Effective Use of Directional Information in Multi-objective Evolutionary Computation

Martin Brown[1] and Robert E. Smith[2]

[1]Department of Computing and Mathematics
Manchester Metropolitan University, Manchester, UK,
m.brown@mmu.ac.uk
[2]The Intelligent Computer Systems Centre
The University of The West of England, Bristol, UK.
robert.smith@uwe.ac.uk

Abstract. While genetically inspired approaches to multi-objective optimization have many advantages over conventional approaches, they do not explicitly exploit directional/gradient information. This paper describes how steepest-descent, multi-objective optimization theory can be combined with EC concepts to produce improved algorithms. It shows how approximate directional information can be efficiently extracted from parent individuals, and how a multi-objective gradient can be calculated, such that children individuals can be placed in appropriate, **dominating** search directions. The paper describes and introduces the basic theoretical concepts as well as demonstrating some of the concepts on a simple test problem.

1 Introduction

Multi-objective optimization is a challenging problem in many disciplines, from product design to planning [2][3][7][9][10]. Evolutionary computation (EC) approaches to multi-objective 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 [4]. 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 theoretical framework and some empirical evidence for these adjustments.

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). The MOG indicates whether the design is locally Pareto optimal, or if the design can be improved further by altering the parameters along the direction defined by the negative MOG.

The main problem associated with the conventional approach to steepest-descent optimization is the need to estimate the local gradient for each design at each iteration. Therefore, viewing the problem from an EC perspective (where a population of de-

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 778–789, 2003.

signs is maintained at every iteration) allows the directional information to be obtained from neighboring samples (mates), thus lowering the number of design evaluations that must be performed. This paper presents theory on how this information should be used. 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.

2 Directional Multi-objective Optimization

In recent years, there have been a number of advances in steepest descent-type algorithms applied to differentiable, multi-objective optimization problems [1][5]. While they suffer from the same disadvantages as their single objective counterparts (slow final convergence, convergence to local minima), they possess both an explicit test for convergence and rapid initial convergence both of which are desirable properties in many practical design problems. This section reviews the basic concepts of these gradient-based, multi-objective algorithms, describing how to calculate a multi-objective gradient, how it can be used to test for optimality and how it can be used to produce a dominating search direction. In addition, insights are given into the structure of the multi-objective EC optimization problem during initial and final convergence, and reasons for the change in the convergence rate are provided. It should be acknowledged that 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 [7][8] are differentiable. Moreover, the theory presented may aid the reasoning used in EC algorithm design on a broader class of problems.

2.1 Single-Objective Half-Spaces

One way to generalize the conventional, single-objective steepest descent algorithms to a multi-objective setting is by considering which search directions simultaneously minimize each objective. For any single objective (for instance, the j^{th} objective, f_j,) a search direction will reduce the objective's value if it lies in the corresponding negative half-space, H^-, whose normal vector is the negative gradient vector, as illustrated in Fig. 1. This fact is exploited when second-order, single-objective optimization algorithms are derived, because (as long as the Hessian is positive definite) the search direction will lie in the negative half-space. Therefore, the objective function will decrease in value when points are moved into this half space. It is interesting to note that this concept of a half-space is independent of the objective function's form and does not depend on whether the point is close to the local minima or not. It simply states that a small step in any direction will either increase or decrease the objective.

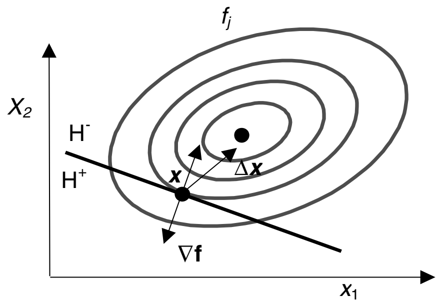

Fig. 1. The half-spaces defined for single-objective f_j. The objective's contours are illustrated as well as the gradient for the current design x.

Also, although one must appeal to probabilistic notions to do so, this idea can be related to modern, real-valued EC, typified by [4]. In such algorithms one can consider those population members selected to survive and recombine to be on one side of an approximate half-space division in the search space, and those deleted from the population without recombination to be on the other side of this division. Note that in the high-performance real-valued EC algorithm introduced in [4], the new individuals generated by GA operators are biased to lie near the selected "parents", thus enforcing the idea of exploiting the preferred side of this approximate half-space.

2.2 Directional Cones and Multi-objective Search

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. This is illustrated in Fig. 2.

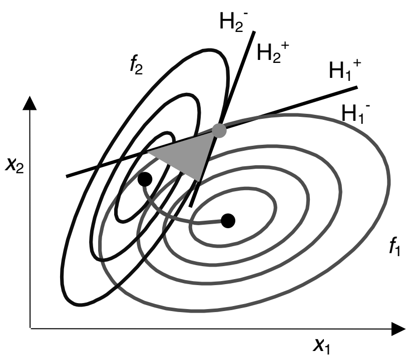

Fig. 2. Directional cones for a 2 variable, 2 objective optimization problem. The Pareto set is the curve between dotted centers, and the directional cone that simultaneously minimizes both objectives is shaded gray.

This region is known as a "directional cone" and in fact, the m half-spaces partition the n-dimensional variable space into 2^m directional cones, within which each objective either increases or decreases in value. This is illustrated in Fig. 3. This interpretation is useful to define search directions that span the Pareto set, rather than converging to it, where some objectives will increase in value and others decrease.

It is also useful to consider the size of the directional cone during initial and final stages of optimization process. When a point is far from the local optima, typically the objective gradients are aligned and the directional 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. However, when a point is close to the Pareto set/front, the individual objective gradients are contradictory and in almost opposite directions. This follows directly from the definition of Pareto optimality, where, if one objective is decreased, another objective must increase. The size of the directional cone is small. Therefore, if a search direction is selected at random, there is only a small probability that it will lie in this directional cone and thus simultaneously reduce all the objectives. The likelihood is that it will lie in a cone such that some of the objectives will increase and the others decrease, thus spanning the Pareto front, rather than converging to it. This is one of the main differences between single and multi-objective design problems.

Appealing once again to probabilistic notions, this reasoning suggests that the children individuals in a multi-objective EC algorithm should be created to lie within the directional cone. Early in the search process, this is likely to be the same as any given single-objective half space, suggesting that children individuals should be placed near the parents, as in [4]. Later in the search process, this is not the case, and one could expect that locating children near parents will not lead to efficient convergence towards the Pareto front.

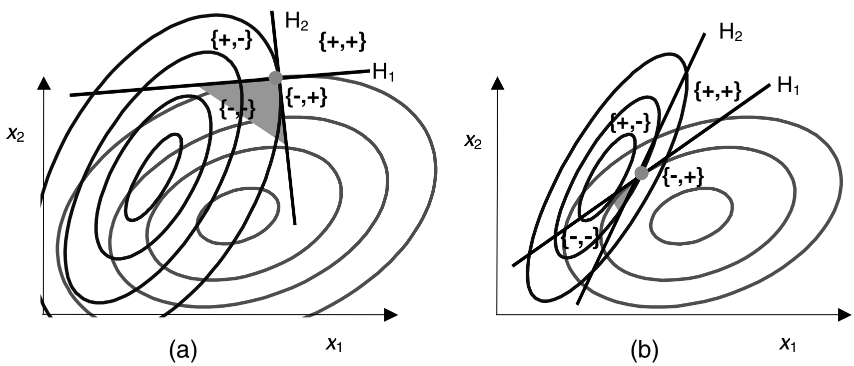

Fig. 3. The directional cones for a 2 parameter, 2 objective design problems during initial (a) and final (b) stages of convergence. The cones are labeled with the sign of the corresponding change in objectives and as can be seen, the descent cone {-,-} shrinks to zero during the final stages of convergence.

2.3 Test for Local Pareto Optimality

The interpretation of multi-objective optimization described in the last section is appropriate, as long as the design is not 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 [5] 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 N(**J**) is the null space of the Jacobian matrix, **J**, and $R(\mathbf{J}^T)$ is the range of $\mathbf{J}^T$. The Jacobian is the matrix of derivative of each variable with respect to each objective. The equation above is equivalent to:

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

The geometric interpretation of this test in objective space (shown in Fig. 4) is that there exists a non-negative combination of the individual gradients that produces an identically zero vector. When this occurs, any changes to the design parameters will affect only $R(\mathbf{J}^T)$, which is orthogonal to λ. Therefore, no changes to the design parameters will produce a descent direction that simultaneously reduces all the objectives. This is the limiting case of the situation described in Section 2.2, 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, so the movement will be along the Pareto front, rather than minimizing all the objectives. In fact, for an optimal design, $R(\mathbf{J}^T)$ defines the local tangent to the Pareto front and thus defines the space that must be locally sampled in order to generate the complete local Pareto set/front.

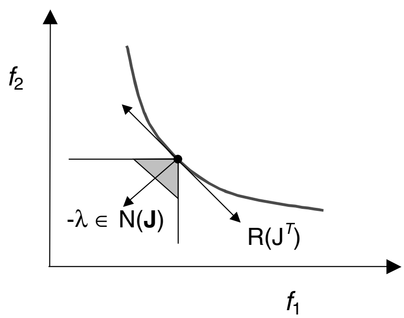

Fig. 4. Geometrical interpretation of the Null space and Range of the Jacobian matrix when a design is Pareto optimal. The vector λ specifies the local normal to the Pareto front.

Once again appealing by analogy to [4], note that concentration of "children" individuals in an EC algorithm near "parents" is not likely to result in individuals within the appropriate directional cone, in a way that is analogous to points being biased to the appropriate half-space in single-objective search. Therefore, to exploit analogous advantages offered by modern, real-valued EC in multi-objective settings, it is appropriate to consider further operations as a part of the search process. These are outlined in the following section.

2.4 Multi-objective Steepest Descent

A multi-objective steepest descent search direction must lie in the directional cone that simultaneously reduces all the objectives. This specification can be made unique [1][5] by requiring that the reduction is maximal which can be formulated as:

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

where **J** is the local Jacobian matrix, **s** is the calculated search direction and α represents the smallest reduction in the objectives' values. This is the primal form of the Quadratic Programming (QP) problem in (n+1) dimensions. It requires as large a reduction in the objectives as possible for a fixed sized variable update. When all the 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° in objective space (assuming that the objectives have been scaled to a common range).

It can be shown that when the current point is not Pareto optimal, this problem has a solution such that α* is negative, and the calculated search direction **s*** therefore lies in the appropriate directional cone. However, it may be easier to solve this problem in the dual form [1][5]:

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

This is now a QP problem in m variables. Once it has been solved, the corresponding search direction is given by:

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

This search direction will simultaneously reduce all objectives and do so in a maximal fashion, as described by the primal problem. Hence, it is known as the multi-objective steepest descent algorithm. The Multi-Objective Gradient (MOG) that is given by:

$$\mathbf{g} = \mathbf{J}\boldsymbol{\lambda}^* \tag{3}$$

is calculated from a non-negative linear combination of the individual gradients. Therefore, the multi-objective search direction will be "aligned" with the individual gradients, although it should be noted that the degree of alignment, λ^*, will dynamically change as the point moves closer to the Pareto set. 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.

In order to implement this calculation, it is necessary to obtain the Jacobian **J**. This can be an expensive operation, especially for practical design problems where it is necessary to perform some form of local experimental design. This is considered further in the next section.

2.5 Dimensionality Analysis

This theory also provides some relevant results about the problem's dimensionality. Firstly, the dimension of both the Pareto set and front has the upper bound min(n,m-1). This can be derived by simply considering the rank of the Jacobian when a point is Pareto optimal. The dimension of the parameter-objective space mapping is locally rank(**J**) = min(n,m). When a point is Pareto optimal, this reduces the dimension of the objective space by one. In fact, rank(**J**) is the actual local dimension which is bounded above by min(n,m-1). This is an important result, as it specifies the dimension of the sub-space that a population-based EC algorithm must sample. An EC population must be large enough to adequately sample a space of this size. It is also important as it provides an approximate bound of how the number of objectives and variables should be balanced. The dimension of the actual Pareto set and front is bounded by min(n,m-1), so it may be unnecessary to have either $n >> m$ or $n << m$. Secondly, it is not always necessary to consider all the design parameters in order to whether a descent direction can be calculated. Suppose that only a single variable is considered for adaptation. As long as this change, either positive or negative, simultaneously reduces all the objectives, it is possible to state that the current point is not Pareto optimal and there exists a descent direction in the current sub-space. This is true for any sub-set of the design parameters considered, and is important as it may allow sparse Jacobian estimates to be used in certain situations.

3 Directional Evolutionary Computation

The work described in this section addresses the question of how to estimate a dominating search direction, based on the information contained in a local neighborhood of designs. In the language of EC, this essentially examines how to perform an "intelligent" crossover in order to produce new designs that dominate the old ones in the population (by placing children in the directional cone). The aim is to minimize the number of actual objective evaluations, by maximally re-using information contained in the population. For each point in the population, a set of local neighbors is used to approximate the local Jacobian and thus calculate the MOG. By searching along this path, a new point will be found that dominates the starting point. In the worst case, each recombination of this sort involves estimating the Jacobian, which is an order $n*m$ computation, and the solution of a quadratic program, which is approximately an O(min$\{m,n\}^3$) calculation. However, while this is computationally expensive compared to many conventional crossover operations, this computational expensive is likely to be vastly outweighed by the expense of evaluating fitness values for the population (as is usually the case in EC).

3.1 Population-Based Estimation of the Multi-objective Gradient

To begin to understand how the local population can help to determine an "intelligent" crossover operation, consider the trivial case where the local population consists of a point that is dominated by the current point, see Figure 6. By moving along the path **s**, where **s** = **x**-$\mathbf{x}_1$, a new design will be found such that it dominates the current design **f**(**x**).

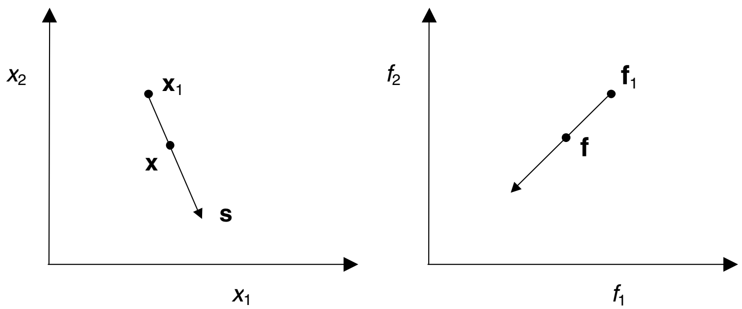

Fig. 5. Calculating a descent direction when the local neighborhood about a design, x, consists of a single dominated design x_1.

In general, the situation is rarely this simple. During the latter stages of convergence, designs rarely dominate each other as they are spread out along the local estimate of the Pareto set/front. In this situation, it is possible to use the primal/dual steepest descent theory described in Section 2.4, as long as the Jacobian information can be generated efficiently. Suppose that a set of r, = min{m,n}, linearly independent designs in the local neighborhood have been identified. Typically, these would be closest set of designs to the current design of interest. As illustrated in Fig. 6, the differences in the design parameters and the objectives along the search directions $\mathbf{s}_i$ can be measured. Two matrices then represent this difference information:

$$\mathbf{X}_s = \frac{\Delta x_i}{\Delta s_j} \tag{4}$$

$$\mathbf{F}_s = \frac{\Delta f_i}{\Delta s_j}$$

and the local Jacobian can be estimated by

$$\mathbf{J} = (\mathbf{X}_x^T \mathbf{X}_s)^{-1} \mathbf{X}_s^T \mathbf{F}_s \tag{5}$$

Therefore, the differences between the point of interest and each member of the local neighborhood is used to estimate gradient information, which is then utilized (2) to calculate the MOG. This relies on the local neighborhood being sufficiently small so that the gradients can be estimated sufficiently accurately.

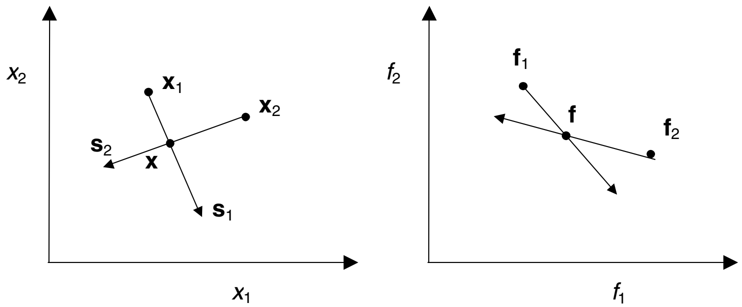

Fig. 6. Estimating the Jacobian matrix by using difference information along the indicated search directions s_i.

3.2 Population-Based Pareto Optimality

When the local Jacobian is calculated using the members of the local population to estimate difference information, it is assumed that the designs are sufficiently well distributed such that they span the necessary *r*, dimensions (note that this may be a subset of the complete design space) and that they are sufficiently close to the central point so that any estimation error is small. When this is true, the test for Pareto optimality is described in Section 2.3 and can be (naively) implemented by testing whether the MOG is sufficiently small.

It should be noted that while this paper has concentrated on describing techniques for testing for optimality and calculating descent directions, it is possible to use the geometric insights about the shape of the Pareto set/front in order to specify search directions that span the Pareto set/front as specified by $R(\mathbf{J}^T)$. This will be described in a later paper that analyses the complete algorithm.

4 Example

The use of directional information for multi-objective optimization will now be demonstrated on a simple test problem with 2 variables and 2 objectives. Specifically, we consider two quadratic objective functions, centered on [0.25, 0.75] and [0.75, 0.25] and with Hessian matrices are [80 40; 40 40] and [40 40; 40 80], respectively. 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.

This concepts described above were implemented as a simple multi-objective optimization procedure where the initial population of 50 points approximately lay along a line with a small amount of random noise added, as illustrated in Fig. 7.

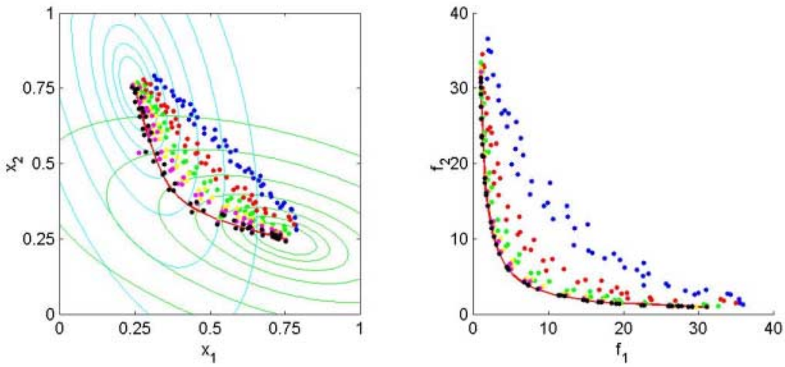

Fig. 7. Using the directional MOG to drive a population towards the Pareto set (left) and front (right). Each progressive population is shown with points in a different color, progressing towards the indicated front.

In Fig. 7, it can be clearly seen that successive iterations produce new "child" values that dominate their parents. Convergence to the Pareto set/front occurred after around 6 iterations. It should be noted that a fixed step size was used here and it would be expected that a more intelligent line search method would produce faster convergence. It should also be noted that the spanning of the Pareto set/front occurs because of the diversity in the initial population. While there is no diversity component in the method described in this paper, it can be clearly seen that the approximate MOG descent calculation projects the points directly towards the Pareto set/front in a rapid and efficient manner. In addition, the development of the diversity component [6] is the subject of current research.

As a final, simple comparison, a naïve genetic multi-objective procedure was developed. A simplex-type scheme was used to adapt the points, where each point was tested against neighboring values, and if the point was dominated or it was too close to a neighboring point (within a threshold distance), it was replaced by a combination of its *n* neighboring values (thus using averaging of parents as a form of recombination), else it was subject to a small random mutation (addition of Gaussian noise), with a standard deviation = 0.01. The results of this procedure are shown in Fig. 8. The same initial population is used as in Fig. 7. It should be noted that the final population now occurs after 60 iterations and the convergence in variable space is poor. While it is readily acknowledged that more sophisticated combination/selection processes could be used, the aim was to demonstrate that an "intelligent" use of directional information can dramatically improve the rate of convergence for suitably smooth, differentiable design problems. Moreover, we feel that this reasoning can be extended for the formulation of recombination operators in a much broader class of problems.

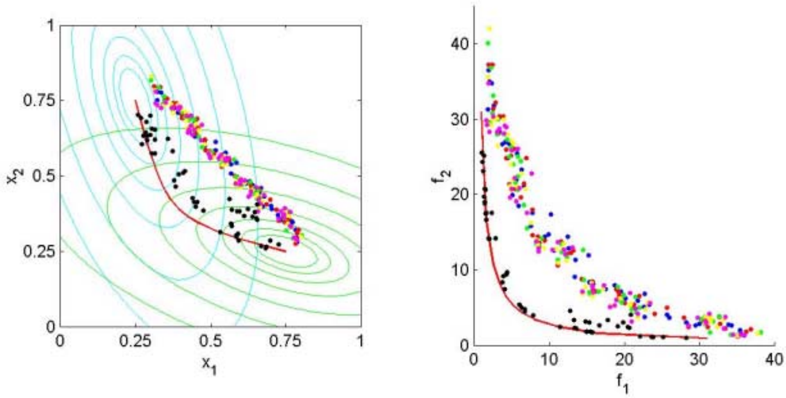

Fig. 8. Using a genetic multi-objective optimization process to solve the simulated problem.

5 Conclusions and Further Work

One must consider how to efficiently estimate a dominating search direction for a multi-objective problem, based on the information contained in a local neighborhood of designs, to properly apply EC to such problems. The theory developed in this paper clarifies how to logically use such information in multi-objective EC. The theory can be both used to explain the performance of current techniques, and gain insights into the problem's dimensionality. Also, the concepts developed (of convergence, directional cones and MOG) are directly applicable to differentiable multi-objective problems, and may be useful in EC algorithm design for a broader class of problems. In particular, a key aspect of EC algorithm design is the specification of recombination and mutation operators. When points are close to the optimal front/set children should be placed in the descent cone, or convergence should be expected to be poor in many cases. This paper shows that appropriate, informed use of direction information in MOEAs will improve performance. Moreover, the authors believe that the theory can both be extended to more logically design operators in non-continuous problems and also to understanding how diversity [6] can be integrated into such schemes, which is a subject of their current research.

References

[1] Brown, M. 3(2002) Steepest Descent Vector Optimization and Product Design, submitted for publication.in preparation

[2] Coello, Carlos A., Van Veldhuizen, David A., and Lamont, Gary B (2002) *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers.

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

[4] Deb, K , Anand, A., and Joshi, D (2002). A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization. KanGAL Report No. 2002003. (to appear in the journal Evolutionary Computation).
[5] Fliege, J. (2000) Steepest Descent Methods for Multicriteria Optimization. *Mathematical Methods of Operations Research*, 51(3), pp. 479–494.
[6] Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002) Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation 10(3) pp. 263–282.
[7] Parmee, I. C., Watson, A., Cvetkovic, D., Bonham, C. (2000). Multi-objective Satisfaction within an Interactive Evolutionary Design Environment. Evolutionary Computation. 8(2). pp 197–222.
[8] Product Formulation using Intelligent Software, http://www.brad.ac.uk/acad/profits/website/
[9] Sobieszczanski-Sobieski, J.; and Haftka, R.T. (1997). Multidisciplinary aerospace design optimization: survey of recent developments. Structural Optimization 14, pp. 1-23.
[10] Vicini, A., and Quagliarella, D. (1997) Inverse and direct airfoil design by means of a multi-objective genetic algorithm. AIAA Journal. 35(9).