Extensions to the Strength Pareto Evolutionary Algorithm.

Jonathan E. Fieldsend, *Student Member*, IEEE,

Richard M. Everson, *Member*, IEEE,

and Sameer Singh, *Member*, IEEE.

PANN Research Group, Department of Computer Science,

University of Exeter, Exeter EX4 4PT, UK

Abstract

The Strength Pareto Evolutionary Algorithm (SPEA) has emerged as one of the leading evolutionary algorithms for multi-objective optimisation. This paper examines the algorithm and shows how significant improvements in the algorithm's performance may be made. It is demonstrated that SPEA, and other population based algorithms, can exhibit both retreating (or oscillatory) Pareto fronts and shrinking Pareto fronts, which can significantly impair optimisation performance. Remedies are proposed including the replacement of SPEA's *external* set with a *frontal* set containing all non-dominated solutions, which actively participates in the search process. The use of a frontal set is shown not only to prevent Pareto front retreat and oscillation, but also to increase optimisation speed. The use of an active frontal set also permits robust criteria for algorithm termination to be used. New data structures are introduced which permit the efficient use of even very large frontal sets.

Index Terms

Optimisation, Genetic Algorithms, Multiple Objectives.

1 INTRODUCTION

Frequently a number of competing objectives have to be traded against one another whilst seeking a viable solution to a given problem, often without any *a priori* knowledge of exactly how the objectives interact with one another. For instance, in product design a firm may wish to maximise the performance of an appliance whilst also trying to minimise its production cost. These two objectives cannot typically be met by a single solution, so, by adjusting the various design parameters, the firm may seek to discover what possible combinations of these two objectives are available, given a set of constraints (for instance legal requirements and size limits of the product). In [1] for example, multi-objective optimisation is applied to four performance measures of a gas turbine and in [2] different loads in trusses are the competing objectives to be minimised.

The curve (for two objectives) or surface (more than two objectives) that describes the optimal trade-off possibilities between objectives is known as the Pareto front [3]. A feasible solution lying on the 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 is to locate the Pareto front of these *non-dominated* solutions.

Multi-Objective Evolutionary Algorithms (MOEAs) represent a popular approach to confronting these types of problem by using evolutionary search techniques (usually, though not exclusively, by incorporating Genetic Algorithms (GAs)). MOEAs have been in use for a considerable length of time now: Beale and Cook in 1978 used a random search technique in an attempt to simultaneously minimise a number of objectives in an aircraft simulator [4]. However, it is the work of Schaffer in 1985 [5], which recognised the need to return a set of solutions, that has been widely quoted as the first MOEA study [6, 7, 8]. The use of Evolutionary Algorithms (EAs) as the tool of choice is due to such problems being typically complex, with both a large number of parameters to be adjusted, and several objectives to be optimised. In addition, EAs, which maintain a population of solutions, are able to explore several parts of the Pareto front simultaneously.

Most recent investigations in the area [9, 1, 6, 10, 11] focus on a MOEA's ability to produce an accurate estimate of the Pareto front. Zitzler et al. [8] present a comparative study, on six test functions introduced by Deb [12], of a number of the most widely used MOEAs, including Fonseca and Fleming's multiobjective EA [1], the Niched Pareto Genetic Algorithm [10], Hajela and Lin's weighted-sum approach [2], the Vector Evaluated Genetic Algorithm (VEGA) [5] and the Nondominated Sorting Genetic Algorithm [11]. Their study suggests that their Strength Pareto Evolutionary Algorithm (SPEA) outperforms the other algorithms, with it consistently recording better results as measured by the $\mathcal{C}$ metric [13, 8, 14] on a number of the test functions. In an earlier paper [14] Zitzler and Thiele also demonstrated SPEA's superior performance in comparison to four other MOEAs on a 0/1 knapsack problem.

Recent work by Laumanns et al. [15] provides a unified model for MOEAs with elitism (called UMMEA), which extends the approach used in SPEA to the entire MOEA genus. This is facilitated by using an archive set of non-dominated solutions in addition to the usual GA population. This archive is limited to a fixed maximum number of individuals.

Despite the performance already proven, this paper will highlight problems in relation to both the speed and stability of the SPEA, manifest in the form of shrinking, oscillating and retreating estimated Pareto fronts. It is shown that these artifacts are a consequence of efforts to represent the Pareto front by a restricted number of solutions. These problems will therefore also occur in models designed using the UMMEA, and other existing population based MOEAs. Remedies are therefore suggested based on retaining all the non-dominated solutions found: rather than merely keeping the non-dominated solutions in a dormant offline store, they are used as an active input to the continuing search process. This *frontal* set of non-dominated solutions is used in the binary tournament selection phase of SPEA as a replacement for that algorithm's *external* set. Furthermore, it is shown that this approach, along with 'pinning' of the extremal solutions, not only increases the algorithm's efficiency by solving the problems of shrinking, oscillating and retreating estimated fronts, but that it also enables the introduction of a robust algorithm termination methodology, which has until now been largely absent from the MOEA literature.

The methods introduced are applicable to any population based multi-objective search, and a new data structure is introduced that permits rapid searching of the frontal set, allowing even very large active frontal sets to become feasible.

The paper takes the following structure: in Section 2 Pareto optimality will be formally introduced and described; in Section 3 the SPEA algorithm as developed by Zitzler et al. will described. In Section 4 a number of problems with the SPEA will be highlighted and demonstrations of their effect shown, followed by the introduction of a number of solutions. In Section 5 implementational considerations of the proposed extensions are discussed and in Section 7 a robust set of stopping criteria are introduced, to replace the current methodology of *ad hoc* algorithm termination.

The results of a set of experiments quantifying the effects of the extended algorithm are reported in Section 6 together with the performance measures used. The paper concludes with comments in Section 8.

2 PARETO OPTIMALITY

Most recent work on MOEAs hinges on the notions of non-dominance and Pareto optimality, which is 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 succinctly stated as:

$$\begin{aligned} \text{Minimise} \quad \mathbf{y} &= f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_D(\mathbf{x})) \quad (3) \\ \text{subject to} \quad e(\mathbf{x}) &= (e_1(\mathbf{x}), e_2(\mathbf{x}), \ldots, e_J(\mathbf{x})) \geq 0 \quad (4) \end{aligned}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_P)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_D)$.

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 it is clear that solutions 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* [7] and the set of all Pareto optimal solutions are said to form the Pareto front.

The notion of *dominance* may be used to make Pareto optimality more precise. A decision vector $\mathbf{u}$ is said to *strictly dominate* another $\mathbf{v}$ (denoted $\mathbf{u} \prec \mathbf{v}$) iff

$$f_i(\mathbf{u}) \leq f_i(\mathbf{v}) \quad \forall i = 1, \ldots, D \quad \text{and} \quad f_i(\mathbf{u}) < f_i(\mathbf{v}) \quad \text{for some} \quad i \tag{5}$$

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 estimate of the Pareto front) 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 THE STRENGTH PARETO EVOLUTIONARY ALGORITHM

As evinced by a number of comparative studies [13, 8, 14] the Strength Pareto Evolutionary Algorithm provides an effective methodology for multi-objective optimisation problems. The SPEA of Zitzler and Thiele is now described, before some of its deficiencies and improvements to it are discussed. SPEA is outlined in Algorithm 1.

At each genetic generation, t, the SPEA algorithm has two main populations, an *internal* search population, P_t, where crossover and mutation take place, but also, unlike other multi-objective optimisation algorithms, it incorporates an active *external* set, E_t, of non-dominated individuals, which is updated at each generation. The external set forms the estimate of the Pareto front. The non-dominated individuals are referred to as *elite* by analogy with single objective optimisation problems. Both the internal and external sets are limited in size, with E_t being reduced in size by clustering if its size exceeds the predetermined limit M (part 3c of Algorithm 1). At the end of each generation the internal population for subsequent mutation and crossover is created by binary tournament selection between the existing internal and external populations (part 5 of Algorithm 1). A single *strength* value is attributed to each individual in both populations (part 4 of Algorithm 1), which determines the individuals rank in the binary tournament selection. For a non-dominated individual this value lies in the range $[0, 1]$, and is calculated as the fraction of the internal population that the selected individual strictly dominates. The strength of a dominated individual is one plus the sum of the strengths of the non-dominated individuals in the external set which dominate it (and therefore lies on the range [1,N]. More formally, where $\mathbf{b} \in E_t$

Algorithm 1 Strength Pareto Evolutionary Algorithm [13].

Input: N, Internal population size.

M, maximum size of External set.

T, number of generations for which SPEA will run

Output: A non-dominated set of M individuals

1) **Initialisation:** Generate Internal population P_0 of N decision vectors chosen at random and satisfying the constraints eq. 2. Initialise the external set of non-dominated solutions to be the empty set, $E_0 = \emptyset$.

Set generation counter $t = 0$.

2) **Fitness assignment:** For each member of P_t evaluate the objectives eq. 1. Mark any individuals which are non-dominated by other members in the population.

3) **Update of External set:**

a) Insert into E_t individuals from P_t which are non-dominated by any individuals in $P_t \cup E_{t-1}$.

b) Remove individuals from E_t which are dominated by the new entrants.

c) Reduce the External non-dominated set size to M (if larger than M), by the imposition of a clustering Algorithm (where M representative centroid solutions are selected, with the remaining individuals removed).

4) **Calculate Strengths:** Assign scalar 'strengths' to members of P_t and E_t based upon how many individuals they dominate or are dominated by.

5) **Selection:** Use binary tournament selection to fill P_{t+1} with individuals from P_t and E_t. N pairs of individuals are selected from $P_t \cup E_t$ with replacement, with the N individuals with the lower strength value of the pair inserted into P_{t+1}.

6) **Evolution:** The evolution of individuals in the internal set is implemented using standard GA mutation and crossover operators.

7) **Termination:** If T generations have passed, terminate; otherwise $t := t + 1$, goto 2.

and $\mathbf{a} \in P_t$, strengths $S(\mathbf{b})$ and S($\mathbf{a}$) are defined as:

$$S(\mathbf{b}) = \frac{|\mathbf{a}' \preceq \mathbf{b}\,;\, \mathbf{a}' \in P_t|}{N+1} \tag{8}$$

and

$$S(\mathbf{a}) = 1 + \sum_{\mathbf{b}' \succeq \mathbf{a}, \mathbf{b}' \in E_t} S(\mathbf{b}') \tag{9}$$

As such the elite external set contains an estimate of the Pareto front that is actively involved, through binary tournament selection, in the search process.

However, it is through the limitation of this external set's size that two distinct problems can arise, that affect both the extent of the final estimate of the Pareto front and the efficiency and stability of the search process within SPEA.

4 EXTENSIONS TO SPEA

4.1 Shrinking Pareto fronts

For computational efficiency the SPEA limits the size of the external set to M individuals. As illustrated in Figure 1, this is achieved by clustering (in objective space) the individuals comprising the external set and replacing clusters by the individual closest to the cluster centroid. The clustering algorithm used in SPEA is shown in Algorithm 2.

Figure 1 illustrates a consequence of using centroid replacement on the external set. Figure 1a shows 12 non-dominated individuals which are to be reduced to $M = 4$ individuals by clustering. The (hyper-) planes bounding this estimated Pareto front on each dimension are marked as A and B. The individuals that define these boundaries of the estimated Pareto front are referred to here as the *extremal individuals* (containing what are referred to as component minima and maxima in [15], in the case of the true Pareto

Algorithm 2 Centroid clustering algorithm used in SPEA [13].

```
      $E_t$ , External set at generation $t$.
      $M$, maximum size of external set.
1)    Initialize cluster set $C$, each individual residing in $E_t$ being a distinct cluster in $C$, $C := E_t$.
2)    If $|C| \leq M$, goto 5, else goto 3.
3)    Calculate all possible cluster pair distances $d_c$, where $l_1, l_2 \in C$
      $d_c\left(l_1, l_2\right) = \frac{1}{|l_1||l_2|} \cdot \sum_{\mathbf{a} \in l_1, \mathbf{b} \in l_2} d\left(\mathbf{a}, \mathbf{b}\right)$, and $d$ is the distance in objective space between individuals $\mathbf{a}$ and $\mathbf{b}$.
4.    Combine the two clusters with minimum distance
      $d_c$: $C = C \setminus \{l_1, l_2\} \bigcup \{l_1 \bigcup l_2\}$. Goto 2.
5.    Select a representative individual from each cluster and insert into $E_{t+1}$. The representative is the centroid, the individual with the minimal average distance to all other individuals in the cluster in objective space (if there is more than one individual in a cluster sharing a minimal distance, one of these is chosen as the representative at random).
```

set.). Figure 1b illustrates how the estimated front is represented by clusters, and Figure 1c shows the external set after the centroid individuals have been selected to represent the clusters. The result of this process is seen to be a 'shrinking' of the the Pareto frontier, with the extremal individuals shifting from A and B and C and D. If subsequent evolution fails to rediscover the extremal individuals repeated clustering will shrink the Pareto front and the final estimate Pareto set will lie across a narrow subset of the true frontier, as shown in Figure 2.

This is true even in an offline 'dormant' estimated Pareto front (that is one that acts as a passive store for individuals which are separate from the search process), as search will not have be directed towards the extremal values. It is interesting to note that after the criticism [9, 11, 8] of Schaffer's VEGA [5] because of its bias towards extremal values, that its replacements (not just SPEA) should in turn be biased towards search in the centre of the front.

The shrinking front effect as illustrated can be detrimental in two ways. The main consequence is the narrow extent of the estimated front; secondly, extra search time is required in order to rediscover the extremes of the estimated Pareto front.

These problems are easily circumvented by removing the extremal individuals from the clustering process and passing them directly to the binary tournament selection stage. In this study this approach is referred to as the *pinning* of extremal individuals, and is illustrated in Figure 3. As can be seen, the hyper-planes bounding the original and reduced external non-dominated sets are identical. Consequently in the next generation information from individuals on the extremes of the estimated Pareto front from the previous generation will be used in the GA search process, and may aid further expansion on the extremes of the estimated Pareto front.

Since up to D individuals may be required to define the extremes of the external set, there may be as few as $M - D$ individuals available for clustering. However, in Section 4.2 it is argued that clustering should be dispensed with altogether.

4.2 Persistent 'frontal' set

The external set of SPEA is in essence a memory of where the algorithm has reached in previous generations in its estimation of the Pareto front and should contain the 'best' estimate of the Pareto front at any stage. The estimated front should 'advance' in the sense that no individual in E_t should be dominated by any member of an earlier external set, $E_0, \ldots, E_{t-1}$. Informally, it is said that an individual $\mathbf{x}$ lies *behind* the front if a member of the external set dominates $\mathbf{x}$. However, the requirement that E_t be limited to M members can produce 'retreating' or, more commonly, 'oscillating' estimates of the Pareto front. In these cases members of E_t may lie behind the earlier estimates of the external set. The origin of this behaviour is now described.

An illustration of a retreating front is shown in Figure 4. Figure 4a illustrates an external set with a maximum of $M = 6$ members. In Figure 4b a new non-dominated member (drawn as a filled circle) has entered the set. Since there are now 7 elements in the external set, one must be removed by clustering;

the pair of solutions nearest each other form a cluster of two and one of them (chosen at random) is deleted, resulting in the external set shown in Figure 4c. If at a subsequent generation a new element enters the external set (Figure 4d), the clustering process will reduce the external set as shown in Figure 4e. This results in an external set (Figure 4e) containing an element that lies behind (is dominated by) elements of the original external set (Figure 4a). Repeated occurrences of this process can lead to the estimated front retreating or, more commonly, oscillating as the front advances in the GA search stage but retreats during clustering.

This artifact has two main consequences. First, search time is wasted 'rediscovering' individuals and regions that have been eliminated by clustering. Secondly, convergence to the true Pareto front is impaired. Numerical simulations show that this oscillation is particularly serious when the estimated front lies close to the true front; the oscillation can prevent convergence to the true front leading to poor estimates and difficulties in assessing convergence.

It should also be noted that this artifact is not just associated with SPEA; other population-based MOEAs have no secondary elite set, and therefore have no externally-stored active 'memory' at all, their search population alone containing the active estimated front. Consequently they are even more susceptible to oscillating and retreating estimated fronts, due to the stochastic nature of the GA search process which drives them. If an offline dormant non-dominated set is being used, it will simply remain unchanged for long periods, until the search population converges once more on the estimated front it had previously discovered.

Zitzler et al. [8] advance elitism embodied in the external set as being an important factor in SPEA's preeminence among MOEAs, as SPEA was the only algorithm that incorporated elitism as a central component. Indeed Zitzler et al. demonstrate than when elitism in the form of an external set is incorporated in the Nondominated Sorting Genetic Algorithm, its performance rises to the standard of SPEA. However, as demonstrated, a second active population with a size limit can cause problems.

Recognising that the stochastic nature of the EAs means that non-dominated solutions may not remain in the search population, Van Velduizen and Lamont [7] note the need for a secondary non-dominated population to hold all estimated Pareto solutions found by the search process, even if the secondary population takes no further part in the search. They suggest that information held within this population may be useful to the search process, and cite SPEA as a MOEA that implements this. However, Van Velduizen and Lamont do not make any explicit recommendations beyond keeping a dormant store of all discovered non-dominated individuals. In the light of the artifacts discussed here, it is recommended that a secondary population of *all* currently non-dominated individuals found during the evolutionary search is used actively within the search process. This set of all currently non-dominated individuals is referred to as the *frontal set*, F_t. Elimination of the SPEA clustering stage and use of the frontal set effectively removes the oscillation / retreat problem, as well as ensuring that extremal individuals are retained throughout the search. It should be noted that, while Laumanns et al [15] recommend the truncation of the external set to a fixed size, one of their numerical studies retains all non-dominated individuals as recommended here.

The estimated Pareto front of E-SPEA is no longer susceptible to oscillatory or retreating behaviour; new members of the frontal set can only advance the estimated Pareto front or increase its resolution.

5 IMPLEMENTATION OF AN ACTIVE FRONTAL SET

Since the frontal set contains all the currently non-dominated decision vectors found, it may become very large as the search progresses. In order for E-SPEA to be computationally viable, efficient ways of selecting elements of F_t for binary tournament selection and of storing F_t to permit rapid query, insertion and deletion of its elements must be found. These problems and their solutions are now discussed.

5.1 Selection for binary tournament

In SPEA all members of the external set enter the binary tournament selection. In E-SPEA many more individuals than necessary are available for selection. However, uniform random selection of individuals from F_t artificially concentrates the search on already densely populated regions of the front. It is therefore helpful to select a number of representative individuals. An approach is to use the SPEA clustering method to find representatives, however, this proves to be too time consuming for large frontal populations. Here Partitioned Quasi-Random Selection (PQRS) is introduced as the selection routine used in E-SPEA. Suppose that M elite individuals are required for the binary tournament. In PQRS the objective space is partitioned into $M - D$ bins of equal width across one of the D objective dimensions, and an individual is selected at random from each of these bins. This ensures that individuals are selected uniformly across the extent of the front in the selected dimension. The objective dimension selected for partitioning rotates through the D dimensions with the generation, t. Pinned individuals, on the extremes of the front, are automatically entered into the binary tournament.

An example of PQRS in a $D = 2$ objective problem is shown in Figure 5. Here $M = 5$ individuals are to be selected from a frontal population of $\widehat{M} = 24$ (where $\widehat{M}$ denotes the current size of the frontal set). As 2 extremal individuals are passed straight to binary tournament selection, this leaves 3 more individuals to be selected. Having selected an objective coordinate, the frontal set is partitioned on that coordinate into $(M - D)$ equally spaced bins. In Figure 5 this can be seen to be three bins for the selected dimension, each spanning 1/3 of the range of the front on that dimension. An individual in each bin is selected by generating a uniformly distributed random number across the range of the bin, and selecting the closest individual. If a bin is empty (for instance due to a discontinuity in the Pareto front), additional individuals are found by randomly selecting from the entire frontal set. In addition, no individual is selected twice.

Note that in SPEA clustering is used to reduce the external set size before individuals are selected for binary tournament selection, in the E-SPEA the frontal set population is not reduced, PQRS only selects individuals for breeding and does not remove them from F_t.

Rapid selection from the frontal set is enabled by maintaining D balanced binary trees, one for each objective dimension. This means that each selection takes $\mathcal{O}\left(2\lg\widehat{M}\right)$ as opposed to $\mathcal{O}\left(\widehat{M}\right)$ for a linear search. Since the frontal set is constantly changing it is implemented using self balancing trees (e.g. AVL or Red-Black trees [16]).

Figure 5 shows how a two objective frontal set is stored in balanced binary trees for PQRS. In the example shown, when finding an individual with a given objective value the maximum number of comparisons needed is 4. If the sought value, say y_1^*, does not exist in the tree, then the closest value has to be selected. In this case two traversals will occur to find the two individuals that lie either side of y_1^*: the individual closest to y_1^* and lying in the correct bin is selected. If neither individual lies in the bin an individual is returned at random. If more than one individual has $y_1 = y_1^*$, one of them is returned at random.

5.2 Efficient storage of the frontal set

The second and greater constraint on using a large active frontal set is the number of comparisons that must be made with individuals in the frontal set at each generation. When the external set is small, for instance $M = 20$ (as in [8]), the time for a linear search of E_t is negligible. However, with no limits to the size of the frontal set in E-SPEA, the linear search of 5000 individuals (for instance) before assigning an individual as non-dominated, is simply too high to make the method practical. Therefore intelligent storage is needed before the frontal set approach is viable.

The frontal set must be searched at three distinct junctures:

1. When representative individuals are selected for binary tournament selection, as discussed in Section 5.1.

2. When strengths are assigned to external and internal set individuals prior to binary tournament selection. In E-SPEA strengths are calculated based on the representatives of F_t selected by PQRS. As there are relatively few of these, the strength calculation does not impose a great computational burden.

3. When individuals which are non-dominated with respect to P_t are compared with F_t to determine whether they dominate or are dominated by members of F_t. A data structure to facilitate searching F_t in logarithmic time is the *dominated tree*, which is now discussed.

5.3 Dominated trees

To determine whether an individual, $\mathbf{y} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_D(\mathbf{x}))$, should become a member of the frontal set, F,[1] it must first be checked that $\mathbf{y}$ is not dominated by any element of F. At the same time any elements of F that are dominated by $\mathbf{y}$ should be deleted from F. When the frontal set is small a simple linear search is sufficiently cheap to perform these checks. However, as the size of the frontal set grows the cost of querying the frontal set becomes prohibitive. In this section we describe two data-structures – *dominated trees* and *non-dominated trees* – for storing, rapidly querying, and updating the frontal set[2].

Here it is convenient to regard members of the frontal set and individuals from the internal population as points $\mathbf{y}$ in D-dimensional space. Geometrically, finding individuals in F that dominate $\mathbf{y}$ amounts to finding frontal individuals that lie to the 'south-west' or 'left and below' $\mathbf{y}$. More formally, the set of dominating individuals is:

$$\{\mathbf{z} \in F \; : \; \mathbf{z}_i < \mathbf{y}_i \text{ for all } 1 \leq i \leq D \text{ and } \mathbf{z}_j < \mathbf{y}_j \text{ for at least one } 1 \leq j \leq D\} \tag{10}$$

[1] Since the genetic generation plays no role here, the subscript t has been dropped.

[2] An example implementation of dominated and non-dominated trees is available from `http://www.dcs.ex.ac.uk/academics/reverson/moea`

It would be possible to use kd-trees [17, 16] or range trees [18, 16], but these are both suited to querying F for elements which lie in bounded (hyper-) rectangles. Priority trees, developed by McCreight [19], are suited to rectangular queries in which the rectangle is unbounded on a *single* side.

As noted in section 2, the 'dominates' relation imposes a partial order on individuals. However, since the elements of F are mutually non-dominating, this relation cannot be used directly to construct, for example, a binary tree to enable fast searching.

The dominated tree consists of an ordered list of *composite points* (usually stored as binary tree). Each composite point represents (upto) D elements of F and composite points are defined so that they are ordered by the weakly-dominates relation, $\preceq$. An example of a dominated tree is shown in figure 6.

The essential property of dominated trees is that the composite points are ordered:

$$\mathbf{c}_i \preceq \mathbf{c}_j \quad \text{iff } i > j \tag{11}$$

Usually, the stronger condition, $\mathbf{c}_i \prec \mathbf{c}_j \quad \text{iff } i > j$, will hold. In addition, if $\mathbf{c}_j \succ \mathbf{c}_i$ then the constituent points of $\mathbf{c}_j$ also dominate $\mathbf{c}_i$. Thus, for example, in Figure 6 the constituent points of $\mathbf{c}_4, \mathbf{c}_5$ and $\mathbf{c}_6$ dominate by $\mathbf{c}_3$. Note, however, that they do not necessarily dominate the constituent points of $\mathbf{c}_3$, namely $\mathbf{y}_3$ and $\mathbf{y}_6$.

It should be emphasised that the points forming the tree in Figure 6 do not form a non-dominated set. This is for expository purposes, because non-dominated sets of two-dimensional elements have the peculiar property that listing the points in order of increasing first coordinate (objective), y_1, is equivalent to listing them in order of decreasing second coordinate, y_2. With more than two objectives this is no longer the case and the points illustrated in figure 6 are more akin to the general case. The dominated tree also has applications to general sets (that is, not non-dominated sets), such as answering queries about enclosing rectangles; see, for example, [19, 20].

Further details regarding the construction and update of dominated trees can be found in Appendix A.

6 EXPERIMENTS

In order to evaluate the efficiency of E-SPEA, results of a comparison with SPEA are presented in this section. Deb has proposed a number of test objective functions for estimating the performance of MOEAs [12, 8]. In this section a brief critique of the Deb functions is presented, before five alternative multi-objective test functions are introduced.

The Deb test functions involve two objectives and have the following structure:

$$\begin{array}{lll} \text{Minimise} & T\left(\mathbf{x}\right) & = \left(f_1\left(x_1\right), f_2\left(\mathbf{x}\right)\right), \\ \text{where} & f_2\left(\mathbf{x}\right) & = g\left(x_2, \ldots, x_P\right) \cdot h\left(f_1\left(x_1\right), g\left(x_2, \ldots, x_p\right)\right), \\ \text{and} & \mathbf{x} & = \left(x_1, \ldots, x_P\right). \end{array}$$

Although the six Deb functions represent many different features and levels of difficulty, the first objective (f_1) is always a function solely of the first decision parameter (in fact for the first four test functions, $f_1\left(x_1\right) = x_1$). This simple form of f_1 means that when genes describing x_1 are initialised as uniformly distributed random numbers, the initial estimate of the Pareto front extends over the full range of f_1. Consequently optimisation chiefly consists of minimising f_2, rather than the combined minimisation of both objectives. For the purposes of this study, where part of the focus is on searching for the extent of the Pareto front, the Deb functions unfortunately are of limited use. Consequently, five new test functions are introduced, where all objectives are dependent on all decision parameters. All five are combinations of the following five base functions, $B_i\left(\mathbf{x}\right)$ (see Table 1).

- Base functions

$$B_1 = \sum_{i=1}^{P} \left| x_i - \frac{1}{3} \exp\left((i/m)^2 \right) \right|^{\frac{1}{2}} \tag{12}$$

$$B_2 = \sum_{i=1}^{P} \left(x_i - \frac{1}{2} \left(\cos\left(10\pi \left(i/m\right)\right) + 1 \right) \right)^2 \tag{13}$$

$$B_3 = \sum_{i=1}^{P} \left| x_i - \sin^2\left(i-1\right) \cos^2\left(i-1\right) \right|^{\frac{1}{2}} \tag{14}$$

$$B_4 = \sum_{i=1}^{P} \left| x_i - \frac{1}{4} \left(\cos\left(i-1\right) \cos\left(2\left(i-1\right)\right) + 2 \right) \right|^{\frac{1}{2}} \tag{15}$$

$$B_5 = \sum_{i=1}^{P} \left(x_i - \frac{1}{2} \left(\sin\left(1000\pi \left(i/m\right)\right) + 1 \right) \right)^2 \tag{16}$$

Where m=30 and $x_i \in [0, 1]$

The problem is therefore to use the SPEA and the E-SPEA to approximate the true Pareto fronts described by the five test functions in Table 1.

6.1 Comparing Pareto fronts

Comparison of Pareto front estimates is difficult as there are several ways in which a front can be inferior or superior to another. Indeed it is unlikely that any one measure will be sufficient to encompass all desired information when evaluating the output of an MOEA, if there was, then that measure would become the objective to minimise (or maximise), and the problem would become a single objective one. In this situation, where the objective properties of the true Pareto front are known *a priori*, a single objective optimiser should be used, using the performance measure as its objective.

Here a number of alternatives to the popular $\mathcal{C}$ metric [13, 8, 14] are discussed.

- The $\mathcal{C}$ metric.

$$\mathcal{C}(A, B) = \frac{\left|\{\mathbf{b} \in B :\ \exists\, \mathbf{a} \in A\,,\ \mathbf{a} \preceq \mathbf{b}\}\right|}{|B|}, \tag{17}$$

where A and B are two sets of decision vectors and $A, B \subseteq X$.

The $\mathcal{C}$ metric measures the fraction of members of B which are (strictly or weakly) dominated by members of A. As such it measures the quality of A with respect to B. When $\mathcal{C}(A,B) = 1$ all the individuals in B are dominated by solutions in A; $\mathcal{C}(A,B) = 0$ represents the situation in which none of the individuals in B are dominated by any of those in A.

It should be noted that $\mathcal{C}(A,B)$ is not technically a metric, since $\mathcal{C}(A,A) \neq 0$ and $\mathcal{C}(A,B)$ is not symmetrical in its arguments and it doesn't satisfy the triangle inequality. Furthermore, $\mathcal{C}$ has the following undesirable property: suppose that W is a non-dominating set and $A \subseteq W$ and $B \subseteq W$, then $\mathcal{C}(A,B)$ can take on any value in $[0,1]$.

In this study the following modified version of the $\mathcal{C}$ measure is used:

$$\tilde{\mathcal{C}}\,(A,B) = \frac{|\{\mathbf{b} \in B :\ \exists\, \mathbf{a} \in A\ ,\ \mathbf{a} \prec \mathbf{b}\}|}{|B|} \tag{18}$$

Now $\tilde{\mathcal{C}}(A,A) = 0$ and, in addition, it measures two mutually non-dominating sets as equivalent. That is, if A and B are each subsets of a non-dominating set, then $\tilde{\mathcal{C}}(A,B) = 0$.

Nevertheless, $\tilde{\mathcal{C}}$ and $\mathcal{C}$ fail to account for the either the difference in the extent of the fronts being compared or the uniformity of the distribution of points along the front. For example, Figure 7a illustrates two fronts with similar extent, but points describing A are uniformly distributed along the front whereas those describing B are clustered in one region. However, $\mathcal{C}(A,B) = \mathcal{C}(B,A) = \tilde{\mathcal{C}}(A,B) = \tilde{\mathcal{C}}(B,A) = 4/12$ even though elements of A dominate elements of B along the majority of their extents. In Figure 7b although B has much greater extent, $\mathcal{C}(A,B) = \tilde{\mathcal{C}}(A,B) = 2/12$, whereas $\mathcal{C}(B,A) = \tilde{\mathcal{C}}(B,A) = 0/12$.

As the some of the problems highlighted with the $\mathcal{C}$ and $\tilde{\mathcal{C}}$ measure show: there are a number of properties which are usually desired of estimated Pareto fronts. These include that the distance of the estimated Pareto front to the true Pareto front should be minimised, and that the extent of the estimated front be maximised (a wide range of solutions in objective space be returned). New measures which are designed to include such information are now introduced.

In [8] Zitzler et al. suggest a number of new measures to encapsulate this information. One of these, the $\mathcal{M}_3$ function, measures the extent of a front, and inspires the introduction in this study of the following $\mathcal{J}_1$ and $\mathcal{J}_2$ measures (the $\mathcal{M}_3$ function itself is not used as it only measures the extent of a single front).

- The $\mathcal{J}_1$ Measure.

$$\mathcal{J}_1(A,B) = \frac{1}{D}\sum_{i=1}^{D} \frac{\max\{|a_i - a_i'| \ ; \ \mathbf{a},\mathbf{a}' \in A\}}{\max\{|b_i - b_i'| \ ; \ \mathbf{b},\mathbf{b}' \in B\}} \tag{19}$$

The $\mathcal{J}_1(A,B)$ measure is an average ratio of the extents of two fronts in each objective dimension, where each objective is normalised by the extent of the front in B (the measure lies on the range $[0,\infty]$). If $\mathcal{J}_1(A,B) = 1$ the two fronts are on average proportionately equal in extent, whereas $J_1(A,B) > 1$ means that on average the extent of A is greater than the extent of B. Again however the $\mathcal{J}_1$ measure by itself does not provide all the contextual information that is desired, therefore the $\mathcal{J}_2$ measure is introduced and defined below, which should be used in tandem with it.

- The $\mathcal{J}_2$ Measure.

$$\mathcal{J}_2(A,B) = \frac{1}{D}\sum_{i=1}^{D} \frac{(\min\{b_i \ ; \ \mathbf{b} \in B\} - \min\{a_i \ ; \ \mathbf{a} \in A\})}{|\min\{a_i \ ; \ \mathbf{a} \in A\}| + 1} \tag{20}$$

The $\mathcal{J}_2$ measure confronts the problems caused by situations when a front may be of wider extent than another, but still dominated (where the minimum feasible value of each objective dimension is 0, the measure lies on the range $[-\max\{X\}, \max\{X\}]$, where $\max\{X\}$ is the sum of the maximum feasible values of each objective dimension divided by D). If $\mathcal{J}_2(A,B) = 0$ the average minimum objective values defined by the two sets are equal, a value in greater than 0 corresponds to the average minimum objective values defined by set A being less than B, and a value less than 0 corresponds to the average minimum objective values defined by set A being greater that B. Therefore where $\mathcal{J}_1(A,B)$ is greater than 1 and $\mathcal{J}_2(A,B)$ is greater than 0, the front defined by A is of a larger extent than that defined by B, and its minimum objective values are also on average lower. If however $\mathcal{J}_1(A,B)$ is greater than 1 and $\mathcal{J}_2(A,B)$ less than 0, a situation exists where A is wider than B, but this extra width is dominated by B.

- The $\mathcal{V}$ measure.

The final measure introduced in this study is similar conceptually to the performance measure used in [15], and is a measure of the objective space volume that is dominated by one front but not the other. Loosely $\mathcal{V}(A,B)$ is the fraction of the volume of the minimum hypercube containing both fronts that is strictly dominated by members of A but is not dominated by members of B (and therefore lies on the range $[0,1]$). An illustration of this provided in Figure 8 where two continuous fronts A and B have differing extents and also dominate each other in different region of the objective space.

$\mathcal{V}(A,B)$ is defined in the following manner. For any set of D-dimensional vectors Y, let Hy denote the smallest axis-parallel hypercube containing Y:

$$Hy = \left\{ \mathbf{z} \in \mathbb{R}^D \ : \ a_i \leq z_i \leq b_i \text{ for some } \mathbf{a},\mathbf{b} \in Y, \ i = 1,\ldots,D \right\} \tag{21}$$

Now denote by $h_Y(\mathbf{y}) : H_Y \longmapsto [0,1]^D$ the normalising scaling and translation that maps H_Y onto the unit hypercube. This transformation comes to remove the effects of scaling the objectives. Let

$$D_Y(A) = \left\{ \mathbf{z} \in [1,0]^D \ : \ \mathbf{z} \prec h_Y(\mathbf{a}) \ \text{ for some } \mathbf{a} \in A \right\} \tag{22}$$

be the set of points in the hypercube defined by Y which are dominated by the normalised A.

Then $\mathcal{V}(A,B)$ is defined as

$$\mathcal{V}(A,B) = \lambda\left(D_{A\bigcup B}(A) \setminus D_{A\bigcup B}(B)\right) \tag{23}$$

where $\lambda(A)$ denotes the Lebesgue measure of A [21].

Despite this rather cumbersome description $\mathcal{V}(A,B)$ and $\mathcal{V}(B,A)$ are easily calculated by Monte Carlo sampling of $H_{A\bigcup B}$ and counting the fraction of samples that are dominated exclusively by A or B. In this study sampling was terminated when the standard deviation (over the last 1000 samples) of the estimate fell below 10^{-5} - approximately 6×10^{5} samples for the objective functions used here.

The benefit of the volume measure $\mathcal{V}$ is that it will reward sets that are of greater extents when those extents are in front of the comparison set, and not when they are behind, it is not effected by the distribution of points across a front, and it also gives information regarding how far one set is (on average) in front of another.

Unfortunately the $\mathcal{V}$ measure, like the original $\mathcal{C}$ metric, has the property that, if W is a non-dominating set, and $A \subseteq W$ and $B \subseteq W$, $\mathcal{V}(A,B)$ and $\mathcal{V}(B,A)$ may be positive.

6.2 Results

The implementation of SPEA is almost identical to that in [8], the difference lying in the use of floating point representation of parameters in the individual chromosomes instead of 30 bit binary representation. In order to compare SPEA and E-SPEA the two EAs were each executed 30 times on each test problem, and the resultant non-dominated solutions saved at the end of each run. In the case of E-SPEA these were simply those individuals residing in F at the end of run, whereas an off-line store of the non-dominated solutions discovered by SPEA was kept. Each simulation was performed using the parameters shown below:

Number of generations : 500, 1000 & 2500

Search population size : 80

Max. external population size (SPEA) and

Max. No. of representative frontal individuals for breeding (E-SPEA) : 20

Crossover rate : 0.8

Mutation rate : 0.01

Single point crossover was used and the mutator variable was drawn from a zero-mean, symmetric, leptokurtic distribution (kurtosis ≈ 10) generated by the product of two uniform distributions covering the range [0,1], and a Gaussian distribution with unit variance and zero mean. In each of the 30 different runs E-SPEA and SPEA were initialised from identical decision vector populations. Initialisation of decision vectors was from $U(0,1)$. The experiments were repeated with the EAs terminated after 500, 1000 and 2500 generations.

Results on the five test functions, in relation to the $\tilde{\mathcal{C}}$, $\mathcal{J}_1$, $\mathcal{J}_2$ and $\mathcal{V}$ measures, are presented in Figure 9 in the form of *box plots*. These box plots provide distribution information in relation to the 30 samples for each generation length. In each plot three boxes are presented, which (from left to right) represent the sample runs for 500, 1000 and 2500 generations. The solid box represents the central 50% of the data, the top and bottom of the box are the upper and lower quartiles, with lines from the box top and bottom stretching to the maximum and minimum sample values.

The E-SPEA and SPEA are tested for significant variation (on the various error measures) using the nonparametric Wilcoxon Signed Ranks Test [22] at the 2% level (1% in each tail)[3]. After 500

[3] t-tests cannot be used as the samples cannot reasonably be assumed to be drawn from Normal distributions, neither are the samples independent (each pair being correlated). The Wilcoxon Signed rank test does not assume normality, and its independence assumption is only in relation to the paired values (that each pair be independent in relation to all other pairs).

generations the E-SPEA is seen to be significantly better on the extent measures of $\mathcal{J}_1$and $\mathcal{J}_2$ on all five test problems, as well as the dominated volume measure $\mathcal{V}$. After 1000 generations the E-SPEA is also seen to be significantly better according to the $\tilde{\mathcal{C}}$ measure. Even so, after 2500 generations $\tilde{\mathcal{C}}\left(S,E\right) > 0$ and $\mathcal{V}\left(S,E\right) > 0$, meaning that in some region the set returned by SPEA is in front of that returned by E-SPEA. This is due to the SPEA's search being concentrated on a smaller section of the front than E-SPEA, due to the shrinking front effect in SPEA's external set. As SPEA's search is focused on a smaller objective volume than E-SPEA it has the local advantage of recombining proportionally more individuals in from this space than E-SPEA.

However, even though this is the case, $\tilde{\mathcal{C}}\left(S,E\right)$ and $\mathcal{V}\left(S,E\right)$ are generally decreasing as the search progresses: the local search advantage of not pinning the extremes is outweighed by the inefficiency caused by the oscillating external front.

These results support the hypothetical effects of shrinking and oscillatory fronts as presented earlier in the study. The search process of both SPEA and E-SPEA, in terms of genetic manipulation used, are identical, as are their approaches to the Pareto ranking of individuals, the number of search individuals used, and the number of 'elite' individuals used for breeding each generation. It is only through the implementation of pinning and the active use of a frontal set that E-SPEA is seen to perform significantly better on the $\mathcal{J}_1$ and $\mathcal{J}_2$ measures which measure extent (and therefore shrinking effects), and the $\widetilde{\mathcal{C}}$ and $\mathcal{V}$ measures which measure dominance (and therefore oscillation effects).

An example of the performance is provided in Figure 10, where the two estimated Pareto fronts generated by SPEA and E-SPEA after 2500 generations on the $F1$ test function are shown (each front contains all the non-dominated individuals from each of the 30 runs of the MOEA, with the dominated individuals removed from the two union sets). The front generated by E-SPEA is shown to extend further than that of SPEA (highlighted by the two minimum hypercubes that contain each front), and to also lie ahead of the SPEA front for a large portion of that front's extent.

The potential size of a frontal set is determined by the nature of the problem and the granularity of the decision parameters used in the genotypic representation of solutions. Figure 11 shows the increase in frontal set size up to 10000 generations for a single run of the E-SPEA on the F1, F2, F3, F4 and F5 test functions used in this study. Although only limited inference can be made from this Figure with regard to the domain as a whole, it is encouraging to note that the growth of frontal set sizes all follow linear (or less than linear) paths with gradients from 2 to 0.1. The maximum possible growth of the frontal set is N extra individuals per generation. In this study $N = 80$, therefore the growth of the fastest growing frontal population (that of the four objective problem $F5$) is only a 40th of this rate.

7 STOPPING CRITERIA

Robust stopping criteria are largely missing from the MOEA literature. Beale and Cook [4] include a fitness-based stopping criterion in their study, in which the algorithm is terminated if the fittest individual has remained unchanged for 1000 consecutive generations. As Coello [9] points out, MOEAs since Schaffer [5], which carry a set of non-dominated individuals, are usually terminated after a fixed number of generations, or the population is monitored at intervals and a decision made on a visual basis.

The use of a frontal set which can only 'advance' however allows a number of robust stopping methodologies to be introduced. Examples of these are:

- When no individual that dominates a member of F is discovered after a given number of consecutive generations.

This is similar to the approach taken by Beale and Cook [4], however in E-SPEA there is a set of elite solutions instead of a single elite individual. It is to be emphasised that this sort of criterion can fail with an external set which is prone to oscillation.

Note that new individuals may have been found that are non-dominated by the frontal set - and therefore are inserted into F. However, these individuals can be seen to be filling in the front (increasing its resolution), as opposed to pushing it forward.

- When there has been no change in the extremal values for a given number of consecutive generations.

The previous stopping criterion, when taken by itself, may lead to sub-optimal stopping in that, although the front may have ceased moving forwards, it may still be moving outwards towards the extremes. This second criterion takes this form of search into account and can usefully be combined with the previous criterion.

- When the average distance in objective space between neighbouring individuals in F reaches a specific threshold.

The practitioner may wish the front to be defined to a particular resolution, therefore they may not solely wish to stop the search process after the front has finished moving, but also after this resolution is reached. With two objectives this can be achieved by calculating the maximum over F of the nearest neighbour distances[4]. With more than two objectives a similar termination criterion can be defined based on the maximum area of any triangle of a Voronoi tessellation [16] of F.

In practice the third criterion alone may lead to stopping before a good estimate of the true Pareto front has been found, however in conjunction with the first two criteria a good estimate of the true Pareto front to any desired resolution can be achieved.

If a practitioner prefers a small number of evenly distributed individuals to be returned after algorithm termination, the final frontal set may be clustered using the method employed in standard SPEA described earlier (the computational cost is not too high as it only needs to be performed once).

[4] This can fail in the pathological case that the true front contains isolated points in objective space.

The stopping criteria defined above are not readily applied to methods which do not the use active frontal set because, even if a dormant offline store is used, these methods are susceptible to oscillating active estimates of the Pareto front, which may cause spurious early termination.

8 CONCLUSION

A number of improvements have been made to the SPEA multi-objective algorithm, in order to rectify observed problems, to which a large number of other population based MOEAs are also susceptible. These extensions have first been justified theoretically, and then supported by empirical evidence on on a number test problems, using new measures introduced in this study, as well as measures used previously in the literature. The impact of the extensions is shown to increase with time.

New data structures based on balanced binary trees have been introduced in order to make the use of an active frontal set practical, and allay concerns of time costs. The concerns of robust algorithm termination have also been addressed, an area that has been largely dormant in recent research in the MOEA domain.

Current areas of research of interest to the authors include the parallel implementation of MOEAs and the use of structural information in local regions of objective space to improve search efficiency.

E-SPEA is currently being applied to the evolution of neural network forecasting models [23], information retrieval, and the optimisation of safety critical systems.

APPENDIX A

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

$$\mathbf{c}_{1,1} = \max_{F} \mathbf{y}_{m,1} \tag{24}$$

This point $\mathbf{y}_m$ is now associated with $\mathbf{c}_1$ and deleted from F. Likewise, the second coordinate of $_1$ is the maximum second coordinate of the points remaining in F, thus $\mathbf{c}_{1,2} = \max_F \mathbf{y}_{m,2}$. In this manner each coordinate of $\mathbf{c}_1$ is defined in turn. The process is then repeated to construct $\mathbf{c}_2$ and subsequent points until F is empty. Note that in construction of the final composite point, (that is, the composite point that dominates all other composite points) F may be empty before the D coordinates of the composite point have been defined. As illustrated by $\mathbf{c}_7$ in Figure 6, in this case the $\mathbf{y}_m$ ($\mathbf{y}_{13}$ in Figure 6) already comprising the composite point serve to define the coordinates in any unfilled dimensions.

Since (except possibly for the dominating composite point) D elements of F are used in the construction of each composite point, the maximum number of composite points is $\hat{M}/D + D - 1$.

Query. Given a test point $\mathbf{q}$, the properties of dominated trees can be used as follows to discover which points in F dominate $\mathbf{q}$. Although the dominated tree may conveniently be implemented as a binary tree, the query procedure is most easily described in terms of an ordered list of $L = \hat{M}/D + D - 1$ composite points. First, the list is searched to find the indices h and l of composite points $\mathbf{c}_h$ and $\mathbf{c}_l$ that dominate and are dominated by $\mathbf{q}$ respectively:

$$l = \begin{cases} 0 \text{ if } \mathbf{c}_1 \prec \mathbf{q} \\ \max\{i \,:\, \mathbf{c}_i < \mathbf{q}\} \text{ otherwise} \end{cases} \tag{25}$$

and

$$h = \begin{cases} L+1 \text{ if } \mathbf{c}_i \succ \mathbf{q} \\ \min\{i \,:\, \mathbf{c}_i \succ \mathbf{q}\} \text{ otherwise} \end{cases} \tag{26}$$

Also donote by $\mathbf{c}_H$ the 'least' composite point that strictly dominates $\mathbf{c}_h$:

$$H = \min\{i : \mathbf{c}_i \prec \mathbf{c}_h\} \tag{27}$$

For the query point illustrated in Figure 6, $l = 2$, $h = 5$ and $H = 6$. (Note that it is not necessarily true that $H = h + 1$.) Since $\mathbf{c}_h \succ \mathbf{q}$ it is clear that all the constituent points of the composite points $\mathbf{c}_i$ that dominate $\mathbf{c}_h$, $H \leq i \leq L$, also dominate $\mathbf{q}$. (Note that the constituent points of $\mathbf{c}_h$, and indeed and $\mathbf{c}_i$ that only weakly dominate $\mathbf{c}_h$, need not dominate $\mathbf{q}$; in Figure 6 $\mathbf{c}_5 \succ \mathbf{q}$, but $\mathbf{y}_9 \not\succ \mathbf{q}$.) Also, since $\mathbf{q} \succ \mathbf{c}_l$ and the constituent points of $\mathbf{c}_l$ have at least one coordinate equal to a coordinate of $\mathbf{c}_l$, it may be concluded that $\mathbf{q}$ is not dominated by any of the constituent points of $\mathbf{c}_1, \ldots \mathbf{c}_l$. Each constituent point $\mathbf{c}_i$ with $l < i < H$ must be checked individually to determine whether it dominates $\mathbf{q}$; in Figure 6 the points $\mathbf{y}_3, \mathbf{y}_6, \mathbf{y}_4, \mathbf{y}_8, \mathbf{y}_9, \mathbf{y}_{10}$ must be individually checked.

Note that when determining whether $\mathbf{q}$ is to be included in F, it can be immediately rejected if $h < L$ because it is certainly dominated by at least one of the constituents of $\mathbf{c}_L$.

Since the composite points are arranged as a sorted list, determination of l and h takes $\mathcal{O}(\lg(\hat{M}/D))$ comparisons each. Hence the total number of comparisons required is $\mathcal{O}(2\lg(\hat{M}/D) + K)$, where K is the number of points that have to be checked individually. Clearly, certain configurations of F and $\mathbf{q}$ can result in all elements of F being checked — in linear time. However, such arrangements are seldom encountered in practice and the logarithmic query time permits the use of very large frontal sets.

If it is determined that $\mathbf{q}$ is to be included in F (because it is not dominated by any element of F), those elements of F which are dominated by $\mathbf{q}$ must be identified and deleted from F. Queries about which elements of F are dominated by $\mathbf{q}$ can be answered using the dominated tree, however, it may be inefficient. This is because although $\mathbf{q} \succ \mathbf{c}_i$ for $i \leq l$, $\mathbf{q}$ need not dominate the constituent points of these $\mathbf{c}_i$ and the constituent points must therefore be checked individually. Thus in Figure 6, $\mathbf{q} \succ \mathbf{c}_2 \succ \mathbf{c}_1$, but $\mathbf{y}_5$ and $\mathbf{y}_7$ are not dominated by $\mathbf{q}$. The *non-dominated tree* is a data structure which permits this sort of query to be answered efficiently.

Non-dominated are analogous to dominated trees. A non-dominated tree consists of a ordered composite points, $\mathbf{c}_i \succeq \mathbf{c}_j$ iff $i < j$, with the additional property that if $\mathbf{c}_i \succ \mathbf{c}_j$, then the constituent points of $\mathbf{c}_j$ are also dominated by $\mathbf{c}_i$. An example of a non-dominated tree is shown in Figure 12. Construction and querying of non-dominated trees is analogous to dominated trees and they are not discussed further here.

Insertion and deletion. Elements are continually added and deleted from the frontal set during the course of an optimisation. It is therefore important that the data structure used to support F can be modified dynamically. Online insertion of a new point $\mathbf{y}$ is straight-forwardly accomplished by creating a new composite point, say $\mathbf{c}'$. Let $\mathbf{c}_j$ be the composite point that dominates $\mathbf{y}$, with $\mathbf{c}_j \succ \mathbf{y} \succ \mathbf{c}_{j-1}$. Then the coordinates of $\mathbf{c}'$ are determined by:

$$\mathbf{c}'_k = \max(\mathbf{y}_k, \mathbf{c}_{j,k}) \qquad k = 1, \dots, D \tag{28}$$

Figure 13 shows the dominated tree resulting from the insertion of a new point $\mathbf{y}_{14}$ in the tree shown in Figure 6. Note that the tree resulting from insertion is less than optimal in the sense that a point (e.g., $\mathbf{y}_6$) may contribute to more than one composite point.

Deletion of a point $\mathbf{y}$ from the tree is slightly more complicated because the composite point to which it contributed, say $\mathbf{c}_j$, must continue to be dominated by $\mathbf{c}_{j+1}$. Let Y_j be the set of points which contribute to $\mathbf{c}_j$, and $\overline{Y}_j = Y_j \setminus \mathbf{y}$. Then, if $\mathbf{y}$ defined coordinate k of $\mathbf{c}_j$, the new kth coordinate is given by:

$$\mathbf{c}_{j,k} = \max\left(\max_{\mathbf{q} \in \overline{Y}_j} (\mathbf{q}_k), \mathbf{c}_{j+1,k}\right) \tag{29}$$

Deletion of $\mathbf{y}_4$ from the tree shown in Figure 6 is illustrated in Figure 13. Note that if $\mathbf{y}$ contributes to more than one composite point, then each of the composite points must be dealt with in turn beginning with the one which dominates all the others; on the other hand, if $\mathbf{y}$ is the sole contributing point to $\mathbf{c}_j$ (e.g., $\mathbf{c}_{13}$ in Figure 6) then $\mathbf{c}_j$ is deleted from the tree.

Cleaning. Insertion and deletion operations lead to some points contributing to more than one composite point. The dominated trees therefore contain more composite nodes than necessary and hence increased time is needed to search them. This can be alleviated by periodically 'cleaning' the tree in the following manner. Starting with $j = 1$, points contributing to $\mathbf{c}_j$ are deleted from $\mathbf{c}_{j+1}, \ldots, L$ in the manner just described. j is then incremented to sweep through the entire list of composite points, thus ensuring that any point contributes to exactly one composite point. A cleaned tree is shown in Figure 13. In practice it is not efficient to clean the tree following every insertion and deletion, but occasional cleaning may be triggered when the number of composite points exceeds a threshold, say $1.5 \times (\hat{M}/D + D - 1)$.

ACKNOWLEDGMENTS

Jonathan Fieldsend gratefully acknowledges support from Invensys Climate Controls Europe and an Exeter University Studentship. The software for this work used the GAlib genetic algorithm package, written by M. Wall at the Massachusetts Institute of Technology (`http://lancet.mit.edu/ga`) and the SPEA code provided online by E. Zitzler at the Swiss Federal Institute of Technology, as the base of the original comparative SPEA algorithm (`http://www.tik.ee.ethz.ch/~zitzler/testdata.html#source`).

Table 1: Test function composition.

	Objectives			
	f_1	f_2	f_3	f_4
Test Function F1	B_1	B_2	–	–
Test Function F2	B_3	B_4	–	–
Test Function F3	B_2	B_3	B_5	–
Test Function F4	B_1	B_4	B_5	–
Test Function F5	B_1	B_3	B_4	B_5

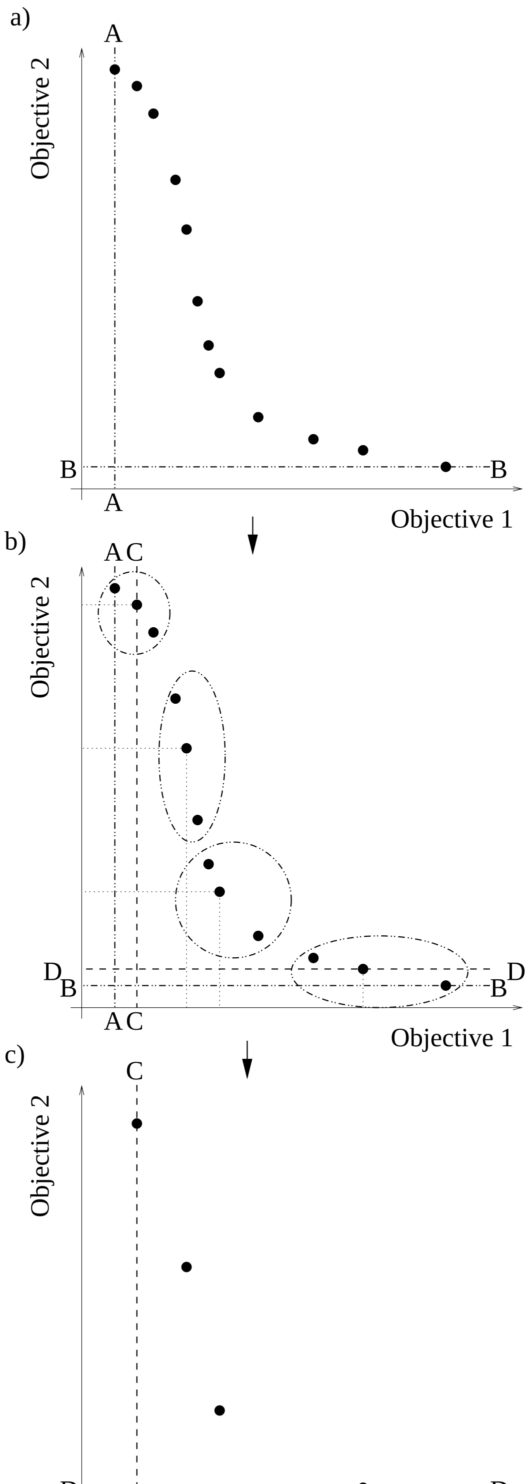


Figure 1: a),b) and c). Clustering in SPEA. The 12 individuals in (a) are to be clustered to find 4 representative individuals to form the external set at the next generation. Clusters are shown in (b) and the resultant external set is shown in (c).

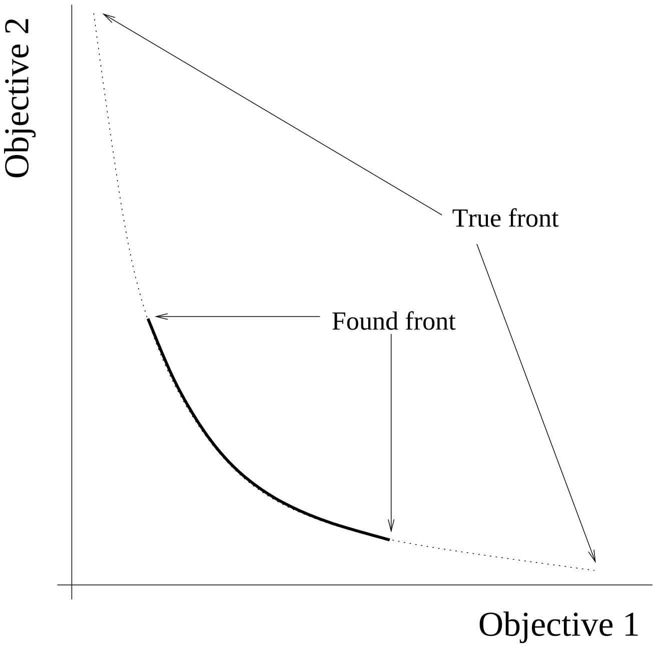


Figure 2: The 'Shrinking' Pareto front effect, where the final front returned lies on a local section of the true front.

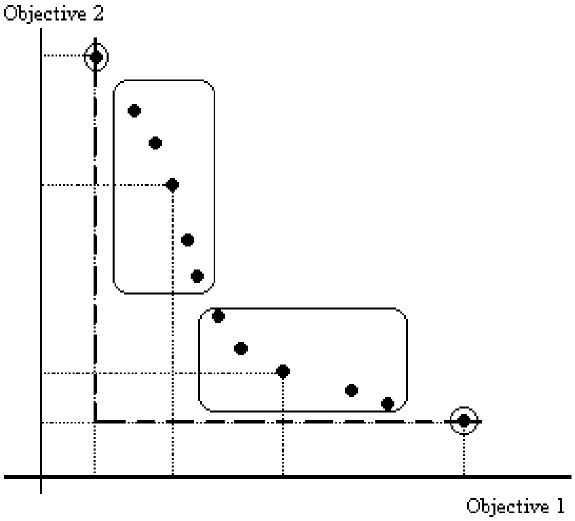


Figure 3: Pinning of the extremal individuals in Figure 1, and resultant clusters.

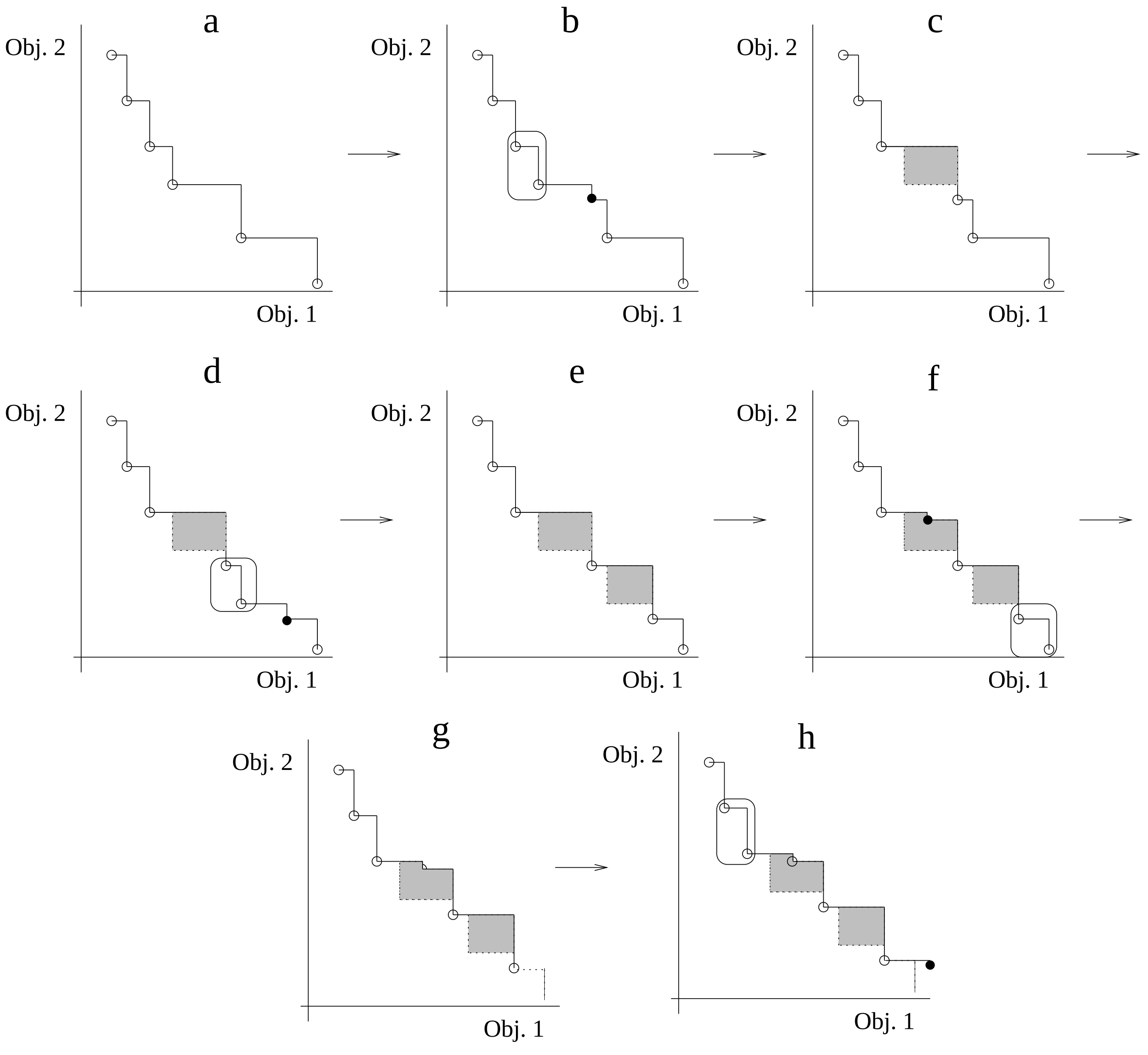


Figure 4: Example of a retreating estimated Pareto Front, as stored in the External set of the SPEA.

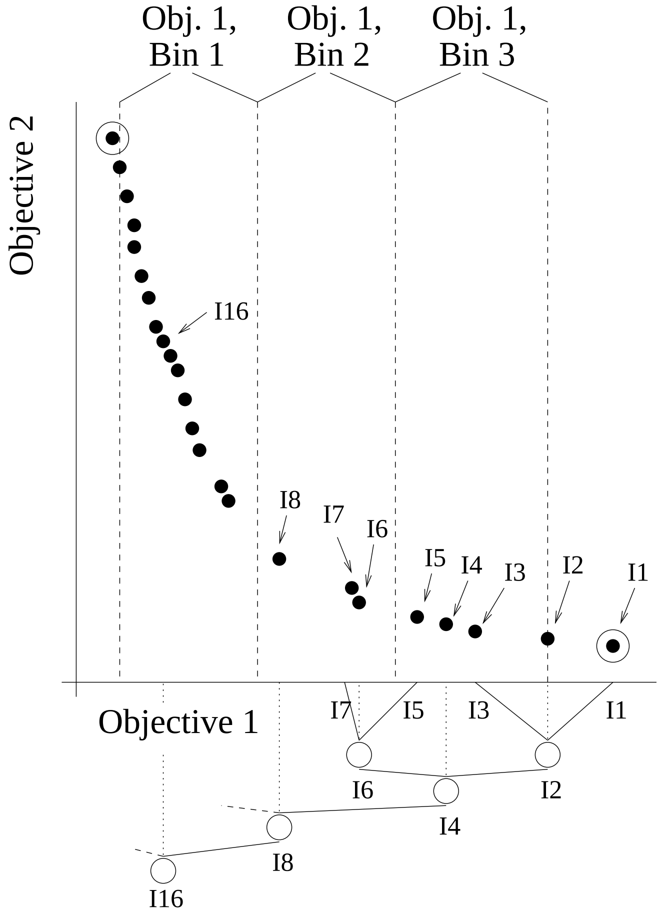
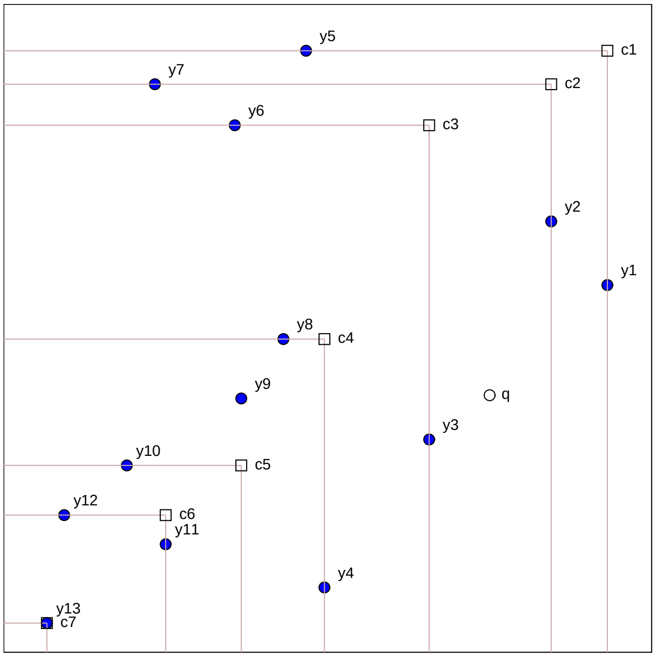


Figure 5: Two objective example of Partitioned Quasi-Random Selection, with the Objective 1 dimension partitioned. $M = 5$ representative individuals required so selection from 3 bins $(M - D)$, after pinning of extremal values. Storage in balanced binary tree also illustrated, with node and leaf individuals I1 to I8 marked, as well as root node I16.



Composite node	Constituent points
$\mathbf{c}_1$	$(\mathbf{y}_1, \mathbf{y}_5)$
$\mathbf{c}_2$	$(\mathbf{y}_2, \mathbf{y}_7)$
$\mathbf{c}_3$	$(\mathbf{y}_3, \mathbf{y}_6)$
$\mathbf{c}_4$	$(\mathbf{y}_4, \mathbf{y}_8)$
$\mathbf{c}_5$	$(\mathbf{y}_9, \mathbf{y}_{10})$
$\mathbf{c}_6$	$(\mathbf{y}_{11}, \mathbf{y}_{12})$
$\mathbf{c}_7$	$(\mathbf{y}_{13}, \mathbf{y}_{13})$

Figure 6: **Dominated tree.** Top: 13 points $\mathbf{y}_m$ in two dimensions and the composite points $\mathbf{c}_i$ (squares) forming a dominated tree. The open circle, q, marks a query point. Bottom: Composite nodes listed as ordered by $\prec$.

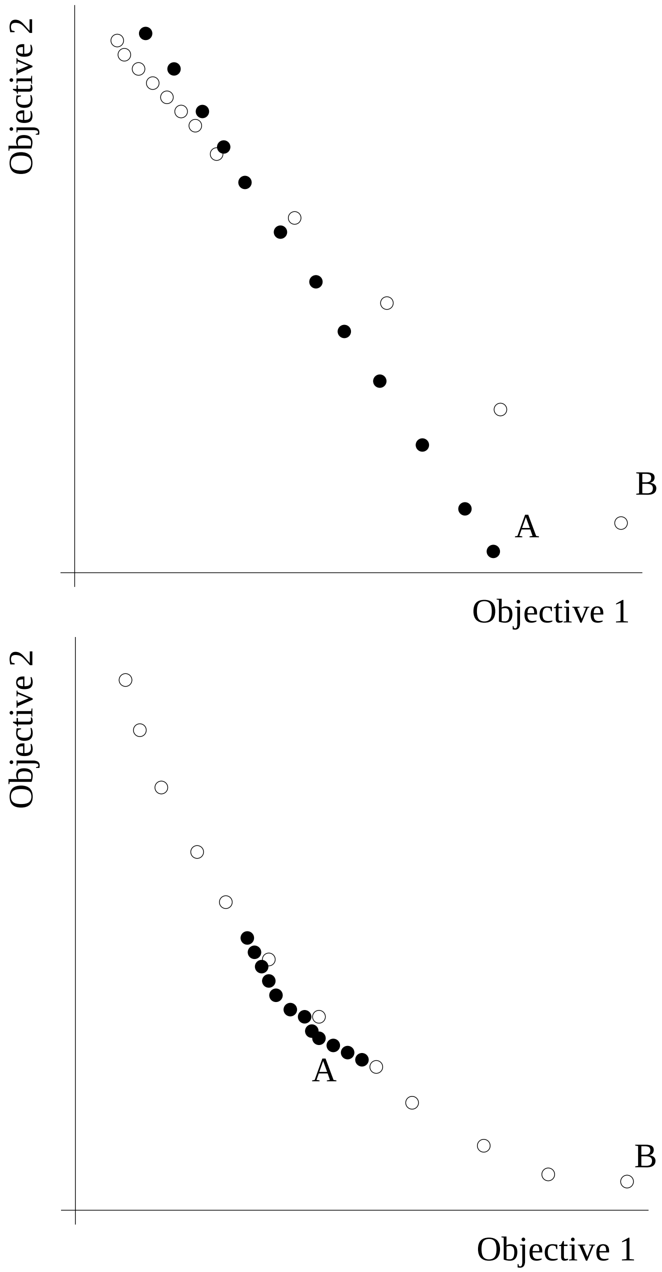


Figure 7: (a) Illustrates dense and evenly distributed estimated Pareto fronts. Front A dominates a larger extent of Front B, but both fronts dominate an equal number of each others' constituent members. (b) Illustrates estimated Pareto fronts of differing extents. Front B is of far greater extent than A, but will receive a lower $\mathcal{C}$ metric value.

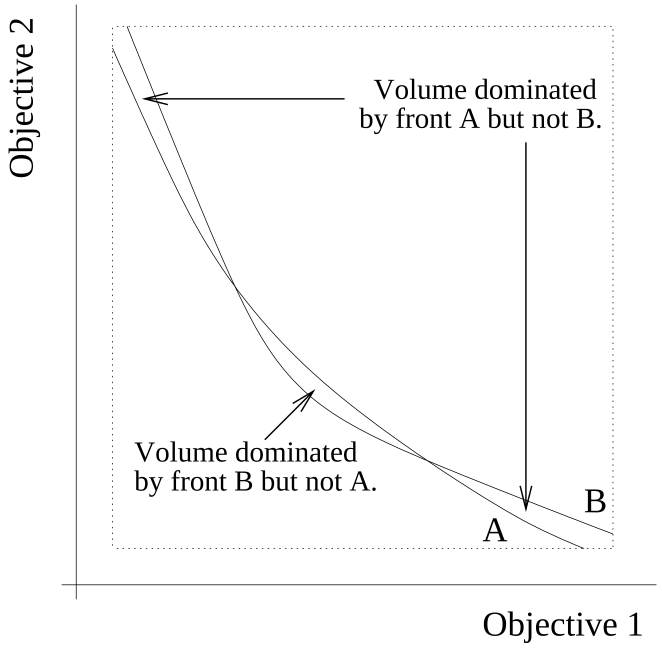

Objective 2
Volume dominated by front A but not B.
Volume dominated by front B but not A.
B
A
Objective 1

Figure 8: Two dimensional illustration of minimum surrounding hypercube volume dominated by two fronts (hypercube boundaries marked with dashed lines).

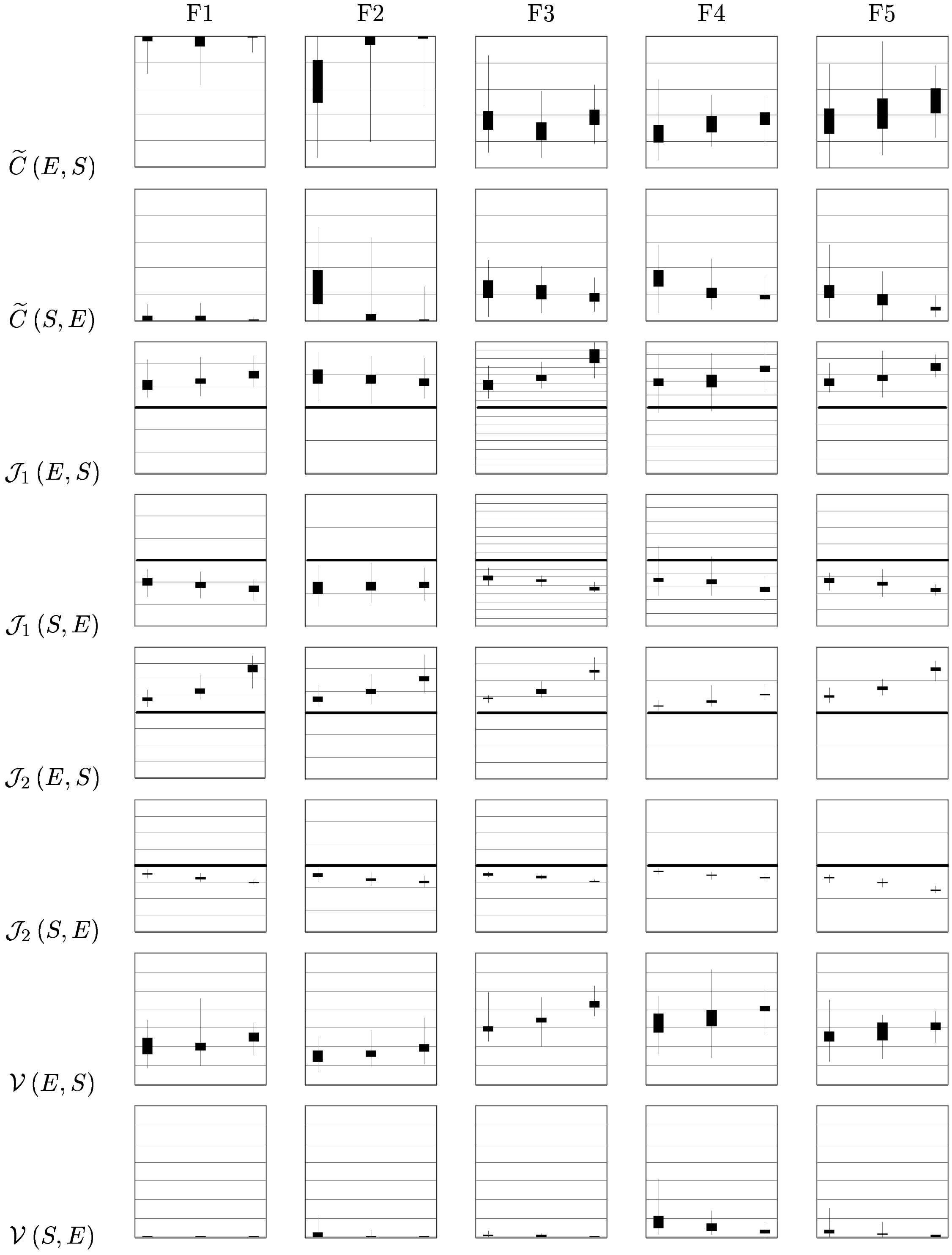


Figure 9: Box plots of $\widetilde{C}$ measure, $\mathcal{J}_1$ measure, $\mathcal{J}_2$ measure and $\mathcal{V}$ measure. Grid lines in $\widetilde{C}$ box-plots mark 0.2 intervals on the range [0,1]. Grid lines in $\mathcal{J}_1$ box plots mark 0.1 intervals, with the mid value 1.0 in bold. Grid lines in $\mathcal{J}_2$ box-plots mark 0.5 intervals, with the mid value 0.0 in bold. Grid lines in $\mathcal{V}$ box plots mark 0.02 (2%) intervals (boxes shown cover the range [0%,14%]). All results are significantly different on the Wilcoxon test (5% two-tailed), except that of the $\widetilde{C}$ measure for 500 generations, in relation to the $F3$, $F4$ and $F5$ test functions.

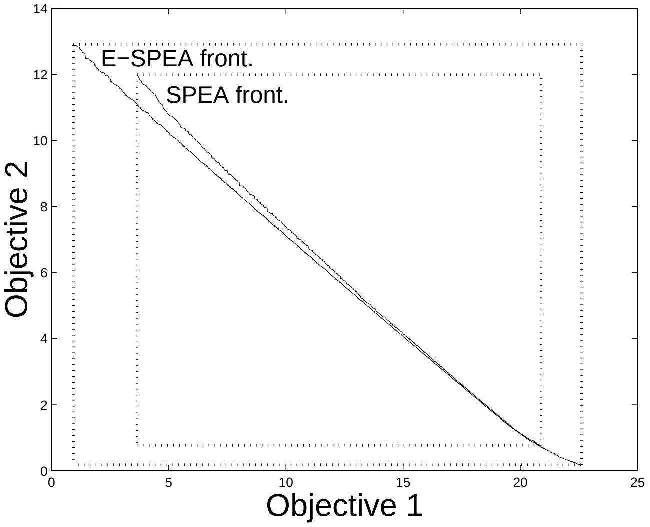


Figure 10: Empirical example. Figure shows the two fronts generated by E-SPEA and SPEA on test function $F1$ after 2500 generations (the non-dominated individuals from the 30 separate runs).

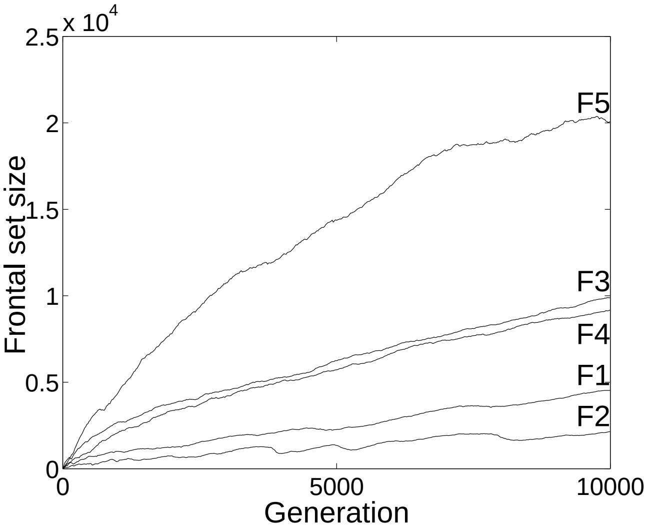


Figure 11: Growth of frontal set $\widehat{M}$. Shows the increase in frontal set size of a single E-SPEA run up to 10000 generations on the F1, F2, F3, F4 and F5 test problems. It should be mentioned again that search of the frontal set for a particular individual is only $\mathcal{O}\left(2\lg\widehat{M}\right)$ in PQRS and $\mathcal{O}(2\lg(\hat{M}/D)+K)$ in dominated trees.

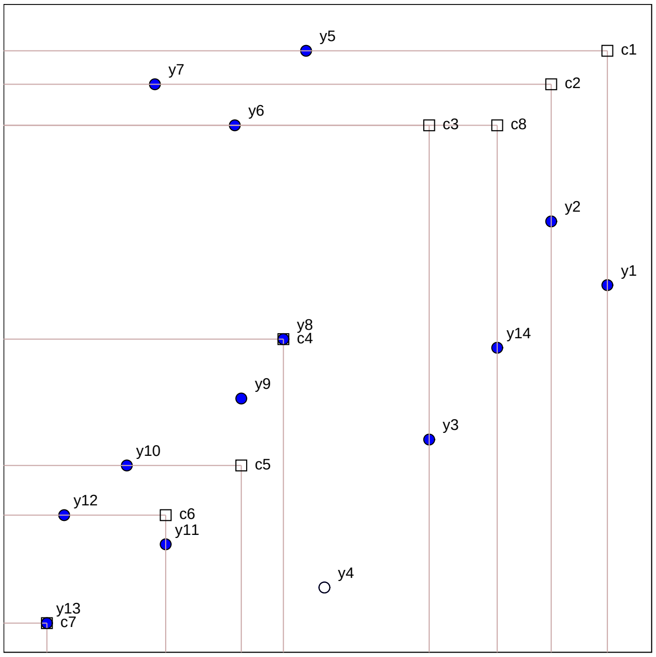


Figure 12: **Non-dominated tree.** Left: 13 points $\mathbf{y}_m$ in two dimensions and the composite nodes $\mathbf{c}_i$ (squares) forming a non-dominated tree.

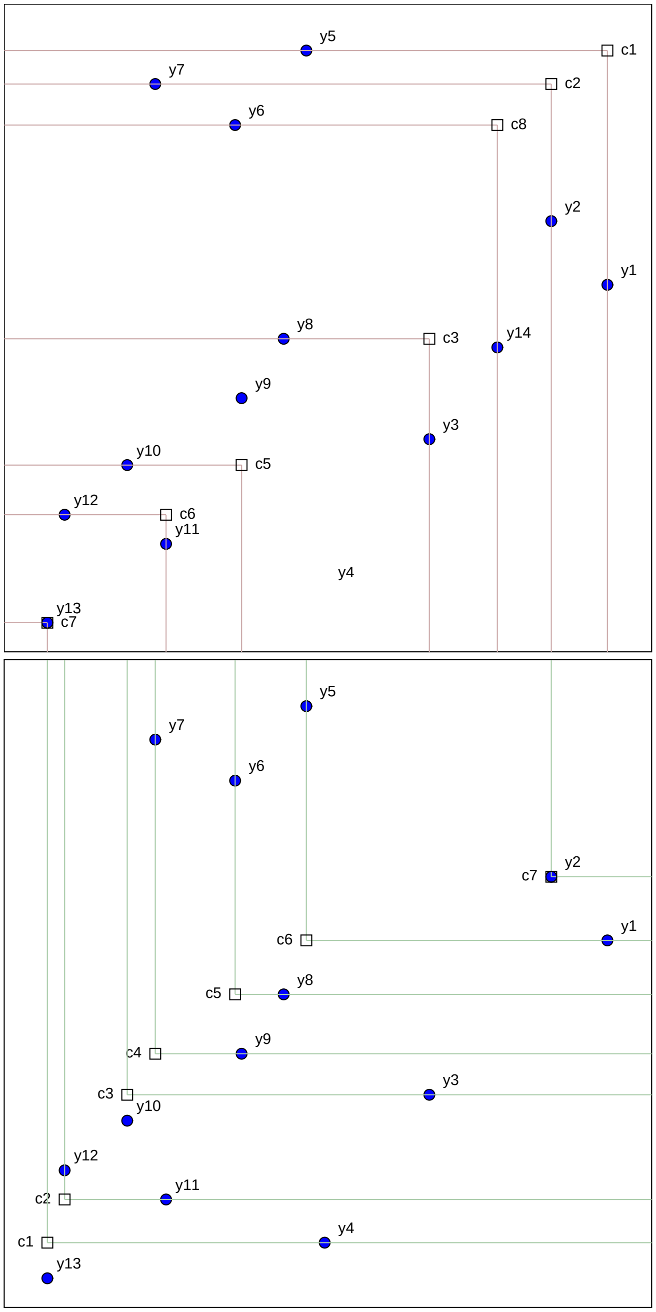


Figure 13: **Insertion and deletion.** Top: Dominated tree shown in Figure 6 following the insertion of $\mathbf{y}_{14}$, and the deletion of $\mathbf{y}_4$. Bottom: Dominated tree shown at the top following 'cleaning'.

References

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

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

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

[4] G.O. Beale and G. Cook. Optimal Digital Simulation of Aircraft via Random Search Techniques. *AIAA Journal of Guidance and Control*, 1(4):237–241, 1978.

[5] 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.

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

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

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

[9] 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.

[10] 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.

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

[12] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.

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

[14] 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.

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

[16] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational geometry : algorithms and applications.* Springer, Berlin, 1997.

[17] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, pages 509–517, 1975.

[18] J.L. Bentley and J.H. Friedman. Data structures for range searching. *Computing Surveys*, 111(4):398–409, 1979.

[19] E.M. McCreight. Priority search trees. *SIAM Journal on Computing*, 14:257–276, 1985.

[20] E.M. McCreight. Efficient algorithms for enumerating intersecting intervals and rectangles. Technical Report CSL-80-9, Xerox Palo Alto Research Center, Palo Alto, CA, USA, 1980.

[21] F. Jones. *Lebesgue Integration on Euclidean Space.* Jones and Bartlett, Boston, 1993.

[22] F. Wilcoxon and R.A. Wilcox. *Some Rapid Approximate Statistical Procedures.* Lederle Labs, New York, 1964.

[23] J.E. Fieldsend and S. Singh. Pareto Optimisation of Multi-Objective Neural Network Regression Models. Technical report, University of Exeter, 2001.