set (see Equation 2) and also the population forms a uniform sample across the complete Pareto set. The first point could be addressed by simply requiring that all the points form a descending/dominating sequence. However, this would not generally result in a uniform sample across the complete Pareto set. The diversity of the population can be addressed by considering the position of each point, relative to its neighbours, and considering how the local diversity of each point can be improved. In Sections 2.4 and 3.2, it was argued that, for each point, updates lying in the negative gradient simplex are optimal because it spans the set of all convergent updates and is optimal because these updates dominate any other potential updates, in a first order sense. Therefore, for each point in a population, the negative gradient simplex forms a constrained sub-space and the aim is to make the update calculation well-posed by imposing additional criteria related to the population.

Consider the two situations shown in Figure 10, where the local neighbours of a point have been projected down onto its negative gradient simplex. If the point lies in the interior of the projected neighbours, then the diversity of the population can be locally improved by calculating an update such that the search direction, **s**, lies at the centre of the projected simplex, equidistant from the neighbouring points. This will ensure the population's points become more uniformly spaced. Alternatively, if the point lies outside of the projected simplex of neighbouring points, the population's diversity can be improved by choosing a search direction that lies on the nearest edge of the negative gradient simplex. This will increase the range of the population, by forcing points to move towards the edges/vertices of the Pareto set.

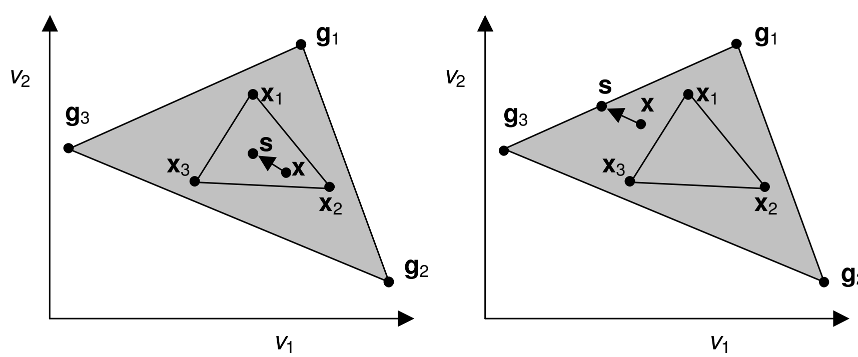

Figure 10. Locally improving the population diversity for a point *x*. When it lies in the interior of its local neighbourhood, the diversity is improved by distributing evenly (left). When it lies outside the neighbourhood, the diversity is improved by extending the range.

Such a scheme results in an approximate ordering of the points when the population is evolved. Points at the edge concentrate on one, or a sub-set of, objectives and they remain on the boundaries of the Pareto set, as illustrated in Figure 7.

4.2. Population-based Jacobian Estimation

Crossover is an important operation in many EC algorithms, and there are a wide variety of different schemes for implementing it. For real-valued parameter spaces, probably the most common implementation of crossover is by randomly selecting the child to lie along the extended line that connects the two parents. As shown in Figure 11, this scheme works well when one parent dominates the other, as the extended line segment includes a portion that should locally dominate either parent (however, it also includes another segment that is dominated by both parents). This situation may occur frequently during initial stages of MOO, before selection has significantly reduced the number of dominated points. When the majority of points lie in diverse quadrants, the extended segments will tend to produce diverse, rather than dominating, updates. This is also illustrated in Figure 11. Instead of simply considering two parents, there may be some benefit in looking at the local neighbourhood of a point and investigating how new update directions can be calculated such that the child point lies in the space spanned by its parents.

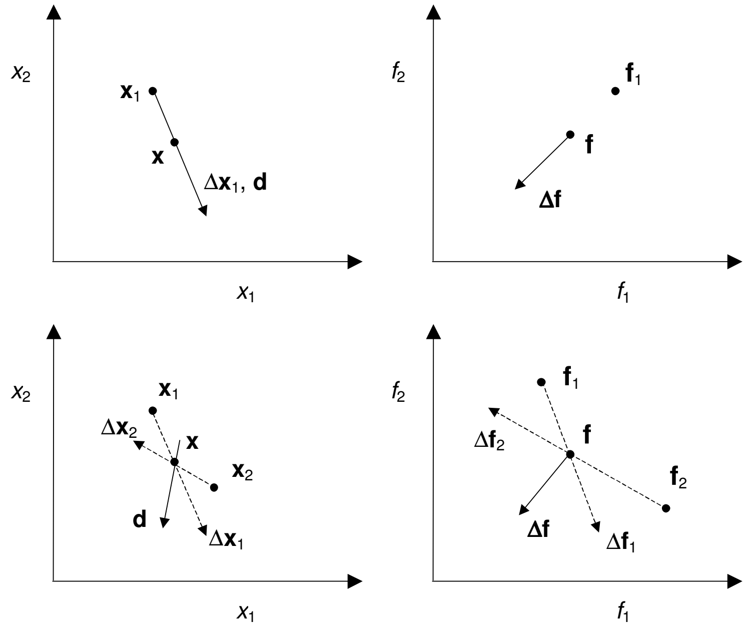

Figure 11. Calculating descent directions for a point, x, when the local neighbourhood consists of a single dominated point (top) and two non-dominated points (bottom).

This can be achieved by using the geometrical insights obtained in Sections 2 and 3, but using the local neighbourhood of points to approximate the local Jacobian, for each point. As illustrated in Figure 11, the difference information between neighbouring points can be used to provide directional information along a particular search direction, and this can be combined to approximate the local Jacobian as follows.

The minimum number of (linearly independent) neighbouring points that need to be considered is $r = \min\{m,n\}$. This is because the rank of the Jacobian is at most r. In some cases, it may be possible to use less than r pieces of directional information in order to calculate a descent direction and in other cases, the gradient estimates may be improved by using more than r neighbouring points. So consider when the difference between each neighbouring point and the point of interest is defined by:

$$\Delta\mathbf{x}_i = \mathbf{x} - \mathbf{x}_i$$
$$\Delta\mathbf{f}_i = \mathbf{f} - \mathbf{f}_i$$

Then the local difference matrices are defined by:

$$\Delta\mathbf{X} = [\Delta\mathbf{x}_1,\ \Delta\mathbf{x}_{2,\ \ldots,}\Delta\mathbf{x}_r]^T$$
$$\Delta\mathbf{F} = [\Delta\mathbf{f}_1,\ \Delta\mathbf{f}_{2,\ \ldots,}\Delta\mathbf{f}_r]^T$$

and the local Jacobian can be approximated by:

$$\mathbf{J} = \left(\Delta\mathbf{X}^T\Delta\mathbf{X}\right)^{-1}\Delta\mathbf{X}^T\Delta\mathbf{F} \tag{11}$$

Therefore, the difference between the point of interest and each member of the local neighbourhood is used to estimate gradient information. This relies on the local neighbourhood being sufficiently small so that the gradients can be estimated sufficiently accurately. In addition, when **J** is rank deficient, an inverse in the non-singular subspace is used.

In many ways, the descent calculation, Equations (4) and (6), can be regarded as an intelligent local crossover operation that combines the information in neighbouring points to calculate a descent direction. It illustrates why convergence may initially be fast when points may dominate its neighbours and also explains why performing a simple linear crossover may be insufficient to drive convergence, especially when the points are close to the Pareto set/front. It should also be noted that in the single objective case, this amounts to a simple Euler approximation of the derivative. It should be noted that while the calculated descent update lies in the space spanned by **x** and its local neighbours, it is not a simple linear combination of these points. Therefore, when static, linear crossover operators are used in EC algorithms, the performance may be sub-optimal in many situations.

4.3. Directed Multi-Objective Optimization Algorithm

These elements can be combined to produce a directed MOO algorithm that works according to:

1. Randomly select the initial population's members in a small area of the parameter space, using prior knowledge as appropriate.
2. For each point in the population, calculate the approximate local Jacobian as described in (11)
3. If the MOG is sufficiently large (i.e. it has not converged), calculate the negative gradient simplex and the update direction, as described in Section 4.1, so that it locally moves towards the optimal Pareto set, while improving the population's diversity.
4. Continue until all the points in the population have locally converged.

The main advantage of such an approach is that the directional information can give rapid convergence when the MOO problem is reasonably well conditioned. The rate of convergence is also predictable, given certain smoothness assumptions about the objectives. In addition, using the local population to approximate the local Jacobian for each point means that the procedure is efficient, in the sense that the number of objectives' evaluations is minimized, although it is acknowledged that a more computationally costly procedure is being used to calculate each point-wise update.

The main disadvantage of such an approach is that it is inherently local. Convergence to a local Pareto set can be guaranteed, but unless some form of re-starting or initializing the population with local islands of points is used, there will be little confidence that the calculated local optimal set is also the global set. In practice, a coarse exploration of the parameter space may be performed to identify areas of interest. In addition, this may not be such a problem for some industrial design problems, where previous work/products may have already identified area(s) of interest. Other problems may occur in looking at the robustness of such approach, especially using the local population to approximate the local Jacobian. However, this is outside the scope of the current paper.

There are many variations on the basic algorithm just described. Instead of having a fixed sized local neighbourhood, this could be dynamic and only use enough points to calculate a dominating direction. Similarly, initial optimization may just concentrate on a sub-space in either parameter/objective space and then refine from there. Other refinements could include an element of "exploration" in the update calculation, where specific directions are randomly searched, if there is insufficient information in the population about how they affect the solution's quality. Also, while the diversity calculation mentioned in Section 4.1 will inherently produce a uniform sample in population space, it may be desirable to have a uniform sample in objective space. Finally, the issue of calculating the step size along the calculated update direction has not been addressed, but in Section 5, a pre-determined step size is used. All of these points are subject to current investigation and will be addressed in future work.

5. EXAMPLES

The use of directional information for multi-objective optimization will now be demonstrated on two test problems. It should be noted that the aim is not to provide a full and rigorous comparison with other approaches, as a complete algorithm has not been described in this paper. Rather the aim is to demonstrate how the theory provides the basis for such an approach, and to give an indication of the power of using directional information.

Consider the following multi-objective design problem that has two objectives and two design parameters, as illustrated in Figure 12. The objectives are specified by:

$$f_1(\mathbf{x}) = (x_1 - 1.5)^2 + (x_2 + 1.5)^2$$
$$f_2(\mathbf{x}) = (x_1 + 1.5)^2 + (x_2 - 1.5)^2$$

so the Pareto set is a straight line between the points [1.5, -1.5] and [-1.5, 1.5] and the Pareto front is a single convex curve between the points [0, 9] and [9, 0], respectively. A population of size 7 was used and the initial population was randomly distributed around [6, 6], as shown in Figure 12. While this initial population is quite different from normal EC algorithms, it should be emphasized that the aim here is to exploit local, directional information. Five iterations of the descent and diversity calculation, using a fixed step size of 0.25, were performed and it can be clearly seen that the designs rapidly move towards the Pareto set/front, searching only those parts of the parameter space where the directional information indicates that convergence will occur. The points at the edge of the population converge towards the minima of the individual objectives and the points in the interior of the population descent towards the local Pareto set/front while increasing their diversity with respect to each other.

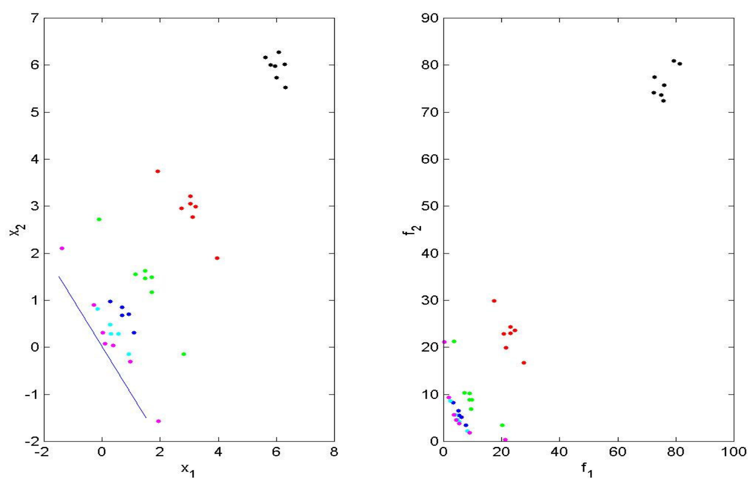

Figure 12. Using directional information on a simple 2 parameter, 2 objective design problem. The 5 iterations are plotted and denoted by different colours on the graph.

It is readily acknowledged that this is a very simple evaluation problem. There exists a single global optimal Pareto set/front and the condition of each objective function is very good. It has also been assumed that a suitable step size has been determined before learning commences. However, even with these reservations, it can be clearly seen that convergence is rapid and that descent and diversity updates are both forcing the population of points to efficiently converge to the Pareto set/front.

Another multi-objective optimization problem that has been investigated by researchers (Fonseca and Flemming, 1995), is the two parameter, two objective system described by:

$$f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{2} \left(x_i - 1/\sqrt{2} \right)^2 \right)$$
$$f_2(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{2} \left(x_i + 1/\sqrt{2} \right)^2 \right)$$

This is an unconstrained optimization problem, although range constraints of the form $-4 \le x_i \le 4$ have been used. The main difference from the last problem is that the Pareto front is now concave and so normal weighted optimization would only find the two extreme designs. The Pareto set for this problem lies between the points and the relationship between the two functions on the Pareto front is described by the concave curve:

$$f_2 = 1 - \exp\left(-\left(2 - \sqrt{-\ln(1 - f_1)}\right)^2\right)$$

The initial population of 7 designs was randomly distributed around the point [1,-1] as illustrated in Figure 13, and the first three iterations as well as the 12^{th} are also shown in this figure. A fixed step size of 1 was used to update the designs along the descent/diversity directions by the algorithm. As can be seen from Figure 13, initial convergence is slower than the previous example, although satisfactory convergence has occurred after approximately a dozen iterations. The slower initial convergence is due to the large change in curvature as a design moves away from the Pareto set. Having an adaptive step size could reduce this effect, but as previously mentioned, the step size is fixed in this work. In addition, it should be noted that while the convergence towards the Pareto front appears smooth, the convergence towards the Pareto set is less smooth. This is due to the multi-objective gradient approaching zero near the Pareto set.

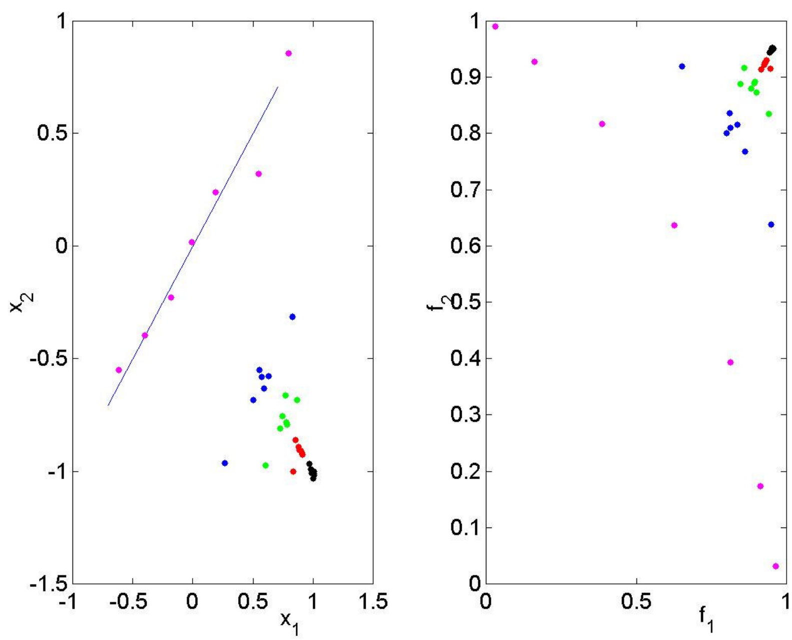

Figure 13. The parameter (left) and the objective (right) spaces for the concave FON optimization problem. The first three iterations are shown in red, green and blue and the magenta designs represent the 12^{th} iteration.

6. CONCLUSIONS AND FURTHER WORK

The central issue addressed by this paper is how to efficiently re-use the local directional information contained within a population for population-based, multi-objective design problems. The aim has been to minimize the number of objective evaluations, by performing more complex/intelligent calculations that produce updates in an appropriate descent or diverse direction. The theory developed shows how population-based descent and diverse updates can be calculated and how they should both lie in the negative gradient cone to ensure convergence to the local Pareto set/front. The theory clarifies how to logically use such information in multi-objective EC algorithms and explains why initial convergence may appear rapid, but during latter stages, updates due to the crossover rarely produce dominating search directions. While it is readily acknowledged that the theory can only be applied to smooth problems, it may be possible to relax some of the conditions and extend the concepts to a wider class of problems.

Subjects for further investigation clearly include assessing how using approximate derivative information affects convergence, the robustness of the procedures, assessing different methods for calculating diversity updates, automatically calculating appropriate step sizes and incorporating parameter and range constraints. In addition, it will be interesting to examine if the concepts can be extended using second-order directional information to improve the convergence on stiff systems.

REFERENCES

Coello, C.A., Van Veldhuizen, D.A. and Lamont, G.B., (2002) *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers.

Deb, K. and Jain, S. (2002). Multi-Speed Gearbox Design Using Multi-Objective Evolutionary Algorithms, *Journal of Mechanical Design*, **125**(3), 609-619.

Deb, K. (2000). *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley.

Deb, K., Anand, A., and Joshi, D (2002). A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization, *Evolutionary Computation*, **10**(4), 371-395.

Fliege, J. (2000) Steepest Descent Methods for Multicriteria Optimization. *Mathematical Methods of Operations Research*, **51**(3), 479-494.

Fonseca, C.M. and Flemming, P.J. (1995). An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computing Journal* **3**(1), 1-16.

Laumanns, M., Thiele, L., Deb, K. and Zitzler, E. (2002) Combining Convergence and Diversity in Multi-Objective Optimization, *Evolutionary Computation* **10**(3), 263-282.

Lahanas, M., Baltas, D. and Zamboglou, N. (2003) A Hybrid Evolutionary Multiobjective Algorithm for Anatomy-Based Dose Optimization Algorithm in HDR Brachytherapy. *Phys. Med. Biol.* **48**, 399-415.

Miettinen, K. (1999) *Nonlinear Multiobjective Optimization*, Boston, Kluwer.

Parmee, I. C., Watson, A., Cvetkovic, D. and Bonham, C. (2000). Multi-objective Satisfaction within an Interactive Evolutionary Design Environment. *Evolutionary Computation*. **8**(2), 197-222.

Sobieszczanski-Sobieski, J. and Haftka, R.T. (1997). Multidisciplinary aerospace design optimization: survey of recent developments. *Structural Optimization*, **14**, 1-23.

Quagliarella, D. and Vicini, A. (1997). Combining genetic algorithms and gradient based optimization techniques. In J Periaux et al (eds.) *Genetic Algorithms in Engineering and Computer Science*, 2nd edition, Wiley.