Multi-objective particle swarm optimisation methods.

Jonathan E. Fieldsend

1st March 2004

Abstract

This study compares a number of selection regimes for the choosing of global best *(gbest)* and personal best *(pbest)* for swarm members in multi-objective particle swarm optimisation (MOPSO).

Two distinct *gbest* selection techniques are shown to exist in the literature, those that do not restrict the selection of archive members and those with 'distance' based *gbest* selection techniques. Theoretical justification for both of these approaches is discussed, in terms of the two types of search that these methods promote, and the potential problem of particle *clumping* in MOPSO is described. The popular *pbest* selection methods in the literature are also compared, and the affect of the recently introduced *turbulence* term is viewed in terms of the additional search it promotes, across all parameter combinations. In light of the discussion, new avenues of MOPSO research are highlighted.

Keywords: Multi-objective optimisation, particle swarm optimisation.

1 Introduction

Since its inception in 1995 [15] the particle swarm optimisation (PSO) heuristic has gained rapid popularity as a technique to facilitate single objective optimisation.[1] Like the standard evolutionary algorithm (EA) methods of genetic algorithms (GAs) and evolutionary strategies (ESs), PSO was inspired by nature, but instead of evolution it was the flocking and swarm behaviour of birds and insects that motivated its development.

A population of individual solutions is maintained in PSO, whose representation is similar to that of ES, that is a string of floating point *decision* parameters, which are used in an individual's evaluation. During the optimisation process of PSO (following initialisation), members of this population are *flown* (have their parameters adjusted) according to their previous *flying* experience. This flying experience is both in terms of the particle as an individual, and as a member of a wider group (the entire population or a subset of it). The general PSO model implements this by adjusting an individuals decision parameters to make them 'closer' to the decision parameters of two other solutions;

[1] 91 separate articles are listed in the web PSO bibliography at `http://www.computelligence.org/pso/bibliography.htm`.

the best evaluated individual found so far by the population, *gbest* (or that from a local subset, *lbest*), and the best evaluated individual found previously by that individual, *pbest*.

Until recently PSO had only been applied to single objective problems, however, in a large number of design applications there are a number of competing quantitative measures that define the quality of a solution. For instance, in designing the ubiquitous widget, a firm may wish to minimise its production cost, but also maximise/minimise one or more widget performance properties. These objectives cannot be typically met by a single solution, so, by adjusting the various design parameters, the firm may seek to discover what possible combinations of these objectives are available, given a set of constraints (for instance legal requirements and size limits of the product). The curve (for two objectives) or surface (more than two objectives) that describes the optimal trade-off possibilities between objectives is known as the true Pareto front. A feasible solution lying on the true Pareto front cannot improve any objective without degrading at least one of the others, and, given the constraints of the model, no solutions exist beyond the true Pareto front. The goal, therefore, of multi-objective algorithms (MOAs) is to locate the Pareto front of these *non-dominated* solutions.

Multi-objective evolutionary algorithms (MOEAs) are a popular approach to confronting these types of problem by using evolutionary search techniques [1, 4, 7, 5, 9, 8, 10, 12, 13, 17, 16, 19, 31, 21, 23, 25, 27, 28, 30, 32, 29]. The use of EAs as a tool of preference is due to such problems being typically complex, with both a large number of parameters to be adjusted, and several objectives to be optimised. EAs, which can maintain a population of solutions, are in addition able to explore several parts of the Pareto front simultaneously. PSO similarly has these characteristics, so, given the promising results reported in the literature comparing PSO to EA techniques in the uni-objective domain, a transfer of PSO to the MO domain seems a natural progression.

In 2002 this progression occurred, with a number of different studies published on multi-objective PSO (MOPSO) [2, 6, 14, 24]. However, although most of these studies were generated in tandem, each of these studies implements MOPSO in a different fashion. Given the wealth of MOEAs in the literature this may not seem particularly surprising, however the PSO heuristic puts a number of constraints on MOPSO that MOEAs are not subject to. In PSO itself the swarm population is fixed in size, and its members cannot be replaced, only adjusted by their *pbest* and the *gbest*, which are themselves easy to define. However, in order to facilitate an MO approach to PSO a set of non-dominated solutions (the best individuals found so far using the search process) must replace the single global best individual in the standard uni-objective PSO case, in addition, there may be no single previous best individual for each member of the swarm. Interestingly the conceptual barrier of *gbest* and *lbest* tends to get blurred in the MO application of PSO. A local individual may be selected for each swarm member, however these *lbest* individuals may all also be non-dominated (representing local areas of the estimated Pareto front maintained by the swarm), making them all also *gbest*. Choosing both which *gbest*, *lbest* and *pbest* to direct a swarm member's flight therefore is not trivial in MOPSO. The principle divergence within [2, 6, 14, 24] has therefore been on how these are selected, with a separate

divergence on whether an elite archive is maintained.

This paper provides a theoretical comparison of a number of different *gbest*/*lbest* and *pbest* selection methods [2, 6, 14, 20, 24]. The inclusion of a turbulence variable within MOPSO algorithms is also discussed.

The paper takes the following structure: in Section 2 Pareto optimality is reviewed; in Section 3 the general PSO model is briefly described, as are the current applications in the literature of MOPSO. In Section 4 the different *gbest*/*lbest* and *pbest* selection methods are defined.

A discussion on further work concludes the paper in Section 5.

2 PARETO OPTIMALITY

Most recent work on MOAs is formulated in terms of non-dominance and Pareto optimality, which we now briefly reviewed.

The multi-objective optimisation problem seeks to simultaneously extremise D objectives:

$$y_i = f_i(\mathbf{x}), \qquad i = 1, \ldots, D \tag{1}$$

where each objective depends upon a vector $\mathbf{x}$ of P parameters or decision variables. The parameters may also be subject to the J constraints:

$$e_j(\mathbf{x}) \geq 0, \qquad j = 1, \ldots, J. \tag{2}$$

Without loss of generality it is assumed that the objectives are to be minimised, so that the multi-objective optimisation problem may be expressed as:

$$\text{Minimise } \mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_D(\mathbf{x})) \tag{3}$$

$$\text{subject to } e(\mathbf{x}) = (e_1(\mathbf{x}), \ldots, e_J(\mathbf{x})) \geq 0 \tag{4}$$

where $\mathbf{x} = (x_1, \ldots, x_P)$ and $\mathbf{y} = (y_1, \ldots, y_D)$.

When faced with only a single objective an optimal solution is one which minimises the objective given the model constraints. However, when there is more than one objective to be minimised solutions may exist for which performance on one objective cannot be improved without sacrificing performance on at least one other. Such solutions are said to be *Pareto optimal* [28] after the 19th century Engineer, Economist and Sociologist Vilfredo Pareto, whose work on the distribution of wealth led to the development of these trade-off surfaces [22]. The set of all Pareto optimal solutions are said to form the true Pareto front.

The notion of *dominance* may be used to make Pareto optimality clearer. A decision vector $\mathbf{u}$ is said to *strictly*

dominate another $\mathbf{v}$ (denoted $\mathbf{u} \prec \mathbf{v}$) iff

$$\begin{aligned} f_i(\mathbf{u}) \leq f_i(\mathbf{v}) \quad \forall i = 1, \ldots, D \quad \text{and} \\ f_i(\mathbf{u}) < f_i(\mathbf{v}) \quad \text{for at least one } i. \end{aligned} \tag{5}$$

Less stringently, $\mathbf{u}$ *weakly dominates* $\mathbf{v}$ (denoted $\mathbf{u} \preceq \mathbf{v}$) iff

$$f_i(\mathbf{u}) \leq f_i(\mathbf{v}) \quad \forall i = 1, \ldots, D. \tag{6}$$

A set of M decision vectors $\{\mathbf{w}_i\}$ is said to be a *non-dominated set* (an estimated Pareto front $\mathcal{E}$) if no member of the set is dominated by any other member:

$$\mathbf{w}_i \not\prec \mathbf{w}_j \quad \forall i, j = 1, \ldots, M. \tag{7}$$

3 PSO

In this section the traditional uni-objective PSO is briefly described, followed by descriptions of the various MOPSO models in the literature.

3.1 Uni-objective PSO

The PSO heuristic was first proposed by Kennedy and Eberhart [15] for the optimisation of continuous non-linear functions. A fixed population of solutions is used, where each solution (or particle) is represented by a point in N-dimensional space. The ith particle is commonly represented [2, 6, 24, 26] as $X_i = (x_{i,1}, x_{i,2}, \ldots x_{i,N})$, and its performance evaluated on the given problem and stored. Each particle maintains knowledge of its best previous evaluated position, represented as $P_i = (p_{i,1}, p_{i,1}, \ldots p_{i,N})$, and also has knowledge of the single global best solution found so far, in the traditional uni-objective application indexed by g. The rate of position change of a particle then depends upon its previous local best position and the global best, and its previous velocity. For particle i this velocity is $V_i = (v_{i1}, \ldots v_{iN})$. The general algorithm for the adjustment of these velocities is:

$$v_{i,j} := wv_{i,j} + c_1 r_1 (p_{i,j} - x_{i,j}) + c_2 r_2 (p_{g,j} - x_{i,j}) \tag{8}$$

$$x_{i,j} := x_{i,j} + \chi v_{i,j}, \qquad j = 1, \ldots, N. \tag{9}$$

Where $w, c_1, c_2, \chi \geq 0$. w is the inertia of a particle, c_1 and c_2 are constraints on the velocity toward global and local best, χ is a constraint on the overall shift in position, $r_1, r_2 \sim U(0, 1)$. In [15], the final model presented has w and χ set at 1 and c_1 and c_2 are set at 2. In a later study by Shi and Eberhart [26] more explicit guidelines for the setting

of these parameters are presented.

3.2 MOPSO

In this section brief descriptions and critiques of preceding works in this area are provided.

3.2.1 Hu and Eberhart (2002)

A considerable degree of *a priori* knowledge in terms of test function properties is used in the implementation of the $D = 2$ MOPSO in Hu and Eberhart [14]. Instead of a single *gbest* a local *lbest* is found for each swarm member selected from the 'closest' two swarm members. The concept of closeness is calculated in terms of only one of the evaluated objective dimensions, with the selection of the local optima from the two based upon the other objective. The selection of which objective to fix (used to find the 'closest') and which to optimise is based on the knowledge of the test function design – the relatively simple objective function being fixed.

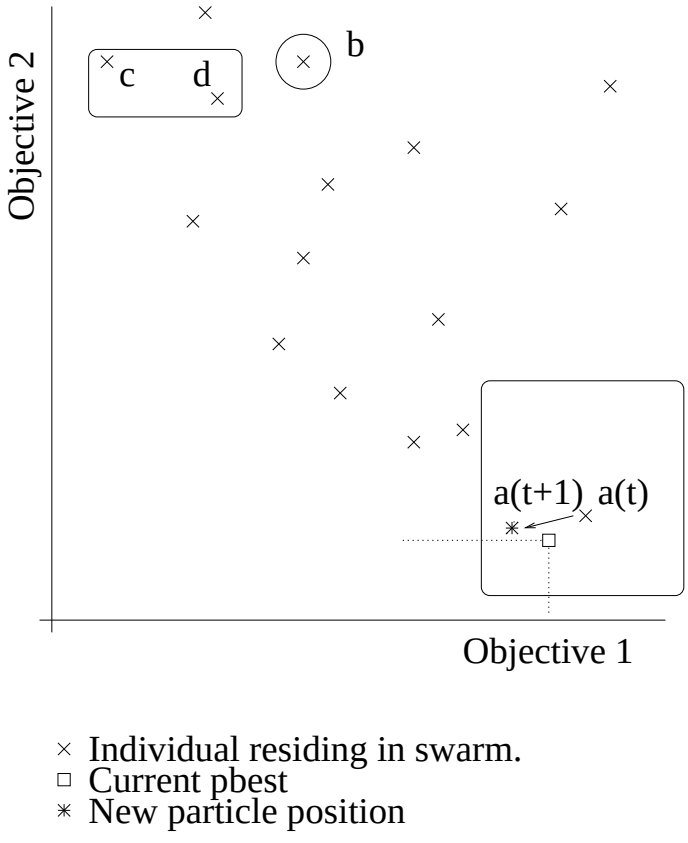

Figure 1: The multi-objective particle swarm optimisation method of Hu and Eberhart [14].

This is shown in Figure 1 with the nearest particles to b highlighted (in terms of the 'simpler' objective 2), meaning the *lbest* for b is c (the fitter of the two neighbours in terms of objective 1). A single *pbest* is maintained for each swarm member, which is only replaced when a new solution is found which dominates it (identical to the 'conservative'

preservation of efficiency selection rule described in the earlier work by Hanne [11]). This is demonstrated in Figure 1 with particle a moving to a fitter position at generation $t+1$ (one that dominates its previous position). This new position is mutually non dominating with the *pbest* of a, however, as the multi-objective evaluation of the new particle does not lie in the lower quadrant of the *pbest* (represented in Figure 1 with a square), P_a remains unchanged.

The performance of the MOPSO was demonstrated on a number of test functions from the literature (including the ZDT test functions from [30]), however no comparison was made with any other models, or the true Pareto fronts for the problems.

3.2.2 Parsopoulos and Vrahatis (2002)

Parsopoulos and Vrahatis [24] introduce two methods that use a weighted aggregate approach and another that is loosely based on Schaffer's MOEA [25]. These were compared on a number of two dimensional problems. In the first two approaches the weighted aggregate algorithms needed to be run K times to produce K estimated Pareto optimal points (meaning each run had a single global best). Although [24] states that this approach has a low computational cost, the need for a separate run for each solution found does not necessarily support this. Their final method - the Vector Evaluated Particle Swarm Optimiser (VEPSO), uses one swarm for each objective (as illustrated in Figure 2, where the two swarms are shown pushing toward the opposing axis). The best particle of the second swarm is used to determine the velocities of the first swarm (act as its global best), and vice-versa.

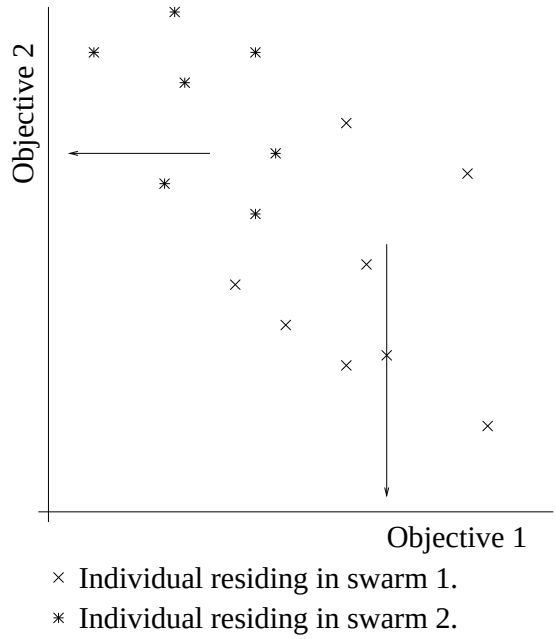

Figure 2: The multi-objective particle swarm optimisation model of Parsopoulos and Vrahatis [24].

Comparison between the algorithms was qualitative (based on visual inspection of the found fronts), with no comparison was made to recent competitive methods in the MOEA domain. In addition the current VEPSO model is

only designed for $D = 2$ problems.

3.2.3 Coello and Lechunga (2002)

The main problems with [14, 24] are their formulation purely for 2-dimension problems, and that, by taking their inspiration from early work in the MOEA domain, they themselves are susceptible to the problems that beset these early models, which research in the 1990s highlighted, and in a large part rectified. For instance, by using a swarm for each objective the VEPSO model of [24] will tend to suffer from the same problem of biasing its search toward the optimising of the individual solutions as Schaffer's VEGA does. The degree of prior knowledge needed by the MOPSO of [14] severely restricts its application, and inspection of the plots provided in their paper show that the model experiences problems discovering solutions over the full extent of the front. The constraints of needing m swarm members to even have the potential of having m estimated Pareto optimal solutions at the end of the search process is also very restrictive; typically only a small proportion of solutions in the population at the end of the process with be estimated Pareto optimal. This in turn necessitates large swarm sizes and therefore increased number of fitness evaluations, which may be costly in many applications (let alone inefficient). In addition the oscillating phenomena described in [5] is exacerbated when the Pareto front 'memory' is solely contained in X and P.

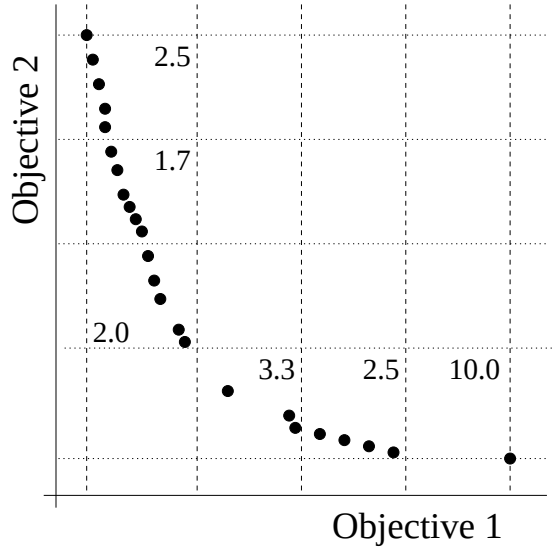

Figure 3: 2D Illustration of grid based selection scheme used in [2], with the 'fitness' of populated hypercubes highlighted.

In the previous two studies the maximum number of estimated Pareto points returned at the end of the search process equalled the swarm size, meaning large swarms were typically used. In comparison Coello and Lechunga [2] propose a method which is inspired by more recent developments in the MOEA literature. Two repositories are maintained in addition to the search population. One of the global best individuals found so far by the search process, F, and one containing a single local best for each member of the swarm. A truncated archive is used to store the

(global) elite individuals. This archive uses the method from [17] to separate the objective function space into a number of hypercubes (an adaptive grid), with the most densely populated hypercubes truncated if the archive exceeds its membership threshold. The archive also facilitates the selection of a global best for any particular individual in [2]. A *fitness* is given to each hypercube that contains archive members, equal to dividing 10 by the number of resident particles. Thus a more densely populated hypercube is given a lower score, an illustration of which is given in Figure 3.

Selection of a global best for a particle is then based on roulette wheel selection of a hypercube first (according to its score), and then uniformly choosing a member of that hypercube. This method therefore biases selection toward under-represented areas of the estimated Pareto front (unlike the original method developed in [17]). Only one local best solution is maintained for each swarm member however; if a particle X_i is evaluated and found to be mutually non-dominating with P_i, one of the two is randomly selected to be the new P_i.

An illustration of the swarm is shown in Figure 4, again particle a is highlighted in its generational move. However in this model, unlike [14], $a(t+1)$ has a 50% probability of becoming the new *pbest* of a.

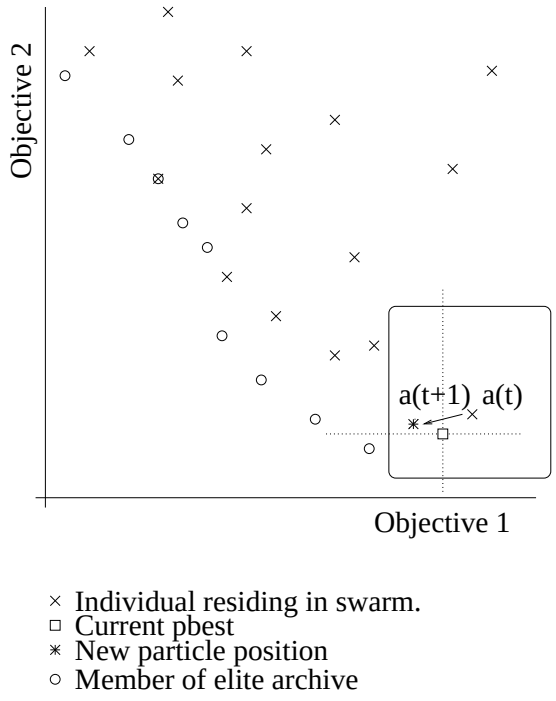

Figure 4: The multi-objective particle swarm optimisation model of Coello and Lechunga [2].

The MOPSO method in [2] was compared with two highly regarded MOEAs, the Pareto Archived Evolutionary Strategy (PAES) [17] and the Non-Dominated Sorting Genetic Algorithm II [3], with promising results. On the two dimensional test functions used the MOPSO either outperforms or is not significantly different to the competing algorithms (using the M_1^*measure [30]).

3.2.4 Fieldsend and Singh (2002)

Fieldsend and Singh [6] argued that the approach taken in [2] could be improved even further, as it is not a full transference of the PSO heuristic to the multi-objective domain. In uni-objective PSO the swarm is concerned with improving the fitness of each of its members with respect to a single objective. The formulation of Coello and Lechunga's model transfers this across by having each particle concerned with improving on *all* objectives. That is, selection of the *gbest* from the archive takes no consideration of where in the fitness landscape a particle is; in randomly choosing a *gbest,* a particle may be pushed toward an area in decision space whose fitness evaluation may be fitter in respect to one objective than the particles present position, but worse on one or more of the other objectives. In contrast an approach closer to the original model would try and push a particle toward and area in decision space which was evaluated as dominating its current position, better on at least one objective and no worse on any other objective. This can be taken even further by attempting to push the particle toward the member of the archive that not only (at a minimum) weakly dominates it, but that it is closest to in objective space. In this interpretation of MOPSO each swarm member is therefore concerned with improving a particular region of the estimated Pareto front, however only a single swarm is used. A shift in the relative position of a particle is not problematic as 'memory' of its search in a particular multi-dimension objective area is retained in the archive should any other particle become concerned with it (or indeed if it moves back in subsequent generations).

Although this focused or 'directed' form of MOPSO seems an appealing transference of PSO to the multi-objective domain the costs of implementation are prohibitive with existing methods. To find the closest archive member to a swarm individual X_i takes $\mathcal{O}(D \cdot |F|)$ objective comparisons, meaning $\mathcal{O}(|X| \cdot D \cdot |F|)$ each generation! However by using the ordering of individuals caused by the composite point data structure discussed in [5] this approach was made viable.

Dominated trees and PSO *gbest* selection Recent studies have highlighted the theoretic inefficiency caused by representing a non-dominated set with a limited number of solutions [11, 18]. This in turn led Fieldsend et al. [5] and Everson *et al.* [4] to empirically demonstrate the inefficiency caused by truncation of estimated Pareto archives in MOEAs, and develop a number of data structures to facilitate the maintenance of unconstrained archives. In this section the properties of one of these, the dominated tree, shall be briefly described. For a far more detailed definition and proofs please refer to [5]. The dominated tree consists of a list of $L = \lceil |F|/D \rceil$ *composite points* ordered by the weakly-dominates relation, $\preceq$:

$$\mathcal{T} = \{\mathbf{c}_L \preceq \ldots \preceq \mathbf{c}_2 \preceq \mathbf{c}_1\} \tag{10}$$

Usually, the stronger condition, $\mathbf{c}_i \prec \mathbf{c}_j \quad \text{iff } i > j$, will hold. The coordinates of each composite point are defined by (up to) D elements of F, the *constituent points* of a composite point. An example of a dominated tree in two dimensions is shown in Figure 5.

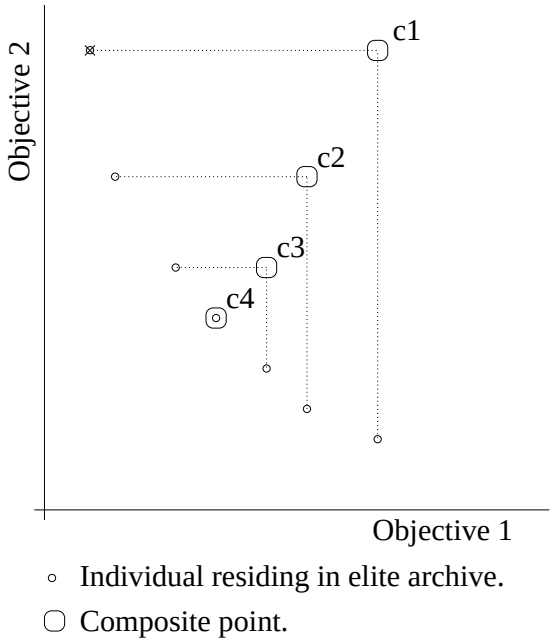

Figure 5: A 2D dominated tree.

Construction of a dominated tree from $|F|$ points $F = \{\mathbf{y}_m\}_{m=1}^{|F|}$, where $\mathbf{y}_m \in F$, proceeds as follows. The first composite point $\mathbf{c}_1$ is constructed by finding the individual $\mathbf{y}_m$ with maximum first coordinate; this value forms the first coordinate of the composite point:

$$\mathbf{c}_{1,1} = \max_{\mathbf{y}_m \in F} (\mathbf{y}_{m,1}) \tag{11}$$

This individual $\mathbf{y}_m$ is now associated with $\mathbf{c}_1$ and removed from F. Likewise the second coordinate of $\mathbf{c}_1$ is given by the maximum second coordinate of the points remaining in F: $\mathbf{c}_{1,2} = \max_{\mathbf{y}_m \in F \setminus \mathcal{T}} (\mathbf{y}_{m,2})$. This procedure is repeated to construct $\mathbf{c}_2$ and subsequent composite points until all elements of F are associated with the tree. In general the dth coordinate of the ith composite point is given by:

$$\mathbf{c}_{i,d} = \max_{\mathbf{y}_m \in F \setminus \mathcal{T}} (\mathbf{y}_{m,d}) \tag{12}$$

Note that in construction of the final composite point (that is, the composite point that dominates all other composite points) the $|F|$ elements of F may have been used before all the D coordinates of the final composite point $\mathbf{c}_L$ have been defined. The last remaining point in F is reused to define the remaining coordinates (as shown in Figure 5).

Implementation in MOPSO The directed MOPSO uses the properties of the dominated tree archive to select the 'closest' archive member to act as a swarm particle's *gbest*. For any member of the swarm, X_i, the first non-dominated composite point, c^j, of the global non-dominated set is sought (i.e. where $c^j \not\succ s \prec c^{j-1}$), this takes $\mathcal{O}(\lg(M+1))$ domination comparisons to find (where M is the number of composite points). The global best for an individual X_i is

that archive member of the composite point c^j contributing the vertex which is less than or equal to the corresponding objective in X_i. An illustration of this is provided in Figure 6.

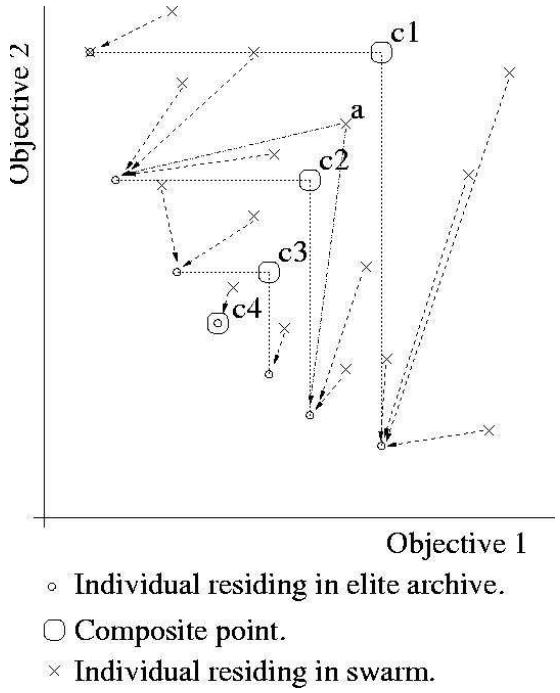

Figure 6: Selection of local *gbest* for each swarm member used in Fieldsend and Singh [6].

In the case of a composite point c^j with more than one vertex less than or equal to the corresponding objectives of an individual X_i (as is illustrated in figure 6 between composite point $c2$ and individual a) one of the vertex that meets the condition is selected at random to provide the global best (F_i) for the swarm individual X_i.

3.2.5 Mostaghim and Teich (2003)

In partial response to work discussed in [6], Mostaghim and Teich [20] introduced a new method called the *Sigma method* as alternative approach to directing particles toward the front. Again an elite archive was maintained (though truncated), and the *pbest* method was similar to that of [2], except the most recent non-dominating instance was maintained. In this approach each particle in the swarm is assigned a vector $\boldsymbol{\sigma}$, where $|\boldsymbol{\sigma}| = D$, which defines the gradient of a line connecting that point with the origin. For D dimensions this is calculated as

$$\boldsymbol{\sigma} = \begin{pmatrix} f_1(\mathbf{x})^2 - f_2(\mathbf{x})^2 \\ f_2(\mathbf{x})^2 - f_3(\mathbf{x})^2 \\ \vdots \\ f_D(\mathbf{x})^2 - f_1(\mathbf{x})^2 \end{pmatrix} / \left(f_1(\mathbf{x})^2 f_2(\mathbf{x})^2 + \ldots + f_D(\mathbf{x})^2 \right). \tag{13}$$

The $\boldsymbol{\sigma}$ values are calculated for the swarm and archive at each iteration (an $\mathcal{O}(D \cdot |X|)$ operation) and $\mathcal{O}(D \cdot |F|)$ operation) before the $\boldsymbol{\sigma}$ distances between all the swarm members and all the archive members are calculated (an $\mathcal{O}(|X| \cdot D \cdot |F|)$ operation). The *gbest* for a swarm member is selected as the archive member whose $\boldsymbol{\sigma}$ distance is the smallest. An illustration of this is shown in Figure 7.

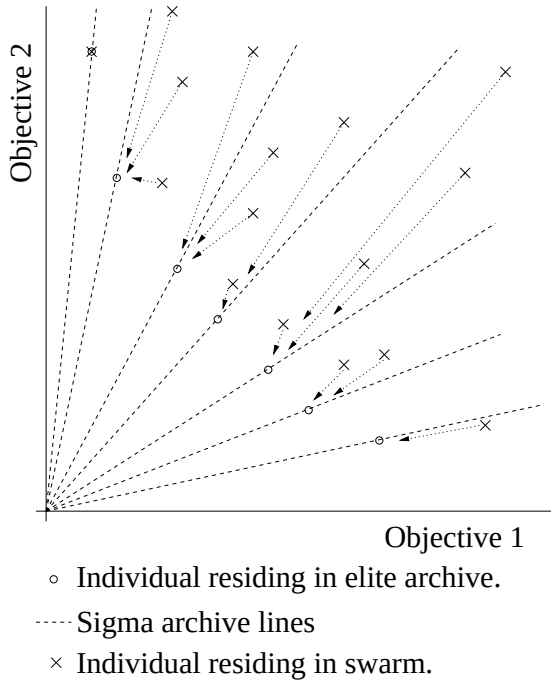

Figure 7: Selection of local *gbest* for each swarm member used in Mostaghim and Teich [20].

The method is intuitive - though, like the S-metric [29], its results are heavily influenced by where the point of comparison is placed (in the case of the Sigma method, at the origin). [20] compares the Sigma method to the methods from [6] and [31] on four test functions (for a single run) which seems to indicate that the Sigma method was an improvement over the directed-MOPSO of [6].

4 A theoretical discussion on the effect of the different search methods currently in use in MOPSO

In this section the effect of the different *pbest* and *gbest* selection methods on MOPSO search are discussed. However, first I shall look closer at the interaction of a turbulence term in MOPSO.

4.1 Turbulence

The studies that have used turbulence [6, 20] have noted the positive effect it has on the search process. When considering why this may be the case one must view the effect of turbulence in the context of the general PSO arithmetic form. In the general case, a particle X_i is pushed/pulled towards its *pbest* and *gbest* operating points, as well as along its current velocity. This means that a hypercuboid is generated in decision/particle space containing these four points, the bounds of which are defined by the sum of the absolute distances from X_i to the three other points (each distance multiplied by its relevant constraint from equation 8). X_i can therefore effectively move to any point within this hypercuboid, but not *outside* it. An illustration of this is shown in Figure 8 where the decision space is comprised of two variables. The highlighted area in Figure 8 shows the bounding hypercuboid where particle X_i can move to, given its previous best P_i, global best P_g and velocity V_i. Therefore, as illustrated, it is feasible for X_i to be moved to 'a', but impossible for it to shift to 'b'.

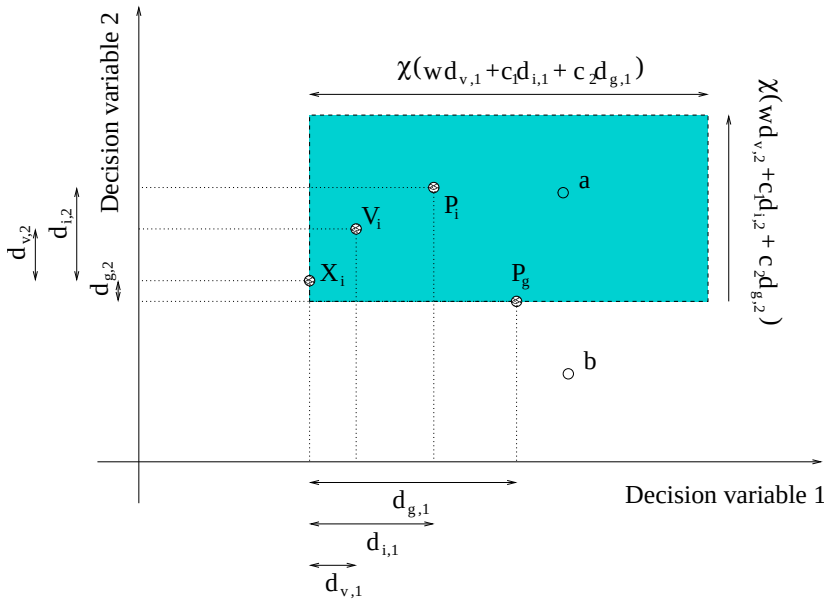

Figure 8: Illustration of the PSO search process, and the volume in which a particle X_i can move at each iteration.

This restriction on a particles movement means that local optima within this bound may be found, but any global optima outside will not be found on that iteration, and may never be attainable. Turbulence has the effect of increasing the volume of this bounded region, indeed, if the turbulence is drawn from distributions that extend beyond the range of the variables then in effect there is no bound on the search process at all. The turbulence term can therefore be seen to operate as a stochastic process, dislodging particles from local optima they may become stuck in.

4.2 Selection and maintenance of *pbest*

Four distinct *pbest* maintenance and selection strategies can be derived from the literature.

- In the first method, $\mathcal{P}_{random}$, a single *pbest* is maintained [2]. P_i is replaced if $X_i \succ P_i$, otherwise, if particle X_i is evaluated and found to be mutually non-dominating with P_i, one of the two is randomly selected to be the new P_i.

- In the second method $\mathcal{P}_{newest}$, a single *pbest* is maintained [20]. P_i is replaced if $X_i \succ P_i$, otherwise, if a particle X_i is evaluated and found to be mutually non-dominating with P_i, X_i (the most recent evaluation) is selected to be the new P_i.

- In the third method $\mathcal{P}_{dominating}$, a single *pbest* is maintained [14]. P_i is only replaced if $X_i \succ P_i$.[2]

- In the fourth method $\mathcal{P}_{Pareto}$, a set of *pbest*s is maintained [6]. This set contains the mutually non-dominating set of solutions that describe the estimated Pareto front found by X_i during the search process.

It is immediately clear that the computational cost of the $\mathcal{P}_{Pareto}$ is significantly higher than that of $\mathcal{P}_{random}$, $\mathcal{P}_{newest}$ and $\mathcal{P}_{dominating}$. The benefit of $\mathcal{P}_{Pareto}$ would be that it would promote a greater degree of search by the heuristic, however this could probably be achieved equally as well through the use of an appropriate *gbest* method or through turbulence. There is not great difference between $\mathcal{P}_{random}$, $\mathcal{P}_{newest}$ and $\mathcal{P}_{dominating}$, although the restrictive update nature of $\mathcal{P}_{dominating}$ may be detrimental in the later stages of the search process. Once the true Pareto front has been reached by a particle the *pbest* using $\mathcal{P}_{dominating}$ will no-longer be changed. As such the subsequent *coverage* of the true Pareto front may be impeded, as the *pbest* term will be constantly pulling the particle back to a fixed point.

4.3 Selection of *gbest* or *lbest*

From the previous section six different types of *gbest* selection can be identified.

- The first method, $\mathcal{G}_{random}$, is simple uniform selection of an instance from F. The principle benefit of this approach is that this selection is rapid, $\mathcal{O}(1)$, however it biases selection to already densely represented areas of the estimated Pareto front.

- The second method, $\mathcal{G}_{partitioned}$, is based on unbiased selection of the front. Here the front is partitioned (for example into grids in [17] and bins of equal width in [4, 5]), with selection first uniformly of a partition, and then uniformly from that partition. If a method such as partitioned quasi random selection (PQRS) [5] is used this selection takes $\mathcal{O}(\lg(|F|))$ objective comparisons.

- The third method, $\mathcal{G}_{biased}$, also uses partitioning, but biases selection toward those partitions which have fewer members. This can be achieved, for example, by using roulette wheel selection of partitions (as used by the MOPSO in [2] and one of the MOPSOs in [6]).

[2] N.B. This is equivalent to $\mathcal{P}_{newest}$, but always selecting the oldest of two mutually non-dominating points.

- The fourth method,$\mathcal{G}_{directed}$, is the local-global method of [6], where a *gbest* individual from an elite archive is selected locally to each swarm member. As stated previously, this takes $\mathcal{O}(\lg(M+1))$ domination comparisons

- In the fifth approach, $\mathcal{G}_{Sigma}$, the Sigma method of [20] is used to determine the *gbest* for each swarm member, needing $\mathcal{O}(|F| \cdot D + |X| \cdot D + |F| \cdot D \cdot |X|)$ calculations.

- The sixth approach, $\mathcal{G}_{Euclidean}$, also attempts to project the swarm members towards their nearest *gbest*, using the Euclidean distance method described (but not implemented) in [6]. Here the weighted Euclidean distances between the swarm members and the archive members are calculated and the closest archive members are used as the *gbest*. The distance between swarm member X_j and archive member F_k is calculated as

$$dist = \sum_{i=1}^{D} \left(\frac{f_i(X_j) - f_i(F_k)}{\max(f_i(F)) - \min(f_i(F))} \right)^2 \tag{14}$$

  By dividing each of the individual objective distances by its range in the archive, the distance space is normalised, thereby mitigating any objective scaling differences. This is illustrated in Figure 9, which again uses the same set of archive points and swarm members as the previous figures. This approach is an $\mathcal{O}(|F| \cdot D \cdot |X|)$ calculation.

On examination the six *gbest* methods described here can be seen to fall into two categories. The first 'unrestricted' group, $\mathcal{G}_{random}$, $\mathcal{G}_{partitioned}$ and $\mathcal{G}_{biased}$, allow for the selection of a *gbest*, for a given swarm member X_i, from anywhere in the archive F (albeit with differing probabilities). The second 'restricted' group, $\mathcal{G}_{directed}$, $\mathcal{G}_{sigma}$ and $\mathcal{G}_{Euclidean}$, by using some form of distance measure for selection, restrict the possible *gbest* for a given member of X_i. This restriction, although initially thought by researchers in the area to be desirable [6, 20] can also be seen to have three detrimental effects. Firstly, if it is assumed that that there is a mapping between objective and decision space domains - such that closeness in objective space relates to closeness in decision space,[3] then the 'unrestricted' methods can be seen to promote greater search than the 'restricted' methods. This is because in the 'unrestricted' methods, the objective hypercuboids generated by the *pbest*, *gbest*, X_i and V_i will tend to be larger, *ceteris paribus*, than that formed by distance based approaches. This is illustrated in Figure 10 by considering only X_i and its *gbest*, $F_{(i)}$.

Figure 10a shows the objective hypercuboids containing the swarm individuals and their *gbest* values where there is no restriction on where a *gbest* may be selected from. Figure 10b illustrates an identical situation, where *gbest* selection is restricted to the 'closest' archive member. The hypercuboids in the second case are clearly smaller - so assuming that objective distance maps to decision space distance this infers that at each iteration particles in a distance based *gbest* method will tend to search smaller areas. Relating this to the previous discussion on the use of turbulence, this means that *gbest* selection methods which do not restrict where on the front an $F_{(i)}$ may be drawn, methods like $\mathcal{G}_{random}$, $\mathcal{G}_{partitioned}$ and $\mathcal{G}_{biased}$, promote a greater degree of search within MOPSO.

[3] N.B. Of course in certain problems, or certain parts of the decision/objective space this is not necessarily the case. However it is an acceptable general assumption for the discussion here.

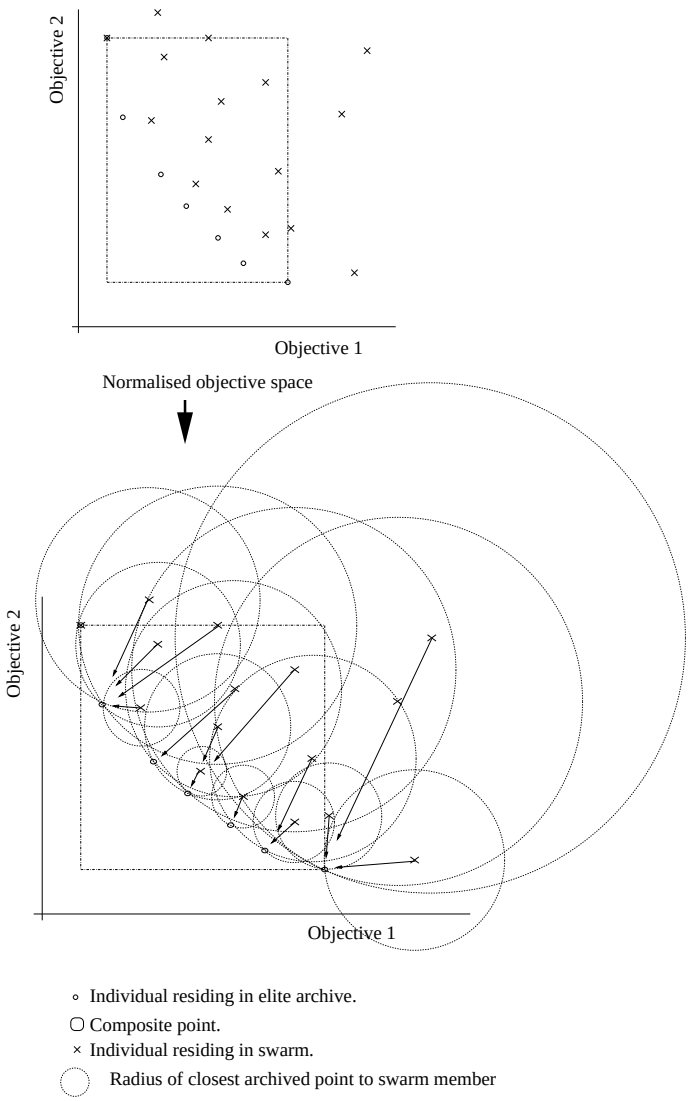

Figure 9: Selection of local *gbest* for each swarm member.

The second effect that using an 'unrestricted' selection routine has opposed to restricting the choice of *gbest* to close archive members (however one may define 'close') is that they actively work against *clumping* of particles; that is, preventing particles focusing on limited areas of the search front. Figure 11 illustrates a situation mid way through a search process. Here the particles have converged to two distinct clumps within objective space. The 'unrestricted' *gbest* selection method acts to pull these clumps apart by pushing them towards different parts of the estimated Pareto front (Figure 11a). The 'distance' based *gbest* selection method however does not do this, as its simply pushes the particles towards *gbest* individuals which are already in their area. As such it may even promote the clumps to become even tighter (Figure 11b).

If the *pbest* and/or velocity of a particle does not overcome this effect then search may only be directed towards to sub areas of the Pareto front, possibly duplicating efforts and reducing the overall efficiency of the algorithm.

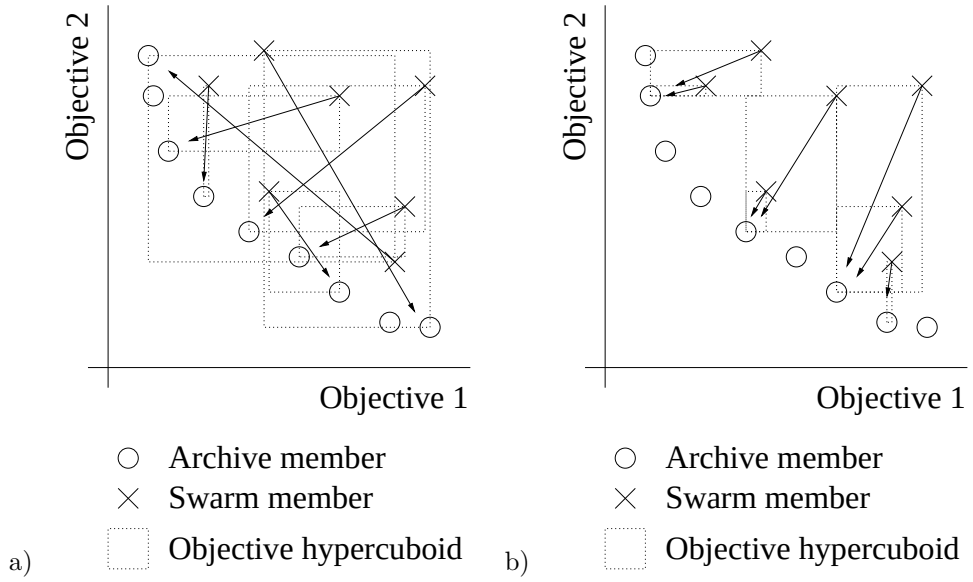

Figure 10: Search in 'unrestricted' (a) and 'distance' (b) based selection methods

The third effect can be viewed as a subvariant of either of the previous two effects, but merits discussion in its own right. If a swarm member reaches the estimated Pareto front (as must be the case every time a new estimated Pareto point is discovered) it is entirely possible that $X_i = P_i = F_{(i)}$. This means that the second and third term in Equation 8 equal zero, so that the only variable moving the point (barring turbulence if it is used) is the velocity term V_i, which may itself be very small at that point. This means a considerable amount of time may needed to move X_i and promote search further. By using an 'unrestricted' *gbest* selection term however the likelihood of this occurring is greatly reduced.

5 Further work

The different *gbest* and *pbest* selection methods described and examined in this study can been coarsely seen to either promote convergence or search. At the current time there does not seem to be any principled investigation into how these should effectively interact within MOPSO. Whether convergence should be promoted in the *pbest* term and search in the *gbest* term, or vice versa. In addition there could be convincing arguments made that their respective levels should be changed over time. An empirical comparison of combinations of these different methods is part of the author's current research interests.

Acknowledgements

The author would like to gratefully acknowledge support from the EPSRC during the writing of this report.

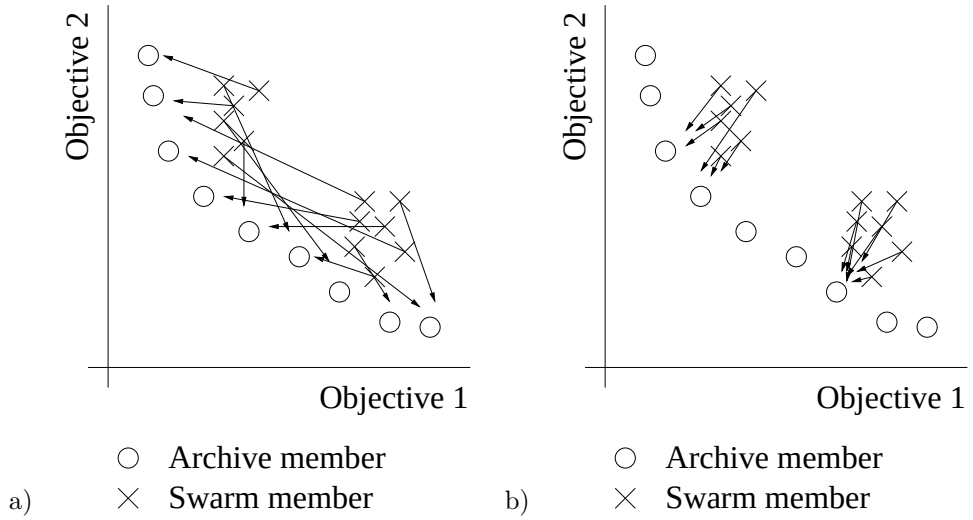

Figure 11: Particle clumping in 'unrestricted' (a) and 'distance' (b) based selection methods.

References

[1] C.A.C Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, 1999.

[2] C.A.C. Coello and M.S. Lechunga. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In *Proceedings of the 2002 Congess on Evolutionary Computation, part of the 2002 IEEE World Congress on Computational Intelligence*, pages 1051–1056, Hawaii, May 12-17, 2002. IEEE Press.

[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Proceedings of Parallel Problem Solving from Nature - PPSN VI*, pages 849–858. Springer, 2000.

[4] R.M. Everson, J.E. Fieldsend, and S. Singh. Full Elite Sets for Multi-Objective Optimisation. In I.C. Parmee, editor, *Adaptive Computing in Design and Manufacture V*, pages 343–354. Springer, 2002.

[5] J.E. Fieldsend, R.M. Everson, and S. Singh. Using Unconstrained Elite Archives for Multi-Objective Optimisation. *IEEE Transactions on Evolutionary Computation*, 7(3):305–323, 2003.

[6] J.E. Fieldsend and S. Singh. A Multi-Objective Algorithm based upon Particle Swarm Optimisation, an Efficient Data Structure and Turbulence. In *Proceedings of UK Workshop on Computational Intelligence (UKCI'02)*, pages 37–44, Birmingham, UK, Sept. 2-4, 2002.

[7] J.E. Fieldsend and S. Singh. Pareto Multi-Objective Non-Linear Regression Modelling to Aid CAPM Analogous Forecasting. In *Proceedings of the 2002 IEEE International Joint Conference on Neural Networks, part of the 2002 IEEE World Congress on Computational Intelligence*, pages 388–393, Hawaii, May 12-17, 2002. IEEE Press.

[8] C.M. Fonseca and P.J. Fleming. Genetic Algorithms for Multiobjective Optimizationi: Formulation, Discussion and Generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. Morgan Kaufmann.

[9] C.M. Fonseca and P.J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[10] P. Hajela and C-Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.

[11] T. Hanne. On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research*, 117:553–564, 1999.

[12] T. Hanne. Global Multiobjective Optimization Using Evolutionary Algorithms. *Journal of Heuristics*, 6:347–360, 2000.

[13] J. Horn, N. Nafpliotis, and D.E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, 1994. IEEE Service Center.

[14] X. Hu and R. Eberhart. Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. In *Proceedings of the 2002 Congess on Evolutionary Computation, part of the 2002 IEEE World Congress on Computational Intelligence*, Hawaii, May 12-17, 2002. IEEE Press.

[15] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *Proceedings of the Fourth IEEE International Conference on Neural Networks*, pages 1942–1948, Perth, Australia, 1995. IEEE Service Center.

[16] J. Knowles and D. Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 98–105, Piscataway, NJ, 1999. IEEE Service Center.

[17] J.D. Knowles and D. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[18] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. On the Convergence and Diversity-Preservation Properties of Multi-Objective Evolutionary Algorithms. Technical Report TIK-Report 108, Swiss Federal Institute of Technology Zurich (ETH), June 2001.

[19] M. Laumanns, E. Zitzler, and L. Thiele. A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 46–53, Piscataway, NJ, 2000. IEEE Service Center.

[20] S. Mostaghim and J. Teich. Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In *IEEE 2003 Swarm Intelligence Symposium*, 2003.

[21] T. Murata and H. Ishibuchi. MOGA: Multi-objective genetic algorithms. In *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation*, pages 289–294, Perth, November, 1995. IEEE Press.

[22] V. Pareto. *Manuel D'Économie Politique*. Marcel Giard, Paris, 2nd edition, 1927.

[23] G. T. Parks and I. Miller. Selective Breeding in a Multiobjective Genetic Algorithm. *Lecture Notes in Computer Science*, 1498:250–259, 1998.

[24] K.E. Parsopoulos and M.N. Vrahatis. Particle Swarm Optimization Method in Multiobjective Problems. In *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002)*, pages 603–607, 2002.

[25] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms*, pages 99–100, 1985.

[26] Y. Shi and R. Eberhart. Parameter Selection in Particle Swarm Optimization. In *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pages 591–601, 1998.

[27] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.

[28] D. Van Veldhuizen and G. Lamont. Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation*, 8(2):125–147, 2000.

[29] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology Zurich (ETH), 1999. Diss ETH No. 13398.

[30] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

[31] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report TIK-Report 103, Swiss Federal Institute of Technology Zurich (ETH), May 2001.

[32] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.