An Analysis of Multiobjective Optimization within Genetic Algorithms

Peter J. Bentley
Division of Computing
and Control Systems Engineering
The University of Huddersfield
Huddersfield HD1 3DH, U.K.
p.bentley@hud.ac.uk

Jonathan P. Wakefield
Division of Computing
and Control Systems Engineering
The University of Huddersfield
Huddersfield HD1 3DH, U.K.
j.p.wakefield.hud.ac.uk

Abstract

This paper focuses on the problem of how to rank a population of solutions into order of fitness within a genetic algorithm for multiobjective optimization applications. Attention is paid to the fact that the set of acceptable solutions to a problem is usually only a small sub-set of all Pareto-optimal solutions to the problem. Two key concepts essential to the solution of this problem are identified and explained: range-independence and importance. Six methods (three old and three new) for solving this multi-fitness ranking problem are described in detail and applied to five test problems for comparison: four established test functions and an example solid object design task with ten separate constraints and objectives. Results show that all five methods allow the generation of Pareto-optimal solutions, but all have different distributions of solutions within the Pareto-optimal range(s). The bias of each distribution and the resulting quality of solutions generated by each method is examined and compared. The paper concludes that the new ranking method 'Sum of Weighted Global Ratios' (SWGR) created as part of this work allows the most consistent generation of acceptable solutions, whilst also being fully independent of the multiobjective problem.

Keywords

genetic algorithm, multiobjective optimization, ranking, range-independence, importance, solution distribution

1. Introduction

The genetic algorithm (GA) has been growing in popularity over the last few years as more and more researchers discover the benefits of its adaptive search. Many papers now exist, describing a multitude of different types of genetic algorithm, theoretical and practical analyses of GAs and huge numbers of applications for GAs (Goldberg, 1989; Holland, 1992). A substantial proportion of these applications involve the evolution of solutions to problems with more than one criterion. More specifically, such problems consist of several separate objectives, with the required solution being one where some or all of these objectives are satisfied to a greater or lesser degree. Perhaps surprisingly then, despite the large numbers of these multiobjective optimization applications being tackled using GAs, only a small proportion of the literature explores exactly how they should be treated with GAs.

With single objective problems, the genetic algorithm stores a single fitness value for every solution in the current population of solutions. This value denotes how well its corresponding solution satisfies the objective of the problem. By allocating the fitter members of the population a higher chance of producing more offspring than the less fit members, the GA can create the next generation of (hopefully better) solutions. However, with multiobjective problems, every solution has a number of fitness values, one for each objective. This presents a problem in judging the overall fitness of the solutions. For example, one solution could have excellent fitness values for some objectives and poor values for other objectives, whilst another solution could have average fitness values for all of the objectives. The question arises: which of the two solutions is the fittest? This is a major problem, for if there is no clear way to compare the quality of different solutions, then there can be no clear way for the GA to allocate more offspring to the fitter solutions.

1.1 Defining a Fit Solution

The approach most users of GAs favour to the problem of ranking such populations, is to weight and sum the separate fitness values in order to produce just a single fitness value for every solution, thus allowing the GA to determine which solutions are fittest as usual. However, as noted by Goldberg: "...there are times when several criteria are present simultaneously and it is not possible (or wise) to combine these into a single number." (Goldberg 1989). In other words, the separate objectives may be difficult or impossible to manually weight because of unknowns in the problem. Additionally, weighting and summing could have a detrimental effect upon the evolution of acceptable solutions by the GA (just a single incorrect weight can cause convergence to an unacceptable solution). Moreover, some argue that to combine separate fitnesses in this way is akin to comparing completely different criteria; the question of whether a good apple is better than a good orange is meaningless.

The concept of Pareto-optimality helps to overcome this problem of comparing solutions with multiple fitness values. A solution is Pareto-optimal (i.e., Pareto-minimal, in the Pareto-optimal range, or on the Pareto front) if it is *not dominated* by any other solutions. As stated by Goldberg:

Definition 1.1. A vector $\mathbf{x}$ is partially less than $\mathbf{y}$, or $\mathbf{x} <_p \mathbf{y}$ when:

$$(\mathbf{x} <_p \mathbf{y}) \Leftrightarrow (\forall_i)(\mathbf{x}_i <= \mathbf{y}_i) \wedge (\exists_i)(\mathbf{x}_i < \mathbf{y}_i)$$

$\mathbf{x}$ *dominates* $\mathbf{y}$ when $\mathbf{x} <_p \mathbf{y}$. (Goldberg, 1989)

However, it is quite common for a large number of solutions to a problem to be Pareto-optimal (and thus be given equal fitness scores). This may be beneficial should multiple solutions be required, but it can cause problems if a smaller number of solutions (or even just one) is desired. Indeed, for many problems, the set of solutions deemed acceptable by a user will be a small sub-set of the set of Pareto-optimal solutions to the problems (Fonseca and Fleming 1995b). Manually choosing an acceptable solution can be a laborious task, which would be avoided if the GA could be directed by a ranking method to converge only on acceptable solutions. For this work, an *acceptable solution* (or champion solution) is defined:

Definition 1.2 A solution is an *acceptable solution* if it is Pareto-optimal and it is considered to be acceptable *by a human*.

1.2 Background

Existing literature seems to approach this ranking problem using methods that can be classified in one of three ways: the aggregating approaches, the non-Pareto approaches and the Pareto approaches. Many examples of aggregation approaches exist, from simple 'weighting and summing' (Syswerda and Palmucci, 1991; Goldberg 1989) to the 'multiple attribute utility analysis' (MAUA) of Horn and Nafpliotis (1993). Of the non-Pareto approaches, perhaps the most well-known is Schaffer's VEGA (Schaffer 1984, 1985), who (as identified by Fonseca and Fleming, 1995a) does not *directly* make use of the actual definition of Pareto-optimality. Many other non-Pareto methods have been proposed (Linkens and Nyongesa, 1993; Ryan 1994; Sun and Wang, 1992). Finally the Pareto-based methods, proposed first by Goldberg (1989) have been explored by researchers such as Horn and Nafpliotis (1993) and Srinivas and Deb (1995). In addition, many researchers are now introducing 'species formation' and 'niche induction' in an attempt to allow the uniform sampling of the Pareto set (Goldberg 1989; Horn and Nafpliotis, 1993). For a comprehensive review, see Fonseca and Fleming (1995a).

1.3 Aims of the Paper

The problem of ranking a population of solutions into order of fitness within a GA is an often overlooked, but fundamental problem when using a GA to search for solutions to multicriteria problems. The concept of Pareto-optimality allows a broad definition of which solution is fitter than another, but not all Pareto-optimal solutions are acceptable solutions.

Additionally, despite the existence of literature on the subject, there appears to have been little exploration of the actual nature of this multiobjective solution ranking problem. Many researchers point out the difficulties of handling noncommensurable objectives and then give their own multiobjective optimization algorithms, often it seems, giving little thought to whether their methods actually solve the true problem at all. All too often, such algorithms seem to have been created with most of the emphasis on *whether* the method will work, and little on *why* the method works.

Consequently, this paper will initially focus on the difficulties posed by these problems to GAs, and will explore exactly why separate criteria can cause problems in a genetic algorithm. A technique to guide the GA to converge on the smaller subset of acceptable solutions will be introduced. In the light of this, six different ranking methods will then be described, explored and compared in detail: three aggregating variants ('sum of weighted objectives' and two novel alternatives), one novel non-Pareto approach, one non-Pareto approach based on Schaffer's VEGA, and one Pareto approach (Goldberg's 'non-dominated sorting'). As well as assessing the quality of solutions produced, this paper will examine the previously unknown distribution of solutions produced in the Pareto-optimal range(s) by each method.

2. Range-Independence

In nature, every living creature must satisfy a large number of objectives sufficiently in order to be successful (e.g. avoid predators, find food, survive illnesses, reproduce). However, nature cannot (and has no need to) determine precisely which creature will be more successful (or fitter) than another. Whilst nature does sometimes attempt to improve the chances of successful creatures passing on their genes by allowing the physically stronger members of a group to breed more than weaker ones (e.g. male lions, deer, walruses), criteria such as physical strength can only be indirect approximations to the overall fitness (ability to produce good offspring) of the creature. Indeed, the criteria used for mating selection sometimes seems to bear little relation to the fitness of the creature (e.g. the tail of the peacock). This does not matter greatly, for in nature, the definition of a 'successful' creature is one which has managed to produce potentially successful offspring. (Note that this is *not* a tautology of 'Survival of the Fittest' - the fact that a creature has survived doesn't necessarily make it genetically fit.) Thus the true 'fitness' of a creature tackling the problem of life, can only be determined after its death.

With genetic algorithms, however, a solution is not considered 'fit' if it has produced good offspring - quite the reverse: a solution is allocated a greater chance of having offspring if it is identified as being fit. In these algorithms, fitness now becomes a measure of how well the solution satisfies mathematical objectives and bears little relation to the real measure used in nature. Therefore, because a mathematical function is being used to judge fitness, this problem of ranking multiobjective solutions is purely artificial.

Consequently, the problem has more to do with mathematics than nature. Throughout the evolution by the GA, every separate objective (fitness) function in a multiobjective problem will return values within a particular range. Although this range may be infinite in theory, in practice the range of values will be finite. This 'effective range' of every objective function is determined not only by the function itself, but also by the domain of input values that are produced during evolution. These values are the parameters to be evolved by the GA and their exact values are normally determined initially by random, and subsequently by evolution. The values are usually limited still further by the coding used, for example 16 bit sign-magnitude binary notation per gene only permits values from -32768 to 32768. Hence, the *effective range* of a function can be defined:

Definition 2.1.	The *effective range* of $f(x)$ is the range from $\min(f(x))$ to $\max(f(x))$ for all values of x generated in one run of the GA

Although occasionally the effective range of all of the objective functions will be the same, in most more complex multiobjective tasks, every separate objective function will have a different effective range (i.e. the function ranges are noncommensurable; Schaffer 1985). This means that a bad value for one could be a reasonable or even good value for another, see Fig. 1. If the results from these two objective functions were simply added to produce a single fitness value for the GA, the function with the largest range would dominate evolution (a poor input value for the objective with the larger range makes the overall value much worse than a poor value for the objective with the smaller range).

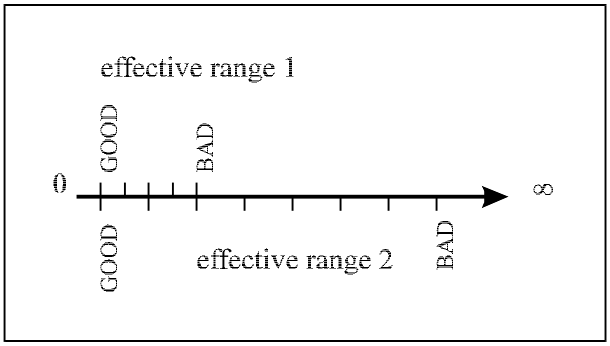

Figure 1. Different effective ranges for different objective functions (to be minimised)

For example, consider the two objective functions:

$$f_{11} = x^2$$
$$f_{12} = (x - 2)^2 / 1000$$

(both to be minimised).

Given a non-optimal input value, the output value from f_{11} will normally be three orders of magnitude worse than that from f_{12} (i.e. the second function will be approximately one thousand times closer to the minimum of zero). As can be seen in the simplest of tests, if the outputs from both were simply summed, the first function would completely dominate the second, resulting in the effective evolution of a good solution only to the first function.

Thus, the only way to ensure that all objectives in a multiobjective problem are treated equally by the GA is to ensure that all the effective ranges of the objective functions are the same (i.e. to make all the objective functions commensurable), or alternatively, to ensure that no objective is directly compared to another. In other words, either the effective ranges must be converted to make them equal, and a range-dependent ranking method used, or a range-independent ranking method must be used. Typically, range-dependent methods (e.g. 'sum of weighted objectives', 'distance functions', and 'min-max formulation') require knowledge of the problem being searched to allow the searching algorithm to find useful solutions (Srinivas and Deb, 1995). Range-independent methods require no such knowledge, for being independent of the effective range of each objective function makes them independent of the nature of the objectives and overall problem itself. Thus, multiobjective ranking methods that are *range-dependent* or *range-independent* can be defined:

Definition 2.2. Given objective function(s) of a problem: $f_{1..n}(x)$
and a solution vector to the problem: **s**

A multiobjective ranking method is *range-dependent* if the fitness of **s** changes when the effective range(s) of $f_{1..n}(x)$ change (and **s** is scaled correspondingly).

A multiobjective ranking method is *range-independent* if the fitness of **s** *does not* change when the effective range(s) of $f_{1..n}(x)$ change (and **s** is scaled correspondingly).

For example, the standard 'sum of weighted objectives' method favoured by so many, uses the weights to make the effective domains of each objective equal, then provides a single fitness value by summing the resulting values. This is a range-dependent method, for it relies completely on the weights being set precisely for every problem. Should any of the objectives be changed, or the allowable domain of input values be changed (perhaps by a change in coding, or seeding the initial population with anything other than random values), then these weights may have to be changed.

Alternatively, the non-dominated sorting method, and variants of it, is a range-independent method. It requires no weighting of the objective values, for the fitness values from each objective function are never directly compared with each other. Only values from the same objective are ever compared in the process of determining the non-dominance of solutions (Goldberg 1989). For complex multiobjective problems, this range-independence is extremely advantageous: good results do not depend on the ability of the user to fine-tune weights correctly. However, a disadvantage of non-dominated sorting is that all Pareto-optimal solutions are considered equally good, regardless of what the user actually regards as being acceptable.

Hence, there is one other vital, and usually overlooked requirement that a good ranking method should satisfy: the ability to increase the importance of some objectives with respect to others in the ranking of solutions, to allow search to be directed to converge on the smaller subset of acceptable solutions.

3. Importance

On separate occasions, many researchers have independently noted that with highly complex search problems, searching efficiency can be increased, and time can be reduced, by increasing the importance of a particular part, or objective(s) of that problem (Dowsland, 1995; Marett and Wright, 1995). This is often achieved either by introducing objectives to the search algorithm one at a time (or in distinct 'stages') with the most important first, or by simply weighting the most important objectives more heavily. Indeed, experience shows that many users of GAs and the 'sum of weighted objectives' ranking method are inadvertently increasing the importance of certain objectives without being aware of it, as they fine-tune their weights to improve evolution.

Intentionally determining which objectives are more important in a problem can be a matter of debate, but to improve evolution time, it seems that often the best results are gained by making the most difficult to satisfy objectives the most important. However, some problems require that certain objectives have differing levels of importance just to allow evolution of an acceptable solution. (For example, the optimization of an electronic device has the design criteria: cost, speed, size and power consumption. For some devices, a low cost is overwhelmingly important, for others, a high speed is of greatest importance.)

Aggregation ranking methods (typically range-dependent) usually guide the GA to converge upon a single 'best compromise' solution. For the purposes of this paper, the *best compromise* solution is defined:

Definition 3.1. A *best compromise* solution is the solution with the sum of (weighted) objective fitnesses minimised.

However, with additional guidance in the form of importance weightings, this best compromise solution can be made the same as the required solution, allowing the GA to converge directly to an acceptable solution. Thus, producing a single best compromise solution is not always a disadvantage. However, to accurately set the values of importance, a range-independent method is perhaps more desirable since a range-dependent method requires objectives to be weighted twice - once to make the function ranges commensurable, and once to specify increased importance.

Nevertheless, the more favoured ranking methods do not employ aggregation (and typically are range-independent). They are usually used with some form of niching and speciation method to allow the GA to generate not one, but a range of non-dominated (Pareto-optimal) solutions. (Niching can also help the quality of solutions by preventing excessive competition between distant solutions; Goldberg 1989.) The user is then required to select the preferred solution from this range of different solutions. However, particularly for problems with many objectives, only a small proportion of Pareto-optimal solutions may be acceptable solutions. This means that even when hundreds of different solutions are generated by the GA, there can be no guarantee that an acceptable solution will be among them. Moreover, for such large problems, it is not always feasible to allow the user to pick the preferred solution from a truly representative range of Pareto-optimal solutions: the number to be considered may be too large. Thus, the ranking method needs further information, to guide the algorithm to converge more closely to truly acceptable solutions *within* the range of Pareto-optimal solutions. This information is 'importance' - by specifying which objectives must be satisfied more than others, the GA can converge more closely to acceptable solutions, not just Pareto-optimal solutions.

Significantly, importance can be used in this way, regardless of what the individual objective functions represent. Hence, objective functions that represent wildly different things can be judged against each other. Often, when two functions represent different things, regardless of the degree of similarity between their effective ranges, they are, perhaps improperly, called noncommensurable. To clarify this term, for this work, *commensurable* functions are defined:

Definition 3.2. Two or more functions are *commensurable* if the difference between the effective ranges of the functions is insignificant (i.e., the differences between the minimum and maximum of each must be negligible), *regardless of what these functions represent.*

Hence, given a problem with two commensurable objective functions, whatever each one represents (be it cars or carrots), solution vectors to the problem can have their fitnesses precisely set, using the relative importance values of the objectives.

For example, consider the problem of packing a bag before going mountaineering. In this simplified example, the person has to choose between the amount of climbing equipment and the amount of food to be packed in the bag. How much of each should be packed? Many researchers would state that the two are non-commensurable and cannot be directly compared by a computer, and so would present a human user with a number of alternative solutions to choose from. Clearly, in reality, the ideal solution depends on the length of time of the trip, and the difficulty of the climb. If the trip is to take two days, and will involve only a hike in some hills, then more food is required than climbing equipment. However, if the climb will involve an hour scaling a vertical cliff, then more climbing equipment is required than food. In other words, a human picks a solution based on the relative *importance* of the two objectives. Moreover, there is no good reason not to specify these relative importance values for the computer, and let the computer pick the *same solution* (without the need for a human to consider potentially hundreds of different Pareto-optimal solutions). Hence, importance can be defined as:

Definition 3.3. *Importance* is a simple way to give a ranking method additional problem-specific information, in order to direct a GA to converge to acceptable solutions within a smaller subset of the Pareto-optimal range, by favouring those solutions closer to the optima of functions with increased importance, in proportion to this increased importance.

Unfortunately, there is no easy way to increase the importance of one objective in relation to another, without the two objectives being directly compared to each other. In other words, whilst it is simple to specify increased importance with a range-dependent method such as 'sum of weighted objectives' (just increase the weights), with a range-independent method such as non-dominated sorting, specifying importance is more complex. (Fonseca forces a kind of importance with his 'preference articulation' method, (Fonseca and Fleming 1995b) but this requires detailed knowledge of the ranges of the functions themselves, involves human interaction, and is not a continuous guide to evolution.) Thus, alternative methods of ranking multiobjective solutions are required, that are ideally range-independent, allow the easy specification of importance, and can accurately judge fitness levels of solutions.

4. Multiobjective Ranking Methods

There follows descriptions of six different ranking methods. The first two are the most commonly used methods: the range-dependent 'weighted sum' (aggregation) method and the range-independent Pareto non-dominated sorting. The next three are novel range-independent methods, developed in an attempt to allow importance to be specified with such methods. The techniques used within these methods are not new, but they have as yet been rarely, if at all, used to rank multiobjective populations within a genetic algorithm. Finally, the sixth method is a range-independent method based on Schaffer's VEGA. Algorithms can be found in the appendix.

Method 1: Sum of Weighted Objectives (SWO)
This is perhaps the most commonly used method because of its simplicity. All separate objectives are weighted to make the effective ranges equivalent (and to specify importance) and then summed to form a single overall fitness value for every solution. These values are then used by the GA to allocate the fittest solutions a greater chance of having more offspring. (Because of the similarity in nature and performance between this method and many of the other 'classical' methods (Srivinas and Deb, 1995), only this classical method will be described and explored in detail.)

Method 2: Non-Dominated Sorting (NDS)
Described by Goldberg (1989), this range-independent method and variants of it are commonly used. The fitnesses of the separate objectives are treated independently and never combined, with only the value for the same objective in different solutions being directly compared. Solutions are ranked into 'non-dominated' order, with the fittest being the solutions dominated the least by others (i.e. having the fewest solutions partially less than themselves). These fittest can then be allocated a greater probability of having more offspring by the GA.

Method 3: Weighted Average Ranking (WAR)
This is the first of the alternative ranking methods proposed. The separate fitnesses of every solution are extracted into a list of fitness values for each objective. These lists are then individually sorted into order of fitness, resulting in a set of different ranking positions for every solution for each objective. The average rank of each solution is then calculated, with this value allowing the solutions to be sorted into order of best average rank. Thus, the higher an average rank a solution has, the greater its chance of producing more offspring. Since all objective fitnesses are treated separately, this method is range-independent. This technique allows the specification of importance by the weighting of average ranking values for each solution.

Method 4: Sum of Weighted Ratios (SWR)
This is the second of the novel ranking methods proposed for GAs and is basically an extension to SWO (method 1). The fitness values for every objective are converted into ratios, using the best and worst solution in the current population for that objective every generation. More specifically:

$$fitness\_ratio_i = \frac{(fitness\_value_i - \min(fitness\_value))}{(\max(fitness\_value) - \min(fitness\_value))}$$

This removes the range-dependence of the solutions, and they can be weighted (for the setting of importance) and summed to provide a single fitness value for each solution as with the first method.

Method 5: Sum of Weighted Global Ratios (SWGR)
This method is the third of the novel proposed ranking methods for GAs, and is a variation of SWR (method 4). Instead of the separate fitnesses for each objective in every solution being converted to a ratio using the *current* population best and worst values, the *globally* best and worst values are used. Again the importance of individual objectives can be set by weighting the appropriate values.

Method 6: Weighted Maximum Ranking (WMR)
This ranking method is based on Schaffer's VEGA (Schaffer 1984, 1985). VEGA forms lists of fitness values of each solution for each objective. The fittest *n* solutions from each list are then extracted, and random pairs are selected for reproduction. This is equivalent to WAR (method 3) except that the *maximum* rank of each solution for all objectives is used to determine the overall rank, instead of the average. Importance levels can be set in a similar way to those in WAR. Note that the additional heuristic used by Schaffer to encourage 'middling' values (Schaffer 1984) was not implemented in WMR.

5. Application of the Ranking Methods: Test Functions *F*1~*F*4

5.1 Test Functions

To explore and compare the performance of the six ranking methods, they were applied in turn to four different test functions: f_1 to f_4. The first three are identical to those used by Schaffer (1984,1985), whilst f_4 is identical to Fonseca's f_1 (Fonseca and Fleming 1995b). Each function was chosen to represent a different class of function (i.e., each has different numbers of Pareto-optimal ranges and/or best compromise solutions). All functions are to be minimised, see Table 1 and Fig. 2.

FUNCTION:	DESCRIPTION:
$f_1 = x_1^2 + x_2^2 + x_3^2$	A single-objective, three parameter function with an optimal at (0,0,0).
$f_{21} = x^2$ $f_{22} = (x - 2)^2$	A simple twin-objective single parameter function with a Pareto-optimal range between 0 and 2, and a best compromise solution of 1.
$f_{31} = -x$ where $x <= 1$ $-2 + x$ where $1 < x <= 3$ $4 - x$ where $3< x <= 4$ $-4 + x$ where $4 < x$ $f_{32} = (x - 5)^2$	A twin-objective single parameter function with two disjoint Pareto-optimal ranges: 0 to 2 and 4 to 5. This has a single best compromise solution of 4.5
$f_{41} = 1 - \exp(-(x_1 - 1)^2 - (x_2 + 1)^2)$ $f_{42} = 1 - \exp(-(x_1 + 1)^2 - (x_2 - 1)^2)$	A twin-objective function this time with two parameters. This has a single Pareto-optimal range along the line (-1,1) to (1,-1), but two equal best compromise solutions at the optima of each function: (-1,1) and (1,-1).

Table 1. The four test functions used to compare the ranking methods.

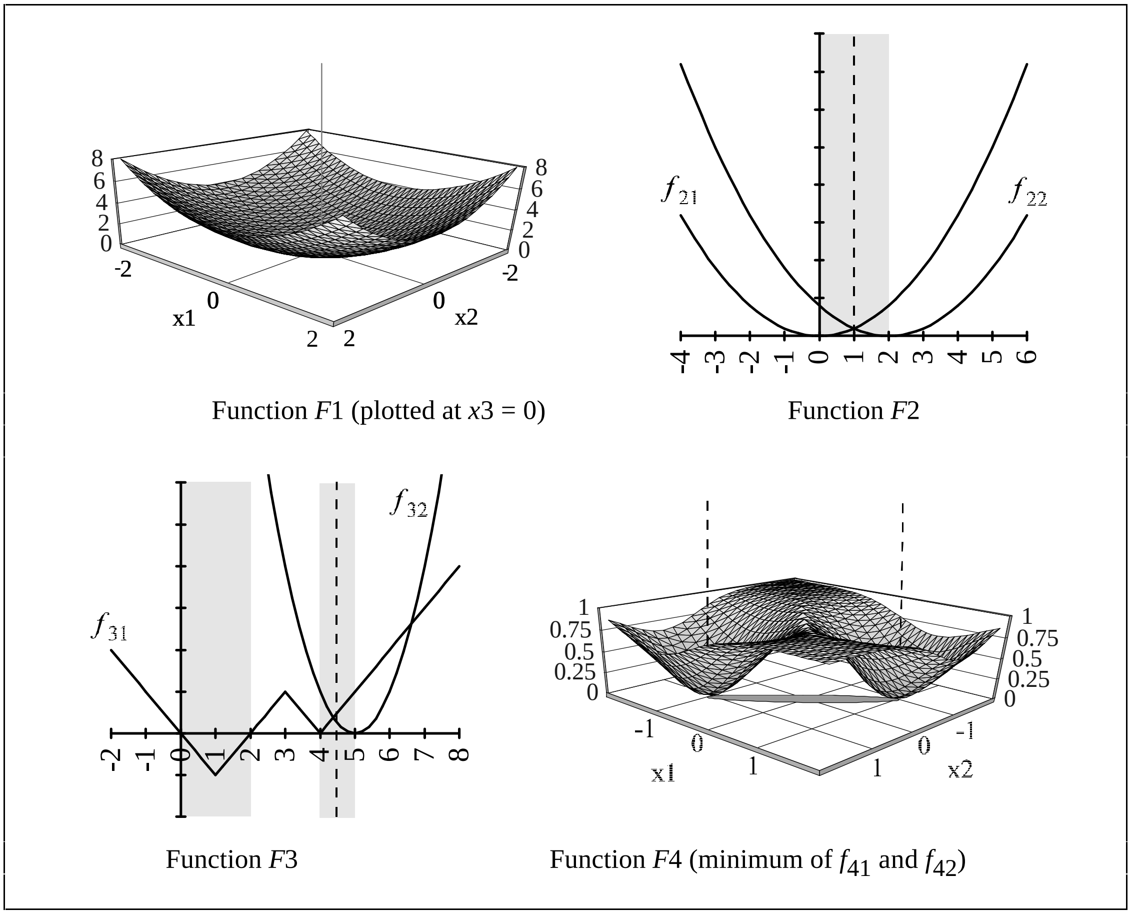

Figure 2. Graphs of the four test functions with Pareto-optimal ranges shown by grey shaded regions and best compromise solutions marked with dotted lines.

All five methods were used with a basic genetic algorithm using binary coding, a population of 50, and running for 100 generations. Probability of crossover was 1.0, probability of mutation was 0.05. Although this GA used elitist selection techniques, with all of the ranking methods described in this paper it is possible to use alternatives.

To the authors knowledge, there has been no previous work investigating the distribution of solutions generated with multiobjective ranking methods. Hence, the distributions produced by methods 1-6 for each function were calculated by running the GA between 1,000 and 10,000 times (depending on the function). It was assumed that the distribution of solutions produced by a series of runs of this algorithm would not differ significantly from the distribution of solutions obtained by an algorithm with niching or other speciation techniques.

5.2 Evolved Results: *F*1

The first experiment performed with each method was simply to allow the GA to minimise f_1. This function was used to validate that each method would rank solutions to single-objective problems correctly (as was done for VEGA by Schaffer, 1985). As expected, every method allowed the GA to converge on, or very near to, the optimal solution of (0,0,0), every time. (The distributions of solutions for this function are all at a single point and hence are not shown.)

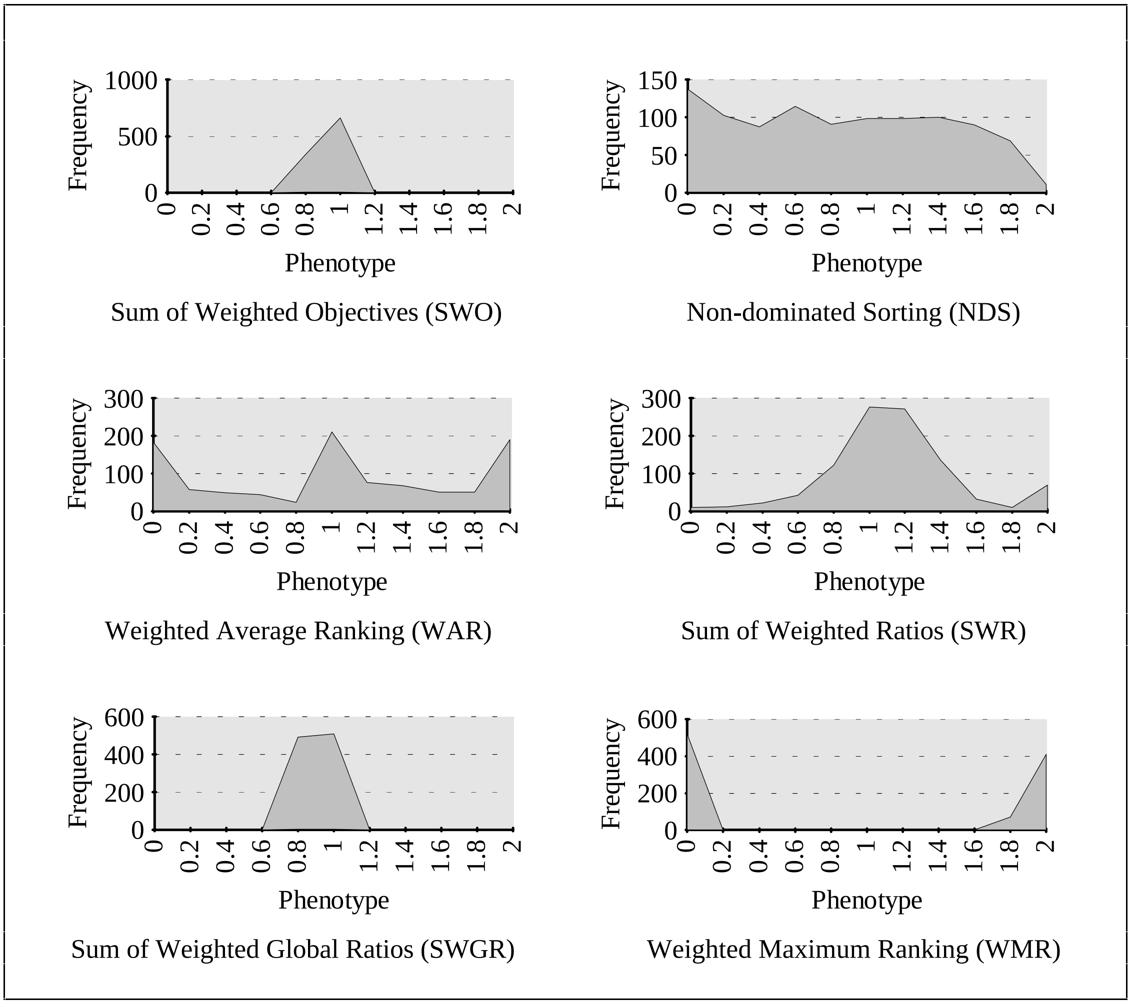

Figure 3. Distributions of solutions within the Pareto-optimal range for function F_2.

5.3 Evolved Results: *F*2

The next experiment involved minimising F_2. To give some idea of the quality and distribution of solutions, 1000 test runs were performed for each method. All methods allowed the GA to produce Pareto-optimal solutions every time, however, as fig. 3 shows, the distribution of these solutions on the Pareto front for this function are very different for each method. SWO and SWGR both produced solutions very close to or exactly the best compromise value of 1.0. SWR also favoured this value, but with a larger 'spread', with the numbers of solution produced falling almost logarithmically the further from the best compromise value they were. NDS showed a fairly even distribution throughout the Pareto-optimal range, and WMR favoured solutions at either function optima, with nothing in between. WAR gave the most unexpected and fascinating distribution, with solutions close to each optima and close to the best compromise value being favoured, all other Pareto-optimal values being less commonly produced, see fig. 3.

Additionally for F_2, the average solution of each method was calculated to give an indication of how balanced these distributions were. In other words, no matter what value(s) of Pareto-optimal solution were favoured, the mean value should be the centre value of 1.0 (for F_2). Table 2 (F_2 test 1) shows that all methods produced mean solutions close to 1.0.

	Best Compromise	SWO	NDS	WAR	SWR	SWGR	WMR
F_2 test 1	1.0	1.00922	0.93999	1.10226	1.21556	0.98763	0.97595
F_2 test 2	1.0	2.01459	0.85992	1.17007	1.22672	0.98825	0.99532
F_2 test 3	1.333	1.37837	N/A	2.01466	1.66141	1.310	1.45757

Table 2. Average solutions for each ranking method in F_2 tests 1-3.

Two further tests were performed using F_2. For the second test, f_{21} was temporarily changed to:

$$f_{21} = x^2 / 1000$$

to demonstrate the range-independence (or lack of it) for each method. As Table 2 (F_2 test 2) shows, after 1000 test runs for each method, SWO (method 1) clearly demonstrates its range-dependence by converging on average to the optimal of f_{22} instead of near to 1.0. All other methods show their range-independence by continuing to give mean solution values close to 1.0.

Finally, for the third test with F_2, the importance of f_{22} was doubled for every method capable of supporting importance (the two objectives being otherwise unchanged from the first test). By increasing the importance, the best compromise solution is changed from 1.0 to 1.333. Only three methods: SWO, SWR and SWGR, all successfully produced values close to this new desired value (see Table 2, F_2 test 3). NDS does not support importance, and WMR just doubled the frequency of optimal solutions to f_{22} (giving a deceptive mean solution), without actually producing any values between the two function optima. Finally, and somewhat unexpectedly, WAR simply converged every time to the optimal of f_{22}.

Upon investigation, it emerged that WAR does not permit the specification of gradual importance values. On the face of it, it would seem that increasing the weighting of the ranking value for more important objectives should introduce some level of additional importance for these objectives. Interestingly though, in practice it does not appear to be possible to gradually increase 'importance' values: either all objectives are treated equally, or the objective with the increased weight dominates all other objectives completely. Somewhat counter-intuitively, it seems that no matter how large or small an increase is made to a weight, it will make that objective dominate all others.

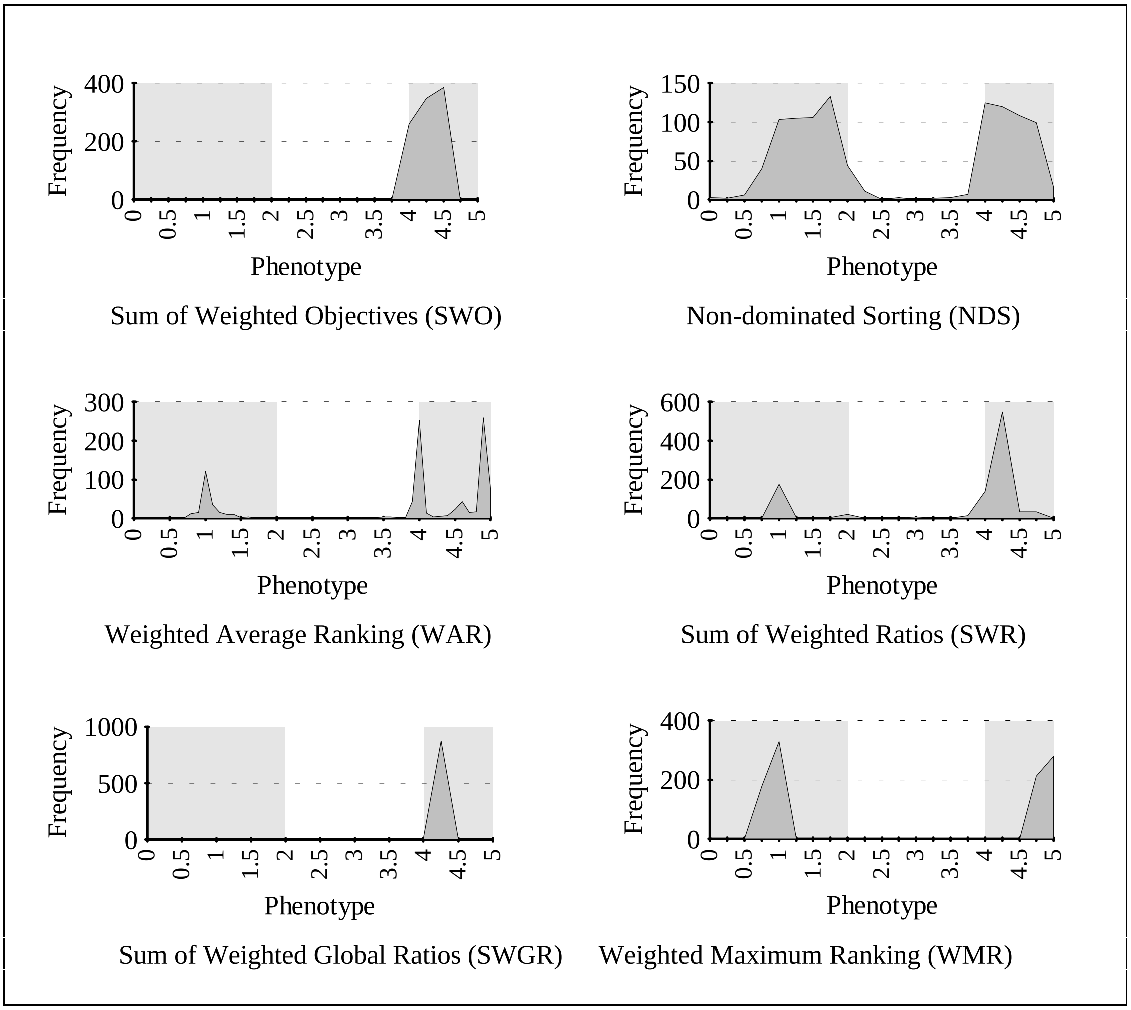

Figure 4. Distributions of solutions within the Pareto-optimal ranges (shown by grey shaded regions) for function F_3.

5.4 Evolved Results: *F*3

Experiments were then performed using F_3 with each method in turn. The function F_3 is significant since it has two disjoint Pareto-optimal ranges. Nevertheless, the distribution of solutions for this function was surprisingly consistent with those for F_2. As before, SWO and SWGR almost always converged to solutions near to the best compromise value of 4.5 (for F_3). Again, SWR favoured the best compromise solution with a slightly larger 'spread', but this time some solutions close to the optimal of f_{31} were also produced. NDS gave a fairly even distribution of solutions within the two Pareto-optimal ranges, and WMR again only generated solutions at the optima of the two objectives, with none in between. Finally, WAR showed its highly unusual distribution once more, by favouring solutions close to the optima of both objectives (including both minima of the multimodal objective f_{31}), and the best compromise solution to a lesser degree.

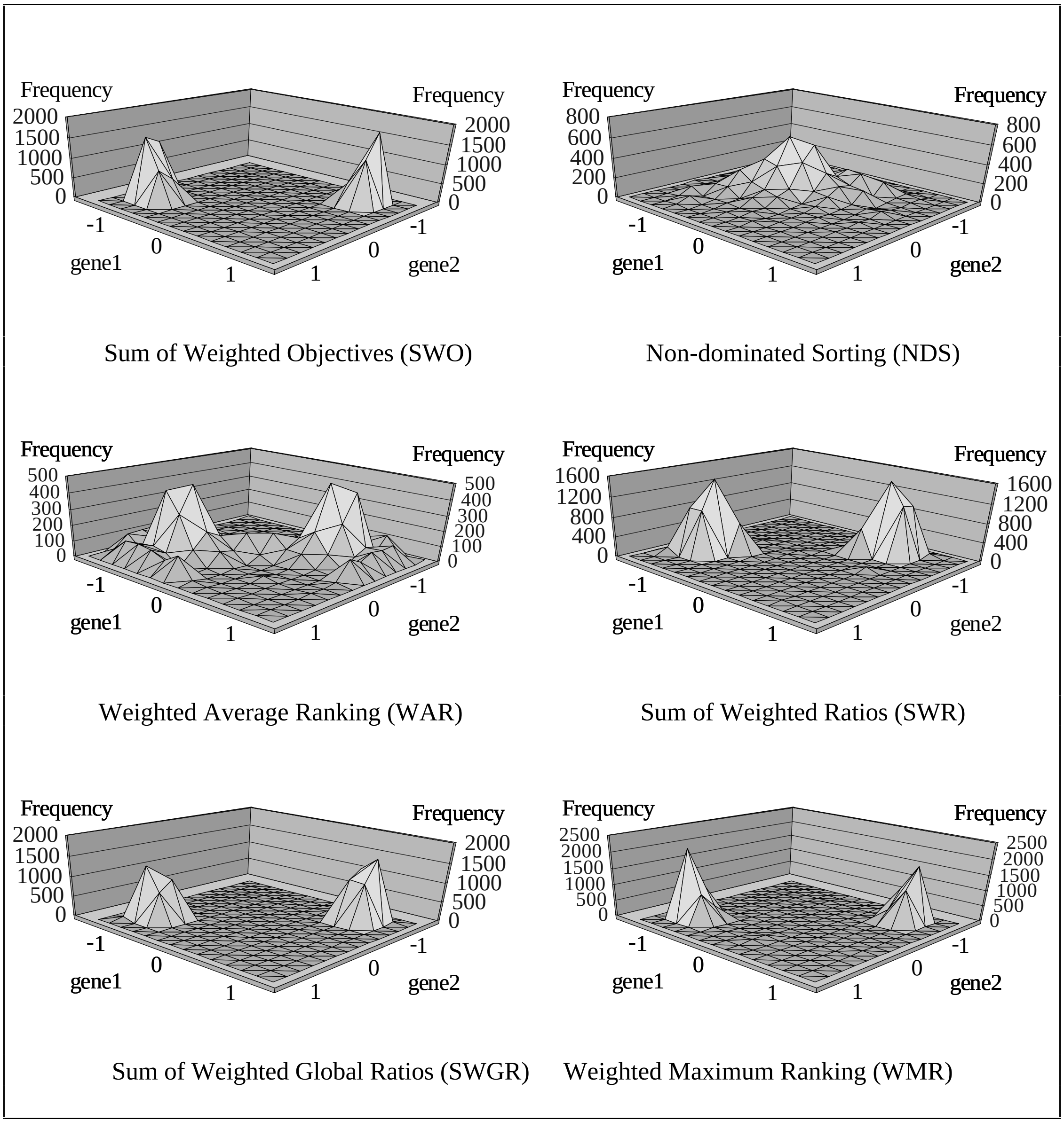

Figure 5. Distributions of solutions within the Pareto-optimal range for function F_4.

5.5 Evolved Results: *F*4

Finally, experiments were performed using F_4 with each method in turn. Again, consistent distributions of solutions were obtained. It should be noted that F_4 is a significant type of function because solutions between the optima of the two objectives are worse than at one optima or the other. This results in two equal best compromise solutions, one at each optima. Hence, although SWO and SWGR this time showed two peaks of distribution, these lie on the best compromise solutions, just as before. Once again, SWR favoured the best compromise solutions with a slightly larger 'spread'. As before, WMR favoured the two optima of the functions with nothing in between. NDS again produced a distribution of solutions covering the entire Pareto-optimal range, but for this function an unexpected and unwelcome bias towards the middle of the range was evident (where most solutions are very poor). Finally, WAR showed its typically unusual distribution, again favouring values close to the optima of the objectives (and the best compromise solutions, as they are the same for F_4), with other Pareto-optimal values being favoured less.

5.6 Explaining The Distributions

It should be stressed that all six of the ranking methods allow a GA to produce almost nothing but Pareto-optimal solutions. It is clear, however, that the distribution of these solutions within the Pareto-optimal range is a highly significant factor in determining whether an acceptable solution will be produced.

As shown above, each ranking method consistently seems to favour certain types of Pareto-optimal solution, based upon three factors: the Pareto range(s), the separate optimum or optima for each objective and the best compromise solution(s) of the function. These patterns of distributions remain consistent even with more unusual functions with multiple Pareto-ranges (F_3) and multiple best compromise solutions (F_4).

Upon consideration, these distributions are explicable. The three aggregation-based ranking methods: SWO, SWR and SWGR must inevitably favour the best compromise solution(s) to a problem, by definition. (The best compromise solution is the solution with *sum* of weighted objectives minimised, so any ranking method that sums objectives in any way, should allow convergence close to exactly the best compromise value.) NDS gives all non-dominated solutions equal rank, so a fairly even distribution throughout the Pareto-optimal range (range of non-dominated solutions) is to be expected. WMR bases the fitness of a solution on the maximum rank the solution has for any objective, so this predictably will result in the generation of solutions only at the optimal of one objective, with nothing in between (a high rank equates to a good value for that objective). Finally, even the unexpected distributions of WAR are explicable. WAR bases the fitness of a solution on the average rank for every objective. This means that a solution with a very high rank for one objective and a low rank for another will be judged equally fit compared to a solution with 'middling' ranks for two objectives. In other words, solutions close to optima of objective functions will be favoured, as will solutions close to the best compromise solution(s).

6. Application of the Ranking Methods: Solid Object Design

6.1 The Problem

To allow the investigation of the multiobjective techniques described previously on a larger and more difficult problem, they were all applied to a design creation problem consisting of 36 parameters and 10 constraints and objectives.

It can be clearly seen, by examining some of the wide range of literature describing the application of genetic algorithms to some of the numerous problems in design, that research in this area is flourishing, with some remarkable results (Adeli and Cheng, 1994; Holland, 1992). Applications range from optimization of existing designs (e.g. a high-bypass jet engine turbine as described by Holland, 1992) to the creation of 'artistic' pictures and shapes (Dawkins, 1986; Todd and Latham, 1992). However, as yet, very little research has been performed in the area of solid object design creation using GAs, i.e. the evolution of entirely new designs of solid objects from purely random beginnings (Bentley and Wakefield 1995a, 1995b, 1996a, 1996b).

To allow a genetic algorithm to evolve such designs in this way, a suitable representation of solids is required. This representation must be capable of describing the geometry of designs with the minimum of parameters, whilst allowing the freedom to manipulate the shape of a represented design in almost any way. To achieve this, a spatial-partitioning representation was created in previous work (Bentley and Wakefield 1995a). For the sake of simplicity, it can be compared to a number of 'building blocks', or primitives each with its own position, width, depth and height (the more advanced aspects of this representation are not being considered in this brief example). Giving a GA this amount of freedom when evolving the geometry of designs allows the generation of new *conceptual* designs (Bentley and Wakefield 1996a) - the most difficult part of the design process to automate.

The evolutionary design task was to evolve a set of portable, free-standing steps from a population initialised randomly. A good solution to this design task would be a reasonably light set of steps, capable of supporting the weight of a heavy person on each step in turn (suitable for use, for example, in a library). Three steps were desired. The number of primitives permitted for each potential design was limited to just six (giving 36 shape-definition parameters). This limitation immediately increases the difficulty of the problem, since inevitably, three primitives are required for the three steps, leaving only three remaining to support the steps sufficiently at the required heights. In total, some ten separate criteria are required to fully specify (and evaluate) the designs: maximum size, minimum size, desired mass, flat surfaces (three criteria), supportive surfaces (three criteria), and being unfragmented. The required design should have the fitnesses for all criteria minimised. More detailed descriptions of these criteria can be found in Bentley and Wakefield (1995b).

Using these ten criteria, a simple genetic algorithm was used to evolve a selection of designs. The GA used populations of 100, binary-coding, elitist selection, crossover and mutation as before and was allowed to run for 1000 generations. Where possible, test runs were performed with and without importance values set for the problem. Because of lengthy execution times, 15 test runs were performed for each algorithm. Figure 6 shows the number of almost fully minimised criteria (values from 0.0 to 10.0), low-valued criteria (10.0 to 50.0) and high-valued criteria (values of 50 upwards) for the best designs produced by each method. Distributions of solutions could not be calculated for this multidimensional problem; solutions are judged more on how acceptable they appear to be (i.e. how well-formed the designs are, as judged by a human).

6.2 Evolved Results

NDS and WMR both produced the most disappointing results. Since neither allow the guidance of the GA to any desirable subsets of Pareto-optimal solutions with importance, most designs were unacceptable. It was common to see designs evolved with fundamental and highly important aspects missing (such as correct size), since a solution is considered optimal by these ranking methods if just a single objective is minimised. Figure 6 shows clearly the small number of objectives that were fully minimised by these methods. Figure 7 (right) shows the best design evolved using NDS, the low quality clearly illustrates that this problem does require a ranking method capable of directing evolution to more acceptable non-dominated solutions.

The GA using WAR (method 3) produced some slightly disappointing results, since it too could not be directed during the search process by importance. However, since it does favour best compromise solutions, unlike WMR, results were slightly improved and more consistent in their quality compared to WMR and NDS. Figure 6 illustrates the increased number of minimised objectives.

SWR allowed the GA to generate better results, as can been seen from the increase in the number of minimised (or nearly so) criteria shown in fig. 6. As expected, results were not consistently good, because of the fairly 'spread-out' distribution generated by this method. However, SWO and SWGR both produced equally good results - better by far than the best generated using any of the other methods. Both consistently converged to acceptable solutions (as defined by relative importance values), see example design shown in fig. 7 (left). The number of minimised objectives was higher (see fig. 6). The only difference between the two was of course, that the range-dependent SWO required a substantial amount of time spent fine-tuning weights, whilst the range-independent SWGR only required the simple setting of importance values (a considerably easier task).

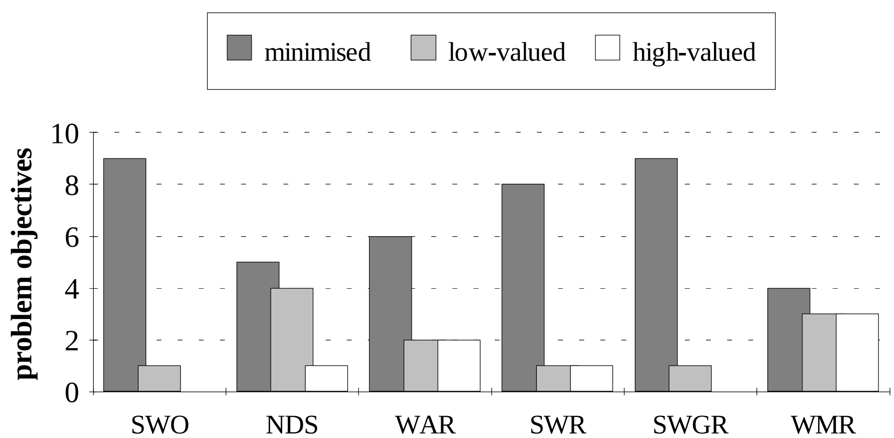

Figure 6. The number of minimised, low and high valued objectives for the best designs produced with each method.

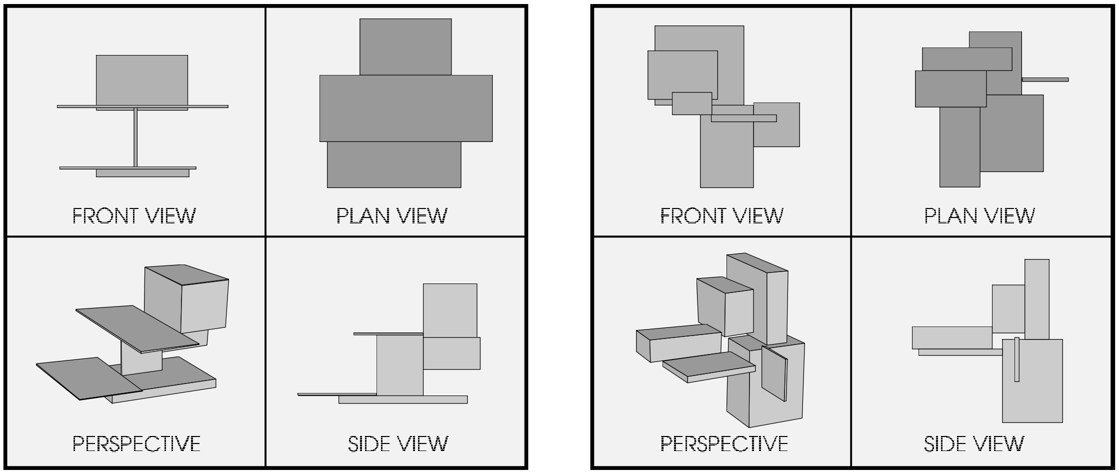

Figure 7. Well-formed evolved design of steps using SWGR (left).
Mal-formed evolved design of steps using NDS (right).

7. Performance Of The Ranking Methods

For all tests, the five ranking methods all guided the genetic algorithm to produce non-inferior (non-dominated) solutions. Indeed, after tens of thousands of test runs, only a tiny handful of solutions were not Pareto-optimal. What varied for each method was the distribution and bias of the results within the Pareto-optimal front, and the ability of 'importance' to guide this bias to allow a GA to converge to a smaller subset of Pareto-optimal solutions.

SWO (sum of weighted objectives), the first and simplest of all methods examined, gave solutions all close to or exactly the best compromise solution(s). This method does not produce a range of Pareto-optimal solutions - it will always force the GA to converge as quickly as possible to a (Pareto-optimal) compromise solution. Although it is possible to make this best compromise solution the same as the desired solution by setting importance weightings, in practice determining the exact values of the weights is often very difficult. As the results show, with correct weights, the results can be very good, but this range-dependent method, despite being the simplest and quickest, is not practical for anything other than simple problems.

NDS (non-dominated sorting) usually produced solutions with no significant bias; if a solution is Pareto-optimal, it is considered acceptable by this method. Although it is a range-independent method with no parameters to fine-tune, it is slow, difficult to implement, and has no obvious way to make use of importance values. Results for this method were surprisingly poor, perhaps because of the lack of niching, but more likely because this method lacks the ability to guide the GA to converge to acceptable Pareto-optimal solutions.

WAR (weighted average ranking) again produced a wide range of non-dominated solutions, but this time with a highly distinct bias towards solutions close to the minimum of each objective function. It also showed a bias towards the best compromise solutions. Compared to the other methods, results using WAR were average for all problems. Again, this method is a range-independent method with no parameters to fine-tune, but it is perhaps simpler to implement and is faster in execution than the second method. Unfortunately, weighting for importance has no gradual effect (either all objectives are treated equally, or one dominates completely).

SWR (sum of weighted ratios) also produced a range of non-dominated solutions, this time with a strong bias towards the best compromise solution, the frequency of solutions falling almost logarithmically the further they were from this solution. Because this method supports importance, this bias can be moved to allow the GA to favour any Pareto-optimal solution. In this way, the problem-specific knowledge encompassed in the importance values can be used to guide the GA to produce higher numbers of acceptable non-dominated solutions. Being range-independent, having the ability to add importance to objectives, and producing a range of solutions makes this perhaps the most versatile of the six. However, despite also being fast and easy to implement, the results were not exceptional for the solid-object design application.

SWGR (sum of weighted global ratios) like the first method, produced just a single non-dominated solution as close to a best compromise solution as possible. Being range-independent, it will normally treat all objectives equally without the need for any fine-tuning of weights. Since it does support importance, it can be made to guide the GA to converge to a single, acceptable solution on the Pareto-optimal front. The method is fast and simple to implement, and as the test results show, the quality of solutions is as good as (and sometimes better than) those produced by the first method - without the lengthy time spent on fine-tuning parameters.

Finally, WMR (weighted maximum ranking), only produced solutions with one objective fully satisfied (i.e. similar numbers of solutions at each objective minimum to the problem). Unlike WAR, there were no other solutions produced 'between' the separate function optima. This behaviour was noted in VEGA by Schaffer (1984), who introduced an additional heuristic in an attempt to increase the numbers of 'middling' values. This range-independent method requires the same implementation and execution time as WAR. Importance changes the relative numbers of solutions at each objective minimum. Results with this method were poor.

8. Conclusions

This paper has made three significant advances in the area of multiobjective function optimisation with genetic algorithms:

1. The problem of ranking solutions to multicriteria problems has been investigated, with the clarification of some commonly used terminology, and the identification of two key factors: range-independence and importance.

   If the ranking method is not *range-independent*, then one or more objectives in the problem can dominate the others, resulting in evolution to poor solutions. Thus, a ranking method should not just be independent of individual applications (i.e. problem independent), as stated by Srivinas and Deb (1995), it should be independent of the *effective ranges* of the objectives in individual applications (i.e. range-independent). When the ranking method is range-independent, it requires no problem-specific knowledge to set and fine-tune parameters before it will work. Thus, a range-independent method will also be independent of the applications themselves.

   Giving certain objectives in a problem greater *importance* allows the GA to produce not just non-dominated solutions, but a smaller subset of *acceptable* non-dominated solutions. As was seen in section 6, ranking methods which support importance allow the consistent evolution of considerably improved solutions compared to those that do not support importance.

2. Using this new understanding of the problem, three novel ranking methods were created, in an attempt to embrace both factors of the problem.

   All three of these new methods: 'weighted average ranking' (WAR), 'sum of weighted ratios' (SWR), and 'sum of weighted global ratios' (SWGR) are range-independent. SWR and SWGR both support importance fully. This compares favourably with most existing ranking methods, such as 'sum of weighted objectives' (SWO) which is range-dependent, and 'non-dominated sorting' (NDS) and VEGA's 'weighted maximum ranking' (WMR) which do not support importance.

3. Three standard ranking methods were compared with the three novel methods, using four well-known test functions and one more complex solid object design task. Significantly, the quality

and *distribution* of results was examined. Until now, the distribution of most types of ranking methods was unknown.

All methods produced Pareto-optimal solutions, but the consistent distributions of these solutions was revealing. For applications where multiple solutions are required, it is clear that NDS, WMR and the new WAR method all give potentially useful distributions of solutions. Should some user-alterable bias in the distribution of solutions be required, SWR would be an appropriate choice. However, for many applications, these methods would still produce hundreds of alternative 'optimal' solutions for a human to laboriously (and unnecessarily) search through, with no guarantee that any would be acceptable to him/her. For this type of application, SWGR should be used. This new range-independent method performs as well as or better than all other methods, allowing a GA to consistently evolve a single *acceptable* optimal solution regardless of the multiobjective problem. Indeed, this multiobjective ranking method has now been used to tackle a wide range of different solid object design problems with great success (Bentley and Wakefield, 1996b).

10. References

Adeli, H. & Cheng, N. (1994). Concurrent Genetic Algorithms for Optimization of Large Structures. *ASCE Journal of Aerospace Engineering*, 7(3), 276-296.

Bentley, P. J. & Wakefield, J. P. (1995a). Generic Representation of Solid Geometry for Genetic Search. *Microcomputers in Civil Engineering* 11(3) (to appear).

Bentley, P. J. & Wakefield, J. P. (1995b). The Evolution of Solid Object Designs using Genetic Algorithms. *Applied Decision Technologies* (ADT '95), London (pp. 391-400).

Bentley, P. J. & Wakefield, J. P. (1996a). Conceptual Evolutionary Design by Genetic Algorithms. Submitted to *Engineering Design and Automation Journal*.

Bentley, P. J. & Wakefield, J. P. (1996b). Overview of a Generic Evolutionary Design System. In *Proceedings of the 2nd On-line Workshop on Evolutionary Computation*, Nagoya University, Japan (pp. 53-56).

Dawkins, R. (1986). *The Blind Watchmaker*. Longman Scientific & Technical Pub.

Dowsland, K. A. (1995). Simulated Annealing Solutions for Multi-Objective Scheduling and Timetabling. *Applied Decision Technologies* (ADT '95), London (pp. 205-219).

Fonseca, C. M, & Fleming, P. J. (1995a). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1), 1-16.

Fonseca, C. M, & Fleming, P. J. (1995b). Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restriction. *Genetic Algorithms in Engineering Systems: Innovations and Applications* (GALESIA 95), Sheffield (pp. 45-52).

Linkens, D. A. & Nyongesa, H. O. (1993). A Distributed Genetic Algorithm for Multivariable Fuzzy Control. *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, Digest No. 199/130, (pp. 9/1 - 9/3).

Marett, R. & Wright, M. (1995). The Value of Distorting Subcosts When Using Neighbourhood Search Techniques for Multi-objective Combinatorial Problems. *Applied Decision Technologies*, London (pp.189-202).

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison-Wesley.

Holland, J. H. (1992). Genetic Algorithms. *Scientific American*, 66-72.

Horn, J. & Nafpliotis, N. (1993). Multiobjective Optimisation Using the Niched Pareto Genetic Algorithm. *Illinois Genetic Algorithms Laboratory (IlliGAL)*, report no. 93005.

Ryan, C. (1994). Pygmies and Civil Servants. Advances in Genetic Programming. In Kinnear, K. (ed), *Advances in Genetic Programming*, MIT Press.

Schaffer, J. D. (1984). *Some experiments in machine learning using vector evaluated genetic algorithms*. PhD dissertation, Vanderbilt University, Nashville, USA.

Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. *Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms*, (pp. 93-100).

Srinivas, N. & Deb, K. (1995). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3), 221-248.

Sun, Y. & Wang, Z. (1992). Interactive Algorithm of Large Scale Multiobjective 0-1 Linear Programming. *Sixth IFAC/IFORS/IMACS Symposium on Large Scale Systems, Theory and Applications*, (pp. 83-86).

Syswerda, G. & Palmucci, J. (1991). The Application of Genetic Algorithms to Resource Scheduling. *Genetic Algorithms: Proceedings of the Fourth International Conference*, Morgan Kaufmann (pp. 502-508).

Todd, S. & Latham, W. (1992). *Evolutionary Art and Computers*. Academic Press.