University of Exeter
Department of Computer Science

Ordering and Visualisation of Many-objective Populations

David John Walker

September 2012

Supervised by Dr Jonathan Fieldsend and Prof Richard Everson

Submitted by David John Walker, to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science, September 2012.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

.....

for my family

Abstract

In many everyday tasks it is necessary to compare the performance of the individuals in a population described by two or more criteria, for example comparing products in order to decide which is the best to purchase in terms of price and quality. Other examples are the comparison of universities, countries, the infrastructure in a telecommunications network, and the candidate solutions to a multi- or many-objective problem. In all of these cases, visualising the individuals better allows a decision maker to interpret their relative performance. This thesis explores methods for understanding and visualising multi- and many-criterion populations.

Since people cannot generally comprehend more than three spatial dimensions the visualisation of many-criterion populations is a non-trivial task. We address this by generating visualisations based on the dominance relation which defines a structure in the population and we introduce two novel visualisation methods. The first method explicitly illustrates the dominance relationships between individuals as a graph in which individuals are sorted into Pareto shells, and is enhanced using many-criterion ranking methods to produce a finer ordering of individuals. We extend the power index, a method for ranking according to a single criterion, into the many-criterion domain by defining individual quality in terms of tournaments. The second visualisation method uses a new dominance-based distance in conjunction with multi-dimensional scaling, and we show that dominance can be used to identify an intuitive low-dimensional mapping of individuals, placing similar individuals close together. We demonstrate that this method can visualise a population comprising a large number of criteria.

Heatmaps are another common method for presenting high-dimensional data, however they suffer from a drawback of being difficult to interpret if dissimilar individuals are placed close to each other. We apply spectral seriation to produce an ordering of individuals and criteria by which the heatmap is arranged, placing similar individuals and criteria close together. A basic version, computing similarity with the Euclidean distance, is demonstrated, before rank-based alternatives are investigated. The procedure is extended to seriate both the parameter and objective spaces of a multi-objective population in two stages. Since this process describes a trade-off, favouring the ordering of individuals in one space or the other, we demonstrate methods that enhance the visualisation by using an evolutionary optimiser to tune the orderings.

One way of revealing the structure of a population is by highlighting which individuals are extreme. To this end, we provide three definitions of the "edge" of a multi-criterion mutually non-dominating population. All three of the definitions are in terms of dominance, and we show that one of them can be extended to cope with many-criterion populations.

Because they can be difficult to visualise, it is often difficult for a decision maker to comprehend a population consisting of a large number of criteria. We therefore consider criterion selection methods to reduce the dimensionality with a view to preserving the structure of the population as quantified by its rank order. We investigate the efficacy of greedy, hill-climber and evolutionary algorithms and cast the dimension reduction as a multi-objective problem.

I would like to thank several people for their help and support during the course of my PhD. My parents have supported me in many ways over the last four years, and it is very much appreciated. I would also like to thank my many colleagues, particularly: Andrew Clark, Jacqueline Christmas, Max Dupenois, Kent McClymont and Zena Wood. I am also grateful to Antony Galton and Joshua Knowles for agreeing to examine this thesis.

Special thanks are due to my supervisors, Jonathan Fieldsend and Richard Everson, who have been a constant source of sound advice and encouragement.

Contents

1. Introduction

Situations in which the relative performance or quality of individuals in a population must be compared are commonplace, and the performance of such individuals are often described by a set of competing criteria. A well known example is the comparison of universities; an individual university is often described by a set of criteria known as *key performance indicators* (KPIs), each of which measures a particular aspect of the university, such as the proportion of students graduating with good honours degrees or research output. In order for a student to identify the university at which they wish to study, they may wish to compare the different universities based on the data provided by the KPIs to determine which is best. League tables that rank universities according to some function of their criteria are commonplace. Another example in which such quality measures are used to compare individuals is in the construction of a maintenance schedule for the components of a technical infrastructure, such as the wireless access points in a mobile telephone network. Here it is important to identify poorly performing components so that they can be given priority in the schedule. The purpose of this thesis is to provide methods that facilitate the visualisation and understanding of a population of such individuals in cases where there is *conflict* between criteria, such that good performance on one criterion does not guarantee good performance on another.

A common way in which university performance has been evaluated is to construct a *league table*, a ranking of individuals from which their relative quality can be easily observed. League tables are frequently used to rank sports teams based on the results of tournaments. In that case, the league table is based on a single criterion, often a maximisation of points accrued in a tournament with other teams. In the case of a population described by a more than one criterion the league table must be constructed by drawing together information from all of the criteria. This can be, and often is, done by aggregating the criterion values into a single value. However, criteria are often measured on different scales, such as a proportions representing graduation rates and monetary values representing the amount spent for every student, and directly aggregating these values does not produce a meaningful overall score for the individual. The criteria can be normalised in some way; however, aggregation functions usually require *a priori* weights to be chosen for the criteria so that an indication of the importance of each criterion is incorporated into the aggregation. Solutions to *multi-objective* optimisation problems, which are analogous to the individuals we seek to compare, are often compared using the Pareto dominance relation [e.g., Deb, 2001], which is defined later. This allows for the comparison of solutions

on their original, unnormalised and unweighted, objective values, and facilitates the partial ranking of solutions. In the *many*-objective case, where a problem comprises four or more objectives, dominance is less discriminative and often cannot differentiate between solution quality. Alternative ranking methods have been developed by the optimisation community for such populations. In Chapter 3 we show how to rank many-criterion populations with some of these methods, demonstrating multi- and many-objective ranking methods in lieu of building a league table by normalising, weighting and aggregating criterion values in order to compare individuals in a principled way.

A complementary method to comparing individuals by ranking a population is to produce a visualisation of the individuals from which their relative performance can be inferred. Multi-criterion populations are often visualised by plotting individuals on two or three dimensional axes, known as a scatter plot [Deb, 2001]. Clearly, this approach will not work for populations described by four or more criteria. For this reason, we adopt the nomenclature of optimisation problems and refer to populations comprising two or three criteria as *multi*-criterion populations; *many*-criterion populations comprise four or more criteria. To visualise a many-criterion population one of two alternatives must be considered: either a method that is capable of visualising the entire population, such that all of the original data can be recovered from the visualisation, is required, or some of the criteria must be discarded so that a conventional tool, such as a scatter plot, can be used. Neither of these two approaches is perfect; methods that can visualise high-dimensional data are often cluttered because they attempt to represent high-dimensional data in two or three dimensions. On the other hand, reducing the dimensionality of the data leads to an inevitable, undesirable, loss of information. An important focus of this thesis is to find methods which can visualise a many-criterion population in a clear and intuitive way that is useful to a decision maker, without having to discard criteria, and we present two approaches. One is a novel visualisation which constructs a graph of a population using dominance and incorporates the rank of individuals visually with colour. The other, seriation, in which an ordering is obtained that places similar items close together, enhances a very well known visualisation method, the *heatmap*. We also show how seriation can be a useful tool in the visualisation of solutions to many-objective optimisation problems by incorporating solution parameters and objective values into the visualisation together. Chapter 3 presents the work on visualising populations as graphs, while Chapter 5 presents the work on seriation.

Whilst we aim to visualise the original population with respect to the full set of criteria, reducing the dimensionality of the population allows for easier visualisation and can also expose informative structural information about the individuals and their relationships with each other. To this end we investigate dimension reduction techniques that are suited to this particular type of data. Dimension reduction methods generally require a method for quantifying the loss of information that their application will incur. Methods in which criteria are discarded have been proposed that, for example, consider correlation between

criteria [Deb and Saxena, 2005] and the changes in pairwise dominance relationships between individuals [Brockhoff and Zitzler, 2007a]. In Chapter 6 we propose a method that calculates an individual's rank based on the original set of criteria and seeks a criterion subset that preserves this rank as closely as possible. This method is incorporated into a bi-objective evolutionary algorithm which explores the trade-off between the number of criteria used to describe a population and the amount of information loss incurred. An alternative to discarding redundant criteria is to project the population onto a new set of coordinate axes. Methods that do this often attempt to preserve metric information; Euclidean distance is a common choice for this, however this does not capture the dominance relations between the individuals in the population and may not be the appropriate metric for cases in which the criteria are on different scales. In this thesis we introduce a new metric based on dominance which captures the relations within the population. This is used in Chapter 3 in conjunction with a well known dimension reduction technique, multi-dimensional scaling (MDS) [Kruskal, 1964; Sammon, 1969], to produce an embedding in which individuals with similar dominance relationships are placed close together.

Understanding the geometry of a population of individuals can provide useful information to a decision maker, allowing them, for example, to determine which individuals are extreme. A recent example of such an idea from the evolutionary optimisation community is the identification of the corners of a non-dominated population [Singh et al., 2011], and we extend this idea to identify the set of individuals that lie on the *edge* of a mutually non-dominating population so that they can be highlighted in a subsequent visualisation. Three methods for doing this are presented in Chapter 4 and we observe that whilst this is a relatively straightforward task for a multi-criterion population, the extension into many-criterion populations brings with it additional complications.

1.1. Thesis Structure

This thesis is organised as follows:

Chapter 2: Background

This work draws on evolutionary methods for solving multi- and many-objective optimisation problems, so the background chapter begins by introducing them. Since the work is closely related to population ranking, we discuss methods for comparing permutations. Finally, methods for visualising high-dimensional datasets are discussed, with a particular focus on their application to many-criterion populations.

Chapter 3: Many-criterion League Tables

The focus of this chapter is the ranking and visualisation of many-criterion populations. Existing methods from multi- and many-objective evolutionary algorithms are reviewed. We show how Pareto sorting and a ranking method such as *average*

rank or the *power index* can be combined to produce and visualise a league table from a many-criterion population and provide demonstrations on several example populations. A new measure of the stability of an individual's rank is presented and the chapter concludes by introducing a similarity measure based on dominance which we use as the basis for visualising populations with MDS.

Chapter 4: Finding the Edge of a Mutually Non-dominating Population

In an effort to enhance the understanding of a many-criterion population we have investigated methods for identifying which individuals lie on the edge of a mutually non-dominating population. Such individuals represent extreme values on specific criteria, and combinations of criteria, and as such may be of interest to a decision maker. We discuss how the "edges" of a set of mutually non-dominating individuals may be defined and present straightforward algorithms for finding the edge of a mutually non-dominating multi-criterion population.

Chapter 5: Seriation of Heatmaps

Heatmaps are an established method for visualising a high-dimensional dataset, however they can often appear cluttered and unintelligible. Seriation has been used to ameliorate this by placing similar individuals and/or criteria close together, and we introduce seriation methods that are designed to take advantage of the structure of many-criterion populations. Seriating heatmaps of such populations results in a visualisation that is significantly clearer and easier to understand. We also extend the work of Pryke et al. [2006] by seriating heatmaps of parameter space and objective space for the visualisation of solutions to many-objective problems simultaneously; we optimise the permutation of solutions so that they are ordered to produce a clear heatmap in both spaces.

Chapter 6: Rank-based Dimension Reduction

While previous chapters have demonstrated methods for visualising many-criterion populations, they are often better visualised and more easily understood if redundant criteria are removed. Using methods developed in earlier chapters that can rank a population based on a set of criteria, we introduce methods for reducing the dimensionality of a population whilst preserving the original rank of individuals as closely as possible. We present evolutionary methods for *criterion selection*.

Chapter 7: Conclusion

We conclude the thesis by drawing together elements from the preceding chapters and discuss potential avenues of future investigation.

1.2. Thesis Contributions

The main novel contributions made by this thesis are the following:

- A method based on Pareto sorting for visualising many-criterion populations (Chapter 3).
- New measures of the robustness of an individual's position in a partial ordering determined with Pareto sorting (Chapter 3).
- A method for describing the outcome of a tournament played between many-criterion individuals which is used to define a graph of the population and extend a ranking method, the power index, to many-criterion use (Chapter 3).
- A method for describing distances between many-criterion individuals using dominance which can be used to produce low-dimensional embeddings with MDS (Chapter 3).
- Three methods for identifying individuals which lie on the edge of non-dominated populations (Chapter 4).
- Seriation methods for enhancing heatmaps of many-criterion populations and optimising a visualisation which uses heatmaps to illustrate both the decision and objective space components of a multi-objective solution set (Chapter 5).
- Methods for reducing the number of criteria such that the overall rank of individuals is preserved (Chapter 6).

1.3. Publications

Some elements of this work have been published or submitted for publication in journals and conference proceedings:

David J. Walker, Richard M. Everson, and Jonathan E. Fieldsend. Visualisation and ordering of many-objective populations. In *IEEE Congress on Evolutionary Computation (CEC 2010)*, pages 3664–3671, CCIB, Barcelona, Spain, July 2010a.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Ordering multi-objective populations with the power index. In *2010 Postgraduate Conference for Computing: Applications and Theory (PCCAT 2010)*, pages 12–18, Exeter, Devon, UK, June 2010b. University

of Exeter.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Rank-based dimension reduction for many-criteria populations. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 107–108, New York, NY, USA, 2011. ACM.

K. McClymont, D. Walker, E. Keedwell, R. Everson, J. Fieldsend, D. Savic, and M. Randall-Smith. Novel methods for ranking district metered areas for water distribution network maintenance scheduling. In *CCWI 2011*, Exeter, 2011.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualising many-objective populations. In *Proceedings of the 3rd Workshop on Visualisation in Genetic and Evolutionary Computation (VizGEC 2012) at GECCO 2012*, Philadelphia, 2012.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualising Mutually Non-dominating Solution Sets in Many-objective Optimisation. *IEEE Transactions on Evolutionary Computation*, (in press).

R. M. Everson, D. J. Walker, and J. E. Fieldsend. Edges of Mutually Non-dominating Sets In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, GECCO '13, (in press), New York, NY, USA, 2013. ACM.

1.4. Summary

In this chapter we have briefly introduced the notion of many-criterion populations and alluded to their prevalence. Having highlighted the difficulties associated with visualising and understanding such populations, we now introduce some relevant background material before presenting methods aimed at enhancing their interpretability.

2. Background

2.1. Introduction

In this chapter we introduce the underlying concepts upon which the work presented in the remainder of the thesis is based. The central theme of the thesis is the understanding and visualisation of many-criterion populations using approaches from evolutionary multi-objective optimisation, so we begin this chapter with an introduction to techniques commonly used in solving single-, multi- and many-objective problems with evolutionary algorithms. An important part of this discussion relates to decision making and population ranking, as does a significant proportion of the rest of the thesis, so we then introduce basic techniques for ranking alternatives in multi-criteria decision making and comparing permutations. We conclude the chapter by describing visualisation methods that are commonly used for presenting high-dimensional data. We illustrate the techniques on example sets of solutions to multi- and many-objective problems.

2.2. Optimising Multiple Objectives

Many industrial and scientific problems require the identification of an optimal set of *parameters*. A parameter is an adjustable variable that defines an aspect of a *solution*; an example problem is the design of a wireless telecommunications network, in which a potential network configuration is a solution and the bearing of a single wireless access point antenna within the network is a parameter that can be adjusted [e.g., Smith et al., 2008]. Such *optimisation problems* consist of a criterion, often called an *objective*, which evaluates the suitability of a candidate solution for solving a problem and is to be either maximised or minimised. When a problem is defined in terms of just one objective it is known as a *single-objective* or *uni-objective* optimisation problem. In the case of the mobile telephone network design problem, a possible objective would be to maximise the coverage provided by the network to ensure the best possible service for mobile telephone users.

A putative solution to an optimisation problem is commonly represented as a P-dimensional vector $\mathbf{x} = (x_1, \ldots, x_P)$. Throughout this thesis we denote by x_p the value of the pth pa-

rameter and by x_{ip} the pth parameter of the ith solution in a collection of N solutions $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$. Generally, an optimisation problem is either *continuous* or *discrete*. In a continuous problem, a parameter can take *any* value within a specified range. A discrete optimisation problem involves the selection of the best combination of a finite set of parameters. An important difference between the two is that while it is possible to place a natural ordering over a set of real valued parameters in a continuous problem, there is no such ordering over the parameters in a discrete problem. The reason for optimising a continuous problem is clear, as an infinite number of parametrisations are possible. Optimising a discrete problem is challenging because of the number of possible combinations of parameters. There are often far too many combinations to enumerate and evaluate all of them. Whatever the problem type, the solution $\mathbf{x}$ is mapped to objective space with an *objective function* f such that:

$$y = f(\mathbf{x}) \tag{2.1}$$

where y is to be maximised or minimised. A problem such as this is known as a *single-objective* problem because the quality of a solution is determined in terms of a single function that returns a scalar value. In practice, however, most problems do not just consist of a single objective, but rather a set of M objectives between which a trade-off exists. In this case, the single-objective value for the ith solution, y_i, becomes a M-dimensional vector $\mathbf{y}_i$:

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i) \tag{2.2}$$

$$= (f_1(\mathbf{x}_i), \ldots, f_M(\mathbf{x}_i)) \tag{2.3}$$

in which y_{im} is the value of the ith solution on the mth objective such that $y_{im} = f_m(\mathbf{x}_i)$. The collection of solutions $\mathbf{X}$ has a corresponding collection of *objective vectors*, $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$. In such a *multi-objective* case, where a problem comprises two or three objectives, the objectives can be maximised, minimised, or a combination of the two. To extend the illustrative problem into the multi-objective domain, the problem is reformulated to consist of two objectives. The original objective, to maximise network coverage, is retained and a second one is added to minimise the number of wireless access points needed. Installing a large number of access points is likely to result in good network coverage, however the cost of deploying the network will be prohibitively expensive. A *trade-off* therefore exists between the two objectives; Figure 2.1 demonstrates this idea, presenting candidate solutions to a multi-objective problem comprising two objectives (a *bi-objective* problem), one of which is to be minimised while the other is maximised. Clearly, the red solutions are inferior to the blue solutions; each red solution is worse on both objectives than at least one of the blue solutions. For example, the solution $\mathbf{y}_2$ is inferior to $\mathbf{y}_1$, and $\mathbf{y}_4$ is inferior to $\mathbf{y}_3$. It is also clear that while the extreme solutions in the blue solution set, those at the extremes of the collection, optimise one objective well, they do so at the expense of their performance on the other objective. The absence of a distinct "knee" in the solution set, a point at which an improvement in either objective would drastically reduce the quality of a solution on the other objective, means that it is not clear which

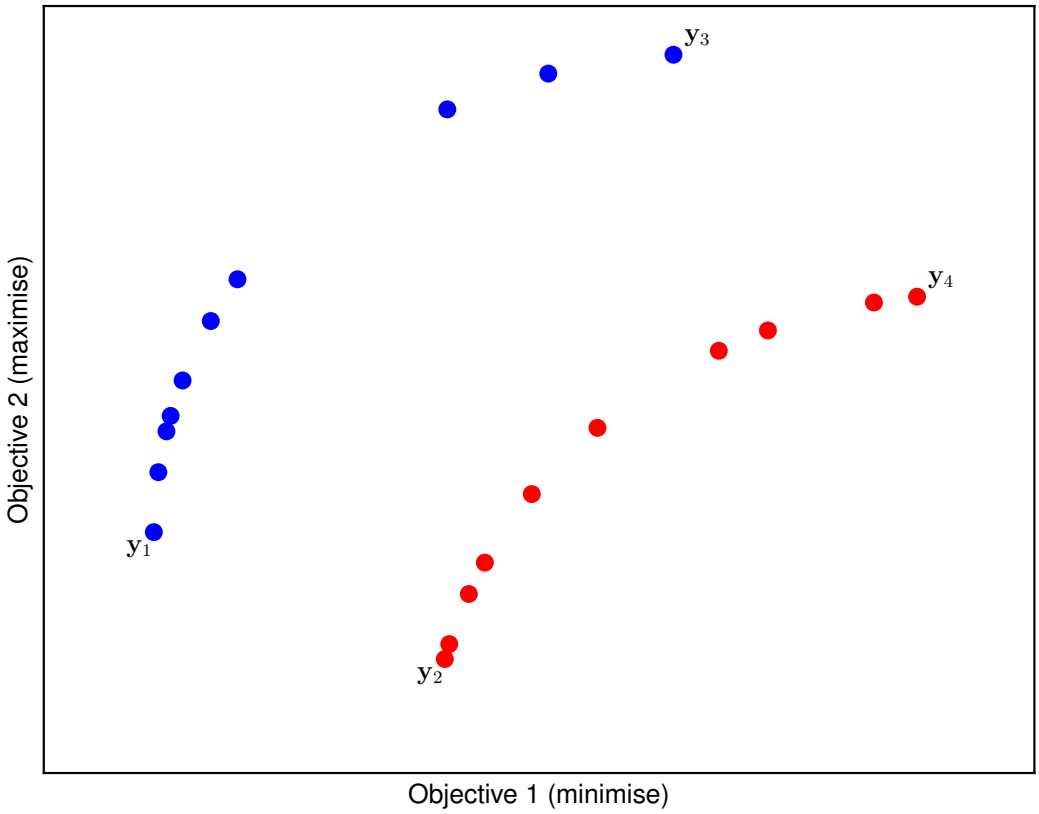


Figure 2.1.: An example of the trade-off between two objectives in a minimisation problem. Each of the red solutions is inferior to a least one of the blue solutions, for example $\mathbf{y}_2$ is inferior to $\mathbf{y}_1$ and $\mathbf{y}_4$ is inferior to $\mathbf{y}_3$. It is not, however, clear which of the blue solutions provides the best operating point for a decision maker to choose as improving the objective value for any solution decreases its performance on the other objective.

solution a decision maker should prefer, assuming that they have no *a priori* preference of one objective over the other. This highlights the most important distinction between a single- and multi-objective problem, that whereas a single-objective problem is solved by a single optimal solution, in general a multi-objective problem is optimally solved by a set of solutions which trade off the problem objectives. Without loss of generality, a multi-objective problem is defined in terms of a minimisation as follows:

$$\text{minimise } y_m = f_m(\mathbf{x}) \qquad \text{for all } m = 1, \ldots, M. \tag{2.4}$$

The set of solutions that solve a multi-objective problem lie on the *Pareto front*. The Pareto front is defined in terms of the Pareto *dominance* relation. Dominance is a binary relation under which an objective vector $\mathbf{y}_i$ *dominates* $\mathbf{y}_j$, denoted $\mathbf{y}_i \prec \mathbf{y}_j$, if it is at least as good as $\mathbf{y}_j$ on all objectives and better on at least one. More formally, assuming that all objectives are to be minimised without loss of generality:

$$\mathbf{y}_i \prec \mathbf{y}_j \iff \forall m(y_{im} \leq y_{jm}) \wedge \exists m(y_{im} < y_{jm}). \tag{2.5}$$

This is known as *strong dominance*. Similar to strong dominance is *weak dominance*:

$$\mathbf{y}_i \preceq \mathbf{y}_j \iff \forall m(y_{im} \leq y_{jm}). \tag{2.6}$$

In this thesis, we use the term dominance to refer to strong dominance. An objective vector that is dominated by no other is said to be *non-dominated* and if neither of two objective vectors dominates the other then they are *mutually non-dominating*. In Figure 2.1, the blue solutions are mutually non-dominating with respect to each other, and are non-dominated. Each of the red solutions is dominated by at least one of the blue solutions. The goal of optimising a multi-objective problem is to find *feasible* solutions whose objective vectors lie on the true *Pareto front*, which optimise the trade-off between problem objectives. A feasible solution is one that does not violate any problem *constraints*, and we denote by $\mathcal{X}$ the feasible decision space that maps to the feasible objective space $\mathcal{Y}$. Objective vectors on the true Pareto front are mutually non-dominating with respect to each other, and non-dominated with respect to any other feasible solutions:

$$\mathcal{F} = \{\mathbf{y} \in \mathcal{Y} \mid \neg\exists \mathbf{y}' \ (\mathbf{y}' \in \mathcal{Y} \wedge \mathbf{y}' \prec \mathbf{y})\} \tag{2.7}$$

The solutions with objective vectors $\mathbf{y} \in \mathcal{F}$ are called *Pareto optimal*. The pre-image of the Pareto front is called the *Pareto set* $\mathcal{S}$:

$$\mathcal{S} = \{\mathbf{x} \in \mathcal{X} \mid \neg\exists \mathbf{x}' \ (\mathbf{x}' \in \mathcal{X} \ \wedge \ \mathbf{f}(\mathbf{x}') \prec \mathbf{f}(\mathbf{x})\}. \tag{2.8}$$

It is important to note that some of the examples in this thesis are collections of individuals, *populations* $\mathbf{Y}$, rather than collections of objective vectors corresponding to many-objective solutions. In these cases we rewrite the definition of Pareto optimality given in Equation 2.7 in terms of the population $\mathbf{Y}$ rather than the feasible objective space $\mathcal{Y}$:

$$\mathcal{F} = \{\mathbf{y}_i \mid \neg\exists \mathbf{y}_j \ (\mathbf{y}_j \in \mathbf{Y} \wedge \mathbf{y}_j \prec \mathbf{y}_i)\}. \tag{2.9}$$

In addition to objectives, a problem can also have *constraints*. These are conditions which must be met in order for a solution to be considered a candidate for solving a problem. A likely example from the example telephone network problem is that an access point must be on dry land, and there must be no pre-existing structures in that location. Otherwise, obviously, the access point cannot be built there. Constraints can be represented as bounds on parameters, and, as described earlier, the *feasible region* for a problem, within which allowable solutions exist, is denoted $\mathcal{X}$ for parameter space. The objective space image of this feasible space is denoted $\mathcal{Y}$, such that $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ for a feasible solution.

Optimisation problems can be *linear* or *nonlinear*, where in the case of a nonlinear problem at least one of the constraint or objective functions is nonlinear [Miettinen, 1999], and a variety of methods exist to solve them. A commonly used family of methods, discussed in the next section, are evolutionary algorithms [Goldberg, 1989]. *Hill-climbers* are also

popular owing to their simplicity. A hill-climber operates by generating an initial solution to a problem, which is modified to move it to a new region of the search space. If the new, modified, solution yields a higher quality according to the problem objective(s) then it is retained instead of the original solution. Otherwise, if the new solution is inferior to that original solution, it is discarded. The local search strategy means that it is possible for a hill-climber to become trapped in a *local optimum*, a region of the search space that offers the best solution quality of any in the local neighbourhood, but that is inferior to the *globally* optimal solution, which is the best possible solution to the problem at hand. It is difficult for a hill-climber to escape from a local optimum because there are no superior solutions within its search neighbourhood to which the algorithm can move. Hill-climbing algorithms are employed in later chapters of this thesis. Other popular methods are the downhill simplex method [Nelder and Mead, 1965], simulated annealing [Kirkpatrick et al., 1983], tabu search [Glover and McMillan, 1986] and nonlinear programming [Kantorovich, 1969]. Since this thesis is concerned with evolutionary approaches these methods, along with the great range of other techniques available, are not discussed further here.

The selection of a method with which to solve an optimisation problem can be influenced by the problem type, specifically with regard to deciding how a solution to the problem should be represented. A variety of solution representations are used, and a common choice in the realm of evolutionary algorithms is to use a binary string. A binary encoding can be used in both continuous problems, where the string represents numerical parameter values, or in discrete problems, where the binary values represent combinations of parameters. This is a particularly popular representation for evolutionary algorithms because, as will be discussed in Section 2.3, those methods develop new candidate solutions by modifying existing solutions; a binary representation facilitates straightforward mechanisms for solution modification. A common alternative is to use a vector of decimal values to represent solutions to a continuous problem, which can provide greater precision. Discrete problems sometimes lend themselves to representation by permutation.

2.2.1. Classical Aggregation Approaches

Much of the early research into solving multi-objective problems focused on how existing techniques, which had been successfully applied to single-objective problems, could be applied to multi-objective problems [Deb, 2001; Coello Coello et al., 2007]. A problem with this is that to solve, for example, a single-objective minimisation problem a single-objective algorithm compares quality between two solutions $\mathbf{x}_i$ and $\mathbf{x}_j$ such that if $f(\mathbf{x}_i) < f(\mathbf{x}_j)$ then $\mathbf{x}_i$ is the better solution. In order to facilitate this sort of comparison in the multi-objective case, early efforts went into identifying methods for aggregating objective functions. A

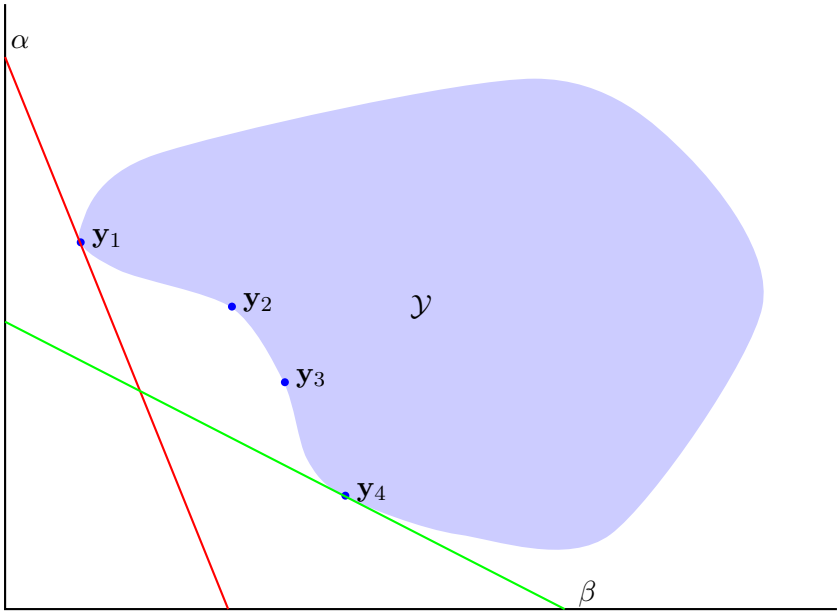


Figure 2.2.: Tangential lines formed by different weightings of two objectives. The minimisation of the weighted aggregation shown by α identifies the solution $\mathbf{y}_1$ as it lies on the region of feasible space with which the aggregation cost line is tangential. Likewise, the aggregation shown by β identifies the solution $\mathbf{y}_4$. Since the two solutions $\mathbf{y}_2$ and $\mathbf{y}_3$ lie in a non-convex region of the solution space, it is not possible to locate them using weighted aggregation, since the cost line must form a tangent with the search space but cannot intersect with $\mathcal{Y}$. To identify these two solutions, the tangent would intersect with other regions of the search space.

simple technique for doing this is to take a weighted sum of the objective functions

$$\hat{y} = \sum_{m=1}^{M} \gamma_m f_m(\mathbf{x}) \tag{2.10}$$

so that $\hat{y}$ is minimised instead of $\mathbf{y}$; $\boldsymbol{\gamma}$ is a vector of weights in which γ_m is the weight of the mth objective. This requires *a priori* knowledge to choose appropriate weights for the objectives. It may also be necessary to normalise objective ranges, which requires that the upper and lower bounds of the objectives be known; this information is not always available [Deb, 2001]. Additionally, since a weighted aggregation such as this results in a single solution, we must run multiple repeats with different weightings to explore the trade-off between objectives. The most important problem, however, is that weighted aggregation methods can only solve convex multi-objective problems (see Das and Dennis [1997] and discussion in Erghott [2005]). Figure 2.2 shows the feasible (objective) search space of a two-objective minimisation problem. The Pareto front of this problem comprises both convex and non-convex regions. A weighted aggregation operates by placing isocost lines in the space, the gradient of which are decided by the weighting assigned to each objective. The problem is solved, for a given aggregation, when the isocost lines lie at a tangent to the feasible search space, but do not intersect with it. This is shown by the

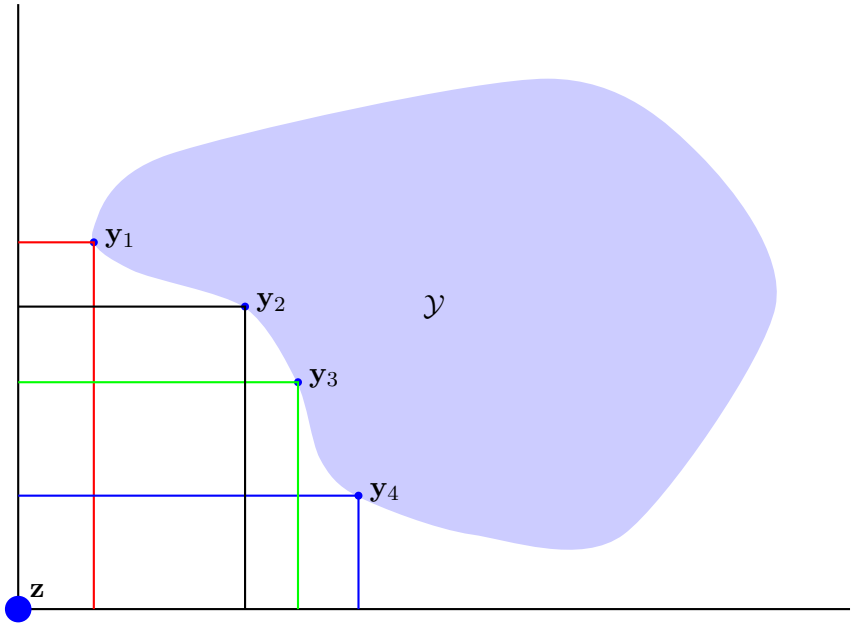


Figure 2.3.: By minimising the worst weighted objective value in the population on a single objective the Tchebycheff metric can find Pareto optimal solutions inside non-convex regions of the Pareto front.

two lines in Figure 2.2, α and β; the result of minimising the aggregation shown by the α isocost line is the solution $\mathbf{y}_1$, and the result of minimising the aggregation shown by β is $\mathbf{y}_4$. Both $\mathbf{y}_1$ and $\mathbf{y}_4$ lie on convex portions of the Pareto front, however the solutions $\mathbf{y}_2$ and $\mathbf{y}_3$ lie on a non-convex region. Although these two points are still Pareto optimal, a weighted aggregation will not be able to place an isocost line at a tangent to either without intersecting another part of the search space. As such, this part of the Pareto front will remain unexplored.

An alternative to using a weighted aggregation approach is to use a weighted metric aggregation, such as the weighted Tchebycheff metric to minimise the aggregate objective values [Deb, 2001]:

$$\hat{y} = \max_{m=1}^{M} \gamma_m |f_m(\mathbf{x}) - z_m|, \tag{2.11}$$

where $\mathbf{z}$ represents an *ideal point*. Since only one weighted objective value is considered (the worst objective value), minimising the Tchebycheff distance can find solutions that reside in convex regions of the Pareto front. Having said this, domain knowledge is still required to choose the ideal point.

2.2.2. Pareto Dominance

As we have already discussed, multi-objective problems are formulated in terms of the dominance relation, and methods that compare solutions using dominance have received

significant attention. Considering solutions to multi-objective problems in terms of dominance provides a much better alternative to aggregating objective values. There is no need to normalise objective values since they are not combined in any way, and there is no need to decide on the relative importance of individual objectives. Recently, however, a good deal of research has focused on *many-objective* optimisation problems, which comprise four or more objectives [Ishibuchi et al., 2008]. It has been noted that dominance is less suitable for comparing solutions to many-objective problems [Hughes, 2005; Farina and Amato, 2003]. This is because as the number of objectives increases, the likelihood that two randomly chosen objective vectors will be mutually non-dominating with respect to each other increases rapidly. Given an M-dimensional individual $\mathbf{y}$ at the centre of criterion space, and assuming a uniform distribution of objective vectors, the proportion of criterion space within which other individuals are mutually non-dominating with respect to $\mathbf{y}$ is determined as follows [Farina and Amato, 2002; Fieldsend, 2003]:

$$1 - \frac{1}{2^{M-1}}. \tag{2.12}$$

As a result, an optimisation method that relies on dominance to compare the quality of objective vectors will be unable to differentiate between all but a small fraction of solutions as M, the number of objectives, becomes large. An alternative to dominance is therefore required so that solutions to many-objective problems can be usefully compared. Since the more generic many-criterion populations that we consider in this thesis are also affected by this issue, we investigate techniques for comparing many-criterion individuals in Chapter 3.

2.3. Evolutionary Algorithms

A common approach to solving a complex optimisation problem is to use an *evolutionary algorithm* (EA). An EA is a computational method inspired by the principles of natural selection and evolution. Such algorithms have been successfully applied to solving single-, multi- and many-objective optimisation problems [Deb, 2001; Coello Coello et al., 2007].

Many different types of EA have been developed, but perhaps the most well known variety is the *genetic algorithm* (GA) [Holland, 1975]. A GA is an algorithm that searches the feasible solution space by generating a group of solutions called a *population*. In a GA, a solution's parameters are often referred to as a *chromosome*; the chromosome of a solution in an evolutionary algorithm is analogous to the genetic material of an organism in the natural world, encoding the 'genetic material' of an 'organism' as a binary string (although real-valued variants have been developed to facilitate more efficient searching in a continuous space [Agrawal and Deb, 1994]). GAs are iterative algorithms, where each iteration is called a *generation*. At each generation parent solutions are combined with a *crossover* operator to produce new *child* solutions that can then be perturbed to a new,

hopefully better, region of the search space with a *mutation* operator. Once all of the new child solutions have been evaluated against the problem objective function, known in the GA nomenclature as a *fitness function*, the parent population for the next generation is selected according to the fitnesses of the parent and child populations; there are various different schemes for doing this, and they are discussed shortly.

The benefit of using a GA comes from its use of a population and operators that swap genetic material between parents. Whereas an approach such as a hill climber might perturb a solution to a local minimum from which it will struggle to escape, the additional change brought about by the crossover operator can place a child solution in a region of space that is far from its locally optimal parents, allowing the algorithm to more efficiently search the feasible space and identify the globally optimal solution.

The *evolution strategy* (ES) is a search algorithm that is similar in operation to the GA and was developed at approximately the same time [Rechenberg, 1973; Schwefel, 1994; Beyer and Schwefel, 2002]. Whereas the GA was initially intended for use with binary representations, the ES was developed with a real-valued solution representation in mind; rather than an individual solution being represented by a binary string, a vector of real valued numbers is used. ESs are commonly categorised according to whether or not they employ *elitism*. The selection operator in an evolutionary algorithm can either be elitist or non-elitist. In a non-elitist algorithm, only those child solutions generated at the current generation can be selected as parents for the next generation, disregarding progenitors whatever their quality. In an elitist selection strategy, progenitors have a chance of being retained. In the traditional ES notation, a parent population of μ solutions is evolved to produce a child population of λ solutions. The non-elitist variant of an ES is called a (μ, λ)–ES while the corresponding elitist version is called a $(\mu + \lambda)$–ES. An ES modifies child solutions using mutation and recombination operators. It is common to find instances of either variety of ES in which both $\mu = 1$ and $\lambda = 1$. These are a special case, known as the $(1, 1)$–ES and $(1 + 1)$–ES respectively, and child solutions are modified using mutation alone, without recombination. A $(1 + 1)$–ES is somewhat similar to a hill climber, as discussed above, in which the superior solution is retained at each generation and a weak mutation operator will not provide enough change to release the solution from a local minimum.

2.3.1. Evolutionary Operators

Evolutionary algorithms generate solutions to problems by combining parameters from existing parent solutions to create new child solutions that are perturbed to a different, hopefully superior, region of the search space. At the end of a generation, the best solutions are retained for future use. It is the task of three evolutionary *operators* to perform these tasks, namely the *crossover*, *mutation* and *selection* operators, each of which can

be implemented in a variety of ways. Operators that best suit a particular problem and encoding can thus be chosen.

Crossover

An operator that is often used in population-based EAs, such as genetic algorithms, is *crossover*. Crossover operates on a set of solutions, a pair in the usual simplest case, and exchanges genetic material between them to create a pair of child solutions. The most basic crossover operator is *single-point crossover* [Goldberg, 1989; Deb, 2001]. Here, a point $1 \leq C < P$ is chosen at random and the first C, $x_1, \ldots, x_c$, parameters are exchanged between parent solutions. An extension of this is to choose two points and exchange the parameters between them; this is known as *two-point crossover* [Goldberg, 1989; Deb, 2001]. In *uniform crossover* [Goldberg, 1989; Deb, 2001] each parameter is exchanged with a given probability.

The crossover operators described so far were originally designed for use with a binary encoding and operators have been developed for use with real valued encodings. A prominent example of this is *simulated binary crossover* (SBX) [Agrawal and Deb, 1994]. SBX uses a probability distribution designed to have the same search capability as that of single-point crossover on a binary representation. The distribution is twin-peaked, with a peak centred on each of the two parent solutions. New child solutions are generated from values drawn from the distribution and their distance from the parent solutions is controlled by a non-negative real number called the *sharing value* that governs the shape of the distribution. With care, crossover can also be applied to a permutation-based population [Eiben and Smith, 2003]. In a permutation on the integers $1, \ldots, M$, each integer in the range must appear exactly once. Here, it is important to ensure that the result of combining two parent solutions is a permutation; *partially mapped crossover* (PMX) [Goldberg and Lingle, 1985], *edge crossover* [Whitley et al., 1989] and *order crossover* [Davis, 1985] are techniques for achieving this.

Mutation

Whereas a crossover operator typically operates on a pair of solutions, a mutation operator operates on a single solution. It is the task of a mutation operator to move a solution to a new region of the search space by modifying some proportion of its parameters.

One of the most basic mutation operators operates on solutions represented as a binary string, and works by flipping the bits on certain parameters so that a 1 becomes 0 and vice versa. The choice of parameters to be mutated can be made in a number of ways; two popular methods are to select a fixed number of parameters at random or to mutate

each parameter with some probability.

When dealing with real-valued parameters, a popular choice is the Gaussian mutation operator [Yao et al., 1999]. Parameters are selected for perturbation with some probability, and each selected parameter is modified with the addition of some Gaussian random noise. The standard deviation for the Gaussian distribution from which the mutation is drawn is a parameter which can either be set *a priori* or learned during the optimisation procedure. Alternatives to the Gaussian are, for example, to use a Laplacian distribution [Yao et al., 1999; Smith et al., 2008] which is favoured for some problems because its ability to generate perturbations more distant from the mean more frequently means that the search space is explored more rapidly. While generally small perturbations allow for more effective evolution, exploiting the high quality solutions that have already been identified, large perturbations can enhance the explorative capability of an algorithm. An alternative is to use a polynomial distribution to generate mutations [Deb and Goyal, 1996]. Here the distribution depends on a parameter, similar to that used in SBX, which generally remains fixed throughout an optimisation process [Deb, 2001]. The ability to control the width of the distribution provides a means of controlling the likely distance of a child solution from its parent in a similar way to setting the standard deviation of a Gaussian distribution, preventing premature convergence [Subbaraj et al., 2011].

When a solution is represented by a permutation, as with crossover, care must be taken to ensure that mutating a solution does not alter the solution's status as a permutation: given a permutation of P items, each item must be represented exactly once. To do this, operators that change the position of items in a solution are used. Examples are the *swap* operator, in which two randomly chosen parameters are exchanged, the *insert* operator, in which two parameters are chosen at random and the permutation adjusted so that the two parameters are placed next to each other, as well as an operator that selects a contiguous block of parameters and shuffles the order of the parameters in the block, therefore modifying the permutation [Eiben and Smith, 2003].

2.3.2. Fitness Assignment

Having generated a putative solution to an optimisation problem using crossover and/or mutation operators, its quality is evaluated with the calculation of a *fitness* value. When dealing with single-objective optimisation problems, a fitness value for the solutions in the population can simply be the objective value for each solution (as discussed in Coello Coello et al. [2007], the notion of an "objective" belongs to the problem, while the notion of "fitness" is specifically used within an algorithm to evaluate the quality of the solutions it generates). That said, objective values for single-objective solutions can be converted to fitness values by means of, for example, ranking or normalisation.

The computation of fitness values in the multi-objective case is more complicated, since a solution may have multiple conflicting objective values. In this case, a common approach has been to employ the dominance relation to produce a single fitness value for each solution. Various approaches to this have been taken, including assigning a rank to a solution according to how many layers of non-dominated solutions must be removed away before the solution itself is non-dominated in a process called *Pareto sorting* [Srinivas and Deb, 1994]; this procedure is described in detail in Chapter 3. Other methods rank the solution according to the number of solutions that it is dominated by [Fonseca and Fleming, 1993; Horn et al., 1994]. As mentioned previously, such a selection scheme is less effective for many-objective problems in which the large number of objectives renders solutions incomparable with the dominance relation. Two widely adopted approaches for resolving this are to use alternative comparison methods and reduce the number of objectives a problem comprises; both of these methods are discussed later in this thesis (Chapters 3 and 6).

Selection

A selection operator decides, based on the fitness of the current parent and child populations, which solutions should be retained as parents in the next generation. Selection operators can be broadly classified in two ways. In the first, the selection operator is restricted to considering only child solutions, while parents, regardless of their quality, are discarded. The alternative is to allow high quality parent solutions to be retained in place of child solutions of poorer quality. This latter elitist approach has been shown empirically to yield better results and is more often used [Deb, 2001].

As with crossover and mutation operators, a range of different selection strategies have arisen in the literature. In the most basic method, a *mating pool* is formed by combining the parent and child populations and selecting the μ fittest solutions in the population. An alternative is *tournament selection* [Deb, 2001]. In this method, two or more solutions are chosen at random and a tournament is played between them. The winner is the fittest solution, and is placed into the new population. The parent solutions then return to the mating pool for further selection. In *roulette wheel selection* [Deb, 2001], fitnesses are converted into probabilities of selection, such that probability of being selected is proportional to the scaled fitness. Then, the population is filled by making random draws from a uniform distribution and selecting the solution corresponding to the interval in which the random value lies.

Algorithm 1 Unified Model for Multi-objective Evolutionary Algorithms [Laumanns et al., 2000]

1: $t := 0$ *Initialise the generation counter*
2: $(E^0, \mathbf{X}^0, p_e^0) :=$ initialise() *Initialise the archive, population and elitism parameter*
3: **while** terminate$(E^t, \mathbf{X}^t, t) =$ false **do**
4: $t := t + 1$ *Update the generation counter*
5: $E^t :=$ truncate(update$(E^{t-1}, \mathbf{X}^{t-1})$) *Update the archive*
6: $p_e^t :=$ adapt$(E^t, \mathbf{X}^{t-1}, p_e^{t-1})$ *Update the elitism parameter*
7: $\mathbf{X}^t :=$ vary(sample(evaluate$(E^t, \mathbf{X}^{t-1}, p^t)$)) *Generate child solutions*
8: **end while**

2.3.3. Multi-objective Evolutionary Algorithms

A plethora of multi-objective evolutionary algorithms (MOEAs) have been developed in order to solve multi- and, more recently, many-objective optimisation problems. Most of these algorithms share common elements, and the Unified Model for Multi-objective Evolutionary Algorithms (UMMEA) was proposed by Laumanns et al. [2000]. This model is shown in Algorithm 1. Briefly, this generalises an MOEA by using an iterative process to generate solutions to a multi-objective problem from existing solutions in a population, as well as potentially those stored in an *archive*. The purpose of an archive is to store the current approximation of the Pareto front. It allows for solutions that have yet to be dominated to remain after they leave the search population in order to provide a better approximation of the true Pareto front. Additionally, depending on the implementation of the algorithm, archived solutions can be used to generate new solutions as well as those in the search population if an elitist approach is taken. Since the UMMEA is intended to generalise a MOEA its steps are high-level processes rather than the specific evolutionary operators discussed previously. EAs are generational algorithms, and so require a counter to track the current generation (Line 1). The next step initialises the elite archive for storing currently non-dominated solutions (E), an initial population of solutions ($\mathbf{X}$), and a parameter to control the amount of elitism (p_e). Setting $p_e = 0$ means that archived solutions are not used in selection, whilst $p_e = 1$ means that only archived solutions are considered (Line 2). The iterative process of evolving solutions then begins, and continues until some termination criterion is met (Line 3). The first step in the process is to update the archive with the elite solutions in the previous population, $\mathbf{X}^{t-1}$. Any of the new solutions that are mutually non-dominating with respect to those that are already in the archive are themselves archived, and any solutions already in the archive that are dominated by a new solution are discarded. It may be necessary to truncate the number of solutions within the archive (Line 5). This can be done, for example, by clustering the solutions [Zitzler and Thiele, 1999] although Fieldsend et al. [2003] point out that truncating the archive can inhibit convergence. In the next step, the elitism parameter is updated prior to its use (Line 6), noting, that in some cases the elitism parameter remains fixed throughout the execution of the algorithm. Finally, a new population of solutions is generated by selecting parent solutions from the population that are used to generate

child solutions which are mutated, or perturbed, and evaluated (Line 7). With this simple framework in mind, we now introduce some of the most prevalent MOEAs and discuss how they align with the UMMEA.

Pareto Archived Evolution Strategy

The *Pareto Archived Evolution Strategy* (PAES) [Knowles and Corne, 1999, 2000] is a $(1+1)$–ES, meaning that each generation consists of a single parent and child solution. A bounded archive of non-dominated solutions found by the algorithm is maintained.

At each generation, the choice between retaining the parent or child as the parent solution in the next generation is based on dominance. If one solution dominates the other, then it is retained as the new parent solution. If they are mutually non-dominating, and the child is not dominated by the archive, the chosen solution is the one which resides in the less crowded region of objective space so that diversity in the solution set is maximised. This information is obtained using a novel gridding technique; the objective space component of the archived solutions is overlayed with a M-dimensional grid and the cell in which each solution resides identified. The cell with the fewest solutions is identified as the most sparse, and whichever of the two solutions resides in it is retained as the next parent. Referring to the definition of the UMMEA provided above, PAES is an elitist algorithm in that, in order to be selected as a parent, a solution must already reside in the archive, also known as the *elite set*.

Other variants of PAES, based on a $(\mu+\lambda)$–ES, were investigated by Knowles and Corne [2000], however they were not found to produce a significant improvement over the original $(1+1)$ algorithm.

Non-dominated Sorting Genetic Algorithm

One of the earliest evolutionary algorithms developed to solve multi-objective problems was the *non-dominated sorting genetic algorithm* (NSGA) [Srinivas and Deb, 1994]. In NSGA, solutions are selected to continue to the next generation based on a two stage process. First, the population is ranked using a method known as *Pareto sorting*. This procedure produces a partial ordering of solutions by peeling off layers of non-dominated solutions; the first Pareto shell comprises the initially non-dominated solutions in the population. These solutions are temporarily discarded leaving a new non-dominated set, which become shell 2. This process continues until all solutions have been assigned to a shell[1]. NSGA uses the resulting ranking of solutions as an intermediate fitness value.

[1] Chapter 3 provides a more detailed discussion of Pareto sorting in the context of constructing many-criterion league tables.

Then, in order to differentiate between those solutions residing in the same Pareto shell the algorithm determines the proximity of solutions to their nearest neighbours by computing a *sharing value* that takes a value of 0 if the solutions are more than a specified distance apart in objective space, σ_{share}, and a value based on their distance if they are closer, and hence part of a niche. The sharing value s_{ij} is calculated as follows:

$$s_{ij} = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{\text{share}}}\right) & : d_{ij} < \sigma_{\text{share}} \\ 0 & : \text{otherwise} \end{cases} \tag{2.13}$$

where d_{ij} is the distance between the objective vectors of solutions $\mathbf{y}_i$ and $\mathbf{y}_j$. A *niche count* for a solution is computed by adding the sharing values between it and the other members of the same Pareto shell, and a final fitness value computed by dividing the intermediate fitness value by the solution's niche count. This niching mechanism is based on the normalised Euclidean distance between solutions.

The original NSGA algorithm received three criticisms: it is computationally expensive, is not elitist, and requires the setting of the parameter σ_{share}. To address these issues, Deb et al. [2000] developed a successor, NSGA-II. NSGA-II used a modified, faster, Pareto sorting procedure which reduced the time complexity of the algorithm from $O(MN^3)$ to $O(MN^2)$, recalling that M is the number of objectives and N the population size. The new algorithm is elitist in that the selection of future parents is based on both the current parent and child population; there is no archive of elite solutions. A parameterless niching method, the *crowding distance*, was also developed. To compute the crowding distance for a solution, a distance is identified such that it defines the largest axis-parallel (hyper-) cube that can be placed in objective space around the solution without intersecting any other solution. NSGA-II was developed in 2000, and is still a staple of the multi- and many-objective optimisation literature against which other algorithms are tested (for example, [Garza-Fabre et al., 2010; Smith et al., 2008; Zhang and Li, 2007; Knowles, 2006; Bader and Zitzler, 2011]). A recent drawback with the algorithm is that it has been found to provide poor approximations to the Pareto front of a many-objective problem; this is because of its use of dominance to evaluate the quality of the solutions it generates, and as previously mentioned dominance cannot discriminate well between many-objective solutions. However, as we will discuss in Chapter 6, research is now considering how the number of objectives comprising an optimisation problem can be reduced, and NSGA-II is commonly used to evaluate the efficacy of such dimension reduction approaches [Deb and Saxena, 2005; Brockhoff and Zitzler, 2007a; Singh et al., 2011].

Strength Pareto Evolutionary Algorithm

The Strength Pareto Evolutionary Algorithm (SPEA) [Zitzler and Thiele, 1999] is an elitist EA, maintaining an elite archive of non-dominated solutions found during the search that

is used in the selection of future parent solutions, as well as a population of solutions. The fitness of an archived solution is defined in terms of its *strength*, where the strength is the proportion of solutions in the main population which it dominates:

$$s_i = \frac{n}{N+1} \tag{2.14}$$

where n is the number of dominated individuals in the population of N solutions. The fitness s'_j of a solution in the main population is the summed strength of the dominating archive solutions, with one added to ensure that population members are never fitter than those in the archive:

$$s'_j = 1 + \sum_{i, \mathbf{y}_i \prec \mathbf{y}_j} s_i. \tag{2.15}$$

Selection is based on the fitnesses of solutions in the union of the population and archive. By incorporating fitnesses defined in these terms the algorithm employs a niching mechanism, designed to preserve population diversity, based on dominance, rather than the distance-based methods used by NSGA and NSGA-II. The archive used by SPEA is bounded, and the removal of solutions is controlled by a hierarchical clustering method.

Zitzler et al. [2001] extended the original SPEA algorithm, making several alterations. The first modifies the fitness assignment procedure so that the number of dominating individuals in *either* the population or the archive, rather than just the archive, is considered when assigning strength. This prevents the solutions in the population receiving the same fitness value if they are all dominated by an archive containing solutions which all have the same fitness value. The selection procedure is also modified, firstly, so that solutions on the extremes of the estimated Pareto front are not removed. The second change concentrates the search on the non-dominated individuals by adding the non-dominated individuals in the population to the archive and selecting from the resulting combined population of elite solutions. If the number of solutions in this combined archive exceeds the specified archive size then clustering is used to discard solutions. If it is less than the specified size, dominated solutions from the population are added to reach the required number.

All three of the methods outlined here provide good approximations to the true Pareto front of a multi-objective problem, however since they rely on dominance as their primary method of selecting parent solutions for the next generation they struggle to differentiate between solutions to many-objective problems. This results in a lack of selection pressure, and the effectiveness of the algorithms tends toward that of a random walk in the search space. A popular current trend in evolutionary optimisation is to develop methods that can provide sufficient selection pressure to locate the Pareto front of a problem comprising a large number of objectives [Ishibuchi et al., 2008], and we will now proceed to briefly outline some of the approaches that have been taken.

2.3.4. *Many*-objective Evolutionary Algorithms

Having observed that the dominance-based EAs discussed so far can suffer from a lack of selection pressure when optimising a many-objective problem, we now turn our attention to selection mechanisms specifically developed for solving such problems.

Rank-based MOEAs

Since one of the principal reasons that multi-objective evolutionary methods are not generally effective for many-objective problems is their reliance on dominance as a measure of quality, another avenue of research is the search for an alternative method for assessing solution quality. A promising line of work in this area is to consider alternative ranking methods that allow an optimiser to maintain sufficient selection pressure to locate the Pareto front. Solutions can then be selected based on their rank using the selection schemes described earlier in this chapter, for example, tournament selection and roulette wheel selection.

Average Rank One of the more prevalent methods that has been proposed is *average rank* [Bentley and Wakefield, 1998; Corne and Knowles, 2007; Garza-Fabre et al., 2009; Li et al., 2010a]. Given that the particular scales of individual criteria are immaterial in determining whether one individual is better than the other, the population is converted to *rank coordinates*. The population is ranked M times, once for each objective, resulting in a set of ranks r_{im} for each individual. Then, the average rank $\bar{r}_i$ of the ith individual is calculated by taking the average of these ranks. The rank of the ith individual on the mth objective is r_{im}, and this has the effect of normalising the objectives by placing each on the range $[1, N]$. The individual with the objective value on a particular objective is assigned rank 1, while the worst is assigned rank N. Assuming that there are no ties, each rank appears M times; ties are dealt with by assigning the average of the range of ranks occupied by the tied individuals. The average rank is then the average of an individuals ranks across all M objectives:

$$\bar{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_{im}. \tag{2.16}$$

This formulation is clearly similar to the weighted sum aggregation discussed earlier in this chapter. The difference is that this method operates on the *ranked* objective values, rather than the raw objective values used in weighted sum methods. Using ranks in this way provides a simple way of normalising the solutions, placing the objectives on the same scale during each stage of the optimisation process.

In a study of various alternative selection operators [Corne and Knowles, 2007], such as *favour relation* [Drechsler et al., 2001] and *preference ordering* [di Pierro et al., 2007] (both

of which we will discuss shortly), a variant of average rank, which operates on the non-dominated solutions in a population, was found to produce a good estimate of the Pareto front in combinatorial problems.

Maximum Rank Also proposed by Bentley and Wakefield [1998] was the *maximum rank* method. Here, as with average rank, the population are converted to rank coordinates, such that r_{im} is the rank of the ith individual on the mth objective. Rather than averaging the ranks those individuals with the best rank on specific objectives are retained as parents in the next generation. Bentley and Wakefield [1998] propose creating a mating pool with this approach and then creating children from random combinations of parents from that pool. Selection based on maximum rank has a drawback that it tends to prefer solutions that are extreme in one or two objectives. This can lead to the selection of solutions with generally poor performance on the majority of the objectives but that are extremely good on one of the problem objectives.

Favour Relation The favour relation is a simple alternative to comparing solutions with dominance that is better able to discriminate between individuals in a high-dimensional space. A solution $\mathbf{y}_i$ is *favoured* over solution $\mathbf{y}_j$ if the number of objectives on which $\mathbf{y}_i$ is superior is greater than the number of objectives for which $\mathbf{y}_j$ is superior. Drechsler et al. [2001] observe that there are a number of advantages to using the favour relation to compare solutions, for example it is scale invariant and does not require weighting. That said, it is possible to weight objectives to impart an indication of their relative importance.

The relation is not transitive [Drechsler et al., 2001] and as such is not a partial order. Despite the increased discrimination between solutions, there are at most M ranks, meaning that the resulting ordering might still provide insufficient information with which to discriminate between individuals.

Preference Ordering An approach taken by di Pierro et al. [2007] is to compute the *preference order* of the non-dominated individuals in a population. Preference ordering is based on the notion of *efficiency of order k*. An individual is efficient of order k if it is non-dominated in all of the $\binom{M}{k}$ k-objective subsets that can describe the population and dominated in at least one of the $\binom{M}{k-1}$ $(k-1)$-objective subsets. This notion is used to refine the ordering of solutions produced with a non-dominated sorting scheme, such as that employed by NSGA-II. Each non-dominated solution is ranked according to its efficiency of order k such that the higher k in which it is efficient, the better the rank it receives. The result is a partially ordered set that provides better discrimination than using Pareto sorting alone. In an extension to this scheme, proposed by di Pierro et al. [2007], an individual is efficient of order k with degree z if it is non-dominated in at least z of the k-objective subsets. This extension facilitates discrimination between individuals

that are efficient of the same order k.

This approach is shown by di Pierro et al. [2007] to offer better convergence to the Pareto front, as well as better coverage of it, when tested on a set of benchmark test problems. Unfortunately, this performance is at the expense of the computational complexity required to evaluate all of the possible objective subsets. As such, the application of preference ordering is generally limited to small many-objective problems. For example, a study by Garza-Fabre et al. [2009] compares the performance of various ranking methods on a set of many-objective benchmark problems, but restricts the use of preference ordering selection to $M \leq 20$.

Global Detriment Three alternative selection operators proposed by Garza-Fabre et al. [2009] are *global detriment*, *profit* and *global best*. Global detriment compares a solution with the others in the solution set $\{\mathbf{y}_i\}$ and measures the amount by which it is inferior on each normalised objective:

$$gd(\mathbf{y}_i) = \sum_{\mathbf{y}_i \neq \mathbf{y}_j} \sum_{m=1}^{M} \max(y_{im} - y_{jm}, 0). \tag{2.17}$$

Global detriment is to be minimised, such that a low global detriment receives a good rank.

Profit *Profit* considers superiority of solutions in terms of the *gain* that one solution generates over another. They define the *gain*, which is to be maximised, of a solution $\mathbf{y}_i$ with respect to $\mathbf{y}_j$ as the sum of the differences between their normalised objective values on objectives where $y_{im} < y_{jm}$. More formally:

$$gain(\mathbf{y}_i, \mathbf{y}_j) = \sum_{m=1}^{M} \max(y_{im} - y_{jm}, 0) \tag{2.18}$$

The overall profit is then calculated with respect to the gain of a solution compared with the other solutions in the population:

$$pr(\mathbf{y}_i) = \max_{j \neq i}(gain(\mathbf{y}_i, \mathbf{y}_j)) - \max_{j \neq i}(gain(\mathbf{y}_j, \mathbf{y}_i)). \tag{2.19}$$

In order to rank solutions, overall profit is maximised.

Global Best The final method presented is *global best*, which seeks to minimise the distance between solutions and the current "ideal" point, an M-dimensional vector $\tilde{\mathbf{y}}$ in which $\tilde{y}_m = \min_i(y_{im})$ is the best (normalised) value for the mth objective in the current

population:

$$gb(\mathbf{y}_i) = \delta(\mathbf{y}_i, \tilde{\mathbf{y}}) \tag{2.20}$$

where δ is a metric; Garza-Fabre et al. [2009] use Euclidean distance.

Having introduced these three measures, Garza-Fabre et al. [2009] perform a comparative study in which the three methods are compared to methods from the many-objective optimisation literature: standard non-dominating; average rank, as well as maximum rank, favour relation, weighted summed aggregation and preference ordering are all evaluated. The three methods proposed in the study, as well as average rank and weighted summed aggregation are shown to provide good convergence, however the diversity of the population of solutions produced in the experiments is not considered.

Preserving Diversity in Rank-based MOEAs

Several recent works have observed, and we have confirmed, that whilst ranking methods such as average rank and global detriment provide good convergence to the true Pareto front, this is at the expense of the diversity of the solution set [Garza-Fabre et al., 2010; Li et al., 2010a,b].

The *Clustering-based Elitist Genetic Algorithm* (CEGA) was presented by Garza-Fabre et al. [2010] and attempted to enhance the diversity of the population using agglomerative clustering in concert with global detriment ranking. A proportion of the next parent population is selected with standard global detriment ranking, while the remainder is selected from a population on which clustering has been used to prune highly similar solutions. This avoids the problem that a solution of very high quality will be selected frequently, creating large numbers of child solutions that are close in objective space, and as such solutions in other regions of objective space are given higher priority. They demonstrate that CEGA retains the good convergence properties of an EA that uses global detriment to rank solutions, while providing a solution set with better coverage of the Pareto front than would be achieved using global detriment alone.

Li et al. [2010a] propose two methods for enhancing diversity in a many-objective optimiser. The first modifies average rank selection with an adaptive grid by introducing a layering mechanism in which only those solutions which are in the current layer are eligible for selection. At the start of the procedure, all of the solutions are in the current layer. The individual with the best average rank is selected, and those solutions that are identified using the grid as being neighbours of the selected individual are relegated to the next layer and can no longer be selected. Once all of the individuals in the current layer have been exhausted, if the new parent population still requires solutions, the next layer becomes the current layer. This continues until a sufficient number of solutions have been selected. A second method using an adaptive grid is proposed by Li et al. [2010b]. In this

work, a grid ranking method is used to determine which solution should be selected. This is based on minimising the sum of the grid coordinates of a solution. Each cell is assigned a rank in each objective, according to how far from the optimal position it is, and the solutions are ranked according to the sum of their ranked grid coordinates in a procedure that is equivalent to average rank. If two solutions are tied according to grid ranking then the adaptive grid crowding method from the authors' earlier work [Li et al., 2010a] is employed. If this does not resolve the tie then a third method, *grid coordinate point distance*, the distance from a solution to the optimal coordinates of the grid element in which it resides, is used. Both of these methods are evaluated on the DTLZ test problems [Deb et al., 2002] and are shown to perform well.

These different methods provide improved diversity enhancement compared to the ranking methods used in isolation, however there is still work to be done in order to produce algorithms that approximate many-objective Pareto fronts as well as dominance-based evolutionary algorithms approximate multi-objective fronts.

Many-objective Aggregation Methods

Research into solving many-objective problems has also considered aggregation techniques. Approaches, such as a multi-objective genetic local search approach [Ishibuchi and Murata, 1996] and Pareto Simulated Annealing [Czyzak and Jaszkiewicz, 1998], use aggregation methods to generate an approximation to the Pareto front of a multi-objective problem in a single run, as opposed to the more common approach of executing multiple runs of an algorithm for different objective weightings. In the many-objective domain, Hughes [2003] presented Multiple Single-objective Pareto Sampling (MSOPS), which is an algorithm that used aggregation methods to optimise a set of arbitrary target vectors $\{\mathbf{q}\}$ initialised by randomly sampling directions in the search space originating from a utopian point. Fitness was determined by the rank of solutions according to how well they optimise each target vector, which was evaluated using a weighted sum aggregation:

$$\hat{f}(\mathbf{x}) = \mathbf{f} \cdot \mathbf{q} \tag{2.21}$$

$$= \sum_{m=1}^{M} f_m(\mathbf{x}) q_m \tag{2.22}$$

Hughes incorporated the MSOPS fitness assignment procedure into a *differential evolution* (DE) algorithm [Storn and Price, 1997]; however, he observed that it can also be incorporated into other evolutionary algorithms; a subsequent paper [Hughes, 2007b] used an evolutionary program. Hughes [2003] applied the method to a bi-objective problem but noted that since it does not rely on dominance for the comparison of solution quality it was a suitable candidate for optimising many-objective problems.

Hughes [2007b] presented MSOPS-II, in which it was shown that a useful extension in some cases is to generate target vectors manually based on *a priori* knowledge of the problem search space. That said, Hughes suggested that it is also useful to use automatically generated target vectors for exploring the search space of a previously unseen problem.

An alternative is to consider the volume of space dominated by an objective vector. The *hypervolume indicator*, also known as the *S metric*, [Fleischer, 2003] is a performance measure which is capable of characterising both convergence and diversity. It measures the volume of objective space dominated by an estimated Pareto front and bounded by a *reference point*. Common strategies for choosing a reference point are to use the worst possible value for each objective, if they are known, or to use the worst value in the population for each objective [Bradstreet et al., 2006]. The hypervolume is maximised when a population is converged to and covers the Pareto front and is computed by summing the volume of hypercubes dominated by the estimated Pareto front. More formally, given a reference point $\dot{\mathbf{y}}$, the hypervolume is defined as [Brockhoff et al., 2008]:

$$I_H(\mathbf{Y}) = \text{vol}\left(\bigcup_{\mathbf{y}_i \in \mathbf{Y}} [y_{i1}, \dot{y}_1] \times [y_{i2}, \dot{y}_2], \times \ldots \times [y_{iM}, \dot{y}_M]\right) \tag{2.23}$$

Without properly converging to and covering the true Pareto front the amount of dominated space between the solution set and reference point will be smaller than if the algorithm had properly converged and covered the true Pareto front. Therefore, given a set of solutions, a selection operator will prefer those that maximise the hypervolume.

Whilst the hypervolume can be measured for any number of objectives, the complexity of its calculation is extremely expensive [While, 2005]. Much work has been done to identify algorithms for reducing the computational complexity of an exact hypervolume calculation [While et al., 2006, 2012]. A promising alternative approach is to use Monte Carlo sampling to produce an estimate of the hypervolume rather than calculating it exactly [Everson et al., 2002]. A set of samples is taken from the region enclosed by an ideal point at which all objectives are minimised and the reference point. The sampled hypervolume is then simply the proportion of samples dominated by the estimated Pareto front. In order to achieve an accurate result, it is necessary to have a large number of samples to ensure sufficient resolution which limits the use of this approach in an iterative algorithm.

Due to its scalability, the hypervolume indicator has seen use as the selection operator in an MOEA. Such examples have calculated the hypervolume [Bradstreet et al., 2006] however a recent advance was the development of a hypervolume-based search algorithm called HyPE which resolves the complexity issue by calculating the hypervolume for multi-objective problems ($M \leq 3$) and estimating it with sampling for many-objective problems ($M > 3$) [Bader and Zitzler, 2011]. The algorithm is shown to provide superior results to the algorithms it is compared against, including NSGA-II and SPEA2.

Beside its use as a selection operator, the hypervolume has been frequently used as a means of comparing experimental results produced by a MOEA. The purpose of an evolutionary optimiser is to converge to and cover the true Pareto front, at which time the hypervolume is maximised. By computing the hypervolume at each generation, it is possible to see how far into the execution of an optimiser, if at all, this happens.

As was the case with the rank-based methods, these approaches have made progress toward the goal of optimising a many-objective problem, however there are still problems to be overcome, for example the complexity of incorporating a technique such as the hypervolume into an iterative algorithm.

Objective Reduction in MOEAs

One of the major fields of interest within the MOEA community currently, discussed in more detail in Chapter 6, is the identification of redundant objectives which can be removed, reducing a many-objective problem to a multi-objective problem that is more easily solved by a standard dominance-based MOEA. As with other types of dimension reduction, the process involves identifying the objectives whose removal will cause the smallest loss of information. This has been done by identifying the objectives responsible for providing the largest proportion of the variance in the objective vectors [Deb and Saxena, 2005]; discarding the objectives which cause the smallest change in the pairwise dominance relationships between objective vectors [Brockhoff and Zitzler, 2007a]; and identifying the corners of the Pareto front for a problem so that the change in the proportion of non-dominated solutions as objectives are discarded can be quantified [Singh et al., 2011]. Objective reduction is the subject of Chapter 6 and these methods are discussed in further detail there.

2.3.5. Test Problems

Throughout this thesis we illustrate the methods we present on example populations, some of which are solutions to test problems. Therefore, we briefly introduce some of the *test problems* that appear in later chapters. An integral part of the development of MOEAs is to apply them to suites of test problems. Test problems are multi-objective problems for which the true Pareto front is known, allowing evaluation of how close to the true front the solution set produced by a MOEA is, and how well it covers the true front. They also facilitate the evaluation of a MOEA on various problem features that are found in real world problems, such as degenerate fronts, in which the true Pareto front of an M-objective problem is Q-dimensional ($Q < M$), or discontinuities, regions of infeasible space within the Pareto set or front.

DTLZ Problem Suite

The DTLZ test problem suite [Deb et al., 2002][2] comprises extensions of the popular ZDT problems [Zitzler et al., 2000] and is designed to aid research with the advent of research into both multi- and many-objective problems. The problems are scalable to any number of objectives and parameters, and provide researchers with a variety of problem features. DTLZ1 and DTLZ3 are *multi-modal*, meaning that the objective space contains deceptive fronts, local optima in which an optimiser can become trapped, facilitating the evaluation of an algorithm's ability to escape from such regions. DTLZ5 contains a degenerate front, where the true Pareto front comprises fewer dimensions than the objective space in which it exists. The problems have a structured parameter space, in which individual parameters are responsible for either the position of a solution on the Pareto front, or its distance to the Pareto front. This information is useful in computing the distance from the Pareto front in order to tell how well the algorithm has approximated the true Pareto front.

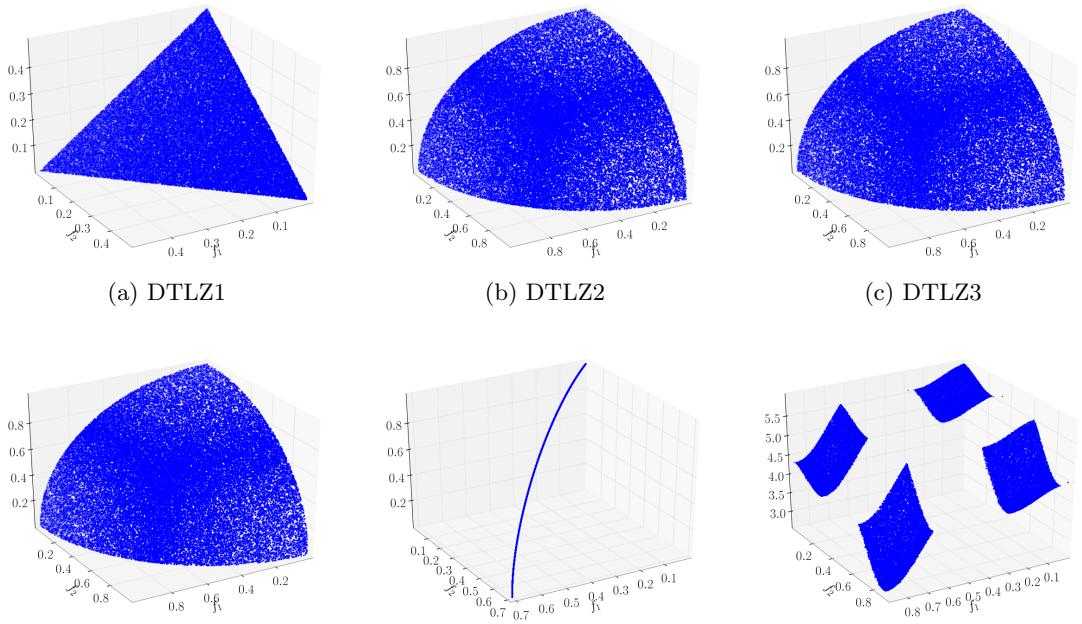
Deb et al. [2002] provide suggested parameterisations which have been widely used in the literature. These define the number of parameters for a given number of objectives; for example, for DTLZ1 the suggested number of parameters is $M - 1 + k$, where $k = 5$. For DTLZ2, $k = 10$ is suggested. Here, k is the number of *distance* parameters, which control how far a solution is from the Pareto front. The first $M-1$ parameters are position parameters, which control in which region of the Pareto front the solution resides. We note that this decomposition of parameters can make the work of an optimiser unrealistically easy, since it is a feature that has been introduced to make the analysis of results easier rather than to produce a more realistic problem. The DTLZ problems themselves have been extended so that problem features beyond those envisaged by Deb et al. can be tested. Examples of this are the creation of the DTLZ5(I, M) [Deb and Saxena, 2005] and DTLZ2$_{BZ}$ [Brockhoff and Zitzler, 2007b]. These problems comprise redundant objectives for testing objective reduction methods.

Figure 2.4 presents example Pareto optimal solution sets for the DTLZ problem suite [Deb et al., 2002]. These sets were produced by evaluating Pareto optimal solutions for 3-objective problem instances. The number of parameters are as proposed by Deb et al. [2002].

WFG Toolkit & Suite

Huband et al. [2005, 2006] conducted a review of multi-objective test problems and found that the DTLZ problems omitted some desirable problem features. For example, they ob-

[2]The work presented by Deb et al. [2002] was preceded by a technical report [Deb et al., 2001] in which the problems were numbered differently. In this thesis we refer to the numbering according to Deb et al. [2002].

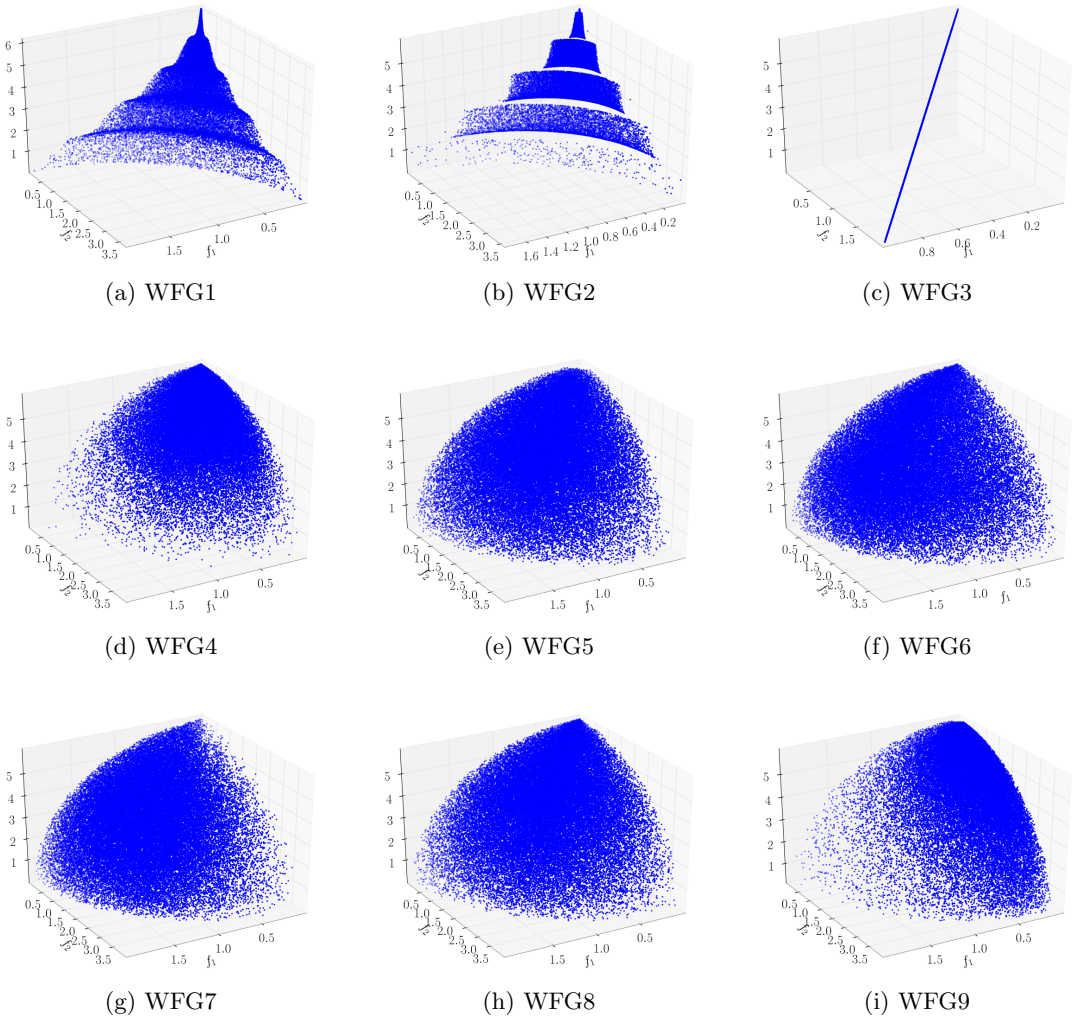


(a) DTLZ1 (b) DTLZ2 (c) DTLZ3

(d) DTLZ4 (e) DTLZ5 (f) DTLZ6

Figure 2.4.: Three-objective instances of DTLZ1-6. The solutions were sampled from the Pareto sets of 3-objective instances of problems; the number of parameters is as specified by Deb et al. [2002]. These plots show the corresponding objective vectors. The problems retain the geometry shown here in $M > 3$ objectives.

served that the suite does not contain problems with *flat regions* in the fitness landscape. In addition, they point out that DTLZ5 is only degenerate when $M < 4$. As a result of their survey, they proposed the Walking Fish Group (WFG) toolkit, a framework for the creation of test problems based on a set of transformation functions, as well as a new set of benchmark test problems, the WFG problem suite. Desired characteristics are introduced into the problems via transformation functions, which operate on the parameters. Shape functions define the shape of the true Pareto front as either linear, concave, convex, mixed (concave and convex) or disconnected. Further transformation functions define the fitness landscape, introducing bias regions and shifting the location of optimal regions, and reduction transformations introduce dependencies between parameters. Unlike the DTLZ problems, the objectives in the WFG problems are on different scales. Figure 2.5 presents example objective space solution sets for the WFG problem suite [Huband et al., 2006]. The solutions are sampled from the true Pareto set of test problems WFG1-9. Note that WFG4–9 (Figures 2.5(d)—2.5(i)) all employ the same shape function and thus the shape of the Pareto front is the same for all six problems.



(a) WFG1 (b) WFG2 (c) WFG3

(d) WFG4 (e) WFG5 (f) WFG6

(g) WFG7 (h) WFG8 (i) WFG9

Figure 2.5.: Three-objective instances of the WFG problems; the number of parameters in each case is as recommended by Huband et al. [2005]. The solutions were sampled from the true Pareto set of the problems. These plots show the corresponding objective vectors. The geometries shown here are retained in $M > 3$ objectives.

2.4. Multi-criteria Decision Making

While the methods on which it is based are drawn from evolutionary multi- and many-objective optimisation, much of the work presented in this thesis relates to *multi-criteria decision making* (MCDM). In particular, we seek to develop methods for visualising populations of alternatives between which a choice must be made by a decision maker. As such, this section presents a short introduction into methods used in that field for differentiating between multi-criterion individuals representing alternatives to an MCDM problem. We begin by outlining how a decision maker's preferences can be incorporated into the MCDM framework, before examining techniques used to rank alternatives in MCDM.

2.4.1. Selecting an Individual based on Decision Maker Preferences

Of crucial importance in the construction of a MCDM problem is the role of the decision maker. The decision maker is a human who will interact with the system developed to solve their problem, and will eventually be responsible for selecting a single solution to it based on the analysis conducted. Generally, a decision maker is a person with some specialist or expert knowledge about the domain in which the problem resides. They will have some problem-specific knowledge relating to why a solution for the decision making problem at hand is required; this specialist knowledge of the problem and its domain are incorporated into the MCDM framework in the form of *preferences*. The decision maker is required to used their specialist knowledge in order to find the *preferred* solution to the problem at hand.

Preferences in a MCDM can be incorporated in three different ways [e.g., Coello Coello et al., 2007]: *a priori* preference articulation; interactive preference articulation; and *a posteriori* preference articulation. In *a priori* preference articulation, preference information is obtained by the decision maker before the multi-criterion analysis is started. The preferences provided by the decision maker at the start of the process remain fixed throughout. This type of preference articulation is popular, but inflexible. Rather than require a fixed set of preferences from the decision maker before the multi-criterion analysis can start, an alternative is to involve the decision maker in the analysis. Additional difficulties might be encountered if the decision maker does not fully grasp the complexities of the MCDM framework employed and may have overly optimistic or pessimistic expectations [Miettinen, 2008]. Interactive preference articulation is perhaps a better alternative to the *a priori* approach because it allows more flexibility. By involving the decision maker in the analysis process, the putative solutions generated can be based on knowledge and intuition that they have gained as the analysis process proceeds. A drawback is that tasks designed to elicit current decision maker preferences (e.g., ranking individuals and adjusting constraints) can be non-trivial [Coello Coello et al., 2007], however as discussed by [Miettinen et al., 2008], involving them as problem solutions are generated avoids generating solutions that, although feasible, are not of interest to the decision maker. As such, the number of solutions from which the final selected solution must be chosen is smaller. The final option, *a posteriori* preference articulation, does not require interaction with the decision maker until a set of putative solutions to the problem has been generated. At this point, the decision maker must select one. A potential issue with such an approach, discussed by Coello Coello et al. [2007], is that at no point in the solution generation process has the problem domain been constrained with the application of decision maker preferences; as such, the number of putative solutions from which the decision maker must make their selection is likely to be large. Also, as noted by Miettinen [2008], the process by which this large set of solutions, many of which may be of little interest to the decision maker, are generated might be prohibitively expensive.

Various techniques exist with which decision maker preferences are incorporated into the MCDM framework. One example, proposed by Magoc et al. [2011], is an *a priori* technique that uses the preferences of the decision maker to construct "preference constraints" before solutions are generated by some optimisation process. This is done by asking the decision maker to score the importance of relative changes in criterion values, quantifying how desirable it is to improve the score of an individual according to one criterion at the expense of a reduction in another criterion value for that individual. The result is a much smaller search space in which the optimisation process must operate, simplifying the optimisation problem.

Another technique, proposed by Fenton and Wang [2006], discussed the problem of data integrity in MCDM tasks. They observe that the data used in such tasks can often be suspect, and propose that preferences can be incorporated into a fuzzy MCDM framework to capture a measure of the uncertainty of criterion values, before preferences are used to take account of the level of risk and confidence the decision maker has about their selection. Their study indicates that such an approach is useful in "human-oriented" decision tasks.

2.4.2. Ranking Alternatives in MCDM

Methods for ranking alternative solutions to MCDM problems are, as in the case of multi- and many-objective evolutionary algorithms described above, an important component of multi-criterion analysis. One of the important reasons for this is that, given a total ordering of the alternatives, the decision maker is presented with a single preferred solution. Also, comparing the rank of an individual according to the different criteria by which it is measured can provide information about the degree of correlation or conflict between the criteria in the population. Three popular approaches to ranking individuals in the MCDM literature are *TOPSIS*, *ELECTRE* and *PROMETHEE*.

The *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) [Yoon and Hwang, 1995] ranks multi-criterion individuals according to their distance to some predefined ideal solution, as well as to the worst possible solution. A preference order is defined over the relative distance to the ideal point, on which basis the multi-criterion alternatives are ranked. This approach has been used in a variety of MCDM problems [e.g., Zavadskas et al., 2006; Triantaphyllou et al., 1998] and has also been used to rank solutions to as part of a multi-objective GA [Coello Coello et al., 2007].

Another common MCDM ranking techinque is ELECTRE (elimination and choice translating algorithm) (Benayoun et al. [1966]; discussed in Coello Coello et al. [2007]). The individuals are representated as a directed graph, which is used to infer a preference order. The graph is defined using the notions of *concordance*, *disconcordance* and *thresholds*. The concordance between a pair of individuals quantifies the number of criteria in which

one individual is superior to the other, while disconcordance is a measure of the number of criteria in which that individual is superior. Two threshold values are then used to place edges between the nodes representing individuals in the graph such that an edge is placed between two individuals if their concordance value is greater than the concordance threshold value and their disconcordance value is less than the disconcordance threshold value. The original version (ELECTRE I) has been modified to create various successors [Coello Coello et al., 2007] and remains a common choice ranking alternatives in MCDM problems [e.g., Thokala, 2011; Ling et al., 2012].

PROMETHEE (Preference Ranking Organisation Method for Enrichment Evaluations) was first demonstrated by Brans et al. [1986], and like ELECTRE various versions exist [Lewi and Hoof, 1992]. Defined in the original work, PROMETHEE I provides a partial ordering of individuals. A *preference index* is defined based on a preference function, which is designed to model the decision maker's preferences for a particular criteria; the preference index is a weighted average of the preference functions for a pair of individuals. PROMETHEE I only considers those cases where two individuals are comparable; a second variant proposed by Brans et al. [1986] is capable of producing a total order by taking into account incomparable individuals, although the authors comment that this may provide a less realistic ordering of individuals in favour of receiving an exact ranking. Like TOPSIS and ELECTRE, PROMETHEE has been used for ranking solutions in multi-objective problems [Coello Coello et al., 2007].

2.5. Comparing Permutations

Many of the methods introduced in this thesis rely on ranking population based on combinations of different criteria. Consequently it is important that we are able to differentiate between these rankings, and various statistical tools exist to do this. Two of these methods, Spearman's footrule [Spearman, 1904, 1906; Diaconis and Graham, 1977] and Kendall's τ metric [Kendall, 1938], are used throughout this work and introduced here.

2.5.1. Spearman's Footrule

Spearman's footrule [Spearman, 1904, 1906] is the city block distance between two permutations. Let $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$ be two permutations, where a permutation is an ordering of the integers $1, \ldots, N$, such that each integer in the range appears exactly once; they might be, for example, rankings of a population according to the mth and nth objectives describing a population. A permutation of length N is used to order N items in a collection, such as the populations we consider in this thesis. Spearman's footrule δ^{SF} is the summed

absolute difference between the positions of individuals in the two permutations:

$$\delta^{\mathrm{SF}}(\boldsymbol{\pi}, \boldsymbol{\rho}) = \sum_{i=1}^{N} |\pi_i - \rho_i| \,. \tag{2.24}$$

Assuming that $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$ are permutations of N elements, the maximum possible value of the metric, $\delta^{\mathrm{SF}}_{\mathrm{max}}$, which occurs when the permutation $\boldsymbol{\pi}$ is the reverse of $\boldsymbol{\rho}$, is:

$$\delta^{\mathrm{SF}}_{\mathrm{max}} = \begin{cases} \frac{N^2}{2} & \text{when } N \text{ is even} \\ \frac{(N+1)(N-1)}{2} & \text{otherwise.} \end{cases} \tag{2.25}$$

Spearman's footrule is a metric, a distance between two permutations, although this is only the case when there are no tied ranks. When tied ranks are present, the metric condition $\delta(i, j) = 0$ if and only if $i = j$ is not satisfied because two elements in the ranking can have the same value.

2.5.2. Kendall's τ Metric

Kendall's τ metric, δ^{KT}, is also a distance between two permutations, but it considers how the pairwise ordering of individuals changes between permutations. If the ordering between a pair of individuals is unchanged ($\pi_i > \pi_j$ and $\rho_i > \rho_j$ or $\pi_i < \pi_j$ and $\rho_i < \rho_j$) then $\tau_{ij}(\boldsymbol{\pi}, \boldsymbol{\rho}) = 0$. Otherwise, if the pairwise ordering is different, $\tau_{ij}(\boldsymbol{\pi}, \boldsymbol{\rho}) = 1$. Then Kendall's τ metric is defined as:

$$\delta^{\mathrm{KT}}(\boldsymbol{\pi}, \boldsymbol{\rho}) = \sum_{ij} \tau_{ij}(\boldsymbol{\pi}, \boldsymbol{\rho}). \tag{2.26}$$

Like Spearman's footrule, the maximum value of δ^{KT}, $\delta^{\mathrm{KT}}_{\mathrm{max}}$, occurs when $\boldsymbol{\pi}$ is the reverse ordering of $\boldsymbol{\rho}$, and is:

$$\delta^{\mathrm{KT}}_{\mathrm{max}} = \frac{N(N-1)}{2}. \tag{2.27}$$

As is the case with Spearman's footrule, Kendall's τ is a proper metric if there are no ties present. In the presence of ties, two different elements in a ranking can have the same value, and therefore a distance of zero.

Spearman's footrule and Kendall's τ are known to be well correlated [Diaconis and Graham, 1977]. A demonstration of this appears in Figure 2.6 in which 1000 random permutations of length 20 have been compared with the ordered version of the permutation using both Spearman's footrule and Kendall's τ. The correlation shown here corresponds with the inequality given in Diaconis and Graham [1977] that $\delta^{\mathrm{KT}} \leq \delta^{\mathrm{SF}} \leq 2\delta^{\mathrm{KT}}$. This relationship is indicated by the two monotonically increasing lines in Figure 2.6.

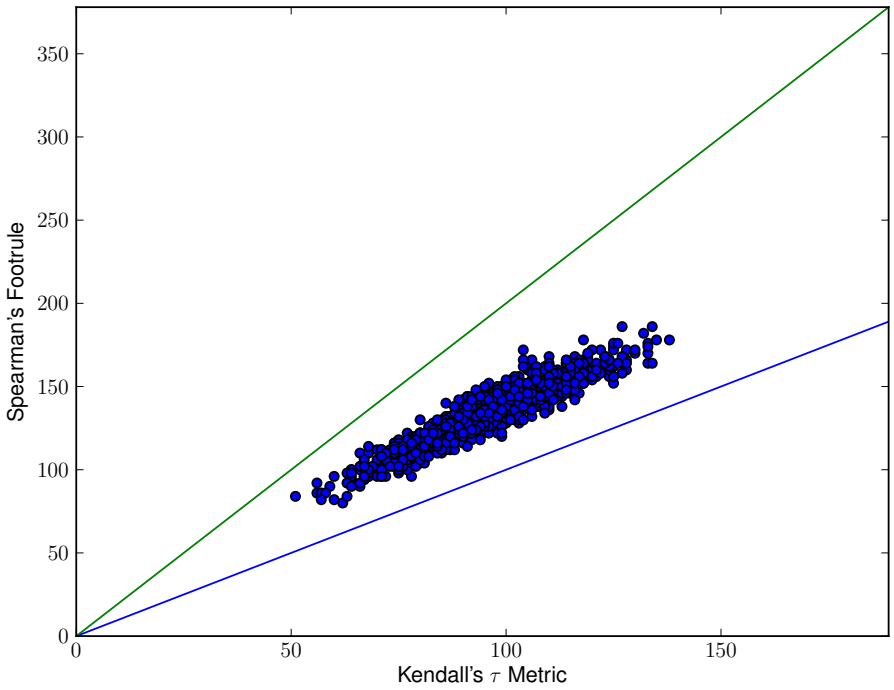


Figure 2.6.: Correlation between Spearman's footrule and Kendall's τ metric.

2.6. High-dimensional Visualisation

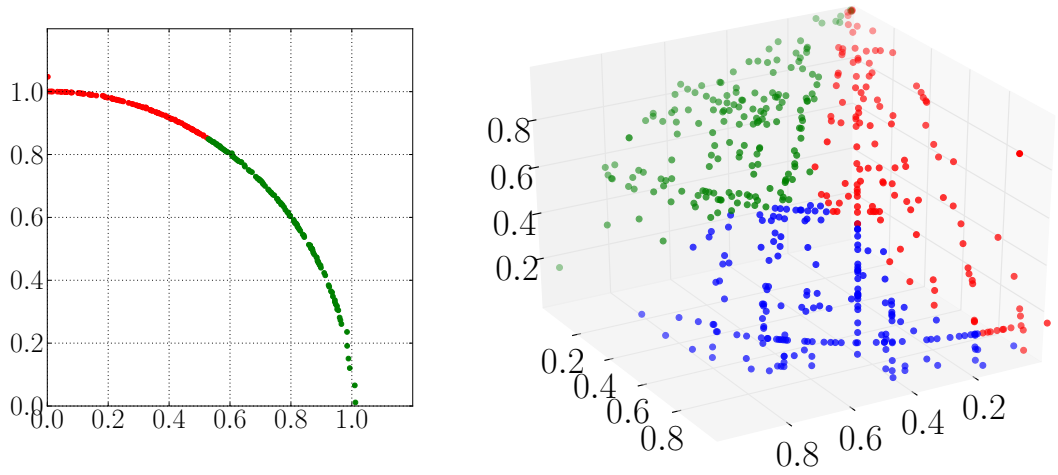
Much of the work in this thesis aims to enhance the understanding of a many-criterion population by a decision maker, and a way in which this is often done for data sets is to represent the data visually. Note, that from now on we use the term *many-criterion* population, rather than the term *many-objective* population. This is because the methods presented in the forthcoming chapters are illustrated on a range of populations including, but not limited to, solution sets to optimisation problems. For example, a running example that will be introduced in Chapter 3 reports on the performance of UK universities according to eight *key performance indicators* rather than solutions to an optimisation problem; a crucial distinction is that there is no parameter space component in that case.

Visualisation is a sensible approach to presenting multi-criterion data for a number of reasons. Lotov and Miettinen [2008] observe that a human decision maker will struggle to keep the alternatives in mind when presented with a long list of numerical values. Visualisation, as noted by Korhonen and Wallenius [2008], provides a better means of quickly obtaining a general view of the population. Human cognition is well suited to evaluating visual information; for example, spatial proximity of points in a visualisation intuitively provides information on the similarity of those points. Given a pair of multi-criterion individuals that are close together in a visualisation, a decision maker would likely interpret them as being "similar" in some way. Likewise, their visual proximity to known optimal or nadir points in the criterion space provides an indication of the

quality of those individuals; an individual that is placed close to an optimal point in criterion space is preceived to be of high quality. The identification of such optimal, or "goal", points in criterion space can also be made with the aid of a visualisation. Lotov et al. [2004] suggest the use of an interactive visualisation method, *Interactive Decision Maps*, to identify goal points by allowing the decision maker to experiment with different combinations of criterion values and find the most satisfactory trade-off possible. It is also possible to identify properties of the population, such as using ordered heatmaps, parallel coordinate plots and pairwise coordinate plots to identify correlations and conflict between criteria. These techniques are discussed in more detail later in this chapter, and this particular use of the heatmap approach forms part of the work presented in Chapter 5.

In order to be of use, the visualisation tools must be suitable for interpretation by a range of audiences. For example, one of the principal examples that this thesis draws on is that of university league tables, and one of the main purposes of such league tables is to enable potential students to select universities at which they may wish to study. In that scenario, the decision maker must be assumed to have little or no specific knowledge regarding higher education. Conversely, another of the examples used in this thesis relates to the construction of maintenance timetables for the components of a wireless mobile telephone network. In that scenario, the decision maker requires a visualisation that will enable them to identify the components which are most in need of maintenance, in order of increasing performance. Multi-criterion decision making is commonly used in concert with GIS systems [e.g., Rinner, 2003] for town planning problems, where, as with the maintenance example, a visualisation is required from which the best option can be selected from the population of alternatives. While these two examples require the decision maker to arrive at a more definite choice of individual than the university league table example, the maintenance engineer or town planner is likely to have more specialist knowledge that will enable them to manipulate the criteria and interact with the visualisation to make an informed decision. In the town planning scenario, some of this knowledge might be based on information collected from a public consultation exercise; multi-criterion visualisation combined with GIS can be useful here too. Jelokhani-Niaraki and Malczewski [2012] conducted a study in which they found that producing a simple interactive visualisation can increase participation in such exercises.

A significant body of work on the visualisation of high-dimensional data exists, some of which we review in this section before going on to describe how it has been used in a many-criterion context. Visualisation methods generally require some form of dimension reduction, and they can be broadly classified by whether information lost as a result of this reduction is recoverable or not. We review examples of both groups, beginning with those in which lost information is recoverable.



(a) 2-objective scatter plot

(b) 3-objective scatter plot

Figure 2.7.: Examples of multi-objective scatter plots for 2 and 3 objective instances of DTLZ2. Individuals are coloured by the objective on which they are best ranked (red = f_1, green = f_2 and blue = f_3). The objective vectors correspond to solutions generated by a $(\mu + \lambda)$–ES which ran for 250 generations.

2.6.1. Visualising All Criteria

It is most desirable to visualise a population with respect to the entire criterion set so that the maximum amount of information is available to the decision maker. Since it is generally not easy for a person to comprehend more than three spatial dimensions a considerable amount of research has been done to find methods which can visualise a many-criterion population that, again, comprises more than four criteria. We are principally concerned with methods drawn from multi- and many-objective optimisation.

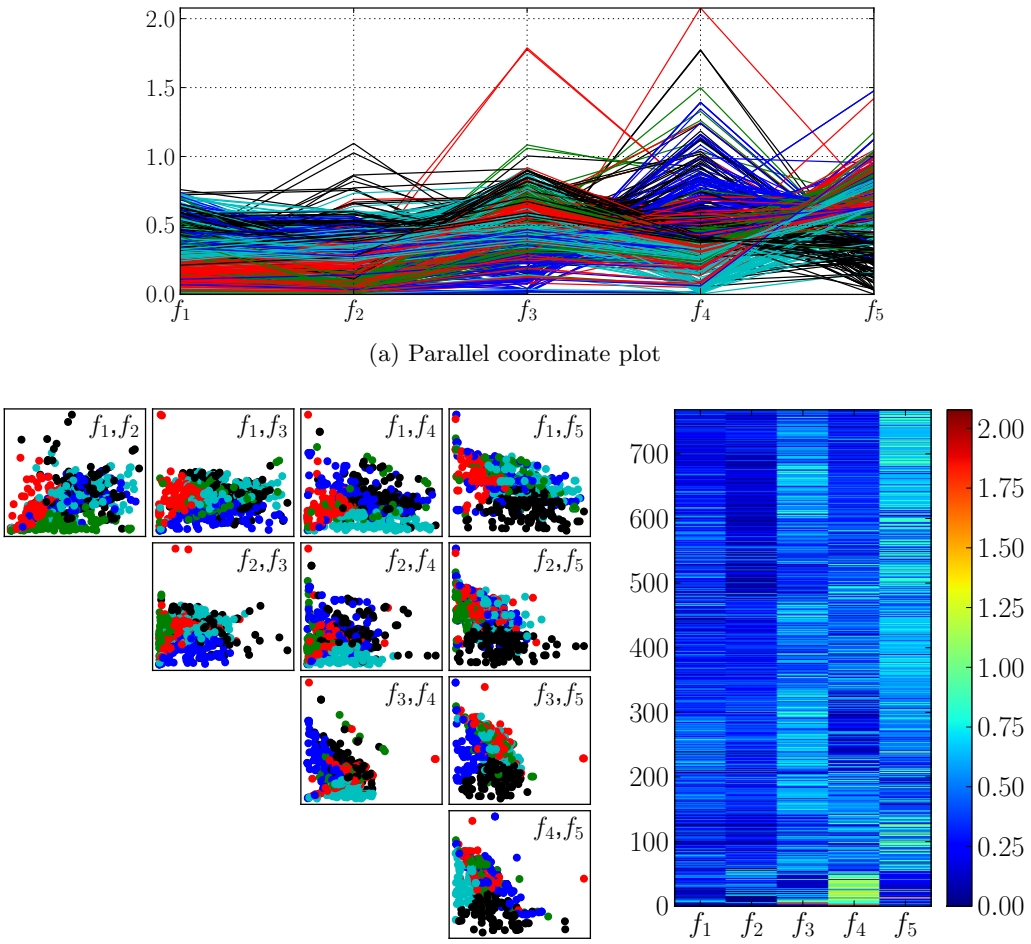
For illustration, Figures 2.7, 2.8 and 2.9 present a set of examples of existing visualisation techniques. The examples all show the objective space mapping of solutions $\hat{\mathcal{F}} = \{\mathbf{y}_i\}$ generated by running a $(\mu+\lambda)$–ES for 250 generations, maintaining an elite archive of non-dominated individuals throughout the process. At each generation, every member of the parent population was copied and an additive Gaussian mutation of standard deviation $\sigma = 0.2$ was applied to exactly one parameter in each child solution. Both μ and λ were set to 20. We used the aforementioned test problem DTLZ2 and generated results for $M = 2, 3, 5$ objective instances of the problem; the number of decision variables was $P = 10 + (M - 1)$ as recommended by Deb et al. [2002]. With the exception of Figure 2.7, all of the examples use 768 objective vectors from the 5-objective archive. By construction, the Pareto front is known to be the portion of a spherical shell of radius 1 lying in the positive orthant and the solutions in the set $\hat{\mathcal{F}}$ approximate this front. The solutions are quite well converged; for the 2-objective instance the median distance from the true Pareto front is 5.48×10^{-4}. For the 3-objective archive the median distance is 3.15×10^{-3}, and

for the 5-objective archive 5.38×10^{-2}. These distances were computed by finding the median distance between the archived solutions and the true Pareto front, which can be determined analytically for this test problem [Van Veldhuizen and Lamont, 1998].

Probably the most common method for visualising solutions in a multi-objective context is to produce a scatter plot on 2- or 3-dimensional axes, where each axis represents an objective. Examples are shown in Figure 2.7a (two objectives) and Figure 2.7b (three objectives) and clearly show the spherical nature of the estimated front. Solutions have been coloured according to the objective m on which the solution is performing "best" as follows. In order to avoid biases due to the differing scales on which the objectives were measured, the solutions were ranked on each objective separately; we denote the rank of solution $\mathbf{y}_i$ on the mth objective as r_{im} with 1 being the best rank and N the worst possible ($N = 768$ in this case). The solution $\mathbf{y}_i$ is then coloured according to the objective for which r_{im} is minimum. As Figure 2.7b shows, this colouring tends to colour neighbouring solutions in the same colour and provides some indication of the nature of a region of objective space. Although this provides relatively little additional information for 2 and 3 objectives, the same device considerably enhances the interpretability of many-objective visualisations.

Two of the earliest methods identified for use in many-objective optimisation were *parallel coordinate plots* [D'Ocagane, 1885; Inselberg, 1980; Fonseca and Fleming, 1993] and *pairwise coordinate plots* [Meisel, 1973; Cleveland, 1994]. A parallel coordinate plot presents each solution $\mathbf{y}_i$ as a line graph of y_{im} versus objective m with the values connected by lines. Whilst this is easily extended to any number of objectives, as Figure 2.8a shows for the 5-objective DTLZ2 solutions, the plots are often too cluttered to be of use. As in Figure 2.7 the solutions are coloured by the objective on which they have the best rank. Zhao et al. [2003] extended parallel coordinate plots to incorporate sequential data, data which changes over time, into the standard parallel coordinate plot with the addition of *trend figures* which are used to show how a particular variable or criterion changes over time. A pairwise coordinate plot compares each pair of objectives as a 2-dimensional scatter plot, as shown in Figure 2.8b. This is useful for revealing correlated and anti-correlated pairs of objectives and provides information on the pairwise interactions between objectives. However, the fact that the points representing a particular solution in each plot are not visually linked means that it is generally difficult to perceive relations between solutions. While it is mechanically easy to extend to any number of objectives, the number of plots $M(M-1)/2$ rapidly becomes overwhelming.

Heatmaps are frequently used to represent large multivariate datasets [Eisen et al., 1998; Grinstein et al., 2001; Wilkinson and Friendly, 2009] and have recently been used for multi-criterion populations [Pryke et al., 2006; Nazemi et al., 2008; Biswas et al., 2009; Hettenhausen et al., 2010; Kiesling et al., 2011]. In a heatmap, criteria are represented by columns, solutions by rows, and relative values as 'heat' represented by colour. Figure 2.8c



(a) Parallel coordinate plot

(b) Pairwise Coordinate Plot

(c) Heatmap

Figure 2.8.: Examples of the many-objective visualisation methods for the example 5-objective DTLZ2 front. With the exception of the heatmap visualisation, in which the colour of a cell indicates the 'temperature' of that value, individuals are coloured by the objective on which they are best ranked (red = f_1, green = f_2, red = f_3, cyan = f_4 and black = f_5).

presents a heatmap of the 5-objective DTLZ2 archive. The arbitrary ordering of solutions means that it can be difficult to observe relationships between the various solutions and objectives. Schemes for reordering the rows and columns in a heatmap to provide a clearer view of a many-criterion population, which we discuss in Chapter 5, have been proposed by Pryke et al. [2006] and Nazemi et al. [2008]. In addition, in order to be of use, the objectives must be on the same scale. One way in which this is done is by normalising values to similar ranges; for example, Nazemi et al. [2008] recommend linearly scaling the solutions for each objective to $[0, 1]$ before assigning colours. In Figure 2.8c the objectives are all on roughly the same scale, so no scaling was done before assigning colours; nonetheless, the heatmap is dominated by the cooler colours because there are just a few individuals with large criterion values. We present methods that use the full range of colours and place similar solutions together to enhance a visualisation of both objective space and parameter space in Chapter 5.

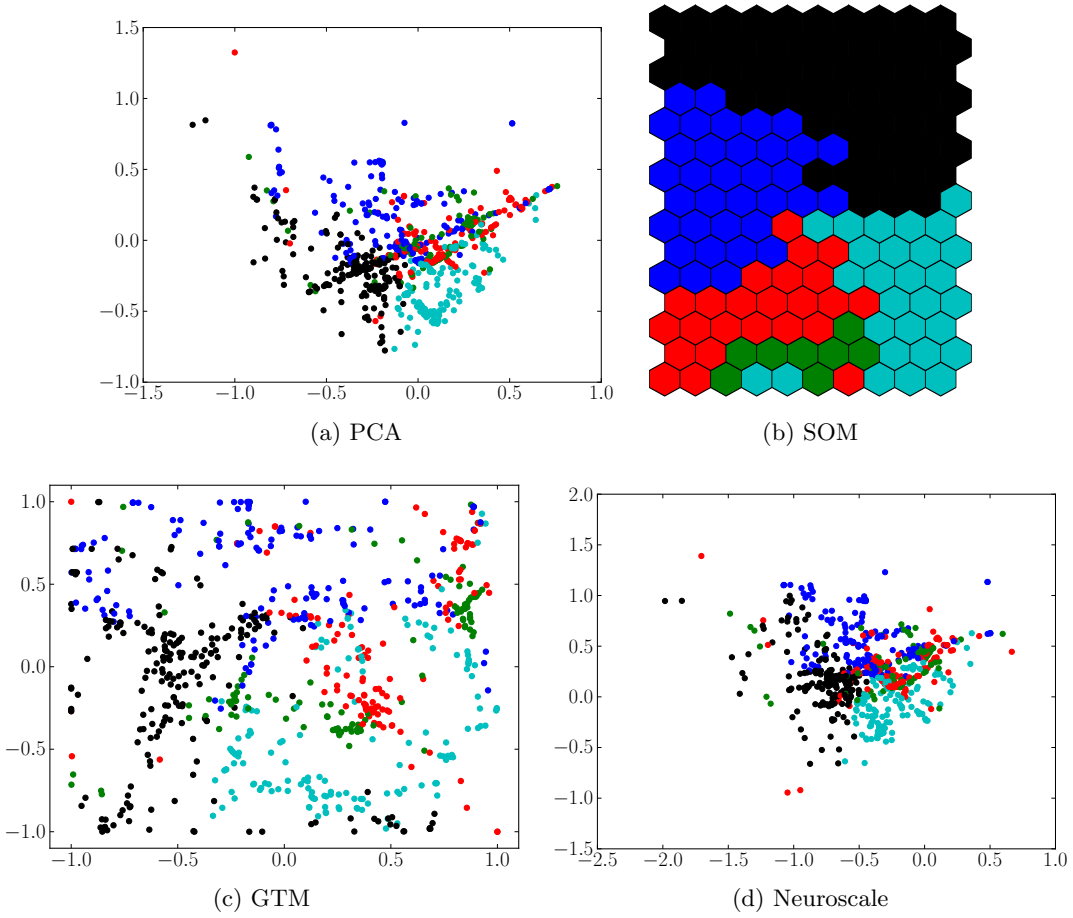


Figure 2.9.: Examples of the dimension reduction visualisation methods. Individuals are coloured by the objective on which they are best ranked (red = f_1, green = f_2, red = f_3, cyan = f_4 and black = f_5).

2.6.2. Visualising a Subset of the Criteria

Since scatter plots provide such an intuitive visualisation of multi-criterion individuals, an obvious course of action is to project the population into 2 or 3 dimensions and draw a scatter plot.

Probably the most common linear dimension reduction technique is principal component analysis (PCA, [Jolliffe, 2002]) which identifies the directions of objective space that capture the maximum amount of variance in the individuals. Figure 2.9a shows the PCA projection of the 5-objective DTLZ2 individuals into the two-dimensional space spanned by the first two principal components. The projection has identified the two directions in objective space that retain the most variance and are therefore the best *linear* approximations to the original population in a mean squared sense. However, when we colour the individuals by the objective on which they achieve the best rank as in Figure 2.7, we see that the clustering illustrated in this multi-objective visualisation is less clear. Although it

is still possible to see clusters of solutions that are best on the same objective, the clusters are beginning to overlap. This is a relatively small example, and the clusters will overlap more as the dimensionality of the population increases. Of course, some information must inevitably be lost by projecting into a lower dimensional space, but we note that PCA is ignorant of the mutually non-dominating nature of these individuals.

Three nonlinear methods that have been used for visualising many-criteria individuals are Self Organising Maps (SOMs) [Kohonen, 1995], Generative Topographic Mapping [Bishop et al., 1998] and Neuroscale [D. Lowe and M. E. Tipping, 1996]. All three of these methods aim to preserve nearest neighbour distances between individuals.

The SOM [Kohonen, 1995] is a topographically-arranged network of interacting transformation functions, whose response (displayed as a degree of *excitation* of all the component nodes) varies depending on the network input. In the basic formulation (as used in e.g. [Obayashi, 2002; Fieldsend and Everson, 2005]), the SOM defines a mapping from the input space onto a two-dimensional array of nodes. Each node in this array has an associated M-dimensional *reference vector* $\mathbf{w}$, and these nodes are compared to any input, $\mathbf{y}$, to the network, in a parallel fashion. The SOM seeks to find some best matching node to $\mathbf{y}$, denoted $\mathbf{w}_c$, whose response should be maximised given the input. Additionally, the SOM learning algorithm seeks to instill a local relationship between neighbouring nodes, such that nodes that are spatially close to one another in the network topology should also be concerned with adjacent regions of input space. One interpretation of this approach presented in [Kohonen, 1995] is to view the trained SOM as a nonlinear projection of the probability density function of the M-dimensional input into the two-dimensional display provided by the network.

Figure 2.9b illustrates the reference vectors associated with a SOM of the 5-objective DTLZ2 archive. Each hexagon represents one of the reference vectors in the trained mapping, and each vector has been coloured according to the objective of the reference vector which has the best value. These ranks were obtained by considering each reference vector in turn as a supplementary individual in the population which has already been ranked by each criterion. From the distribution of colours in the visualisation it is clear that the reference vectors have been distributed across the Pareto front and provides a coarse-grained spatially coherent representation.

The SOM lacks a generative model for the data being visualised; this means that it is difficult to incorporate new data to the visualisation. The Generative Topographic Mapping [Bishop et al., 1998] is a principled alternative to the SOM which represents the data as a nonlinear mapping to the high-dimensional data space of a topographically ordered low-dimensional latent space. The data is then visualised as its projection into the latent space. The nonlinear mapping is achieved by a constrained mixture of radial basis functions and a Gaussian noise model accounts for discrepancies between the noise-free mapping from

latent space to data space and the observed model. The likelihood corresponding to this generative model is then maximised using the expectation-maximisation algorithm [Dempster et al., 1977] in order to learn the model parameters and the latent visualisation. Figure 2.9c shows the visualisation of the 5-objective DTLZ2 archive by Generalised Topographic Mapping; objective vectors have been broadly clustered into similar groups, as indicated by the colouring, but there is a imperfect separation into distinct, topographically coherent regions. This is unsurprising, accounting for the information that has been lost in the compression process.

Neuroscale [D. Lowe and M. E. Tipping, 1996; Lowe and Tipping, 1996] has also been used for many-objective visualisation [Fieldsend and Everson, 2005; Everson and Fieldsend, 2006]. It also uses radial basis functions to form a nonlinear mapping projecting an M-dimensional individual $\mathbf{y}$ into a Q-dimensional individual $\hat{\mathbf{y}}$ using topographical information derived from considering the distances between solutions, where $Q < M$. The aim is for the embedding to preserve pairwise distances between the individuals. The radial basis functions are arranged as a neural network whose inputs are the high-dimensional individuals and the outputs are the corresponding low-dimensional individuals. The network is trained to minimise the Sammon stress [Nabney, 2004]:

$$\sum_{i}^{N}\sum_{j>i}^{N}\left(d_{ij}-\hat{d}_{ij}\right)^2, \tag{2.28}$$

in which d_{ij} is the distance between the objective vectors i and j in M-dimensional objective space and $\hat{d}_{ij}$ is the distance between the corresponding individuals in the Q-dimensional space. This metric is minimised when the distances between pairwise individuals in the original objective space and the embedded space are the same. Figure 2.9d demonstrates the application of Neuroscale to the visualisation of the 5-objective DTLZ2 archive. Individuals have been coloured by the objective on which they achieve the best rank, but like PCA, segregation into distinct regions is less clear than in the multi-objective examples of Figure 2.7. Closely related to Neuroscale is a method proposed by Valdés and Barton [2007] in which individuals are embedded in a 3-objective space by minimising the Sammon error:

$$\frac{1}{\sum_{i<j} d_{ij}}\frac{\sum_{i<j}(d_{ij}-\hat{d}_{ij})^2}{d_{ij}}. \tag{2.29}$$

This embedding is then presented to a decision maker as an interactive virtual environment that can be explored for knowledge discovery.

Clustering approaches have also been used, for example, in order to visualise the results of multi-objective nurse scheduling in which Fuzzy C-Means Clustering was used to cluster the solutions. The axes onto which the data were then projected were identified using Fuzzy Multiple Discriminant Analysis by finding the projection that maximises the ratio between within-class scatter and between-class scatter Yoshikawa et al. [2007].

The *prosection* method [Tušar and Filipič, 2011] visualises a population by compressing the objectives using prosections. A prosection is the projection of individuals within a section of the objective space into a low-dimensional space; in two dimensions, solutions are projected onto a line running through the section and intersecting the origin. The projections of solutions onto the line defining the section are then rotated through the angle between the projection line and one of the axes so that a dimension is removed. Reducing the dimensionality in this way has the advantage that if one solution dominates another then its prosection projection dominates the others' projection. However, two mutually non-dominating solutions may be projected so that one dominates the other. Additionally, it is currently only possible to visualise populations of four objectives or fewer

Another method [Koppen and Yoshida, 2007] which seeks to preserve dominance relationships first projects the non-dominated solutions onto the positive quarter of a circle centred on the origin. The individuals are distributed in a way that attempts to preserve nearest neighbour distances between the solutions. Interestingly, projecting the non-dominated solutions into a two-objective space such as this implies an ordering on the solutions; at one end of the embedded Pareto front the solutions best optimise one of the objectives while at the other end the solutions best optimise the other objective. Having reduced the dimensionality of the non-dominated solutions, a greedy procedure is used to find the position of each dominated solution that best preserves the dominance relationships of the original population. This procedure aims to minimise the number of *implicit* dominance relationships that are created in the two-dimensional embedding. An implicit dominance relationship occurs when two non-dominated solutions, $\mathbf{y}_i$ and $\mathbf{y}_j$ for example, are placed at two points on the circumference of the circle. If a solution $\mathbf{y}_k$ is dominated by $\mathbf{y}_i$ and $\mathbf{y}_j$ in the M-objective space then this should be shown in the visualisation by the placement of the projection of $\mathbf{y}_k$. Care must be taken to ensure that the placement of $\mathbf{y}_k$ in the two-dimensional space does not imply dominance relationships between non-dominated solutions that were not present in the original objective space.

All of the visualisation methods reviewed in this chapter tend to suffer from one of two problems. They are either lossless and present the entire set of objectives, which often results in a lack of clarity, or they incorporate a dimension reduction which loses information about the dominance relations between solutions. Subsequent chapters present methods aimed at addressing both of these issues. Methods for enhancing the clarity of lossless visualisation methods are presented in chapter 5, in which we investigate methods intended to improve the interpretability of heatmaps by reordering the rows and columns. In the next chapter, we introduce methods for compressing the dimensionality of a many-criterion population so that it can be visualised by conventional means, while seeking to minimise the loss of dominance information.

2.6.3. Interactive Visualisation

A common feature of the MCDM literature is the use of interactive visualisations to enhance decision support. Interactive visualisation is an important tool for making decisions based on multiple criteria because it provides an opportunity for a decision maker to experiment with different putative solutions and see the effect almost immediately.

Interactive visualisation has been used in a range of areas. As mentioned earlier, spatial MCDM problems, such as town planning problems like a site selection task, have a range of stakeholders. Providing an interactive visualisation allows such stakeholders, some of whom may not have specific domain knowledge about the entire problem, or may only be interested in the criteria that affect them (rather than the full set of criteria) to make their preferences known. These preferences can then be taken into account by the person with overall responsiblity for selecting a site. Examples in which interaction has been used to collect information on group preferences are Jelokhani-Niaraki and Malczewski [2012]; Rinner [2003].

One of the early interactive visualisation tools for populations of solutions to multi-objective populations, the *Pareto race* system, was presented by Korhonen and Wallenius [1988]. Specifically, they facilitate the exploration of the Pareto front of a linear programming problem by allowing the decision maker to "drive across" the feasible search space. Exploration is controlled by using the keyboard to alter the direction of travel and speed, and goals can be fixed or relaxed depending on the decision maker's preferences as the space is explored. The current solution is shown with a set of bars, one for each criterion. Korhonen and Yu [1997] illustrated a similar method for quadratic optimisation problems.

A nonlinear alternative to the Pareto race, the *Pareto Navigator*, was presented by Eskelinen et al. [2010]. The decision maker begins by selecting a starting point from a small number of generated Pareto optimal solutions and setting preferences. The preferences are used to specify a direction, and the navigator begins to generate more Pareto optimal solutions according to that direction. As the search proceeds, the decision maker has the opportunity to alter the direction of travel by modifying their preferences, and the search continues until the navigator has generated a suitable solution. Tarkkanen et al. [2009] observes that much could be gained by combining interactive visualisation in MCDM with the field of visual analytics, using the Pareto Navigator as an example of a multi-criterion visualisation.

Another well known example of an interactive visualisation is the *Interactive Decision Map* (IDM) [Lotov et al., 2004]. The IDM visualisation is capable of illustrating a trade-off between three criteria. Two of the criteria are shown on the coordinate axes, while the third is shown with areas of colour. A decision maker can also investigate a problem described by four or more criteria; any criteria not shown as part of the visualisation are

represented by scroll bars that a user can adjust to their desired operating point. By allowing a decision maker to investigate different combinations of criterion values in this way better enables them to make an informed decision by illustrating trade-off that is possible for a given problem. An example in which IDM was used to profile a multi-criterion population is Lotov [2002], where visualisation was used to support the choice of water quality improvement strategies according to five critera.

The importance of these interactive visualisation methods should not detract from the usefulness of static visualisation methods. The next chapter introduces a running example of comparing the performance of universities according to multiple criteria, and such data is commonly published in outlets such as newspapers (indeed, the example we employ is published annually in the Times newspaper). Such paper-based presentation will always be static. Some of the classical visualisation techinques we have described, such as parallel coordinate plots, benefit from interaction; by allowing the decision maker to explore different orderings of criteria, the inherent cluttered visualisation can be made clearer to a degree. The static techniques discussed above (e.g., scatter plots, dimension reduction methods and heatmaps) can generally be adapted for use in an interactive environment, although we note that such interactive versions of some methods can be less useful than others. Vetschera et al. [2010] conducted a comparative study in which the performance of interactive heatmaps and parallel coordinate plots was evaluated in a portfolio selection task. The parallel coordinate plots were observed to offer significantly better usability in the decision making process. Interactive methods, such as the Pareto race and Pareto navigator, will have limited usefulness in a static environment, since they require interaction with the decision maker in order to specify preference information.

2.7. Summary

This chapter has provided an introduction into the background material on which the subsequent work is based. We have given a basic introduction to evolutionary multi- and many-objective optimisation, as well as discussing methods for ranking many-criterion populations and comparing such orderings. We have also discussed the problems of visualising many-objective populations: novel methods for resolving the issues highlighted are presented in the next two chapters.

3. Understanding Many-criterion League Table Data

3.1. Introduction

Populations of individuals frequently arise in a variety of fields and, as described in the previous two chapters, it is often necessary to identify which individuals have the highest quality. One way in which this can be done is to construct a *league table* [Goldstein and Spiegelhalter, 1996; Usher and Savino, 2006] so that the individuals are arranged in an ordered list. A common example is ranking the sports teams in a league according to their match outcomes. A league consists of a set of teams, which can be considered a population of individuals (in which each team is an individual), each of which plays against the other teams in the league. It is then common to construct a league table by assigning points to a team based on the number of times they defeat their opponents. The aim is to maximise the number of points, which defines a criterion by which the individuals can be ranked. Such a criterion is often called a *key performance indicator* (KPI) [e.g., Franceschini et al., 2007]. A KPI describes an individual's performance with respect to a single measurable criterion. Constructing a league table from a single KPI is a simple matter of scoring all of the individuals by the KPI and then ranking them based on their score. Clearly, the individual with the best score is assigned rank 1; the individual with the second best score is assigned rank 2, and so on until the worst individual receives rank N (assuming no ties). League tables are also commonly used to compare the performance of schools and universities [e.g., Usher and Savino, 2006; O'Leary, 2009]. Additionally, they are widely used in industry, for example evaluating the performance of the components of a technical infrastructure, such as wireless access points in a mobile telephone network or district metered areas in a water distribution network [McClymont et al., 2011]. This enables the identification of poorly performing nodes and maintenance schedules are devised in which the nodes are given preferential attention.

It is often the case, however, that a population of individuals are described by more than one KPI. In such a population, KPIs can be in *conflict* with one another. For example, in the bi-criterion case, conflict arises where individuals with a good score on

Some of the material in this chapter has been published as Walker et al. [2010a,b]; McClymont et al. [2011]; Walker et al. [2012b,a].

one criterion have a poor score on another, while the individuals with a good score on the second criterion have a poor score on the first, and there is no individual with good performance on both. It is therefore necessary to choose how to combine information from all KPIs in order to arrive at an informative league table. A popular method for producing a league table from multiple KPIs is to aggregate them by taking a weighted sum [Benito and Romera, 2011]. Whilst aggregating criteria in this fashion is mechanically straightforward, selecting the weights for criteria that may well be on different scales is not. One KPI could be a proportion, such as the proportion of students at a school or university to achieve a good qualification, while another could be a monetary cost, such as the amount spent on each student in pounds. To provide a useful overview of the school or university it is necessary to combine these two KPIs in a meaningful way. This means that the KPIs must be rescaled so that they are on the same scale, and once this is done it is not clear how the weights should be applied. Identifying methods for which the need for *a priori* weighting of criteria can be avoided is an issue that this work addresses. League tables and their construction are described in the next section, along with a discussion of advantages and disadvantages of their use.

Fitness assignment methods within multi-objective evolutionary algorithms identify which of the current population of solutions are best suited to being parent solutions in the next generation. The process by which this is often done is inherently similar to the construction of a league table: the solutions can be ranked according to their objective values, which are criteria like the KPIs discussed previously, so that the best ones are selected [e.g., Deb, 2001]. Given the similarity between populations of solutions to optimisation problems and the more general multi-criterion populations discussed in this thesis, a promising line of investigation is to consider the mechanisms from fitness assignment methods in multi-objective evolutionary algorithms for evaluating the quality of individuals. Early multi-objective algorithms aggregated objectives to cast a multi-objective problem as a single-objective problem [e.g., Das and Dennis, 1997; Deb, 2001; Coello Coello et al., 2007], so that conventional fitness assignment methods could be applied, however they were later replaced by dominance-based methods that allow for the comparison of solutions with respect to individual objectives, without the need for weighting or aggregation [Deb, 2001; Coello Coello et al., 2007]. As discussed in Chapter 2, dominance is a fine candidate for comparing multi-objective solutions comprising two or three objectives, but it is less able to discriminate between solutions with four or more objectives (many-objective solutions). As such, a considerable amount of research [e.g., Bentley and Wakefield, 1998; Drechsler et al., 2001; di Pierro et al., 2007; Corne and Knowles, 2007; Li et al., 2010a; Garza-Fabre et al., 2010] is currently aimed at identifying other candidates for solution comparison in a search space formed by a large number of objectives, some of which are employed later in this chapter.

In this chapter we apply methods used to compare solutions in multi-objective evolutionary algorithms to the problem of constructing league tables in the more general context of

many-criterion populations. We investigate various ranking methods. Some of them have been used in the selection operators in multi- and many-objective evolutionary algorithms. One, the power index, has been used for ranking individuals according to a single criterion and we extend it to the multi-objective domain for the first time. Another problem addressed in this chapter is the presentation of a league table. A league table is usually presented as a table of numeric values or a ranked list, but when confronted with several criteria it is often more useful for a decision maker to be presented with a visualisation of the data which is capable of illustrating the members of the population with respect to their performance measures. The various multi-criterion ranking methods are used to visualise a many-criterion population as a graph, and we show that a more useful presentation of the population is achievable by using multiple ranking methods in concert with each other.

This chapter is structured as follows. Firstly, we discuss many-criterion populations in the context of league table construction and introduce the GUG09 population, which will be a running example throughout this thesis. Then, we discuss various multi- and many-criterion ranking methods and demonstrate their application to the ranking and visualisation of a range of many-criterion populations. We then introduce a novel metric, the dominance distance, for computing the distance between individuals in terms of dominance. We project individuals into a low-dimensional space, aiming to preserve their dominance distances, using a well known technique for dimension reduction, multi-dimensional scaling (MDS). We retain the notation for a many-criterion population introduced previously: $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$, $\mathbf{Y}$ is a population of N individuals $\mathbf{y}_i$, each of which is a M-dimensional vector of scores for the M criteria, and we assume that low scores correspond to better performance.

3.2. Measuring Quality with League Tables

League tables are widely used to discriminate between the quality of universities as measured by a set of performance indicators, converting the set of scores for a university into a single value which can be easily ranked [Goldstein and Spiegelhalter, 1996; Usher and Savino, 2006]. The purpose of their construction is to facilitate the understanding of universities based on a set of available data. The data used to construct league tables measure various aspects deemed pertinent to determining university quality, such as indicators of teaching and research quality. The collection of this data is one of several contentious issues surrounding the use of league tables. Whilst it is desirable for information to come from independent third parties, some data inevitably comes from the universities themselves. In some cases, institutions have objected to the very notion of such a ranking exercise and have chosen not to participate [Salmi and Saroyan, 2006; Usher and Savino, 2006]. They are important to a university for a range of purposes, such as goal setting, as well as being used by students in the selection of a university at which to study, governments for

assigning funding and employers in the selection of graduates [Salmi and Saroyan, 2006]. Various different types of league tables have been produced. These include national league tables, such as the Good University Guide published annually by the Times newspaper [O'Leary, 2009], international league tables, such as the Times Higher Education Supplement World Rankings [Baty, 2010], as well as those that measure courses rather than whole institutions [Usher and Savino, 2006].

Clearly, a league table is strongly dependent on the type of KPIs used to construct it, as well as the way in which the data is manipulated prior to construction. A common approach to constructing league tables is to combine the individual scores of a university with an aggregation method, often based on summing the individual indicators [Salmi and Saroyan, 2006; Usher and Savino, 2006; Benito and Romera, 2011]. Often, the indicators are on different scales, and indeed, measured in different units, so to aggregate them in a meaningful way they must first be normalised. Two ways in which this is done are to *rescale* or to *standardise* the data. Rescaling is done by adjusting the values based on the maximum value present for a given criterion, for example:

$$y'_{im} = \frac{y_{im}}{\max_j y_{jm}}, \tag{3.1}$$

while a method of standardising used in league table construction [e.g., Benito and Romera, 2011] computes *z-scores* based on the mean and standard deviation of a criterion across all universities, $\bar{y}_m$ and σ_m:

$$y'_{im} = \frac{y_{im} - \bar{y}_m}{\sigma_m}. \tag{3.2}$$

The scores can then be aggregated, commonly done by taking a weighted sum [Benito and Romera, 2011; Ding and Qiu, 2011]:

$$\hat{y}_i = \sum_{m=1}^{M} \gamma_m y'_{im} \tag{3.3}$$

where γ_m is the weight of the mth criterion.

A difficulty with such an approach is that it is difficult to know how to apply weights to the criteria so that the appropriate amount of weight is assigned to each normalised criterion. The selection of weights is either *subjective* or *objective* [Ding and Qiu, 2011]: subjective weight selection is done by canvassing expert opinion, whereas objective weight selection methods use techniques such as least squares regression to estimate appropriate weights to suit the data.

Despite these difficulties, and the reluctance in some quarters to accept league tables as a measure of institution quality [Salmi and Saroyan, 2006], league tables are becoming more widely used. As a result, techniques are being investigated for constructing more reliable league tables [Usher and Savino, 2006]. These include placing a university in a region of a population to provide an indication of its quality without assigning an exact rank that

would be subject to noise based on the data collection and manipulation methods involved. An extension of this is work by Benito and Romera [2011] in which the robustness of a institution's rank is assessed to provide a more statistically sound illustration of quality. Usher and Savino [2006] describe a league table in which the quality of German universities is conveyed by presenting all of the original criterion values. They note that, whilst this method of presentation is unsuitable for printed media, it is well suited for presentation as an interactive tool as part of a website. Later in this chapter we present methods which can be used to convey university quality without requiring the construction of a league table, but first we introduce the principal example dataset used in this chapter, and indeed throughout the thesis.

3.2.1. The Times Good University Guide 2009

The Good University Guide is an annual league table published by the Times, ranking UK universities. The Times Good University Guide 2009 (GUG09) [O'Leary, 2009] describes the performance of 113 UK universities in 2008. Each university entry is represented by eight KPIs. To produce the GUG09 league table, the values for a university are aggregated to produce the *GUG score*. First, the values are normalised by creating z-scores before they are aggregated with a weighted sum. Student satisfaction and research quality are given weight 1.5, while the remaining KPIs are assigned weight 1. Universities can then be ranked by their GUG score, which is to be maximised. This results in an ordering of universities which corresponds to popular experience; universities such as Oxford and Cambridge, which are accepted as being of a very high quality, are ranked highly by GUG score. The KPIs on which the GUG score is based are as follows:

1. **Student satisfaction** – quantified by the institution's score in the National Students Survey (NSS); this is an annual survey of graduating students. Some universities did not return a value for student satisfaction in 2009 and as a result some individuals have missing data. We present a novel method for imputing these values later in this chapter.

2. **Research quality** – based on the institution's results in the Research Assessment Exercise (RAE) in 2008. Each university is given a score out of 9 based on the amount of 4^* (world-leading), 3^* (internationally excellent), 2^* (internationally recognised) and 1^* (nationally recognised) research conducted at that university. This score is a weighted sum, where 4^* research is weighted highest and 1^* research receives the lowest weight.

3. **Student-staff ratio** – the average number of students to each member of staff with teaching responsibilities.

4. **Services and facilities spend** – average amount spent on facilities for staff and students, per student; expenditure is averaged over two years, measured in pounds.

5. **Entry standards** – the average UCAS score *achieved*, rather than the score required by the institution as part of a provisional offer, by new students under the age of 21.

6. **Completion** – the percentage of students projected to complete their studies.

7. **Good honours** – the percentage of students who achieve either a first class or upper second degree.

8. **Graduate prospects** – the percentage of former students employed in graduate level jobs six months after graduation.

With the exception of student-staff ratio, all KPIs are to be maximised. Throughout this thesis, for ease of interpretation, we arrange the KPIs so that they are all to be minimised. The original GUG09 data is shown in Appendix A.

3.3. Visualising and Ordering Many-criterion Populations

In order to visualise and order a many-criterion population $\mathbf{Y}$ it can be helpful to regard individuals $\mathbf{y}_i \in \mathbf{Y}$ as the nodes in a *weighted directed graph* [Berge, 1962; Knuth, 2011], with edges describing dominance relations between individuals. For example, the presence of a directed edge from $\mathbf{y}_i$ to $\mathbf{y}_j$ implies that $\mathbf{y}_i \prec \mathbf{y}_j$. Representing a population in this way is a useful method of revealing the structure. For example, by examining how the nodes are connected we can discover which are the important individuals in the population. We can discover which are the most dominant, which are the most dominated, and whether the graph is strongly connected (if this is not the case then that might indicate disconnected regions in the population, such as those shown on the true Pareto front of the test problem DTLZ6 in Figure 2.4(f)). In general, the graph $G = (\mathbf{Y}, E, \mathbf{W})$ is defined by the set of nodes $\mathbf{Y}$, the set of directed edges $e_{ij} \in E$ and a set of non-negative weights corresponding to the edges, which we write as an adjacency matrix $\mathbf{W}$; $W_{ij} > 0$ iff there exists an edge e_{ij} from $\mathbf{y}_i$ to $\mathbf{y}_j$, and $W_{ij} = 0$ otherwise.

A disadvantage with the dominance-based graph described above is that when a population comprises a large number of criteria the individuals will tend to be mutually non-dominating. Thus the nodes will not be connected and no structure will be revealed. As an alternative, suppose that an individual $\mathbf{y}_i$ is better than $\mathbf{y}_j$ on some proportion of

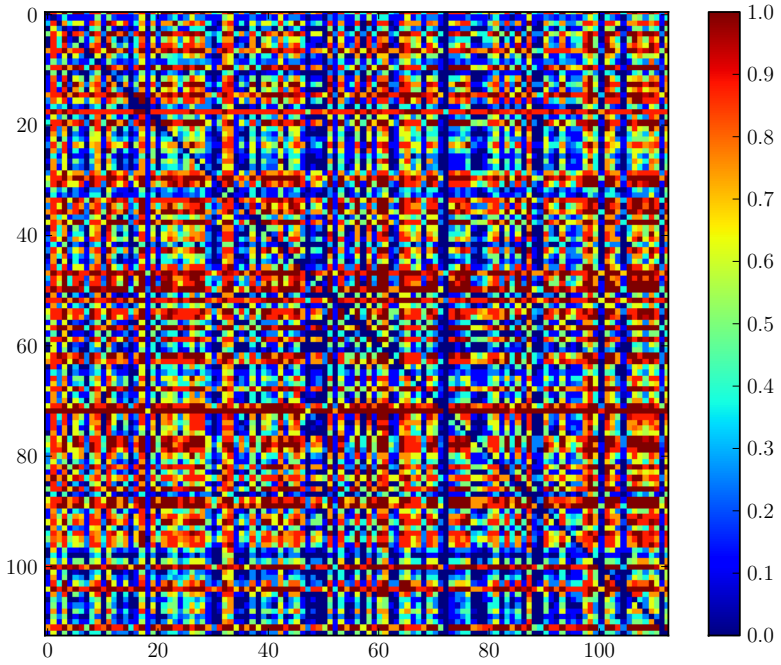


Figure 3.1.: The adjacency matrix $\mathbf{W}$ of the GUG09 population. Universities are in alphabetical order and the colour of W_{ij} indicates the proportion of criteria for which university i (x-axis) is better than university j (y-axis).

the M criteria. We use the proportion to weight the edges between them, W_{ij} and W_{ji}:

$$W_{ij} = \sum_{m=1}^{M} I(y_{im} < y_{jm}) + \frac{1}{2} \sum_{m=1}^{M} I(y_{im} = y_{jm}) \tag{3.4}$$

in which $I(p)$ is the indicator function, returning 1 if the proposition p is true and 0 otherwise. The second term accounts for the possibility of ties, which although unlikely to arise in an evolutionary optimisation scenario can occur in league table data. For example, two universities might have the same entry standards. This may be interpreted in terms of a tournament between the individuals, in which a criterion is selected at random and the dominating individual *on that criterion* wins; the weights are the probabilities that each individual will win such a tournament. Clearly, if $\mathbf{y}_i$ wholly dominates $\mathbf{y}_j$, that is $y_{im} < y_{jm}$ for all m, then $\mathbf{y}_i$ will win every tournament with probability 1. This generalises the idea of the *favour* relation proposed by Drechsler et al. [2001] in which the winning individual in a tournament between two individuals is the one which is better on more criteria. We define $W_{ii} = 0$ for all i, so that $W_{ij} + W_{ji} = 1$ for all $i \neq j$. Moon and Pullman [1970] call matrices with this property *generalised tournament matrices* (GTMs) and they have been used to rank single-criterion populations [Keener, 1993; Slutzki and Volij, 2005]. Figure 3.1 shows the adjacency matrix of the GUG09 population.

Whilst it is not necessary to weight the criteria for the calculation of probability of dominance, if importance weights are provided, as is the case for the GUG09 population, they

can be incorporated into Equation 3.4 as follows:

$$W_{ij} = \sum_{m=1}^{M} \gamma_m I(y_{im} < y_{jm}) + \frac{1}{2} \sum_{m=1}^{M} \gamma_m I(y_{im} = y_{jm}) \tag{3.5}$$

Although weights are available for the GUG09 population we treat all M criteria equally throughout this thesis, unless otherwise stated.

3.3.1. Leagues

Moon and Pullman [1970] show that a GTM may be partitioned into *leagues*;[1] any individual in one league will certainly beat (or dominate) any individual in an inferior league. A GTM may be permuted into a normal form so that blocks containing all ones lie in the top right hand corner. They define a *decomposable matrix* $\mathbf{W}$, one that can be permuted by a permutation $\mathbf{Q}$ so that

$$\mathbf{Q}^T\mathbf{W}\mathbf{Q} = \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \tag{3.6}$$

where $\mathbf{A}$ and $\mathbf{B}$ are square matrices. Then $\mathbf{W}$ is a GTM if there exists a permutation $\mathbf{Q}$ such that

$$\mathbf{Q}^T\mathbf{W}\mathbf{Q} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{1} & \dots & \mathbf{1} \\ \mathbf{0} & \mathbf{W}_2 & & \vdots \\ \vdots & & \ddots & \mathbf{1} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{W}_l \end{bmatrix} \tag{3.7}$$

where each $\mathbf{W}_i$ is a separate league, and every individual in $\mathbf{W}_i$ will beat every individual in $\mathbf{W}_j$, for $j > i$. As Moon and Pullman show, a GTM may be put into normal form by permuting the rows and columns so that the row sums $\sum_j W_{ij}$ are in decreasing order, after which the blocks can be read off. The GUG09 data comprises a single league; that is, no university (or group of universities) dominates every other university.

Partitioning the population into leagues does, however, not provide a total ordering of individuals. An individual in one league can be said to be better than one in a lower league but it does not differentiate between members of the same league. Indeed, it appears that it is very uncommon to find more than a single league in a multi- or many-criterion population. All of the examples in this thesis comprise a single league and we have not found any multi-criterion data comprising more than one league.

[1] Earlier in this chapter the term "league" was used to refer to a population of sports teams. From now on, the term refers to its technical meaning, as defined in this section.

Algorithm 2 Pareto sorting

Require: The population of individuals $\mathbf{Y}$.

1: $s \leftarrow 1$
2: **repeat**
3: $\quad \mathcal{P}_s \leftarrow \texttt{nondom}(\mathbf{Y})$ *Find the current non-dominated set; assign to shell s*
4: $\quad \mathbf{Y} \leftarrow \mathbf{Y} \setminus \mathcal{P}_s$ *Temporarily discard the individuals in the current non-dominated set*
5: $\quad s \leftarrow s + 1$
6: **until** $\mathbf{Y} = \emptyset$
7: **return** $\{\mathcal{P}_s\}$

3.3.2. Pareto Shells

A finer gradation of individuals within a league is provided by sorting individuals into *Pareto shells* of mutually non-dominating individuals, using the non-dominated sorting procedure that is used in some multi-objective evolutionary algorithms [Goldberg, 1989; Srinivas and Deb, 1994]. The common approach of using an evolutionary algorithm requires the incorporation of dominance into a selection operator, and Pareto sorting has been used in prevalent MOEAs such as NSGA [Srinivas and Deb, 1994] and its successor, NSGA-II [Deb et al., 2000], where the first step of the selection operator is to sort the search population into Pareto shells.

The notion of Pareto sorting is relatively simple, and results in each member of the population being assigned to a *Pareto shell*; the better the Pareto shell in which an individual resides, the better its quality as determined with dominance. The procedure by which individuals are assigned to shells is described in Algorithm 2. Those individuals in the population that are not dominated by any member of $\mathbf{Y}$ are assigned shell $\mathcal{P}_1$ (Line 3); for convenience of notation we define a function $\texttt{nondom}(\mathbf{Y})$ which returns the non-dominated individuals in the set $\mathbf{Y}$. The individuals in $\mathcal{P}_1$ are then removed from the population (Line 4) and a new subset of individuals in $\mathbf{Y}$ become non-dominated. These individuals are assigned to $\mathcal{P}_2$ and are themselves discarded. This procedure continues until the entire population has been assigned to a shell. Formally, we define $\mathcal{P}_1$, the set of Pareto optimal individuals, as:

$$\mathcal{P}_1 = \{\mathbf{y}_i \in \mathbf{Y} \mid \neg\exists \mathbf{y}_j \ (\mathbf{y}_j \in \mathbf{Y} \wedge \mathbf{y}_j \prec \mathbf{y}_i)\} \tag{3.8}$$

and the sth Pareto shell, where $s > 1$, is defined as:

$$\mathcal{P}_s = \{\mathbf{y}_i \in \mathbf{Y}'_s \mid \neg\exists \mathbf{y}_j \ (\mathbf{y}_j \in \mathbf{Y}'_s \wedge \mathbf{y}_j \prec \mathbf{y}_i)\} \tag{3.9}$$

where $\mathbf{Y}'_s = \mathbf{Y} \setminus \bigcup_{n=1}^{s-1} \mathcal{P}_n$.

Figure 3.2 illustrates the result of using Pareto sorting to order a small population comprised of two criteria. We interpret such an ordering as a crude league table of the individuals in the population.

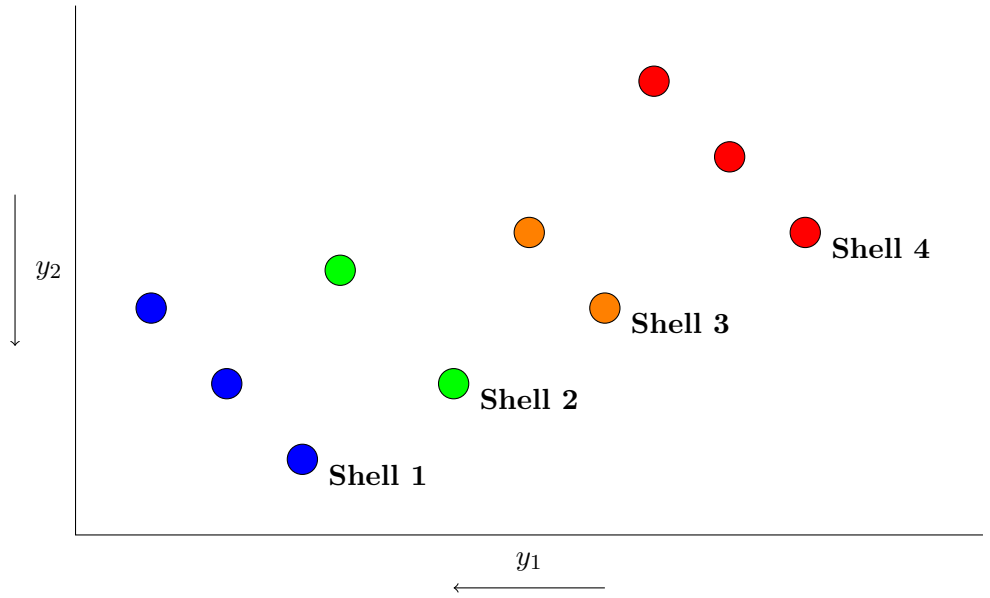


Figure 3.2.: A two-criterion population comprising 10 individuals which have been sorted into Pareto shells. The Pareto shell is shown by the colour of the individual: blue ($\mathcal{P}_1$); green ($\mathcal{P}_2$); orange ($\mathcal{P}_3$); and red ($\mathcal{P}_4$). Both criteria are to be minimised. Each shell $\mathcal{P}_n$ comprises the non-dominated individuals when shells $\mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ are discarded.

Constructing a league table in this fashion provides a means of imputing missing values in the data. To highlight a situation in which such imputation is usefully employed we return to the GUG09 population, in which some of the universities did not return a score for the student satisfaction KPI. If the value y_{im}, the mth KPI for the ith individual, is missing and the criteria are to be minimised the score can be imputed with the following procedure:

1. Assign a very poor value to y_{im}, for example $y_{im} \leftarrow \max_i y_{im}$.

2. Sort the population into Pareto shells.

3. If $\mathbf{y}_i \in \mathcal{P}_j$, then assign $y_{im} \leftarrow \max_{\mathbf{y}_k \in \mathcal{P}_j} y_{km}$.

This conservative imputation uses the worst value of any individual in the same Pareto shell as the incomplete individual for the missing criterion as a surrogate value. Thus, an individual with missing values cannot be promoted to a better shell than justified by the data which is present. We note that the presence of multiple missing criterion values in a single individual does not affect the process of imputation as described above, since under dominance criteria are independent.

Figure 3.3 shows the GUG09 population visualised with a technique based on a Pareto sorting of the universities it describes. The visualisation arranges the individuals in the

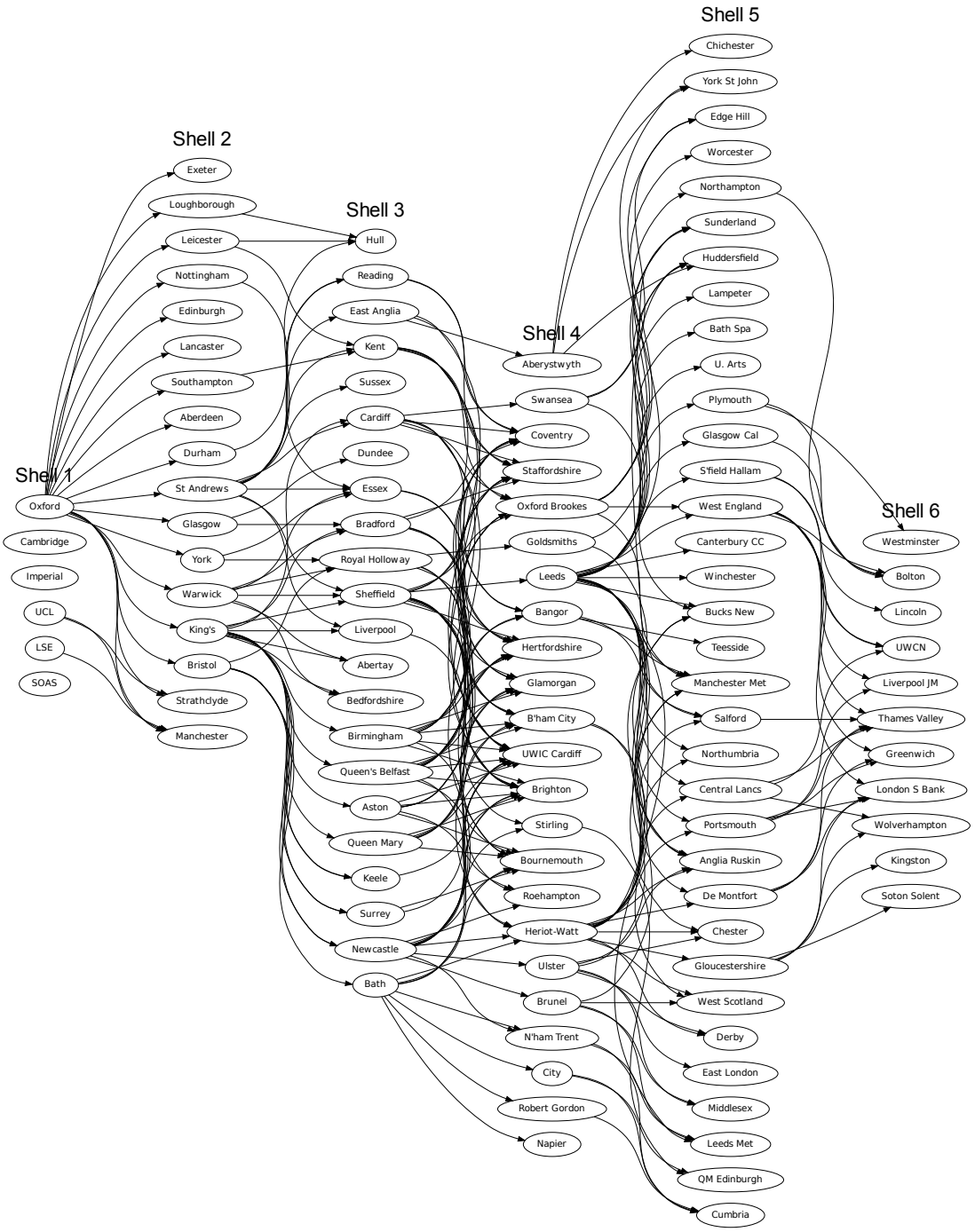


Figure 3.3.: A league table of the GUG09 data produced and visualised using Pareto sorting.

graph according the Pareto shell to which they belong, such that the progression of quality from those individuals that are Pareto optimal in $\mathcal{P}_1$ to those in the poorest Pareto shell is visible. The universities in $\mathcal{P}_1$ are those which are dominated by no other university according to the eight KPIs on which the GUG09 league table is based. The universities in $\mathcal{P}_2$ are dominated by at least one of the universities in $\mathcal{P}_1$, and so on for the remainder

of the population. Universities within a Pareto shell are mutually non-dominating and are therefore incomparable. In this case, the visualisation comprises six Pareto shells. The alignment of the shells from left to right is an aesthetic choice and could justifiably be different, for example aligning the shells in a top-down fashion. The vertical position of the node was determined by the layout algorithm employed by the GraphViz package used to construct the graph [GraphViz, 2012].

In order to preserve the clarity of the visualisation, only those edges between nodes in adjacent Pareto shells are shown; thus, an individual in $\mathcal{P}_1$ could dominate individuals in $\mathcal{P}_3$ but the relationship would be omitted. For example, Oxford is in $\mathcal{P}_1$ and edges are shown between it and the 17 universities it dominates in $\mathcal{P}_2$. However, it also dominates Liverpool in $\mathcal{P}_3$, Oxford Brookes in $\mathcal{P}_4$, Salford in $\mathcal{P}_5$ and Thames Valley in $\mathcal{P}_6$; none of these relationships are shown in the visualisation. Were the edges that represent these relationships to be included they would be overlaid and difficult to follow, which would lessen the information provided by the visualisation.

Visualising the GUG09 data in this fashion allows us to make interesting observations about its structure. We can, for example, determine which are the most dominant individuals in a Pareto shell by examining the number of individuals in the next shell which an individual dominates. Individuals such as Oxford in $\mathcal{P}_1$ and Leeds in $\mathcal{P}_4$ are clearly highly dominant and can be said to define the next Pareto shell because of the number of individuals in the next shell that they dominate. Many of the individuals, for example Exeter in $\mathcal{P}_2$ and Plymouth in $\mathcal{P}_5$, would be promoted to the immediately superior Pareto shell were they not dominated by Oxford and Leeds, respectively. Unfortunately, whilst it is possible to observe some information about the dominant individuals, if individuals within the same shell do not dominate individuals in the next shell they are indistinguishable. It is not possible, for example to infer whether Imperial or Cambridge in $\mathcal{P}_1$ is the superior individual based on this visualisation. Additionally, as mentioned above, it is not possible to take a more global view on who a university dominates because the edges representing domination beyond the immediately inferior Pareto shell are omitted.

As discussed previously, a problem with this proposed visualisation is that Pareto sorting is not capable of constructing a useful league table because dominance is unable to discriminate between individuals in a high-dimensional space. The proportion of space in which resident individuals are mutually non-dominating increases as follows [Farina and Amato, 2003; Fieldsend, 2003]:

$$1 - \frac{1}{2^{M-1}}. \tag{3.10}$$

Given this, the proportion of the space in which individuals are mutually non-dominating increases rapidly with M, and the individuals become generally incomparable. This is a problem which occurs when using multi-objective evolutionary algorithms to solve many-objective problems and we discuss methods that have been proposed to resolve this issue shortly.

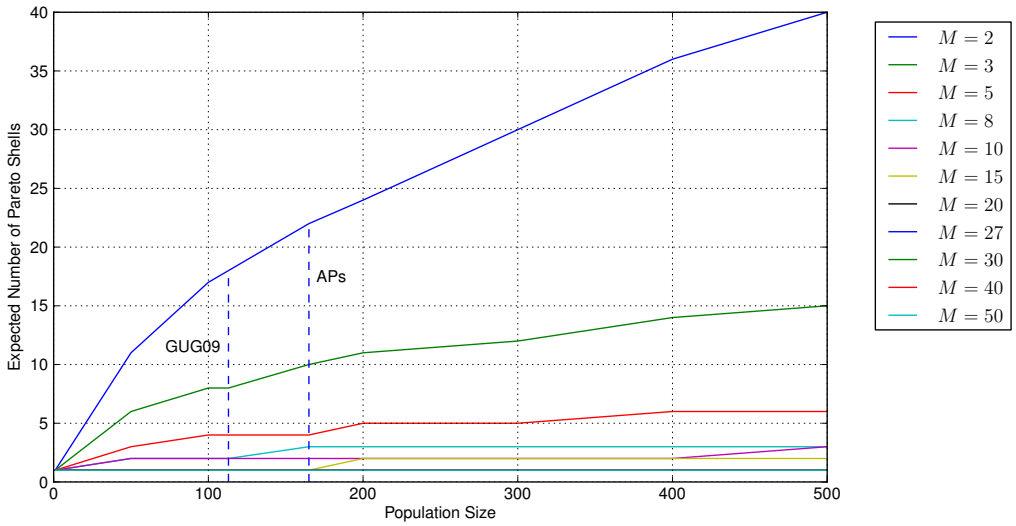


Figure 3.4.: Expected number of Pareto shells for uniform random populations of given individual and criterion size. The points in the graph which correspond to the population size of the GUG09 and Wireless Access Point (AP) populations used later in this chapter are marked.

Figure 3.4 illustrates the number of Pareto shells which can be expected for various population sizes of given numbers of criteria. The populations were produced by sampling uniform random values and sorting them into Pareto shells so that the number of shells could be recorded; the procedure was repeated for 1000 random populations and the figure shows the mean of these repetitions. As can be seen, populations with relatively small numbers of criteria are sorted into a large number of shells. Whilst the number of shells is shown to increase with population size, an 8-criterion population of 100 individuals does not sort into more than 3 shells, which will result in a league table in which it is difficult to discriminate between large numbers of individuals. Interestingly, this is approximately the population size and dimensionality of the GUG09 population which, as shown in Figure 3.3, comprises 6 shells. Pareto sorting thus reveals structure in the GUG09 population that is less prevalent in the 1000 uniform random populations investigated here. We conjecture that this is because of a relationship between the criteria in the GUG09 population that is not present in the uniform random populations, where all criteria are independent. In order to contextualise these results, we mark the GUG09 population in terms of the number of individuals. We also mark a population of 165 individuals representing wireless access points (APs) in a mobile telephone network which we introduce later in this chapter. Figure 3.5 shows a similar set of results for a similar experiment, this time using populations composed of Gaussian random noise instead of uniform random noise.

Having illustrated the difficulty associated with using dominance for comparison in many-criterion populations, we present two methods by which the structural information revealed by Pareto sorting can be increased. The first considers the amount of "rank credit" required to promote an individual to the next Pareto shell, while the second examines how much rank credit an individual can afford to lose without being demoted to the next

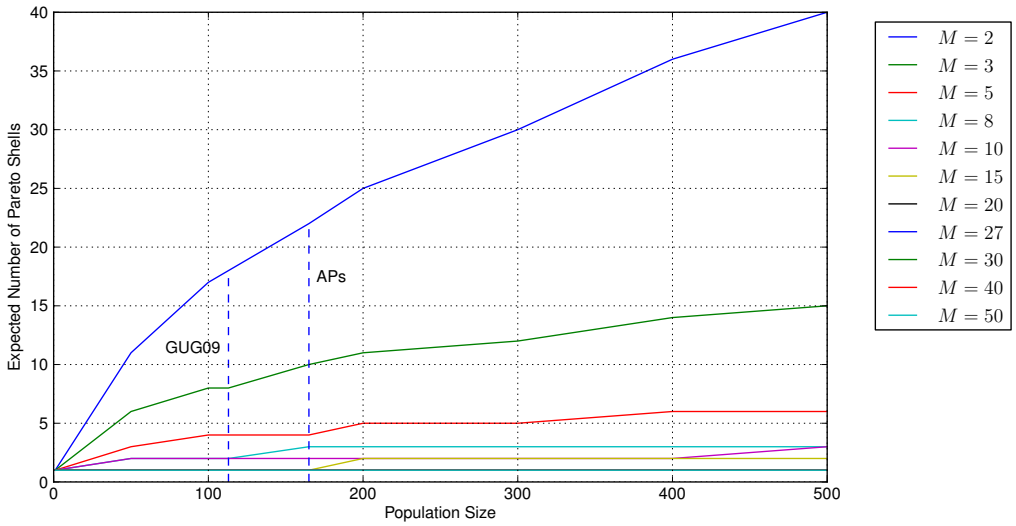


Figure 3.5.: Expected number of Pareto shells for Gaussian random populations of given individual and criterion size.

Pareto shell.

Rank Stability

As we have demonstrated above it is possible to reveal structural information about a population using Pareto sorting. However, since dominance is less discriminative in a high-dimensional space, Pareto sorting can be uninformative for a many-criterion population; the GUG09 population, for example, sorts into six Pareto shells, and we state that an individual in shell $\mathcal{P}_s$ is superior to an individual in shell $\mathcal{P}_t$, where $s < t$. It is not, however, possible using Pareto sorting alone to reveal information about the structure of the population within a Pareto shell. In this section, we investigate methods for using the robustness of an individual's position in a Pareto shell to reveal additional structure in the population based on the Pareto sorting.

One way in which a set of Pareto shells can be used to reveal additional structure within a population is to consider the stability of an individual's place in its assigned Pareto shell. To do this, we consider an approach whereby we evaluate how much of an improvement is necessary in order that an individual be "promoted" to the immediately superior shell. Converting the population to rank space, we imagine an allocation of extra "rank credit" that can be assigned to an individual. We consider that an individual requiring a small amount of additional rank to be promoted to the next shell is likely to be of higher quality than one requiring a large amount of credit. An alternative formulation, which we discuss later in this section, might consider loss of rank and evaluate individual quality in terms of how much rank can an individual lose before it becomes dominated by a member of its own rank and is therefore "demoted" to the immediately inferior shell.

To begin with, we calculate the amount of rank credit required to promote an individual. Consider the example of Strathclyde University, residing in shell 2 and dominated by two universities in shell 1 (Oxford and UCL) as shown by Figure 3.3. When converted to rank coordinates, Strathclyde, Oxford and UCL are described by the rank vectors for the eight criteria:

Strathclyde	32	51	51.5	32	29	64	16	14.5
Oxford	1	3	4	2	2	1	1	4
UCL	32	6.5	1	6	8	13	13	5.5

Clearly, Strathclyde is dominated by both Oxford and UCL and to be promoted to the next Pareto shell it requires enough additional rank credit so that it is mutually non-dominating with respect to both universities. By inspection we can see that Strathclyde and Oxford have the smallest rank difference on criterion 8; Oxford has a rank of 4 while Strathclyde has a rank of 14.5, so they are 10.5 "ranks apart". Thus, if we improve Strathclyde's rank on that criterion by 11.5 it will be superior Oxford and overall the two universities will be mutually non-dominating. Similarly, we can see that Strathclyde and UCL are tied on the first criterion, while UCL is superior on the other 7. An improvement of 1 rank position on the first criterion for Strathclyde will cause those universities to become mutually non-dominating too. The rank credit required to promote Strathclyde to the next shell is therefore the sum of these two differences, 12.5. Had both Oxford and UCL been closest on the same criterion, the larger of the two differences would have been sufficient to promote Strathclyde. We note that this formulation uses the notion of *strong* dominance, however it would be possible for other forms of dominance, such as *weak* dominance or ϵ-dominance to be used instead, however since we have formulated the Pareto shells in terms of strong dominance we do not discuss these alternatives further.

Algorithm 3 describes a procedure of determining the amount of credit required to promote each individual in the population to the next Pareto shell, except, of course, those in shell 1 who cannot be promoted beyond their current position in the best shell.[2] To outline the process, an upper limit of the amount of credit needed to promote an individual to the immediately superior shell is calculated. This upper limit is then refined to see if smaller amounts of credit can be distributed amongst the individual's criteria in such a way that it still promoted. The score of each individual is the smallest amount of rank credit that causes it to be promoted.

The procedure begins by converting the population to rank coordinates (Line 1). Having done this, each Pareto shell can be considered in turn, obviously omitting $\mathcal{P}_1$. For a given member of the current shell, $\mathbf{r}_i \in \mathcal{P}_s$, an upper limit of the credit needed to promote an individual is computed with the `upper_limit` function (Line 3). The upper limit is the

[2] An alternative strategy might consider how much credit is needed for a member of $\mathcal{P}_1$ to dominate the other members of $\mathcal{P}_1$, creating a new shell "$\mathcal{P}_0$".

Algorithm 3 Rank Promotion

Require: $\mathbf{Y}$ (the population of individuals); $\{\boldsymbol{\mathcal{P}}\}$ (the set of Pareto shells into which $\mathbf{Y}$ sorts); s (the current Pareto shell).

```
R := rank_coords(Y)                                   Convert to rank coordinates
for r_i in P_s do
    u_i := upper_limit(R, i, P_{s-1})                 Maximum credit needed to promote individual r_i
    c_i := u_i                                        Initialise current credit to upper limit
    for d := ⌊u_i⌋ − 1, ..., 1 do
        f := false
        for p_t in partitions(d, M) do
            if ¬∃j (r_j ∈ P_{s-1} ∧ r_j ≺ (r_i − p_t)) then
                c_i := d                              Update the credit with the current partition size
                f := true                             Make sure that the next value of d is searched
                break    The smallest credit has been found - move onto the next individual
            end if
        end for
        if f = false then
            break                                     No partition found for this d - stop the search
        end if
    end for
end for
return The credit c_i required to promote each individual r_i in P_s.
```

sum of the largest amounts by which $\mathbf{y}_i$ is dominated by any member of the superior $\mathcal{P}_{s-1}$ on each of the m criteria.

We begin the calculation of the upper limit u_i by first defining a matrix in which each column represents a criterion and each row represents a dominating individual in $\mathcal{P}_{s-1}$, $\boldsymbol{\Lambda}^i$, where all entries are 0. Then, for each dominating individual $\mathbf{y}_j$, we find the smallest difference between criteria for a criterion where $\mathbf{y}_j$ dominates $\mathbf{y}_i$:

$$n = \underset{m, r_{im} - r_{jm} > 0}{\operatorname{argmin}} |r_{im} - r_{jm}| \tag{3.11}$$

and set $\Lambda^i_{jn} = |r_{in} - r_{jn}|$. The upper limit is then determined by computing the sum of the maximum difference for each criterion, the maximum value in each column of $\boldsymbol{\Lambda}^i$:

$$u_i = \left\lceil \sum_m \max_j \left| \Lambda^i_{jm} \right| + 1 \right\rceil . \tag{3.12}$$

The remainder of the procedure then assigns rank credit to different criteria in order to find any better ways of distributing the rank so that the individual is promoted with the smallest possible additional rank, as it is possible that a smaller amount of credit than u_i can be used to promote an individual. Consider the following 3-criterion individuals:

	$m=1$	$m=2$	$m=3$
$\mathbf{r}_1$	8	8	8
$\mathbf{r}_2$	1	6	1
$\mathbf{r}_3$	3	2	1

If we want to promote $\mathbf{r}_1$ we must compute u_1. Using the approach outlined above, $u_1 = (2+1)+(5+1)$; the closest criterion value for $\mathbf{r}_2$ is criterion 2 (2 ranks difference) while the closest value for $\mathbf{r}_3$ is criterion 1 (5 ranks different). Hence $u_1 = 9$. By examining the individuals more closely we can see that having added rank credit of 3 to criterion 2, in order to dominate $\mathbf{r}_2$, we are now only 4 ranks from dominating $\mathbf{r}_3$ on that criterion. Hence, it is possible to promote $\mathbf{r}_1$ with rank credit of at most 8.

In order to search for cases where the upper limit is larger than necessary, we generate m-partitions, where a partition is a M-dimensional vector containing integer values that sum to d. In a partition $\mathbf{p}_t$ the value p_{tm} indicates the amount of rank credit that we assign to the m-th criterion. We examine various partitions to see if they promote an individual with less credit than is indicated by the upper limit. All partitions are generated for $d = \lfloor u_i \rfloor - 1, \ldots, 1$, where u_i is the upper limit needed to promote $\mathbf{y}_i$ and the process stops when no d-partition can promote $\mathbf{y}_i$; this is a computationally expensive task because of the large number of m-partitions for a vector of length M but, for the GUG09 population, was shown to be more efficient than starting with 1 credit and counting up to c_i. We define the function `partitions`(d, M) which returns all of the possible d-partitions of length M, and as each d-partition is generated (Line 7) a check is made to see if any member of $\mathcal{P}_{s-1}$ dominates $\mathbf{y}_i$ with the benefit of the current additional rank credit (Line 8). Note that $\mathbf{r}_i - \mathbf{p}_t$ represents the rank-coordinate version of $\mathbf{y}_i$ with the benefit of the rank credit provided by partition $\mathbf{p}_t$. If the individual is not promoted the next partition is generated. Otherwise, if it is possible to promote $\mathbf{y}_i$ with the current partition, then the current best rank credit c_i is updated accordingly with the current m (Line 9).

Table 3.1 illustrates promotion credit for individuals in two of the Pareto shells ($\mathcal{P}_2$ and $\mathcal{P}_3$) in the GUG09 population. The left-hand side of the table shows the credit needed to promote the individuals in shell $\mathcal{P}_2$ to $\mathcal{P}_1$ and the right-hand column provides a similar view of shell $\mathcal{P}_3$. This provides interesting information about the structure of these two shells. For example, we can see that the position of King's in $\mathcal{P}_2$ is quite strong, since it would require only a small amount of improvement to be promoted to $\mathcal{P}_1$ with the Pareto optimal universities. On the other hand, Glasgow would require a relatively large amount of credit to be promoted. Two other universities requiring large amounts of credit to be promoted to $\mathcal{P}_1$ are Manchester and Strathclyde. This makes intuitive sense since examination of the Pareto shell visualisation in Figure 3.2 informs us that these two universities are dominated by two $\mathcal{P}_1$ individuals (Oxford and LSE in the case of Manchester, Oxford and UCL for Strathclyde) as opposed to the rest of $\mathcal{P}_2$, which are dominated by Oxford alone. In addition to providing an indication of the structure within a shell we can also differentiate

Shell 2		
Rank	**University**	**Credit**
1	King's	2.0
3	Bristol	2.5
3	Loughborough	2.5
3	Leicester	2.5
5.5	St. Andrews	3.0
5.5	Warwick	3.0
7	Edinburgh	4.0
8	York	4.5
9.5	Durham	5.0
9.5	Exeter	5.0
11.5	Nottingham	6.0
11.5	Lancaster	6.0
13	Southampton	6.5
14	Aberdeen	8.5
15	Manchester	10.5
16	Strathclyde	11.5
17	Glasgow	12.0

Shell 3		
Rank	**University**	**Credit**
3.5	Bath	1.0
3.5	East Anglia	1.0
3.5	Birmingham	1.0
3.5	Queen's Belfast	1.0
3.5	Queen Mary	1.0
3.5	Dundee	1.0
7	Cardiff	3.5
9	Keele	4.0
9	Aston	4.0
9	Sussex	4.0
11	Sheffield	4.5
12	Bedfordshire	5.0
13.5	Abertay	6.0
13.5	Newcastle	6.0
15	Hull	7.0
16	Surrey	7.5
17	Kent	9.0
18.5	Royal Holloway	9.5
18.5	Liverpool	9.5
20	Reading	11.0
21	Essex	12.5
22	Bradford	16.5

Table 3.1.: Promotion credit for shells $\mathcal{P}_2$ and $\mathcal{P}_3$ of the GUG09 population.

between more of the universities. Although some ties remain it is now possible to infer a more complete ordering of the universities by considering their promotion credit.

An alternative to calculating the amount of extra rank that an individual needs to be promoted to the next shell is to calculate the amount of rank that they would need to loose in order to be *demoted* to the immediately inferior Pareto shell. Calculating *demotion credit* is mechanically simpler than calculating promotion credit, and computationally much faster. As described above, the promotion credit calculation requires the comparison of each individual in shell $\mathcal{P}_s$ with each individual in $\mathcal{P}_{s-1}$ to see which individuals dominate it and by how much. To demote an individual, all that is necessary is for the individual to become dominated by a member of its own shell. This simply involves finding the difference between an individual and its peers in the shell it resides in, on the criteria that the peers dominate the individual on. More formally, the demotion credit is the smallest distance between any of the M criterion values of an individual $\mathbf{y}_i$ and the corresponding criterion values of the other members of the same shell:

$$\min_{\mathbf{y}_j \in \mathcal{P}_s} \min_{m} |y_{im} - y_{jm}|. \tag{3.13}$$

Consider the following example. The following are the rank coordinate version of Oxford, Cambridge and UCL[3]:

Oxford	1	3	4	2	2	1	1	4
Cambridge	88.5	1	6	3	1	2	2	2
UCL	32	6.5	1	6	8	13	13	5.5

Let us assume that we wish to find the amount of rank that Oxford would have to lose in order to be demoted to shell $\mathcal{P}_2$ (ignoring, for the moment, the other members of $\mathcal{P}_1$). For this to happen, Oxford would have to be dominated by at least one of the other two individuals. Therefore, we simply find the criteria on which Cambridge and UCL are dominated by Oxford (criteria 1, 3, 4, 6 and 7 in the case of Cambridge, and all criteria with the exception of criterion 3 for UCL). We then find the sum of differences between Oxford and the other two individuals on dominating criteria, and the credit required to demote Oxford is the smallest of these, with 1 added[4]. In the example here, the difference between Oxford and Cambridge on dominating criteria is 92.5, while the difference between Oxford and UCL on dominating criteria is 70. Hence, the demotion credit is based on the difference between Oxford and UCL, and is 71. Any less credit would mean that Oxford was still mutually non-dominating with respect to the other two universities.

Table 3.2 illustrates the demotion credit for shells $\mathcal{P}_2$ and $\mathcal{P}_3$ in the GUG09 population. As with the promotion results shown in Table 3.1, ranking the individuals within a shell refines the ordering induced by Pareto sorting. Interestingly, while some individuals occupy similar positions as they did within the promotion rankings (i.e., Bath at the top of $\mathcal{P}_3$ and Strathclyde at the bottom of $\mathcal{P}_2$), some individuals are placed far from their promotion-based rankings. For example, Newcastle in $\mathcal{P}_3$ is in the bottom half of the promotion ranking but has the second highest rank in the demotion ordering. This is unsurprising, since the promotion scores are calculated with respect to the individuals in the immediately superior shell while the demotion scores are concerned with the other residents of the current shell. Both measures provide interesting information about the stability of an individual; an ideal individual is one that is easy to promote and difficult to demote, such as King's which requires little credit for promotion and is difficult to demote. Conversely, Strathclyde requires little credit to be demoted and is difficult to promote. Such information was not available by considering the Pareto sorting alone.

This section has demonstrated the use of Pareto sorting to explore the many-criterion population on which the GUG09 league table is based. We have shown that visualising the population in its Pareto shells provides useful information about the structure of the population and illustrated techniques for evaluating the stability of an individual's position

[3]Note that the Cambridge's rank for the first criterion is based on an imputed value.

[4]The additional 1 rank is to ensure that adjusting an individual's ranks does not result in two equivalent individuals.

Shell 2		
Rank	**University**	**Credit**
1	Warwick	46.0
2	St Andrews	42.0
3	King's	39.0
4	Durham	23.0
5	Bristol	19.0
6	Leicester	18.0
7	York	16.5
8	Southampton	16.0
9	Aberdeen	14.0
10	Lancaster	11.0
11	Glasgow	8.0
12.5	Manchester	7.5
12.5	Edinburgh	7.5
14.5	Loughborough	6.5
14.5	Nottingham	6.5
16	Exeter	5.0
17	Strathclyde	3.0

Shell 3		
Rank	**University**	**Credit**
1.0	Bath	66.5
2.0	Newcastle	38.5
3.0	Sheffield	32.5
4.0	East Anglia	32.0
5.0	Birmingham	30.0
6.0	Queen's Belfast	29.0
7.0	Reading	27.5
8.0	Queen Mary	26.5
9.0	Sussex	26.0
10.0	Aston	24.0
11.0	Royal Holloway	23.5
12.0	Dundee	14.5
13.0	Kent	13.5
14.0	Cardiff	13.0
15.0	Liverpool	11.0
16.0	Keele	7.5
17.0	Surrey	7.0
18.0	Hull	6.5
19.0	Bradford	4.5
20.0	Essex	3.5
21.0	Bedfordshire	3.0
22.0	Abertay	2.0

Table 3.2.: Demotion credit for shells $\mathcal{P}_2$ and $\mathcal{P}_3$ of the GUG09 population.

in its Pareto shell. In the coming sections we consider alternative ranking methods to Pareto sorting for exploring the structure of many-criterion populations.

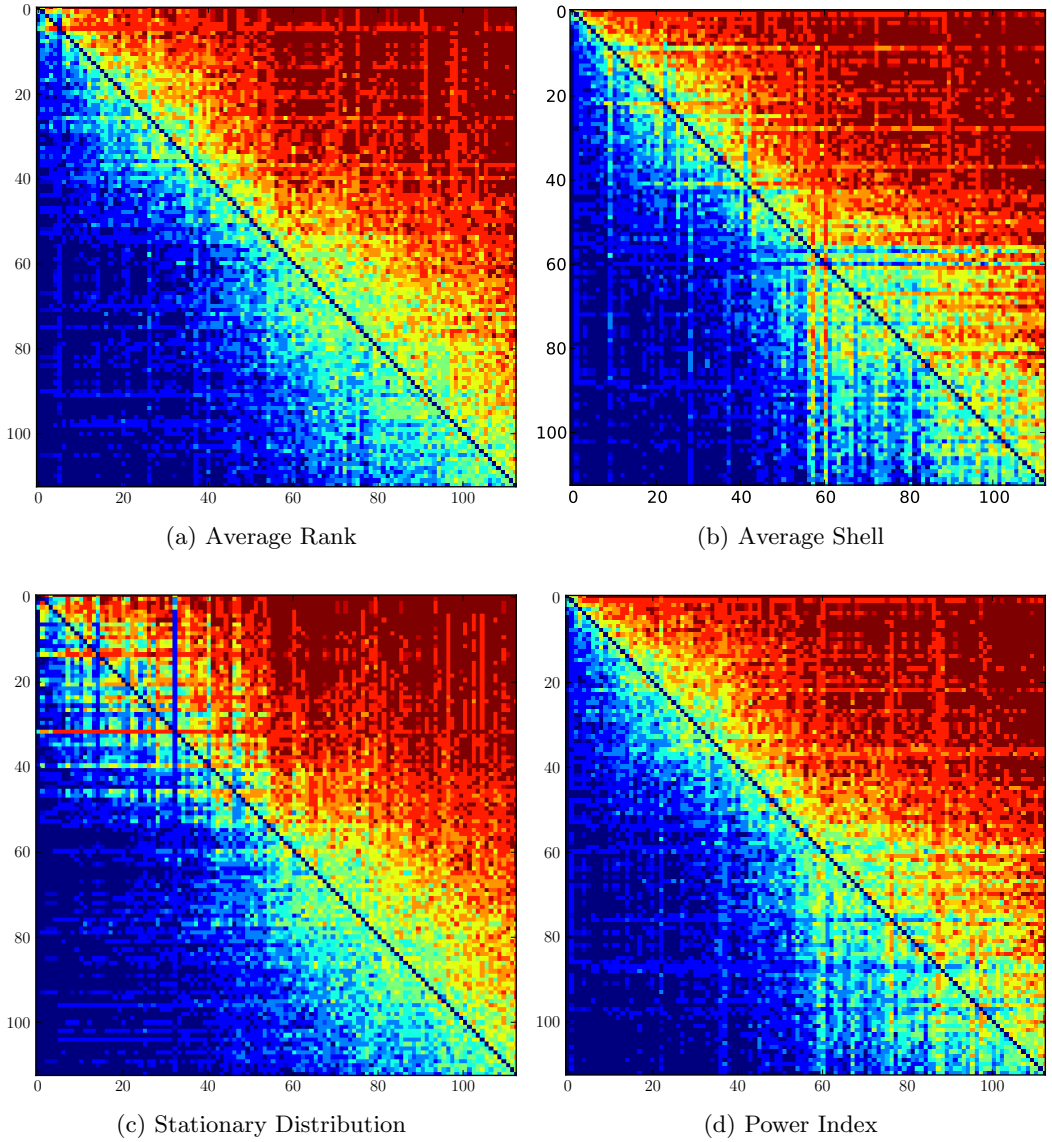
3.3.3. Average Rank

As stated in Chapter 2, the particular scales of the separate criteria are immaterial in determining whether one individual is better or worse than another on a given criterion, so it is natural to rank individuals based on each criterion. Average rank, introduced in Chapter 2, is a method that ranks a population in such a way.

Consider the graph G^m, which describes the mth criterion with an associated adjacency matrix $\mathbf{W}^m$, a GTM with 0, 0.5 and 1 entries only (where $W^m_{ij} = 1$ if and only if $y_{im} < y_{jm}$; if $y_{im} = y_{jm}$ then $W^m_{ij} = W^m_{ji} = 0.5, i \neq j$). Then we compute a *rank vector* $\boldsymbol{\rho}_m$ for each criterion m in which the rank r_{im} of an individual i on criterion m is:

$$r_{im} = N - 1 - \sum_j W^m_{ij}. \tag{3.14}$$

High performing individuals are assigned numerically low scores, with 0 the best score,



(a) Average Rank

(b) Average Shell

(c) Stationary Distribution

(d) Power Index

Figure 3.6.: The adjacency matrix shown in Figure 3.1 re-ordered according to various ranking methods. Recall, that red indicates a high probability that individual i (y-axis) dominates individual j (x-axis)

while the worst individual has a score of $N - 1$ because it is bettered by all others (with the exception of itself).

Ranks for separate criteria can then be averaged in order to produce an overall rank for each individual. The average rank for individual i over all M criteria is:

$$\bar{r}_i = \frac{1}{M} \sum_{m=1}^{M} \gamma_m r_{im} \tag{3.15}$$

where γ_m is a weight indicating the importance of the mth criterion. Whilst it is possible to incorporate weights into the average rank, in this work we treat all criteria equally.

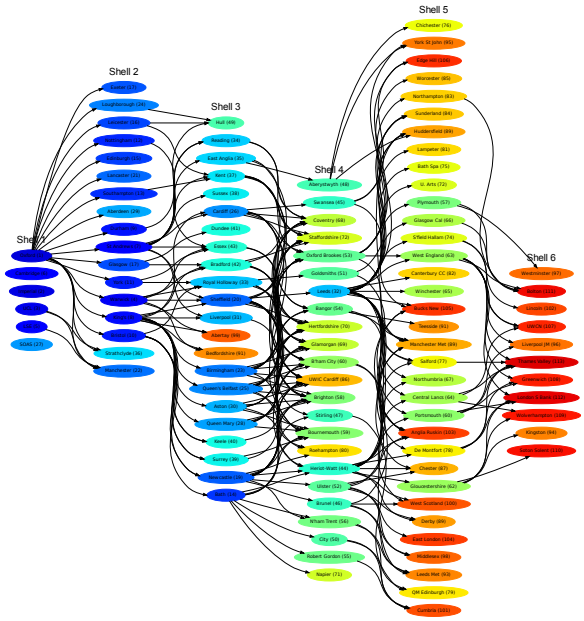


Figure 3.7.: Pareto shell visualisation of the GUG09 population coloured by average rank. Blue indicates a good rank, while red indicates a poor rank.

Figure 3.6(a) presents the GUG09 adjacency matrix with the universities reordered by their average rank, while Figure 3.7 shows the Pareto shell visualisation of the individuals in which the nodes have been coloured by average rank. A university coloured in blue has a good average rank while a red university has a poor average rank. By observing Figure 3.7 it is clear that the average rank agrees with the partial ordering resulting from Pareto sorting in this case.

3.3.4. Graphical Population Ranking

As we are modelling the individuals $\mathbf{y}_i \in \mathbf{Y}$ as a weighted directed graph, we can avail ourselves of methods designed for ranking the nodes in a graph.

One method for constructing such an ordering of nodes in a graph based on adjacency information is the *outflow* method introduced by van den Brink and Gilles [2009]. Outflow is the sum of the weights leaving a node; in terms of the adjacency matrix, the outflow score for the ith individual is defined by:

$$\sigma_i^{\text{out}} = \sum_j W_{ij}. \tag{3.16}$$

Individual i is ranked at least as highly as node j if and only if $\sigma_i^{\text{out}} \geq \sigma_j^{\text{out}}$. It is also noted that, outflow generalises the out-degree and ranking by the Copeland score [Henriet, 1985; van den Brink and Gilles, 2009] which assigns a rank to a player in a tournament based on how many times it beats other players. They also provide axioms which characterise

outflow ranking:

Anonymity: the labelling of a node does not affect its rank.

Positive responsiveness: the performance of an individual is increased if the weights by which it dominates another individual increase.

Outflow monotonicity: the pairwise ordering between individuals does not change unless their outflows change.

Order preservation: the addition of two weighted graphs does not change the order of two individuals where their order is the same in the two original graphs.

An interesting observation is that when a population comprises multiple criteria, each with its own adjacency matrix $\mathbf{W}^m$, the outflow rank is equivalent to the average rank. This is because the overall adjacency matrix is the average of the individual matrices $\mathbf{W}^m$:

$$\mathbf{W} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{W}^m. \tag{3.17}$$

If we denote a vector of ones as $\mathbf{1}$, the vector of outflows for the population may be written as:

$$\boldsymbol{\sigma}^{\text{out}} = \mathbf{W}\mathbf{1} \tag{3.18}$$

$$= \frac{1}{M} \sum_{m=1}^{M} \mathbf{W}^m \mathbf{1} \tag{3.19}$$

$$= N - \mathbf{1} - \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\rho}_m \tag{3.20}$$

where $\boldsymbol{\rho}_m$ is the vector of ranks for the mth objective. It is noted by Corne and Knowles [2007] that these two techniques are also equivalent to another ranking method from multi-objective optimisation, the *winning score* [Maneeratana et al., 2006].

The outflow, and by equality average rank, has many attractive features: it is easy to compute, easy to understand, and, as described previously, is the same method used by Moon and Pullman [1970] to partition a graph into leagues. Nonetheless it only uses information about the weights leaving a node and, as van den Brink and Gilles [2009] mention, this is analagous to ranking countries by their total exports, rather than, for example, the trade balances which would be the result of ranking by the difference between outflow and inflow.

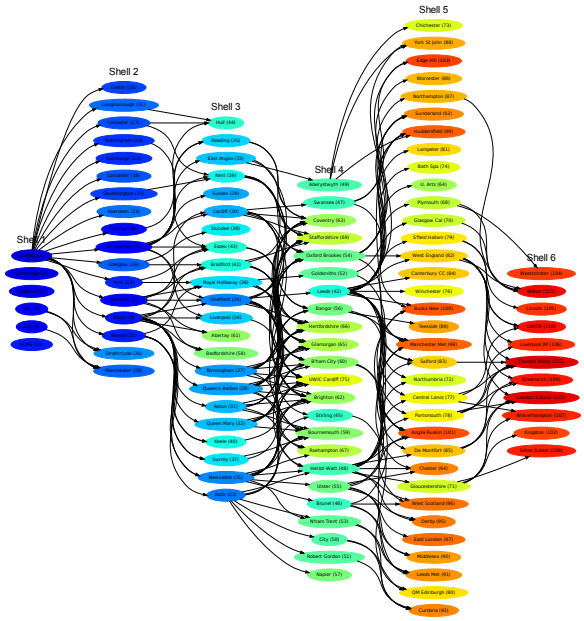


Figure 3.8.: Pareto shell visualisation of the GUG09 population coloured by average shell; blue indicates a highly ranked individual, while blue indicates a poorly ranked individual

3.3.5. Average Shell

The preference order [di Pierro et al., 2007] ranks individuals by how robust they are as criteria are removed, but as noted above tends not to be very discriminative because only M levels are available. A related idea is to track which Pareto shell an individual resides in as criteria are removed. Since with M criteria there are $\binom{M}{k}$ combinations of k taken from the original M, we average over the ranks of these combinations and denote the result by $\bar{S}_k(\mathbf{y})$. Note that $\bar{S}_M(\mathbf{y})$ is just the Pareto shell of the individual using all criteria whereas, providing that criterion values are all distinct, $\bar{S}_1(\mathbf{y})$ is the average rank of $\mathbf{y}$. An individual may then be assessed by the average shell, weighted by the dimension of the reduced criterion space:

$$\bar{S}(\mathbf{y}) = \frac{2}{M(M+1)} \sum_{k=1}^{M} k\bar{S}_k(\mathbf{y}). \tag{3.21}$$

3.6(b) presents the adjacency matrix re-ordered by average shell and Figure 3.8 shows a Pareto shell visualisation for the GUG09 population in which the nodes in the graph are coloured by average shell. By colouring the individuals in the graph we can see that those individuals with the best average shell score are in the Pareto optimal shell. That said, the results are quite similar to those of average rank. Due to its simplicity we prefer to use average rank.

3.3.6. Stationary Distribution

It is straightforward to define a random walk on the graph describing the many-criterion population. Consider a random walker who at each time step jumps from the current node to one of the node's neighbours, a node with which the current node shares an edge, with the probability of transition from the node i to node j of $p_{ij} = W_{ij}/\sum_k W_{ik}$. The transition probabilities are therefore summarised in the matrix

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W} \tag{3.22}$$

where $\mathbf{D}$ is the diagonal matrix of row sums of $\mathbf{W}$ which may be recognised as $\boldsymbol{\sigma}^{\text{out}}$, the vector of outflows (Eq. 3.16) such that $D_{ii} = \sigma_i^{\text{out}}$. If $\mathbf{p}_{t-1}$ is the (row) vector describing the probability of finding the walker at each node at time $t-1$, then the probability distribution at t is $\mathbf{p}_t = \mathbf{p}_{t-1}\mathbf{P}$. A graph is strongly connected when, for every two nodes $\mathbf{y}_i$ and $\mathbf{y}_j$ there is a sequence of directed edges connecting $\mathbf{y}_i$ to $\mathbf{y}_j$. This is true for nondecomposable GTMs, as described by Moon and Pullman [1970] (i.e., GTMs representing a single league). The Perron-Frobenius theorem [e.g., Horn and Johnson, 1990] implies that the transition probability matrix of a strongly connected graph has an eigenvalue 1 and unique left eigenvector $\boldsymbol{\pi}$ with $\pi_i > 0$ for all i which defines the stationary probability distribution of a random walk on G; that is: $\boldsymbol{\pi} = \mathbf{P}\boldsymbol{\pi}$. The stationary probability of finding a walker at a node has been used in ranking applications, notably in the PageRank algorithm [Page et al., 1998] in which frequently visited web pages are ranked highly. For ordering many-criterion populations, directed edges point away from powerful individuals so that powerful individuals are expected to be those with a low stationary probability.

The GUG09 adjacency matrix, re-ordered according to the stationary distribution, and population, in which the nodes are coloured by the stationary distribution, are shown in Figures 3.6(c) and Figure 3.9, respectively. However, although the broad ordering is as expected, there are some prominent anomalies. Oxford is ranked at 1, but King's is ranked at 2 although it is in $\mathcal{P}_2$, because it dominates many universities so that a random walker arriving at King's will easily be transported away from it. Conversely, Cambridge in $\mathcal{P}_1$ is ranked at 33. This is due to it not returning a NSS score so that the NSS value used is the worst value for $\mathcal{P}_1$. Cambridge is thus on the 'edge' of $\mathcal{P}_1$ and does not dominate many others, which means that a probability mass arriving at Cambridge is not easily transported away from it, leading to a relatively high probability in the stationary distribution and consequent poor rank.

3.3.7. Power Index

Earlier in this chapter we showed the equivalence of the average rank and outflow methods. Recall, that the outflow rank is determined by summing the ranks of the dominated

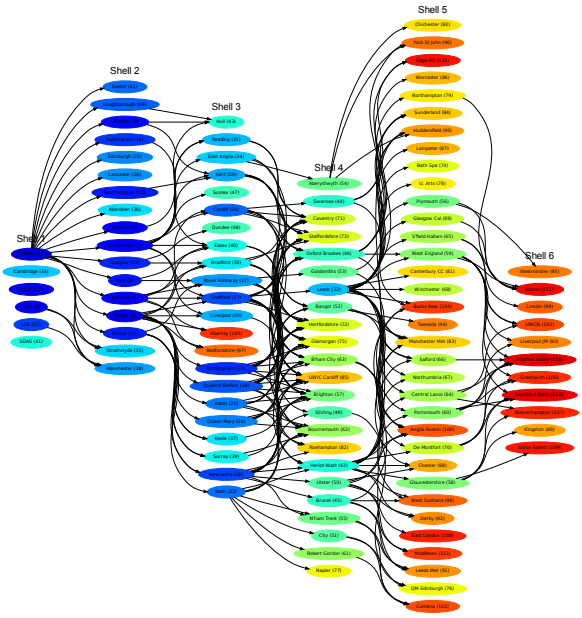


Figure 3.9.: Pareto shell visualisation of the GUG09 population coloured by stationary distribution; blue indicates a highly ranked individual, while blue indicates a poorly ranked individual. Abertay and Bedfordshire are coloured red and orange respectively yet are in a relatively high Pareto shell (shell 3); each has one criterion on which they achieve a good score, making it difficult for other individuals to dominate them and placing them in a superior Pareto shell than would be indicated by considering all of their criterion values (the majority of which are poor). Ranking by stationary distribution does, however, consider all of the criterion values and the individuals are consequently assigned a low rank.

individuals for a given member of the population. It is reasonable to consider that if the dominated teams have themselves dominated strong individuals then that should be reflected in their dominator's rank based on the intuition that they might be more difficult to "beat" using the sports team analogy discussed earlier. The *long path method* can be traced back to Wei [1952] and Kendall [1955] and was called the *power index* by Berge [1962]. It is a method that ranks the individuals in a population by incorporating a notion of the quality of individuals that it dominates by considering the individuals that they in turn dominate, and so on. The power index has been used to rank sports teams [Keener, 1993] based on a GTM describing the outcome of matches between teams. It is defined in terms of the right eigenvector of $\mathbf{W}$ which can be reached by considering the sequence:

$$\mathbf{u}^t = \mathbf{W}\mathbf{u}^{t-1} \tag{3.23}$$

for $t = 1, 2, \ldots$, starting with $\mathbf{u}^0 = \mathbf{1}$. This means that $\mathbf{u}^1$ is the vector of average ranks or outflows, $\boldsymbol{\sigma}^{\text{out}}$; $\mathbf{u}^2$ is the vector that assigns to $\mathbf{y}_i$ the scores of all other individuals in proportion to the weight W_{ij}, that is $u_{ij}^2 = \sum_j W_{ij} u_j^1$. The limit of this procedure is

$$\mathbf{u} = \lim_{t\to\infty} \frac{\mathbf{u}^t}{\sum_i u_i^t}. \tag{3.24}$$

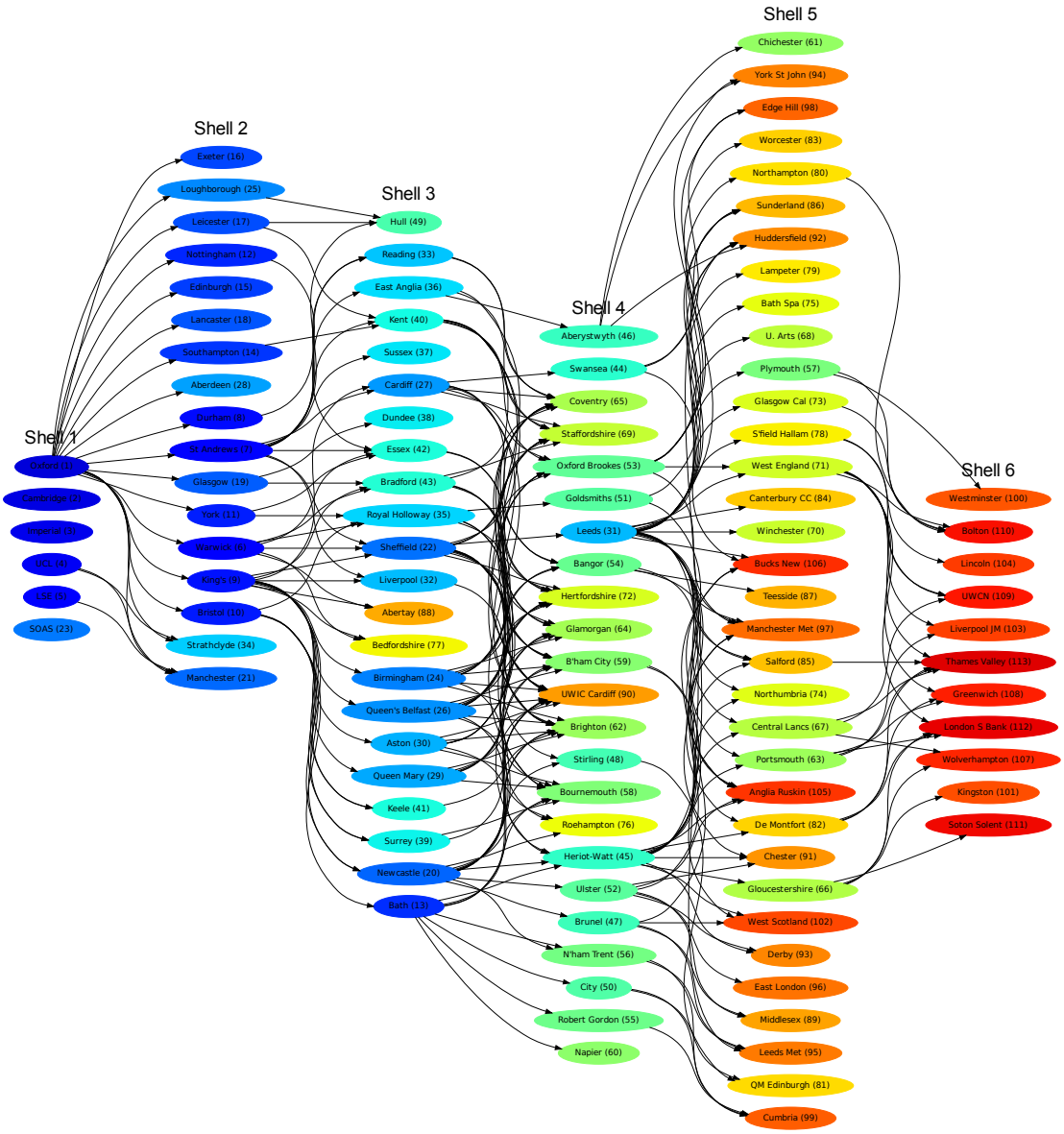


Figure 3.10.: A league table of the GUG09 data produced and visualised using Pareto sorting; universities are coloured by the power index such that a blue university is more powerful than a red one.

The limit exists when G is strongly connected, meaning that every node has a relationship with every other node, and is the eigenvector of $\mathbf{W}$ corresponding to the largest eigenvalue; that is $\mathbf{u}$ solves

$$\mathbf{W}\mathbf{u} = \lambda\mathbf{u}. \tag{3.25}$$

The Perron-Frobenius [Horn and Johnson, 1990] theorem assures that $\mathbf{W}$ has a unique positive eigenvector. The population is then ordered by the values of $\mathbf{u}$; if u_i is the nth largest element of $\mathbf{u}$, then $\mathbf{y}_i$ is assigned rank n.

Figure 3.6(d) shows the adjacency matrix for the GUG09 population ordered by the power

index and Figure 3.10 shows a second visualisation of the population, first seen in Figure 3.3. The adjacency matrix shown in Figure 3.6(d) is a reordering of the unweighted GTM, and whilst the power index does not rely on weightings, weighted criteria can be incorporated into the computation of an individual's power if they are available by using the weighted probability of dominance (Equation 3.5). Turning to the graph, the individuals have this time been coloured by their power index, such that blue indicates the most powerful and red the least powerful. Additionally, the power index rank of each university is shown in its node. Note, that the power index ordering does not necessarily respect the ordering imposed by Pareto sorting, in that a university in shell j, $\mathbf{y} \in \mathcal{P}_j$, may be ranked better than $\mathbf{y}' \in \mathcal{P}_{j-1}$ and vice versa. We observe that this can also occur with the other ranking methods considered so far in this chapter. Clear examples of this shown in Figure 3.10 are SOAS in $\mathcal{P}_1$ which is ranked at 23 by the power index, together with Bedfordshire and Abertay, both in $\mathcal{P}_3$, but ranked at 77 and 88 respectively. As the figure shows, this is because they do not dominate any universities in $\mathcal{P}_4$. We may infer that SOAS, Bedfordshire and Abertay are situated on the periphery of their shells because they do not dominate others. Conversely, those universities that dominate many others are ranked better, for example, Oxford (1), King's (9), Sheffield (22) and Leeds (31) all dominate most of the individuals in the following shell.

Clearly the power index provides a good deal of additional information on the structure of a many-criterion population. We note however, that like average rank, it does not differentiate between members of a bi-criterion non-dominated set because in that case $W_{ij} = 1/2 \;\; \forall i \neq j$ and the power index of each member is $1/\sqrt{N}$. Calculating the entire collection of eigenvalues and eigenvectors for $\mathbf{W}$ can be expensive for a large population. Fortunately, the power index is the eigenvector corresponding to the principal eigenvalue; this can be quickly calculated using the iterative *power method* [e.g., Golub and Van Loan, 1996; Hochbaum, 2006]:

$$\mathbf{u}^{t+1} = \frac{\mathbf{W}\mathbf{u}^t}{\|\mathbf{W}\mathbf{u}^t\|}, \tag{3.26}$$

where $\mathbf{u}^0$ is again an N-dimensional vector of ones. Experimentation showed that N iterations were sufficient to obtain an equivalent ordering of individuals to that achieved by calculating the full collection of eigenvectors.

Figure 3.11 shows the degree of correlation between the average rank (and outflow) and the power index; as can be seen, they are very closely correlated. This is to be expected, since the power index is a refinement of average rank in which, as the iterative process described in Equation 3.23 is followed ties between individual's overall ranks are resolved.

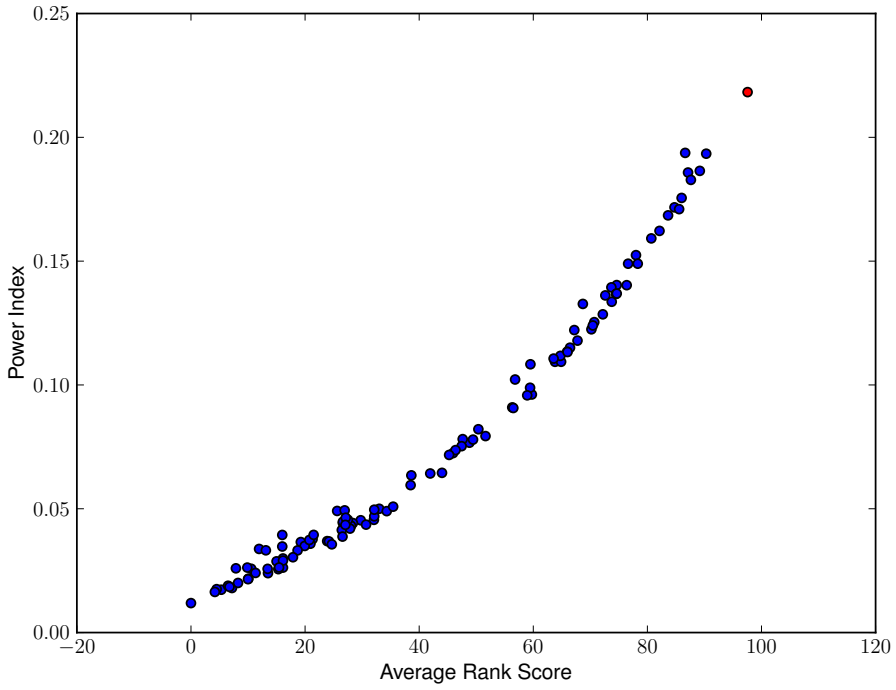


Figure 3.11.: Power index score versus average rank/outflow score for the GUG09 population. High values indicate best performance in both cases.

Illustrations

Figures 3.12 and 3.13 illustrate further applications of the combined Pareto shell power index visualisation method to visualising many-criterion populations. The first, shown in Figure 3.12, is an updated version of the GUG09 population that we have been using throughout this chapter. This version, GUG11, reports on 116 universities on the same criteria as were described earlier in the chapter for GUG09 but for data collected in 2011. An interesting difference between the two is that some universities for which scores were imputed in GUG09 have proper values in GUG11. A notable example of such a university is Cambridge, for whom a student satisfaction score was available in 2011; Cambridge now participates in the National Student Survey. As such, power index provides a more realistic impression of the university, placing it second only to Oxford in the resulting ranking. An example of an individual with an imputed score in this data is the University of Buckingham in $\mathcal{P}_1$. This is a teaching-only institution, and as such does not have a score for research quality. It has an extremely high student satisfaction score, which the other members of $\mathcal{P}_1$ do not dominate.

Figure 3.13 is a series of visualisations of the search population of an evolutionary algorithm as it optimises a multi-objective problem. The problem is a 3-objective instance of DTLZ2 [Deb et al., 2002] which we optimise using a basic multi-objective $(\mu + \lambda)$–Evolution Strategy (ES) where both μ and λ are set to 50. The algorithm was run for 100 generations, although only three of the first 10 generations are shown in the visualisa-

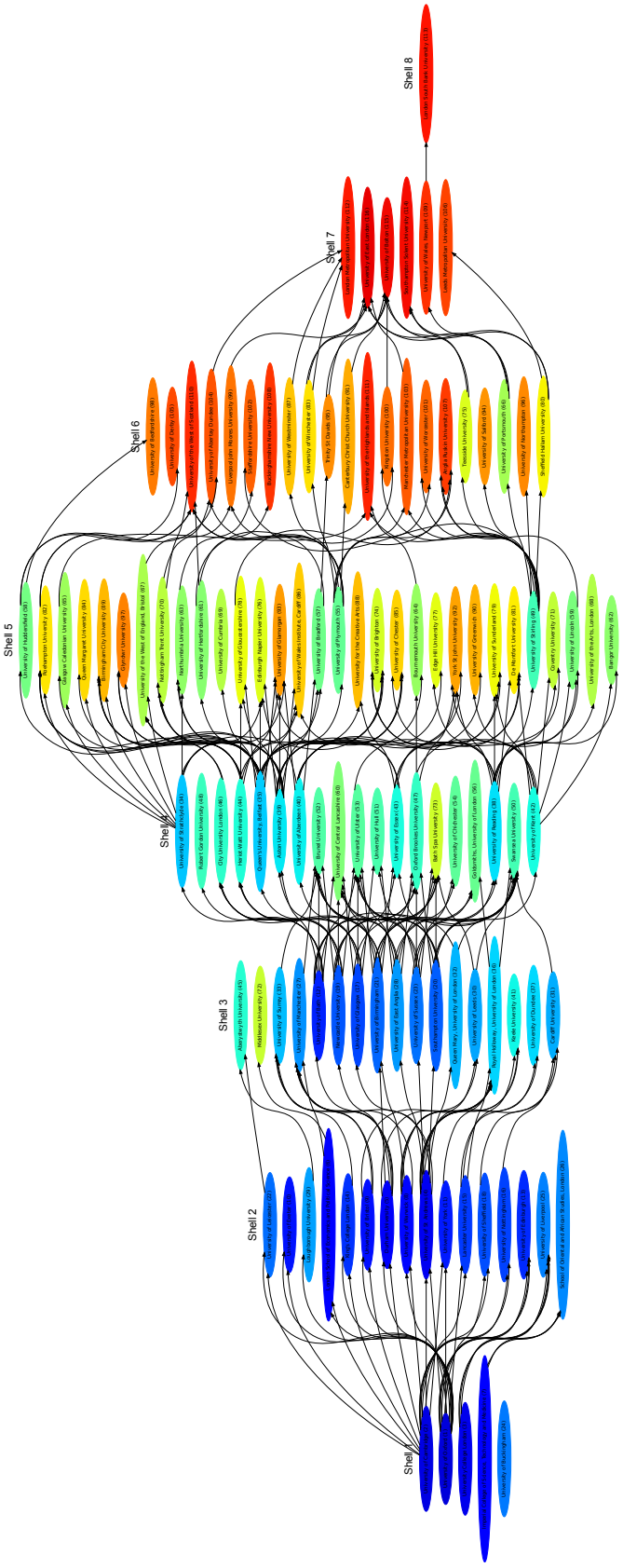
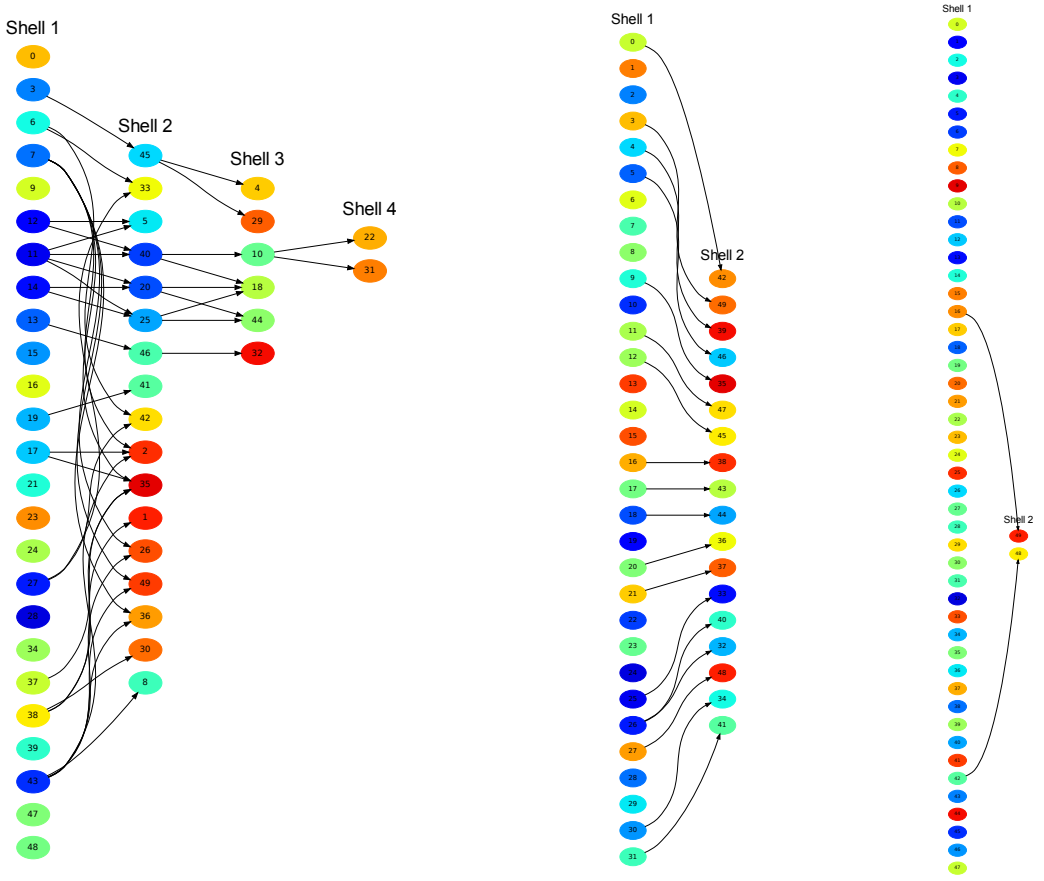


Figure 3.12.: Visualisation of the GUG11 population coloured by power index.



(a) Generation 1 (b) Generation 5 (c) Generation 10

Figure 3.13.: Pareto shell visualisations of the search population in the $(\mu+\lambda)$–ES evolving solutions to the 3-objective instance of DTLZ2. Solutions are sorted into their Pareto shells and coloured by "ranked" average rank (such that colours indicate the ranks $1, \ldots, N$ produced by ranking the individuals' average rank scores $\bar{r}_i$); a blue solution represents one with a good average rank while red indicates a poor rank. The value in the node is an identifier for the solution it represents. Clearly, as the search proceeds and better solutions are found, the population becomes mutually non-dominating.

tion. At each generation the parent population was copied to produce λ child solutions that were perturbed with an additive single-point Gaussian mutation of standard deviation one tenth the feasible range of the decision variables. Selection was based on Pareto sorting of the combined parent and child populations; if a shell contained more solutions than were necessary to fill the remainder of the new parent population a subset of the solutions in that shell were selected at random. The populations in Figure 3.13 are the initial random parent population, as well as the parent populations for generation 5 and generation 10. As we have seen in previous examples, some of the individuals have a higher power index than would be indicated by their Pareto shell. We also find that again the more powerful solutions are those in the Pareto optimal shell that dominate the most individuals. As the search progresses beyond 10 generations and the algorithm generates superior solutions

the population becomes mutually non-dominating. If the child population also exclusively comprises non-dominated solutions then the selection pressure provided by the selection operator is reduced to that of uniform random selection. At this point, the visualisation provides no extra information than would be available from simply inspecting the power index ranking of the solutions.

This section has presented various techniques for ranking a many-criterion population, as well as showing how one of them, Pareto sorting, can be combined with some of the other methods to provide an intuitive graph-based visualisation of the individuals in such a population. Section 3.2 provided a discussion of techniques for constructing league tables, as is often done when the individuals in a many-criterion population must be compared, and the visualisation technique that we have been demonstrating was developed with the aim of addressing some of the problems with aggregating criteria to construct a league table. All of the ranking methods have attractive features, however in general we prefer average rank and the power index. The power index provides a method of considering a multi- or many-criterion population using dominance (or the probability of dominance) and as such is a suitable candidate for exploring and visualising the structure of a population in terms of its individuals' ranks. That said, average rank provides the basis for the power index and was shown to be well correlated in the examples presented above. It also has a significant advantage in that it is simple for non-experts to understand. For this reason, we rely on average rank later in this thesis when occasions arise where it is necessary to rank a population. The visualisation method we have proposed is not, however, a league table. As Usher and Savino [2006] observe, a league table is a total ordering of individuals and although the power index usually produces a total ordering, the Pareto sorting on which the layout of the graph is based does not typically do this in the multi- or many-criterion case.[5] Given that we are now considering methods that visualise the performance of many-criterion individuals in lieu of creating a league table, in the next section we consider methods for illuminating dominance relationships using standard techniques for visualising high-dimensional data.

3.4. Visualisation with the Dominance Distance

In the previous section we demonstrated a method for illustrating the relative quality of many-criterion individuals by considering their rank, providing a useful way of exploring the structure of a many-criterion populations and the relative quality of its individuals without taking the conventional step of constructing a league table. In this section we present another method for visualising the individuals in a population based on their dominance relationships. As mentioned previously, a common method for visualising high-dimensional data is to project the data into a low-dimensional space. We have already

[5]Pareto sorting of a single-criterion population places each non-tied individual in its own Pareto shell.

discussed some of the common methods for doing this in the context of multi-objective populations, namely Principal Component Analysis (PCA) [Jolliffe, 2002], Self Organising Maps [Kohonen, 1995; Obayashi, 2002]. Generative Topographic Mapping [Bishop et al., 1998; Fieldsend and Everson, 2005] and Neuroscale [D. Lowe and M. E. Tipping, 1996; Fieldsend and Everson, 2005].

Metric information is a common part of these projection processes. SOMs, for example, use distance information for clustering reference vectors. Neuroscale tries to preserve the pairwise distances between individuals in the original, high-dimensional, space and the projection space. A common choice of metric is Euclidean distance between objective vectors, however this discards information about the dominance relations between the individuals. These relations provide important information about the structure of a multi-criterion population and it would be useful to capture this structure in a visualisation of the population. It is not, however, clear that the use of a metric in a space defined by criteria that are measured on different scales is sensible. To this end, in this section we investigate a metric that evaluates the distances between many-criterion individuals based on dominance. We call this metric the *dominance distance*, and it is used to visualise various populations, containing both dominated individuals, as well as mutually non-dominating individuals using *multi-dimensional scaling* (MDS) [Sammon, 1969; Webb, 2002].

We begin by defining a new measure of similarity between the individuals in a population, which attempts to capture the degree of dominance between individuals. We regard points $\mathbf{y}_i$ and $\mathbf{y}_j$ as similar if they both dominate a third point $\mathbf{y}_p$, are both dominated by $\mathbf{y}_p$, or both are mutually non-dominating with $\mathbf{y}_p$. Refining this idea, we define the similarity of $\mathbf{y}_i$ and $\mathbf{y}_j$ relative to $\mathbf{y}_p$ as proportional to the number of criteria on which $\mathbf{y}_i$ and $\mathbf{y}_j$ have the same relation (greater than, less than or equal) to $\mathbf{y}_p$. Thus

$$\begin{aligned} S(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p) = \frac{1}{M} \sum_{m=1}^{M} \Big[& I((y_{pm} < y_{im}) \wedge (y_{pm} < y_{jm})) \\ & + I((y_{pm} = y_{im}) \wedge (y_{pm} = y_{jm})) \\ & + I((y_{pm} > y_{im}) \wedge (y_{pm} > y_{jm})) \Big] \end{aligned} \tag{3.27}$$

where $I(p)$ is the indicator function that is 1 when the proposition p is true and 0 otherwise. The second term in Equation 3.27 accounts for exact equality on a criterion. While this occurrence is very rare with real-valued objectives in solutions to a multi-objective problem, it is likely that it will arise in the many-criterion populations, such as the GUG09 population, that we discuss in this thesis due to the presence of categorical and integer-based criteria. We define the distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ relative to $\mathbf{y}_p$ as:

$$D(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p) = 1 - S(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p). \tag{3.28}$$

The dominance distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ is obtained by averaging $D(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p)$ across

all of the individuals in the population:

$$D(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{N-2} \sum_{p \notin \{i,j\}}^{N} D(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p). \tag{3.29}$$

The dominance distance is a metric. It is clear from Equation 3.27 that $D(\mathbf{y}_i, \mathbf{y}_j) = D(\mathbf{y}_j, \mathbf{y}_i)$. Since the maximum value of the sum in Equation 3.27 is 1, $0 \leq S(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p) \leq 1$ and therefore $D(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p) \geq 0$. It is easily checked by direct substitution in Equation 3.27 that $S(\mathbf{y}, \mathbf{y}; \mathbf{y}_p) = 1$ for all $\mathbf{y}_p$, so $D(\mathbf{y}, \mathbf{y}) = 0$. Conversely, $S(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p) = 1$ for all $\mathbf{y}_p$ only if $y_{im} = y_{pm}$ for all m. Thus $D(\mathbf{y}_i, \mathbf{y}_j) = 0$ if and only if $\mathbf{y}_i = \mathbf{y}_j$.

To see that $D(\cdot, \cdot)$ obeys the triangle inequality we associate with $\mathbf{y}_i$ and $\mathbf{y}_j$ strings $\mathbf{b}_i$ and $\mathbf{b}_j$ of length M on an alphabet of the symbols $\{-1, 0, +1\}$ so that a "-1" in position m of the string for $\mathbf{y}_i$ indicates that $y_{im} < y_{pm}$, a "$+1$" if $y_{im} > y_{pm}$, and a "0" if $y_{im} = y_{pm}$. For example, with $M = 7$ criteria:

m	1	2	3	4	5	6	7
$\mathbf{b}_i$	-1	-1	$+1$	$+1$	-1	$+1$	0
$\mathbf{b}_j$	-1	$+1$	-1	$+1$	$+1$	$+1$	-1

Here $\mathbf{y}_i$ is greater than $\mathbf{y}_p$ on criteria 3, 4 and 6, and $y_{i7} = y_{p7}$, while $\mathbf{y}_j$ is greater than $\mathbf{y}_p$ on criteria 2, 4, 5 and 6. Then $M \times D(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p)$ is the Hamming distance between the strings $\mathbf{y}_i$ and $\mathbf{y}_j$, namely the number of positions in which their symbols disagree. In the example $D(\mathbf{y}_i, \mathbf{y}_j) = 4/7$. It is well known that the Hamming distance is a metric, which shows that the dominance distance is also a metric.

A further characterisation of the dominance distance is provided by noting that with one criterion ($M = 1$) the distance is found by calculating the difference in the ranks: $D(y_i, y_j) = |r_i - r_j|$. Given such a method for finding the dominance distance in the single-criterion case, we can find the distance between a pair of individuals $\mathbf{y}_i$ and $\mathbf{y}_j$ on each criterion and take the average to find the overall dominance distance:

$$D(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{M} \sum_{m=1}^{M} |r_{im} - r_{jm}| \tag{3.30}$$

We note that the distance between individuals is measured by the average magnitude of the difference in their ranks on each criterion, not the magnitude of the difference of their average ranks. Equation 3.30 provides an efficient way of calculating $D(\mathbf{y}_i, \mathbf{y}_j)$ compared with a straightforward application of Equations 3.27 and 3.29.

3.4.1. Multi-dimensional Scaling

We now apply the dominance metric developed in the previous section to the problem of constructing a low-dimensional embedding of many-criterion individuals for the purpose of visualising the population.

Both *metric* [Kruskal, 1964; Sammon, 1969; Webb, 2002] and *classical* [Webb, 2002] multi-dimensional scaling (MDS) can be used to create a low-dimensional embedding of a dataset. Since the aim of this work is to construct a visualisation of a many-criterion population based on the dominance distance between individuals, metric MDS is a natural choice. Classical MDS is not discussed further, and for brevity we refer to metric MDS simply as "MDS".

The basic prerequisite for using MDS to reduce the dimensionality of a population of individuals is a matrix of distances. Often this takes the form of a matrix of pairwise Euclidean distances, but for the purposes of this thesis we employ the dominance distance defined above. Whatever metric is chosen, by employing MDS we seek a low-dimensional embedding of the individuals in a population such that pairwise distances in the original, high-dimensional, space are preserved in the low-dimensional space. Two individuals that are close in the high-dimensional space should remain so in the low-dimensional space.

Put simply, MDS obtains a set of coordinates from a set of distances by embedding the individuals into an N-dimensional space from where they are projected into a low-dimensional (e.g., 2) space. According to Schoenberg [1935], and discussed in Gower [1985], a matrix of pairwise distances $\mathbf{\Delta}$, where $\Delta ij = D_{ij}^2$, is a Eucludean matrix iff the matrix $\mathbf{F}$

$$\mathbf{F} = \frac{1}{2}\left(I - \frac{\mathbf{1}\mathbf{1}^T}{N}\right)\mathbf{\Delta}\left(I - \frac{\mathbf{1}\mathbf{1}^T}{N}\right) \tag{3.31}$$

is positive semi-definite. In this case, if $\mathbf{F} = \mathbf{Z}\mathbf{Z}^T$ is a decomposition of $\mathbf{F}$, then the row $\mathbf{z}_i$ of $\mathbf{Z}$ forms the coordinates of the embedding of the ith individual, corresponding to $\mathbf{y}_i$, so that $\|z_i - z_j\|^2 = \Delta_{ij}$. Approximate coordinates in a lower dimensional space (e.g., 2 dimensions) can be found by projecting onto the principal components of $\mathbf{Z}$, which turn out to be the eigenvectors of $\mathbf{F}$. If $\mathbf{F}$ is not positive semi-definite, so that $\mathbf{\Delta}$ does not correspond to Euclidean pairwise distances, an approximate embedding may still be obtained by projecting onto the first principal eigenvectors of $\mathbf{F}$. In all of the work reported here, the eigenvectors of $\mathbf{F}$ are non-negative to within numerical precision.

Spectral decomposition of $\mathbf{F}$ has a computational complexity of $O(N^3)$, however, projections of several hundred points can easily be achieved in a second and if necessary the procedure might be made more efficient by finding only the first few principal eigenvectors of $\mathbf{F}$.

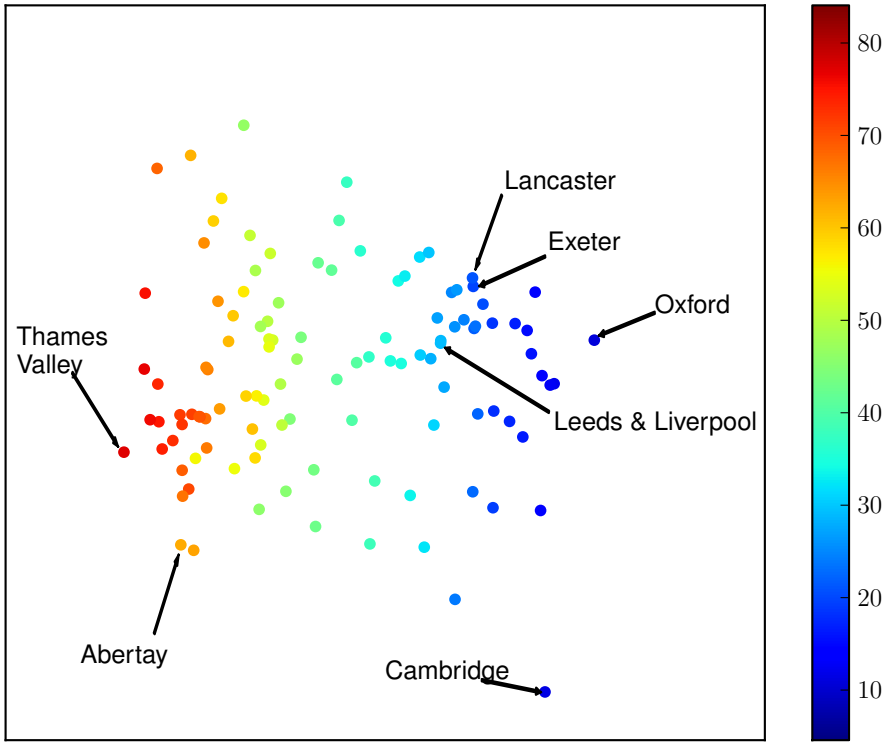


Figure 3.14.: MDS visualisation of the GUG09 population. Universities are coloured by power index. The embedding seeks to preserve pairwise dominance distances. Generally, those universities with a similar power index have been placed close together.

3.4.2. Illustration

One of the advantages of using dominance distance as a basis for compressing the population into a low-dimensional space is that it can be used with both mutually non-dominating populations, such as the elite sets produced by optimising a multi- or many-objective problem, and populations containing dominated individuals, such as the GUG09 running example. In contrast, however, the Pareto shell visualisation does not provide any interesting structural information about a mutually non-dominating population. We illustrate the approach on both types of population for MDS, beginning with the GUG09 population.

Dominated Populations

Recall, that in the GUG09 population, some universities dominate others. Given this, according to the definition of dominance distance provided above, similar universities dominate, or are dominated by, the same universities as each other.

Figure 3.14 and 3.15 present MDS embeddings of the GUG09 population based on domi-

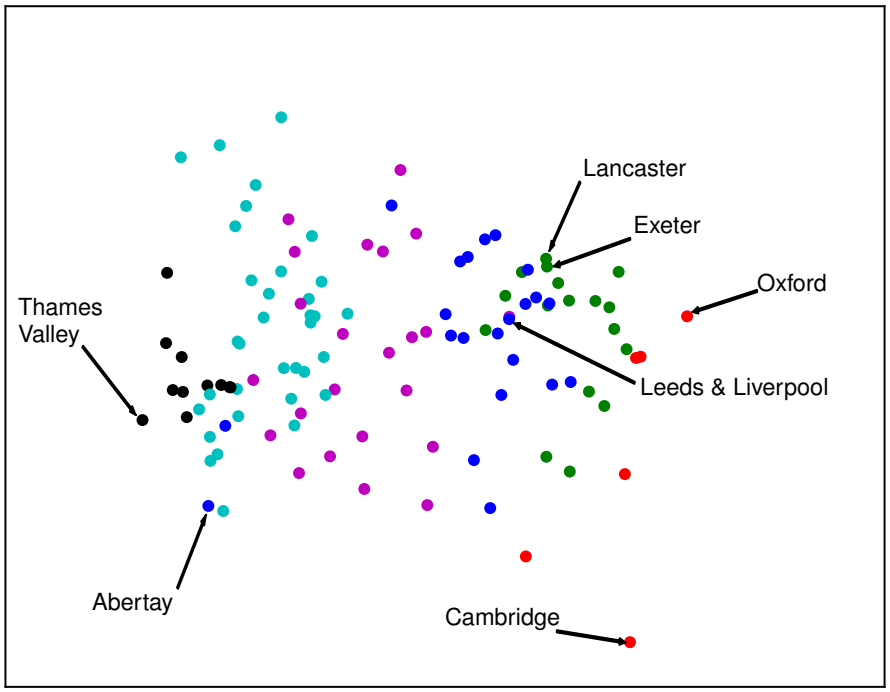


Figure 3.15.: MDS visualisation of the GUG09 population. Universities are coloured by Pareto shell. The embedding preserves pairwise dominance distances. Generally, as with the power index example, those universities residing in the same Pareto shell have been placed close together.

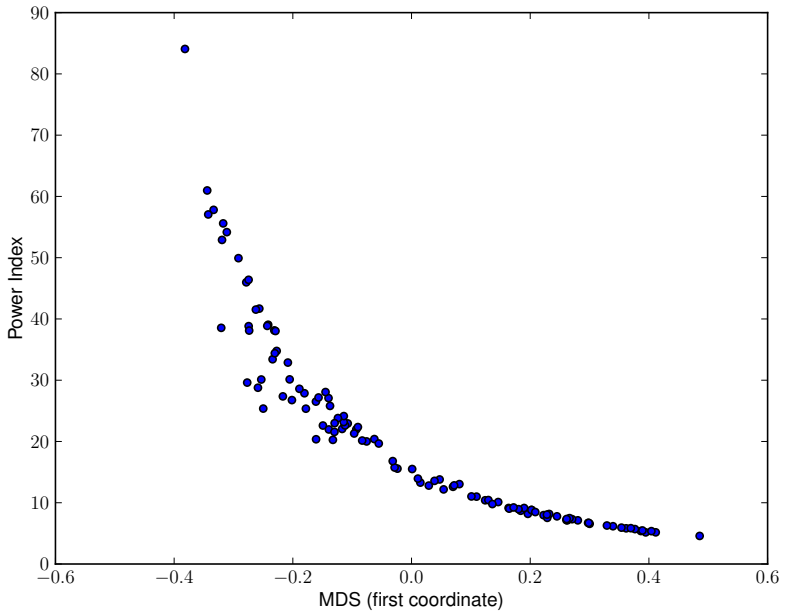


Figure 3.16.: The correlation between the first dimension of the MDS projection of the GUG09 population and the power index values for each university.

nance distance; some universities of interest have been highlighted. Clearly, the horizontal axis has identified the main trend in the data and is aligned with the power index; Figure 3.16 shows the correlation between the horizontal axis values and the power index of each

university. The second axis is used to differentiate between universities of similar power. We observe that the two extreme universities in the ordinal axis are Oxford and Thames Valley, which are the two extreme universities when the population is ranked by power index or average rank. In Figure 3.14 the universities are coloured by their power index, so that a strong university is coloured blue and a weak university is red, and it is clear to see that universities of similar power have (generally) been placed close together. Upon examining Figure 3.15, in which universities are coloured according to the Pareto shell to which they belong, a similar trend is observed, however here some discrepancies can be seen. A good example is Abertay, which is in shell 3 (coloured blue). The universities that it is in close proximity to are those in shells 5 and 6, indicating that they are weaker than Abertay. This is the phenomenon observed in the previous section, in which a university achieves a good score on one of the eight criteria and scores poorly on the others, making it difficult to dominate. The power index is only mildly affected by this, and the university receives a power that better suits its entire set of KPI scores. The placement of Abertay with other universities of low power index implies that it has similar dominance relationships with other universities, based on its generally poor criterion values. Leeds, in shell 4, is placed close to Liverpool, in shell 3. As we have previously observed Leeds is the most powerful member of shell 4 according to the power index, so it is placed close to the more powerful members of shell 3. According to popular perception of university quality, Cambridge would be placed close to Oxford. It is however, placed away from the other members of shell 1. This is because the procedure of imputing a score to replace its missing NSS score has assigned it the worst NSS score in shell 1, making it appear artificially worse than it would likely be with a real score.

Lancaster is placed very close to Exeter. This implies that they have dominance relationships in common with a large proportion of the other universities in the population, and upon examining these relationships this is the case. Exeter and Lancaster have 89 of 111 possible dominance relationships in common. They dominate 28 universities in common, are both dominated by Oxford and there are 60 universities with which both a mutually non-dominating.

Figure 3.17 shows the distribution of dominance distances between all of the universities, ignoring the distance between a university and itself which is 0. As can be seen by the mark showing the distance between Exeter and Lancaster, they are relatively close. Examining the eigenvalues of the Euclidean distance matrix $\mathbf{F}$ shows that the projecting onto the first two eigenvectors preserves 70.4% of the variance in the population. Were we to include an additional third eigenvector in the projection we would recover an additional third 4.84%. This would provide only a relatively small improvement in terms of the amount of variance preserved in the embedding considering the added complexity that a third dimension in the visualisation would incur.

In Figure 3.18 we use the MDS embedding of the GUG09 population to illustrate the simi-

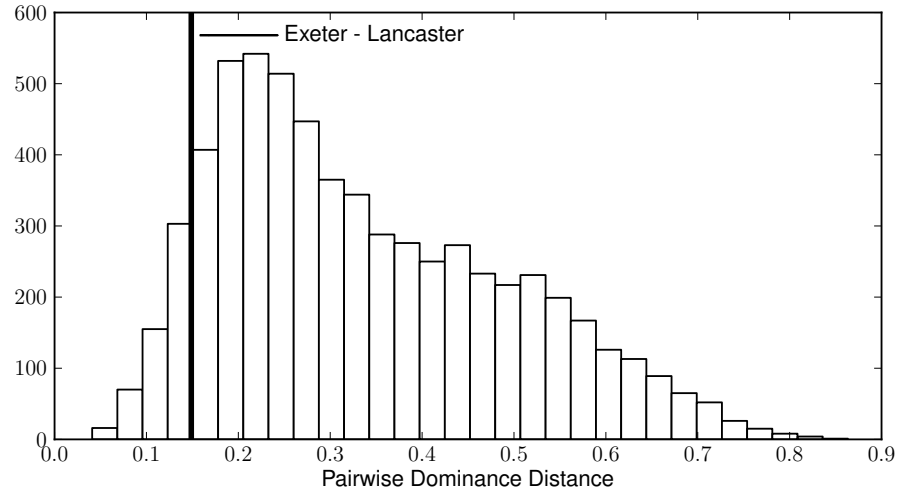
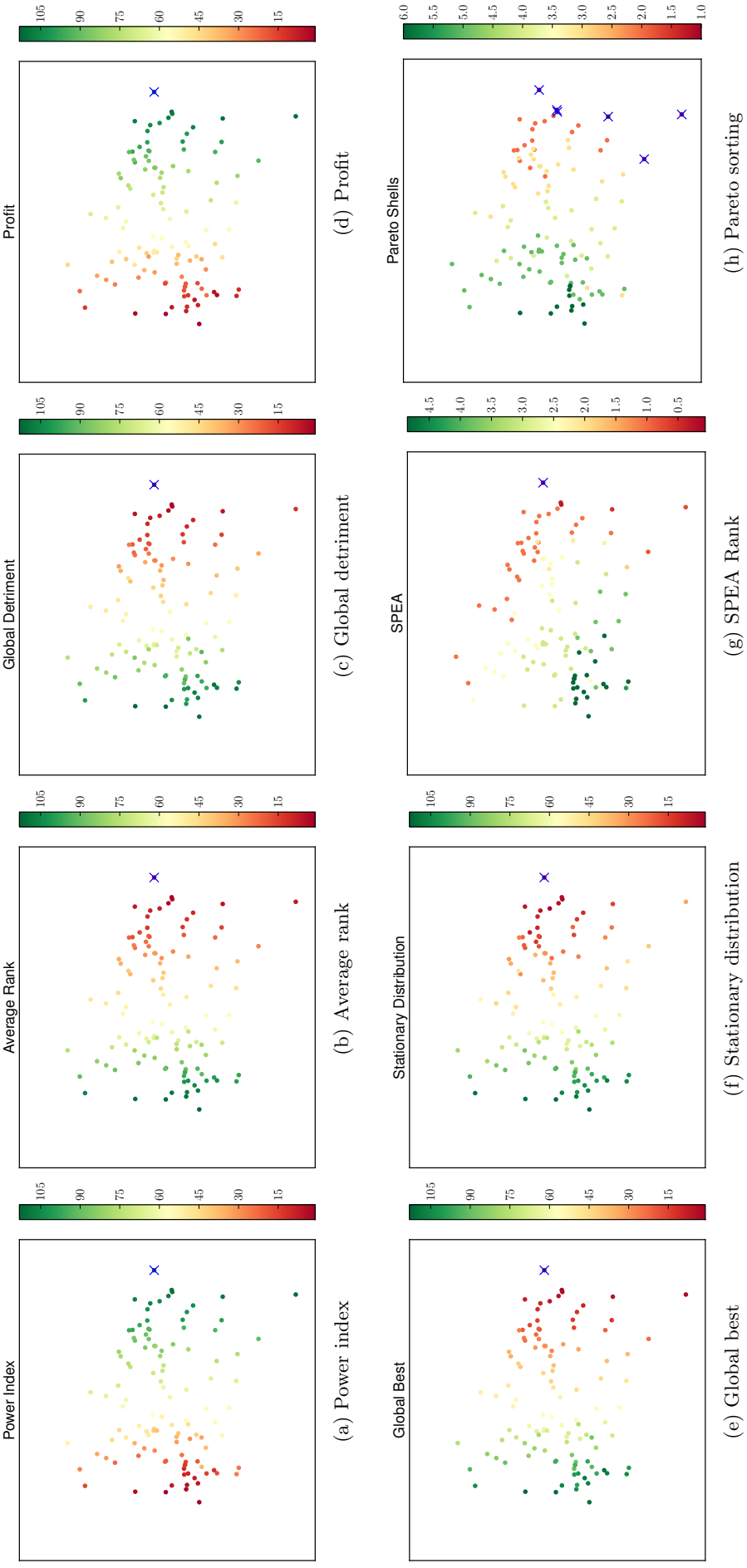


Figure 3.17.: The distribution of dominance distances between universities in the GUG09 population. Note, that the distance between a university and itself is not considered. The distance between Exeter and Lancaster is marked, and the two universities are shown to be relatively close.

larity between some of the various many-criterion ranking methods that we have discussed in this chapter and Chapter 2. We present eight visualisations of the GUG09 population embedded in a 2-dimensional space with MDS, and in each of them the individuals are coloured according to their score as determined by a different ranking method. Whereas previous illustrations have been coloured so as to indicate regions of high and low individual quality in a population, the colour scheme in these figures is used to show the score of the individual according to a particular ranking method. A individual that is coloured red has a low value according to the ranking method in question, while a green individual has a high score. Recall that both power index and profit ranking are to be maximised, while the other ranking methods (average rank, global detriment, global best, stationary distribution, SPEA rank and Pareto shell) are to be minimised. As can be seen by examining the eight figures, the general trend observed in earlier examples, where the high quality individuals are gathered together on the right-hand side of the embedding and the poorest individuals are collected on the left-hand side, is present in all of the ranking methods considered. In each of the figures, the individual with the best score is marked by a cross; in the case of the Pareto sorting example (Figure 3.18(h)) all of the Pareto optimal individuals are marked. As would be expected, in all of the illustrations the best individual is Pareto optimal. Figure 3.19 further illustrates the trends shown in Figure 3.18 by plotting the score of each individual according to each pair of ranking methods. In all cases there is a noticeable agreement between the ranking methods, whether the ranking methods are well correlated (e.g., average rank and global detriment) or anti-correlated (e.g., power index and average rank). Interestingly, there is less agreement between the first six ranking methods in the grid when compared to the SPEA rank and Pareto sorting approaches. This is likely due to the fact that they depend on dominance and are operating in a many-criterion space; as has been discussed previously in this thesis, dominance



(a) Power index (b) Average rank (c) Global detriment (d) Profit

(e) Global best (f) Stationary distribution (g) SPEA Rank (h) Pareto sorting

Figure 3.18.: An MDS embedding of the GUG09 population. In each panel, the individuals are coloured by one of the ranking schemes discussed in this chapter. The highest quality individual in each case is marked with a cross. Each of the ranking methods follows a generally similar trend, in which the high quality individuals are on the right-hand side of the embedding and the low quality individuals are on the left.

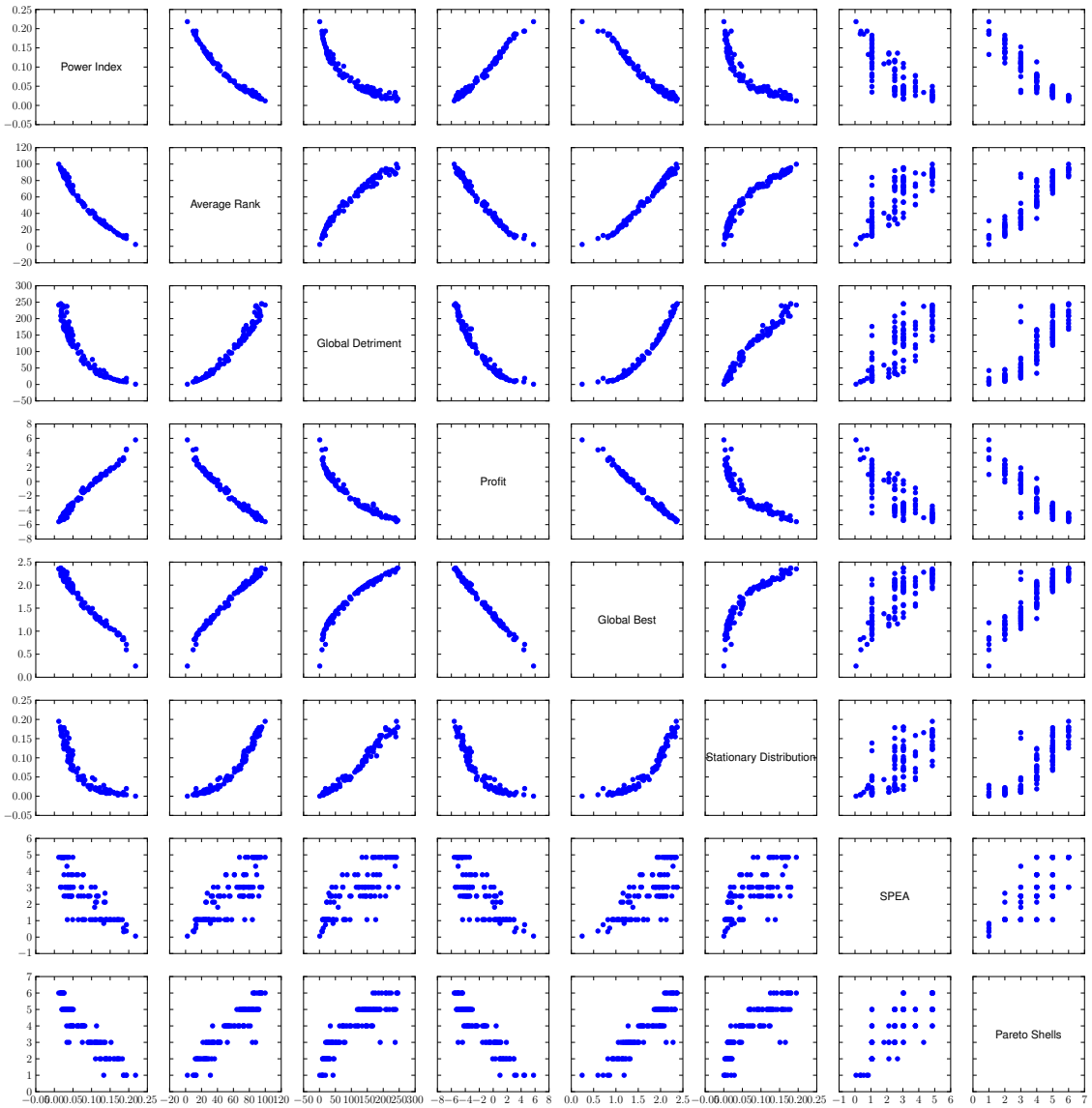


Figure 3.19.: Correlation between ranks of individuals in the GUG09 shown in Figure 3.18 according to: power index; average rank; global detriment; profit; global best; stationary distribution; SPEA rank; and Pareto shell.

does not discriminate well between individuals in such an environment.

We now introduce a new many-criterion population, with which we demonstrate one of the significant advantages of using the dominance distance to visualise a population with MDS. Figure 3.20 illustrates a population of 165 individuals. Each individual represents a wireless access point (AP) in a mobile telephone network. Each access point is responsible for connecting mobile telephones to the network so that consumers can make telephone calls, and the performance of the access points is described by a set of 27 KPIs. These KPIs monitor aspects of the performance of an AP such as the frequency of call drop-outs and bit loss rates. Clearly, a mobile telephone provider wishes to provide the best possible service to a consumer, and so it is important for them to ensure that the maintenance issues are dealt with. Unfortunately, it is not possible for engineers to perform maintenance on

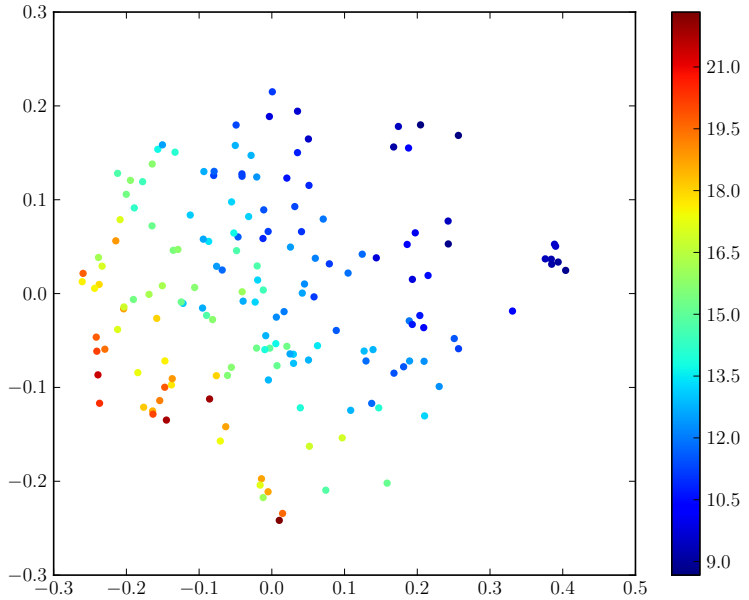
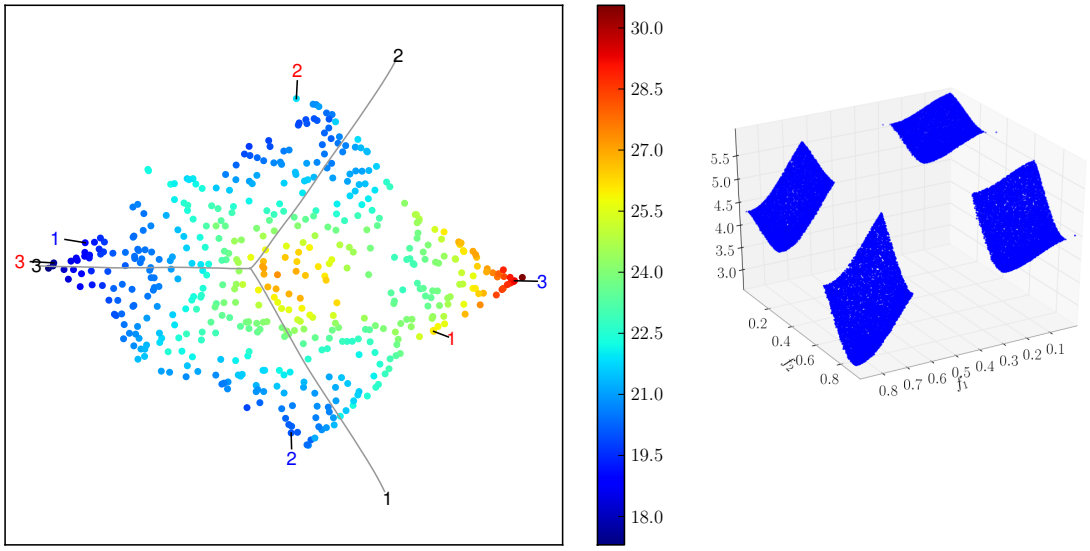


Figure 3.20.: The performance of wireless access points in a mobile telephone network visualised according to 27 KPIs using dominance distance MDS. As with the populations comprising smaller numbers of criteria, the MDS embeds the individuals so that regions of quality can be identified; those individuals with a good power index have been placed on the left-hand side of the embedding, while those with a poor power index have been placed on the right-hand side.

all of the APs simultaneously and their efforts must therefore be guided by which of the APs would most benefit from maintenance work. As such, a visualisation of the KPI data is useful. Unfortunately, with each APs entry residing in a 27-dimensional space, this is not a trivial matter. It is not feasible, for example, to use the Pareto shell visualisation demonstrated in the previous sections because all 165 individuals reside in the Pareto optimal shell (c.f. Figure 3.3), and there is no dominance structure for that method to illustrate. It is, however, possible to illustrate the population using the dominance distance and MDS since the dominance distance considers relations for each distinct criterion. This embedding is shown in Figure 3.20. As with the previous GUG09 visualisations, the individuals are coloured according to their power index (blue again indicates the more powerful individuals), and as with the other examples in this section the individuals are broadly grouped according to their quality. Such a visualisation is useful to an engineer who is responsible for designing a maintenance schedule for the telephone network. By examining Figure 3.20 it is immediately clear which of the APs need attention; they are the APs with a poor power index. That said, examining the additional information provided by the proximity of APs in the dominance distance embedding provides a useful notion of the similarity between them. An AP might not receive the worst rank in the population, but by its proximity to an individual that does have a bad rank would be recognised as a candidate for maintenance.

In examining the variance captured by the MDS projection we observe that 48% is captured by the first two eigenvectors. The addition of the third eigenvector captures an additional



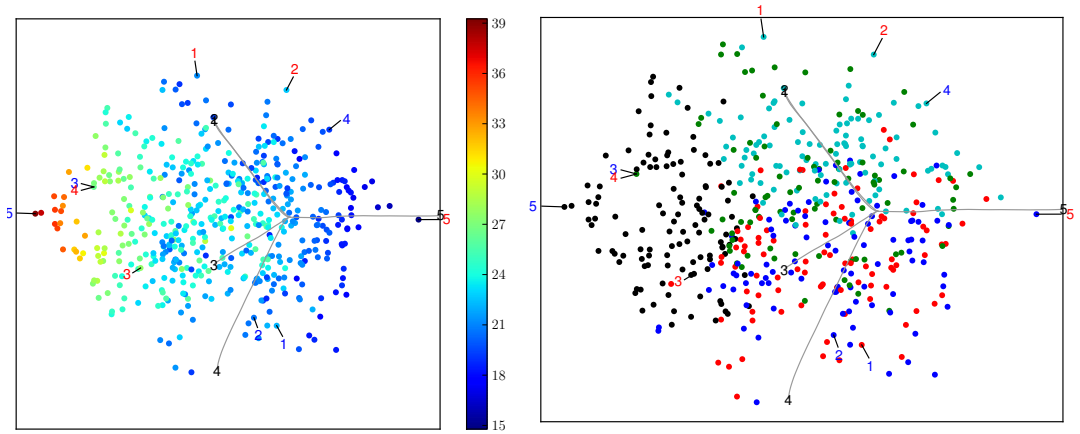
(a) MDS (b) Original objectives

Figure 3.21.: An MDS embedding of 500 individuals drawn from the true Pareto front of DTLZ6 in three objectives (a). Individuals are coloured by their power index. The best and worst individuals are shown in blue and red, and samples from the minimising bounding box have been projected into the embedding. The symmetry of objectives 1 and 2 can be seen, as can patches of high and low quality. The population shown in (b) illustrates the geometry of the 3-objective DTLZ6 Pareto front.

8%. While the projection has captured less variance than in the GUG09 example discussed earlier we draw attention to the fact that this is a larger population, both in terms of individuals and more significantly criteria (recall that the GUG09 population comprises 8 KPIs while this population has 27). That said, the visualisation it results in clearly clusters individuals of similar rank, and an engineer could easily use it as the basis for designing a maintenance schedule.

Non-dominated Populations

In addition to visualising populations of dominated individuals, we also demonstrate its efficacy at reducing the dimensionality of mutually non-dominating sets. Figure 3.21 shows 500 3-objective solutions to the test problem DTLZ6 [Deb et al., 2002] projected onto the principal two eigenvectors of $\mathbf{F}$ (recall that $\mathbf{F} = \mathbf{Z}\mathbf{Z}^T$, where $\mathbf{Z}$ is a matrix in which the rows are N-dimensional individuals between which the Euclidean distances are the same as the distances between the individuals in the population $\mathbf{Y}$ as defined by the metric $\mathbf{D}$). Individuals were generated by drawing samples from the known true Pareto front of the problem, and are coloured by their power index. In order to help orient the decision maker, we also mark the best and worst solution for each objective and the edges of the axis parallel bounding box which contains the solutions which meet at the global best point [Garza-Fabre et al., 2009], namely $(\min_i(y_{i1}), \min_i(y_{i2}), \ldots, \min_M(y_{iM}))$;



(a) Power index (b) Best objective

Figure 3.22.: 500 individuals from the true Pareto front of DTLZ6 in five objectives, coloured by power index (a) and best objective (b). The symmetry seen in Figure 3.21 remains, even given the additional objectives. Additionally, it is possible to observe correlations between some objectives by their placement (e.g., y_1 and y_2, y_3 and y_4; each pair is overlaid).

these edges are parallel to the coordinate vectors in the M-dimensional space. The best individuals are marked in blue and the worst are marked in red. The black numbers at the end of the projected axes indicate which criterion a particular axis belongs to. The visualisation reveals the symmetry between objectives y_1 and y_2, with y_3 a distinguished objective. Note that the worst solutions on each objective are mapped close to the ends of the bounding box axes, while the best solutions are opposite these ends, indicating that the visualisation is providing a topographic representation from the dominance distance, which itself is based only on the greater than, less than or equality relations between individuals. It is clear from the visualisation that a good average rank (dark blue) corresponds to poor values of y_3 and good solutions of y_3 are only obtained by having a poor average rank.

Note that although distinct clusters are not evident, the average rank shows that there are isolated patches of high and low rank. A 3-objective instance of DTLZ6, as shown in Figure 3.21(b), is characterised by the clustering of Pareto optimal solutions into four "cushions". The cushions are disconnected, and the disconnected regions are not visible in the dominance distance MDS embedding because the dominance distance does not incorporate Euclidean information. While it would be best to retain all features of the population, such as discontinuities, it is important to consider the dominance relationships between individuals in order to best understand the population.

Figure 3.22 shows solutions from the 5-objective version of the same test problem. It is apparent that the visualisation has identified a remarkably similar structure and the symmetry in the population remains despite the additional two objectives. Figure 3.22 (a) shows the individuals in the population coloured by their power index, and the distribution is very similar to that shown in the 3-objective case. The last objective, y_5, is again

distinguished. In Figure 3.22(b) the individuals are coloured according to the objective on which they have the best rank, and it is clear the the individuals whose best rank is on objective y_5 have been placed away from the remainder of the population. Also, there is good correlation between objectives y_1 and y_2 (the bounding box axes are mapped almost on top of one another), and between y_3 and y_4. Separation of these three groups (y_1 and y_2, y_3 and y_4, and y_5) can be seen by the colouring in Figure 3.22(b) in which individuals are generally collected together according to the group on which they are best ranked. For example, those individuals whose best rank is on y_1 or y_2 are projected together.

The dominance distance visualisation of solutions to a problem with a degenerate one-dimensional Pareto front, such as WFG3 [Huband et al., 2005], maps the solutions to a single line along the coordinate axis with solutions arranged by power index along it, and there is only a single non-zero eigenvalue of $\mathbf{F}$. We do not show an example of this projection because there is no additional information provided beyond the simple ordering of individuals by their power index.

We also provide a visualisation for a population of solutions to a real world optimisation problem drawn from the literature. Hughes [2007a] applied MSOPS [Hughes, 2003] to the problem of designing an appropriate set of waveforms that can be transmitted by a Pulsed Doppler Radar to simultaneously measure the velocity and distance of a target. To do this, Hughes optimised a schedule of *Pulse Repetition Intervals* (PRIs), which are the times between the transmission of radar pulses. A solution to the radar waveform design problem consists of P parameters. Each parameter is a PRI, determining the amount of time to wait before transmitting the next radar pulse, and Hughes [2007a] has provided results for schedules of $P = 4, 6, 8, 10$ and 12 PRIs. These parameters map onto an objective vector $\mathbf{y}$ comprising $M = 9$ objectives. The first eight objectives characterise different aspects of the radar signal and the final objective is the total transmission time for the waveform:

- Objectives 1 and 2 measure the median range and velocity before the schedule is not decodable.
- Objectives 3 and 4 measure the median range and velocity before the schedule has blind regions.
- Objectives 5 and 6 measure the minimum range and velocity before the schedule is not decodable.
- Objectives 7 and 8 measure the minimum range and velocity before the schedule has blind regions.
- Objective 9 is the time required to transmit the entire waveform, in milliseconds.

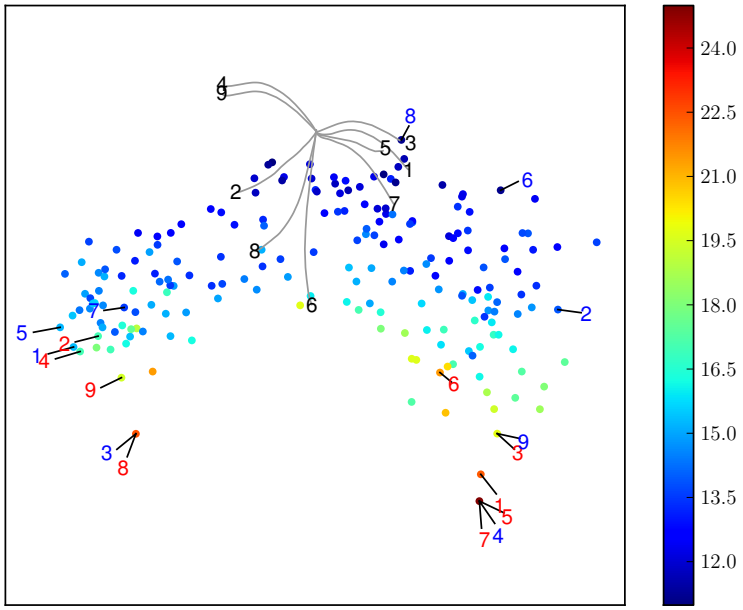


Figure 3.23.: A dominance distance MDS embedding of the radar data. As with the previous examples, individuals have been grouped according to their quality as determined by the power index.

The first eight objectives are to be maximised by good individuals whilst the 9th is to be minimised, however in the datasets [Hughes, 2007b] all objectives have been organised for minimisation. We use 200 randomly sampled solutions of the 11398 solutions in the 12 PRI archive. Figure 3.23 shows the MDS dominance distance visualisation of the 200 solution radar data population. As indicated by the bands of colour, representing bands of similar power, the method has produced a diagram that groups similar solutions together, allowing a decision maker to identify groups of solutions which are similar to each other in the way in which they are related to other solutions in the set. Here the individuals with a low power index are located along the top of the "crescent" of solutions, but with a band of solutions with similar average rank, those at one end of the "crescent" are related to other solutions in ways more similar to each other than those at the other end; that is solutions at one end tend to be greater than or less than other solutions on the same objectives. Note that the bounding box axes, which are more distant from the front than in the DTLZ6 front, are grouped into those associated with range (objectives 1, 3, 5 and 7) and those associated with velocity (objectives 2, 4, 6 and 8). The axis for y_9 has been placed close to the velocity objectives. The visualisation shows that the low rank solutions are associated with good values for y_2, y_6 and y_8, while the best solutions for other objectives, located near the horns of the crescent, have high average rank and are close to (or identical to) solutions which are very poor on other objectives.

We emphasise that this spatial arrangement in the visualisation plane reflects the similarity of order relations among the solutions, rather than their spatial configuration in objective space. Nonetheless, as the colouring by best objective (Figure 3.24) shows, visualisation by MDS with dominance distance tends to group solutions that are close in objective

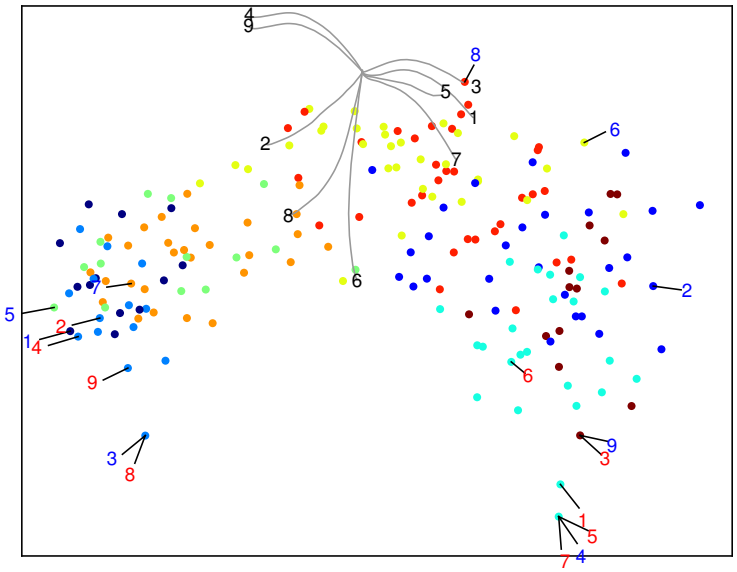
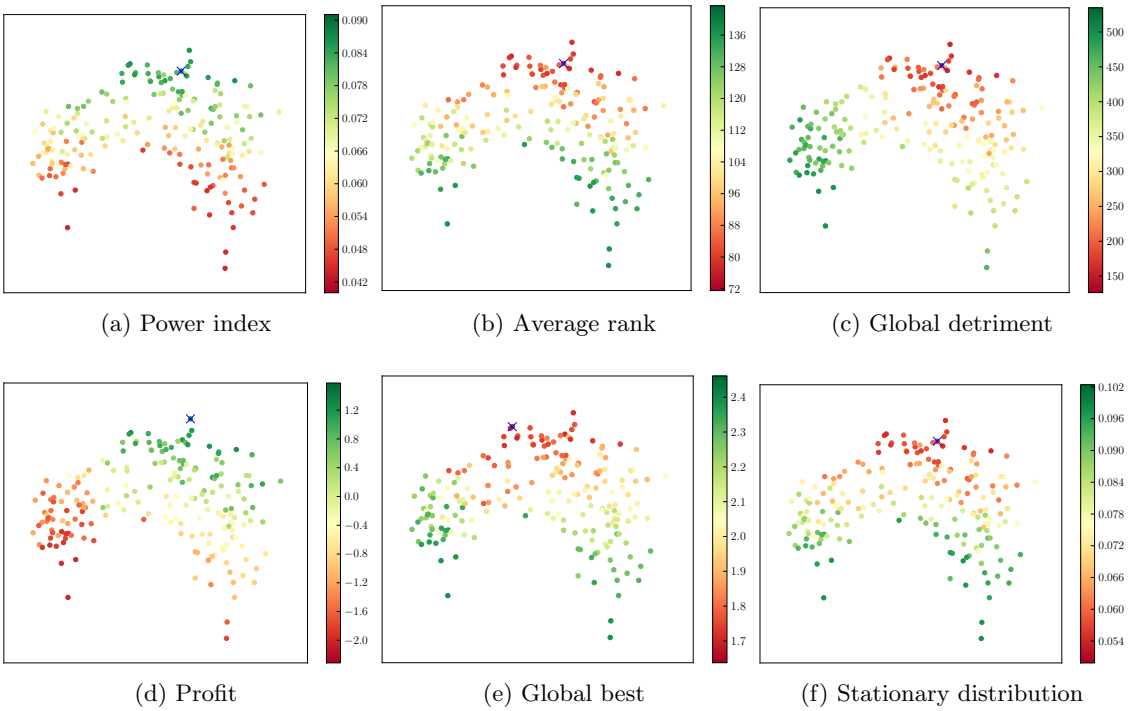


Figure 3.24.: Radar population coloured by best objective. As can be seen by the labels indicating the best and worst individual on each objective, individuals have been placed such that they are grouped according to the type of objective that they best optimise (e.g., range, velocity or transmission time).

space.

Here 54.3% of the variance in the N-dimensional embedding is retained in the 2-dimensional projection onto the plane. Projection onto the third eigenvector of $\mathbf{F}$ captures an additional 5.5%, but visualisation is more cumbersome.

As in the case of the GUG09 examples above, we apply the ranking methods to the radar population and use MDS to visualise which regions have the highest quality (Figure 3.25). Note, that while the previous comparison included Pareto sorting and ranking according to the fitness strategy used in the SPEA algorithm, these examples are of a mutually non-dominating population. As such, these two ranking methods are omitted. In all cases, the ranking methods prefer the region of the population at the top of the crescent shape produced by the MDS embedding. This is away from the regions of the population containing the individuals highlighted in Figure 3.24 as being worst on a particular objective and close to the individuals that are best on objectives y_8 and y_6. Examining the correlation between rankings shown in Figure 3.26, we can see that the general trends are as we have seen in the previous example for the GUG09 population. Some of the correlation plots shown in Figure 3.26 indicate that the individuals belong to one of two "leaves". In that plot, each individual is coloured according to the objective type on which it has the best rank. Thus, an individual whose best rank is a range objective (y_1, y_3, y_5 or y_7) is coloured red, one whose best rank is a velocity objective (y_2, y_4, y_6 or y_8) is coloured blue, and one whose best rank is on the transmission time objective (y_9) is coloured black. A clear grouping is evident in the plots: those individuals that best optimise a velocity objective or the transmission time objective are collected together, and the individuals that best



(a) Power index (b) Average rank (c) Global detriment

(d) Profit (e) Global best (f) Stationary distribution

Figure 3.25.: MDS embeddings of the radar population; individuals are coloured by various ranking schemes.

optimise the range objectives reside in their own distinct cluster. We discuss this grouping further in Chapter 4.

Finally, we use the MDS with dominance distances to illustrate the projection of a population of samples drawn from the Pareto front of a 3-objective test problem and show which regions are preferred by the different ranking methods. Figures 3.27 and 3.29 show a set of 1000 samples from the true Pareto front of a 3-objective instance of the DTLZ2 test problem, the part of a spherical shell in the positive orthant.

From observing the DTLZ2 examples, the MDS has clearly produced an embedding that preserves the geometry of the 3-dimensional objective vectors. As with the DTLZ6 examples shown in Figure 3.21, the effect of projecting the population into a 2-dimensional space with MDS and the dominance distance has been to flatten the population. Turning our attention to the colouring of the individuals we can see that quality is distributed across the population similarly, in that extreme qualities tend to be either in the centre or at the corners of the population. This observation corresponds to existing knowledge about some of the ranking methods that have been used in many-objective selection operators. For example, Garza-Fabre et al. [2010] observed that using a selection operator based on average rank, global detriment or profit offers good convergence properties but leaves large portions of the Pareto front unexplored. A similar point was made by Li et al. [2010a] with regard to average rank. The regions that would be preferred by an MOEA using any of these algorithms for selection can be identified by examining the plots in Figure 3.27

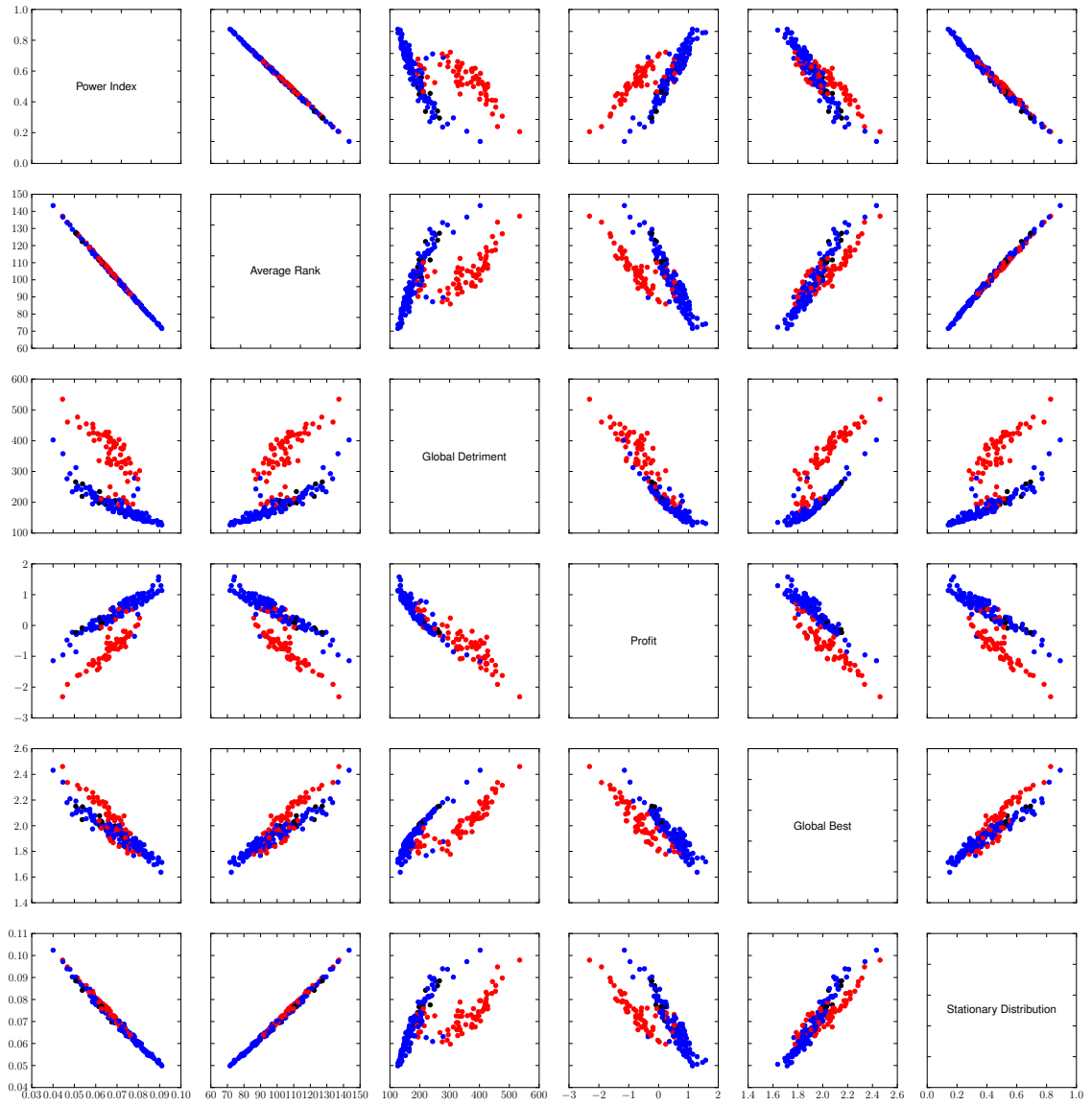
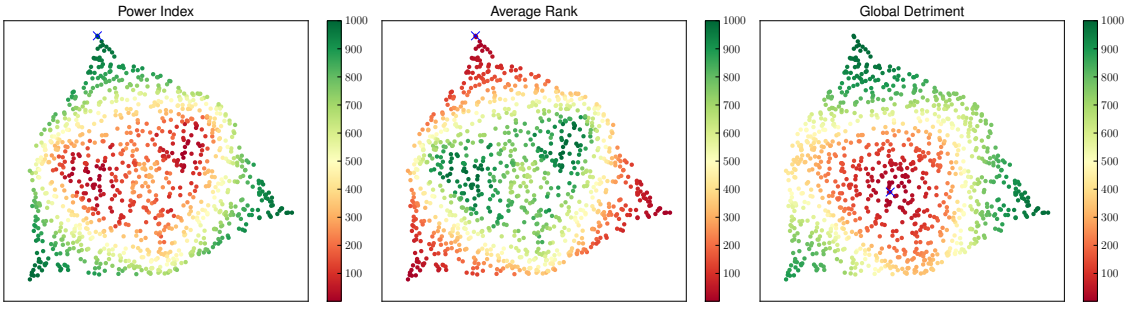
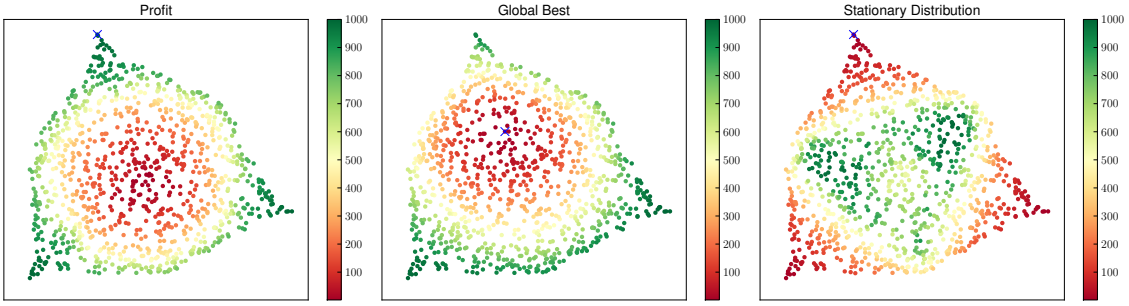


Figure 3.26.: Correlation between ranks of individuals in the radar archive shown in Figure 3.25 according to: power index; average rank; global detriment; profit; global best; and stationary distribution. The colour of an individual indicates the type of objective on which it has the best rank (red indicates a range objective, blue indicates a velocity objective and black indicates the transmission time objective). Colouring in this way reveals a clear relationship between the types of objectives; velocity and transmission time objectives are clustered together, while range objectives generally form a cluster on their own.

for areas of high quality. For example, in the average rank case, the individuals with the highest quality are in the corners of the population. A MOEA using a selection operator based on average rank would therefore select the solutions in the corners. Those in the centre of the population would be discarded, and the diversity in the population would be quickly reduced. This explains the behaviour of MOEAs noted in studies such as Garza-Fabre et al. [2010] and Li et al. [2010a]. From the set of examples shown here, it is clear that many of the commonly used ranking methods will suffer from this problem. A visualisation such as this facilitates diagnostic interaction with the search population or elite



(a) Power index (b) Average rank (c) Global detriment



(d) Profit (e) Global best (f) Stationary distribution

Figure 3.27.: DTLZ2 MDS embeddings coloured by various ranking schemes.

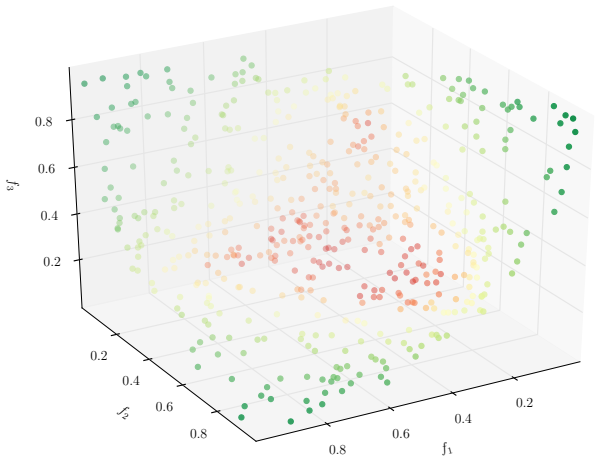


Figure 3.28.: The distribution of average rank across the surface of a concave mutually non-dominating population. Whereas in the case of the equivalent convex population the lowest (best) average rank scores were found in the corners of the population, in this case the centre of the population is preferred.

archive of an evolutionary algorithm to examine how a selection operator is performing and which regions of the search space it prefers.

Figure 3.28 demonstrates the different distribution of average ranks found on a mutually non-dominating concave population. Whereas in the convex equivalent the corners of the front have the lowest, and therefore best, average rank scores, in this case the best scores

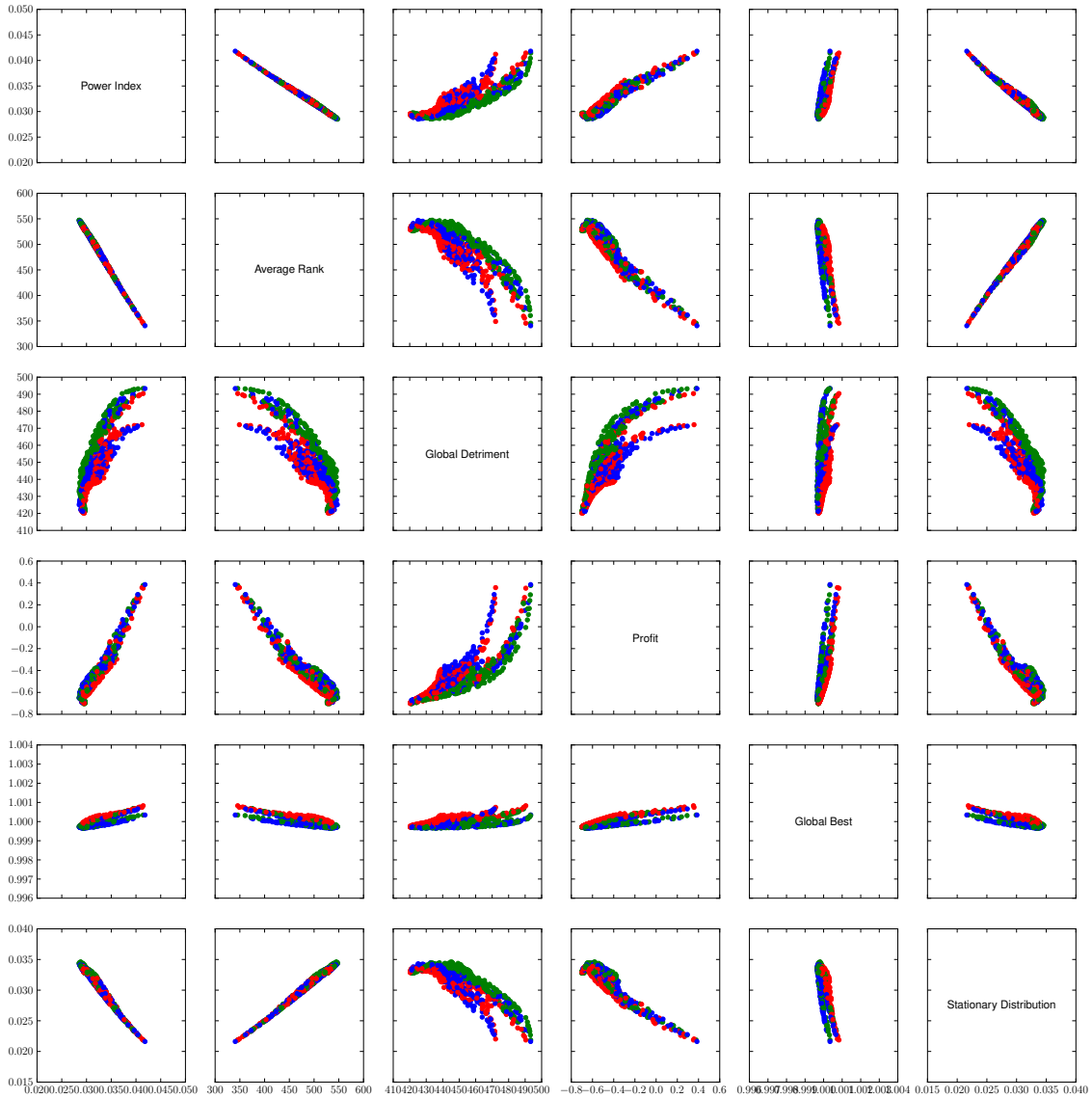


Figure 3.29.: Correlation between ranks of individuals in the DTLZ2 population shown in Figure 3.27 according to: power index; average rank; global detriment; profit; global best; and stationary distribution.

are achieved by those in the centre of the population. A MOEA would therefore prefer the centre of the population and leave the corners unexplored. We also observe that the global best ranking is quite sensitive. The removal of the optimal point could shift to another region of the population if the individual with the best rank is removed.

The correlation shown between the ranking methods, shown in Figure 3.19 is, for the most part, present in the rankings of the DTLZ2 population, shown in Figure 3.29. One notable observation is that there is a degree of correlation between the scores according to global detriment and average rank, power index and profit. According to the definitions provided by Garza-Fabre et al. [2009] we would expect them to be anti-correlated; this was observed to be the case in the earlier two examples. This is because average rank, the power index and profit all prefer values in the corners. Individuals in a corner (for example, the corner

formed by f_1 and f_2) of this population will have one criterion (f_3) on which they have extremely poor scores. When the global detriment is calculated, they have a worse score than the individuals with middling values at the centre of the population. Thus the best individuals according to average rank are the worst according to global detriment and vice versa in this case.

As with the radar example above, the individuals in Figure 3.29 are coloured according to the criterion on which the individual has the best score. Again, the leaf to which an individual belongs indicates the type of objective that the individual best optimises. Although the clustering is less clear than it is in the radar case, it is possible to observe three "leaves", indicating that one corresponds to each objective (this is most clearly seen in the case of the comparison between global detriment and global best).

This section has presented a new metric that considers distances between many-criterion individuals in terms of their dominance relations with the other individuals in a population and used it to project many-criterion individuals into a 2-dimensional space for visualisation with MDS. We have provided various examples of such projections that clearly illustrate the potential behind this approach and described how various aspects of the geometry of a many-criterion population are represented in an MDS projection. While compressing the dimensionality of a many-criterion population with MDS provides a useful visualisation in which features of the high-dimensional space are preserved, it is a linear method that will not capture any nonlinear features present in the data. As such, we intend to extend the work presented here by applying the dominance distance to approaches such as *Isomap* [Tenenbaum et al., 2000], *Locally Linear Embedding* [Roweis and Saul, 2000], *Laplacian Eigenmaps* [Belkin and Niyogi, 2001] and *Maximum Variance Unfolding* [Weinberger and Saul, 2006]. These methods are known to be more adept than MDS at preserving features of multivariate data, such as a low-dimensional manifold on which data points exist. The discovery of such features in multi-criterion populations would yield important information about the population's structure, as well as suggesting that more suitable visualisations can be constructed using nonlinear dimension reduction techniques.

3.5. Conclusion

The construction of a league table from a set of conflicting performance measures is a commonly undertaken task. Such a league table, in which individuals are arranged into a total order, is often constructed by generating a score from a weighted aggregated sum of the performance measures for each individual. This requires the rescaling of the criteria and the selection of weights. Knowing how to weight different criteria can be difficult, and is often subjective. To this end, this chapter has presented methods for illustrating performance data without requiring the construction of a league table or articulation of criterion preferences.

Given the similarity between the task of ranking individuals to convey their quality based on a set of performance measures and the role of the selection operator in a MOEA, which often select the solutions that best optimise a multi- or many-objective optimisation problem, we have applied methods based on dominance to constructing near-total orderings of many-criterion individuals. Having described and illustrated various approaches from the evolutionary optimisation literature, we applied them to the development of a visualisation that requires no rescaling or weighting of criteria. We illustrated the visualisation of various many-criterion populations as a graph in which the individuals are arranged into their Pareto shells; further information is provided by colouring each individual according to its score on a range of ranking methods. We demonstrated that the outflow ranking method is analogous to the average rank method, popular in recent MOEAs, and applied the power index to the multi- and many-criterion domain by developing a probabilistic view of the dominance relation. While these methods were able to visualise multi-criterion and small many-criterion populations, the reliance on Pareto sorting limited their application to populations comprising large numbers of criteria. We address this issue in Chapter 6 where we consider methods for identifying which criteria are the most important for defining the structure of a population so that the remainder can be discarded.

We have also considered a more traditional form of visualising high-dimensional data, compressing the dimensionality of a population with metric MDS. Here, we developed a metric that defines pairwise distances between individuals in terms of their dominance relationships with other members of a population. We demonstrated this technique on a range of populations, both mutually non-dominating and those containing dominated individuals, and illustrated the intuitive fashion in which individuals are organised in the low-dimensional embedding. Since MDS finds a linear approximation, future work will extend this approach to apply dominance distances to methods such as isomap, locally linear embedding, Laplacian eigenmaps and maximum variance unfolding.

Having introduced dominance-based MDS, we will employ it again in Chapter 4. There, we enhance a 2-dimensional embedding of a population by identifying landmarks in the original, high-dimensional, space, and highlight them in the visualisation. Specifically, we develop methods for identifying individuals that lie on the edge of a mutually non-dominating population and show how such information can be useful for data mining and knowledge discovery.

4. Finding the Edge of a Mutually Non-dominating Population

4.1. Introduction

So far in this thesis we have presented methods for visualising many-criterion populations, including methods that are suited to visualising the mutually non-dominating solutions to a multi- or many-criterion optimisation problem. In order to enhance our understanding of such populations, it can be useful to consider landmark individuals, such as individuals that lie on the corner of a population. From the point of view of constructing a visualisation, identifying such individuals can help to understand how the individuals in the population have been represented. For example, visualising a population with a technique such as MDS requires the embedding of individuals in a new coordinate space, with which the decision maker is not familiar; marking those individuals that lie on the edge of the original (M-dimensional) population can help them to navigate the visualisation.

In addition to using landmark individuals to assist a decision maker, the identification of individuals lying on the edge of a mutually non-dominating population is also of interest to the developers of multi-objective evolutionary algorithms. In order to generate an estmiated Pareto front which completely covers the true Pareto front (recall that, along with convergence to the true Pareto front, full coverage of the front is one of the goals of optimising a multi-objective problem) using a fitness strategy that encourages the generation of new solutions from those which form the extent of the current Pareto front is an obvious strategy. Among the algorithms to use this idea are two of the most popular MOEAs, PAES and SPEA2 (both described in Chapter 2).

Recently Singh et al. [2011] proposed a method for finding the corners of a Pareto front. Although we note that this is not strictly the same as the populations of individuals which are the focus of this thesis, their definition provides a useful starting point for the discussion we present in this chapter. According to Singh et al. [2011], a solution lies in the corner of a Pareto front if (assuming a minimisation problem) it minimises all criteria in a $k < M$-criterion subset. Since we are interested in populations of individuals, rather

Some of the material in this chapter has been published as Everson et al. [2013].

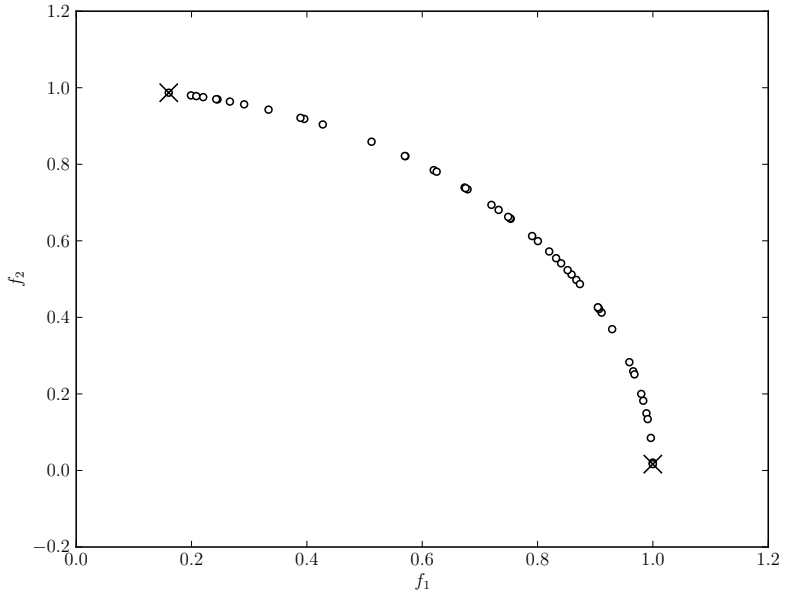


Figure 4.1.: The two individuals which lie on the edge of a mutually non-dominated population comprising two criteria.

than the complete Pareto front of a many-objective optimisation problem,[1] we seek the corners of a set of many-criterion individuals. Given a population $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$ and a set of indices corresponding to some of the M criteria, κ, where $|\kappa| = k$, let $\mathbf{y}_i^{\kappa}$ be the projection of $\mathbf{y}_i$ onto the k criteria specified by κ. In order to find all of the corners in a population it is necessary to consider all of the possible criterion subsets κ. We recall the function $\texttt{nondom}(\mathbf{A})$ which returns the non-dominated members of the population $\mathbf{A}$ and formally state the set of individuals in the corner of a population as:

$$\bigcup_{\kappa} \left(\texttt{nondom}\left(\mathbf{Y}^{\kappa}\right) \ : \ \left|\texttt{nondom}\left(\mathbf{Y}^{\kappa}\right)\right| = 1 \right), \tag{4.1}$$

the union of the non-dominated points for each k-criterion subset, assuming that the k criteria are minimised by a single individual.

An alternative to finding the corners of a mutually non-dominating population is to find the entire edge. We begin by considering the 2-criterion case. Here, the task of identifying the individuals that lie on the edge is intuitively clear. Assuming, without loss of generality, that the two criteria are to be minimised we simply find the individual that minimises each criterion. This is shown graphically in Figure 4.1. Intuitively, it is clear that the two highlighted individuals form the edge of the population, and they can be easily found both by a human decision maker and automatically. That said, we note at this point that well known methods for finding the edge of a cloud of points, such as finding the convex hull [de Berg et al., 2000] are not suitable for this task. In the 2-criterion population shown in Figure 4.1 all of the individuals are on the convex hull.

[1] Although the individuals in a population might be samples from the Pareto front of a many-objective optimisation problem.

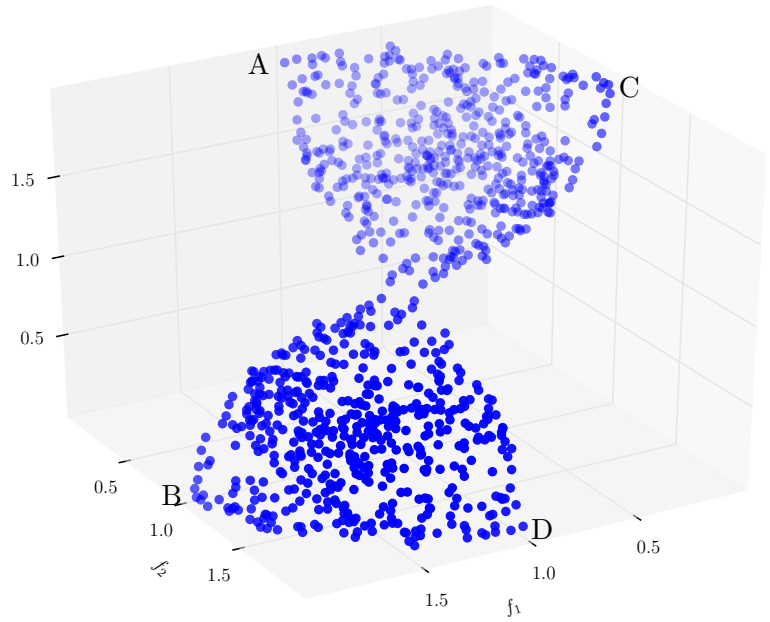


Figure 4.2.: The hourglass population. The corners of the hourglass are marked (A, B, C and D). The radar population is visualised using MDS with the dominance distance.

Unfortunately, the task of identifying the edge in a population comprising three or more criteria is somewhat more challenging, even for a person. Given such a population, defining exactly what is meant by the edge is a difficult task. We therefore seek to formally define when an individual is on the edge, and we provide three candidate definitions. The first relies on the well known definition of the *attainment surface* [Fonseca et al., 2001] while the other two use dimension reduction. All three methods use dominance to identify the individuals on the edge.

We demonstrate the efficacy of the three proposed methods by finding the edge of two example populations. One of them is the radar population introduced in Chapter 3 and the other is an artificial population constructed from a multi-criterion test problem.

The second of the populations, shown in Figure 4.2, is an artificial population constructed with samples from the positive octant of a spherical shell. As described in Chapter 2, Pareto fronts are either *convex*, *non-convex*, *linear*, or a combination of the three; any technique for identifying the edge individuals must be able to operate on all combinations of these geometries. We begin with a convex set of samples. A copy of the samples was taken and inverted so that it was concave rather than convex. Both sets of samples were then relocated so that the top of the convex sample set aligns with the bottom of the concave samples, creating a "pinch point". Figure 4.2 illustrates the hourglass population. The corners of the population have been highlighted (A, B, C and D) for reference later in the chapter.

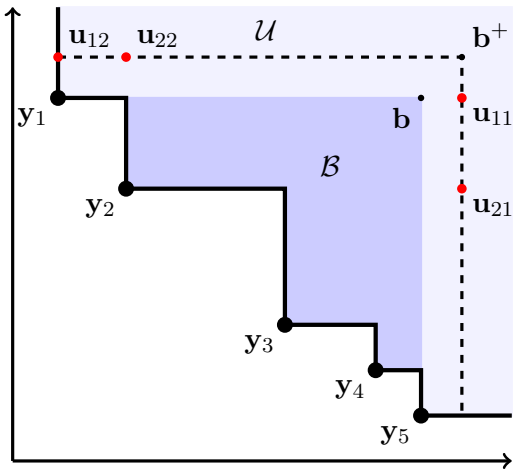


Figure 4.3.: A mutually non-dominating set of points $\mathcal{Y} = \{\mathbf{y}_i\}$ and the region that it dominates $\mathcal{U}$. The region $\mathcal{B}$ is the set of points that dominate the reference point $\mathbf{b}$, whose mth coordinate $\max_{\mathbf{y}' \in \mathcal{Y}} y'_m$. The attainment surface is shown with the thick black line. Candidate points for solo domination by $\mathbf{y}_1$ and $\mathbf{y}_2$ are shown in red.

4.2. Identifying Edges with the Attainment Surface

The basic idea behind the first definition we present uses the attainment surface formed by a mutually non-dominating population to identify individuals which dominate regions of criterion space that are dominated by no other individual. We illustrate this idea graphically in Figure 4.3. It presents a 2-criterion population consisting of five individuals. From inspection, using the intuition presented in the last section, we can observe that the individuals forming the edge of the population are $\mathbf{y}_1$ and $\mathbf{y}_5$.

In order to consider the problem using dominance, we observe that $\mathbf{y}_1$ is the only individual in the population to dominate the region of space formed by $\{ (u, v) \mid y_{11} < u \leq y_{21} \wedge y_{12} < v \}$. We can define a similar region of criterion space such that $\mathbf{y}_5$ is the sole dominator. We can also identify regions of space for which $\mathbf{y}_2$, $\mathbf{y}_3$ and $\mathbf{y}_4$ are the sole dominators. We therefore seek the individuals on the edge by identifying those individuals which contribute to the bounds of the region $\mathcal{B}$ in Figure 4.3 since only an individual lying on the edge can affect the extent of the region $\mathcal{B}$. For example, the addition of a solution above $\mathbf{y}_1$ would lie on the edge of the Pareto front, and would extend the region encompassed by $\mathcal{B}$, however the addition of an individual between $\mathbf{y}_1$ and $\mathbf{y}_2$ would not.

Before we provide a formal definition of an edge according to the intuition outlined above, we first declare some preliminaries. The above intuition behind the proposed definition states that an individual on the edge dominates regions of space, some non-negative distance from the Pareto front, that are dominated by no other individual; the location of this space is defined as follows. To begin with, we define the region dominated by the

population:

$$\mathcal{U} = \{\mathbf{u} \mid \exists \mathbf{y}\ (\mathbf{y} \in \mathbf{Y} \wedge \mathbf{y} \prec \mathbf{u})\}. \tag{4.2}$$

Also, let $\mathbf{b}$ be the point which has the maximum coordinate of any member of $\mathbf{Y}$ in each dimension:

$$b_m = \max_{\mathbf{y}_i \in \mathbf{Y}} y_{im}. \tag{4.3}$$

Then let $\mathcal{B}$ be region the which is dominated by $\mathbf{Y}$ but lies "below" $\mathbf{b}$:

$$\mathcal{B} = \mathcal{U} \cap \{\ \mathbf{y} \mid \mathbf{y} \prec \mathbf{b}\ \}. \tag{4.4}$$

Finally, we define a function that returns all of the members of $\mathbf{Y}$ that weakly dominate a point $\mathbf{x}$:

$$\texttt{doms}(\mathbf{x}, \mathbf{Y}) = \{\ \mathbf{y} \in \mathbf{Y} \mid \mathbf{y} \preceq \mathbf{x}\ \}. \tag{4.5}$$

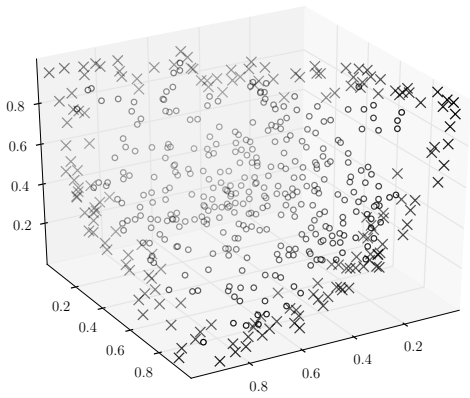
Having established these preliminaries, we say that $\mathbf{y}$ is an edge individual if and only if there exists a $\mathbf{u} \in \mathcal{U} \setminus \mathcal{B}$ such that $\texttt{doms}(\mathbf{u}, \mathbf{Y}) = \{\mathbf{y}\}$. That is, there are points outside of $\mathcal{B}$ (i.e., sufficiently far away from $\mathbf{Y}$ to avoid the inclusion of interior individuals) that are dominated by $\mathbf{y}$ alone.

We observe that any point which is a corner according to Singh et al. [2011] is on the edge. Recall, that in order for an individual to form a corner it must minimise all criteria in a k-criterion subset. Thus, it will dominate space that is dominated by no other member of $\mathbf{Y}$ on that criterion (or criteria) and is therefore part of the edge.

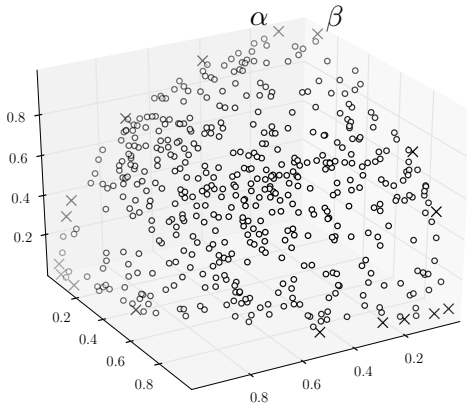
In order to identify which of the individuals in a population are on the edge, according to this population, we consider that an edge individual must dominate regions of criterion space (far enough away from the attainment surface) that are dominated by no other individual in the population. One approach might be to draw samples from the hypercube defined by the point $\mathbf{b}^+$ and the coordinate axes. We pick a hyper-rectangle to project onto, defined in terms of a point $\mathbf{b}^+$ that is definitely in $\mathcal{U} \setminus \mathcal{B}$:

$$b_m^+ = b_m + \epsilon \qquad\qquad \epsilon > 0 \tag{4.6}$$

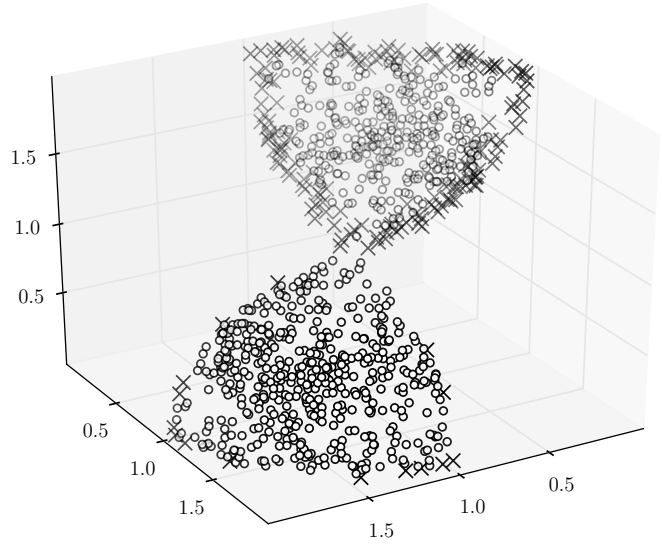
Assuming that it is positive, the value of ϵ is immaterial. A problem with this is that since the edge is defined in terms of weak dominance we can use the point $\mathbf{b}^+$ in conjunction with the individuals in the population themselves to construct a set of candidate points $\{\mathbf{u}\}$ against which we can test each individual. We extend each member of $\mathbf{Y}$ along the coordinate axes into $\mathcal{U} \setminus \mathcal{B}$ so that we can evaluate how many individuals in the population dominate each $\mathbf{u}$. Having defined the hyper-rectangle along which the individuals will be



(a) Concave population



(b) Convex population



(c) Hourglass

Figure 4.4.: The edge of the hourglass population as identified by the attainment surface method. The method has identified the top half of the edge well, but has struggled to identify the bottom half, finding only a few edge individuals.

placed we create the M candidate points for $\mathbf{y}_i$ as:

$$u_{in} = \begin{cases} y_{in} & n \neq m \\ b_m^+ & n = m \end{cases} \qquad m = 1, \ldots, M. \tag{4.7}$$

By construction $\mathbf{y}_i \preceq \mathbf{u}_{im}$, but if any other $\mathbf{y} \in \mathbf{Y}$ also dominates $\mathbf{u}_{im}$ then $\mathbf{y}_i$ does not lie on the edge of the population. In Figure 4.3, the candidate point $\mathbf{u}_{11}$ is dominated by both $\mathbf{y}_1$ and $\mathbf{y}_2$. On the other hand, $\mathbf{u}_{12}$ is dominated only by $\mathbf{y}_1$. Hence, $\mathbf{y}_1$ is a point which lies on the edge.

Figure 4.4 presents three examples of edge identification using the attainment surface. The

first example (Figure 4.4(a)) is of a concave population, and the method has identified a reasonable approximation of the edge. The second example is of a convex population (Figure 4.4(b)) and the method has found a much less complete set of edge individuals when compared with the concave example. The reason for this is explained by considering the two edge individuals at the top of the population (α and β). In terms of the first two criteria, f_1 and f_2, α and β are near-optimal. It is therefore unlikely that another individual will dominate regions with respect to f_1 and f_2 that are not also dominated by α and β, meaning that few of the individuals in the convex population are on the edge. The final example shows the edge of the hourglass according to the attainment surface definition. There, we again see that the edge of the concave region has been well identified while there are large gaps in the edge of the convex region.

Applying the attainment surface approach to finding the edge of the population of individuals to the radar problem proves to be less successful. This is to be expected, as it is a side effect of the inability of dominance to discriminate between individuals in a high-dimensional space. We find that all but two of the 2000 individuals in the population are on the edge in the 9-criterion space; we do not present an illustration of this result.

4.3. Dominance-based Edge Identification with Rotations

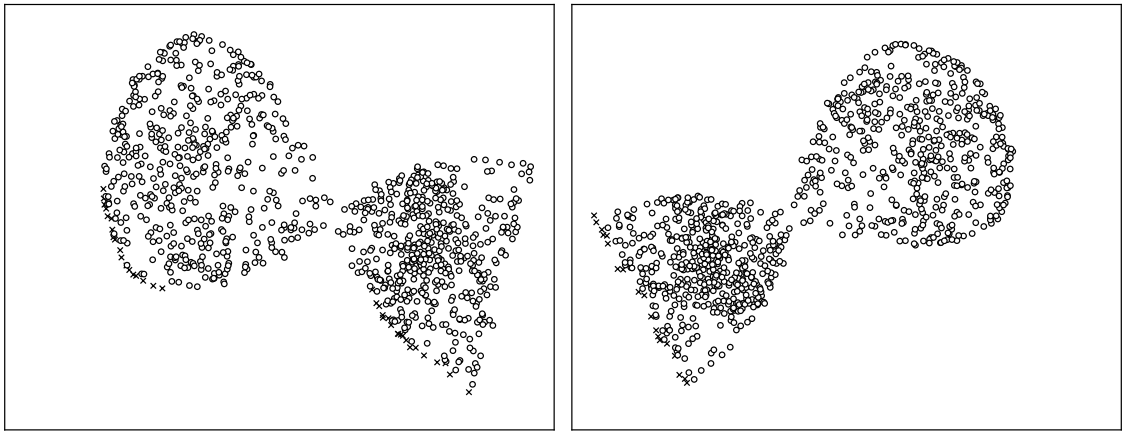
Given a population of mutually non-dominating individuals, we can project them into the plane by embedding the population in the M-dimensional simplex. In the resulting (M-1)-dimensional representation of the population, a region of the edge can be found by identifying those individuals which are non-dominated (in the planar representation, with respect to the rest of the embedded population). Rotating the planar embedding causes other regions of the population to become non-dominated, exposing other stretches of the edge. Given a sufficient number of rotations, it is therefore possible to identify all of the individuals that lie on the edge of the population, and this section presents a method for finding the edge in this manner.

In order to arrive at a planar representation of a mutually non-dominating population, ensuring that the individuals remain mutually non-dominating, we project the individuals onto the flat simplex. The simplex is the region of the plane in the positive orthant defined by

$$\mathbf{n} \cdot \mathbf{y} = d \qquad\qquad y_m \geq 0,\ m = 1, \ldots, M \tag{4.8}$$

where the elements of the unit vector $\mathbf{n}$ normal to the simplex are $n_m = d/\lambda_m$ and the perpendicular distance to the origin d can be found as

$$d^{-2} = \sum_{m=1}^{M} \lambda_m^{-2}. \tag{4.9}$$



(a) (b)

Figure 4.5.: Two random rotations of the hourglass population projected onto the principal $M-1$ components.

We project the mutually non-dominating population onto the simplex by

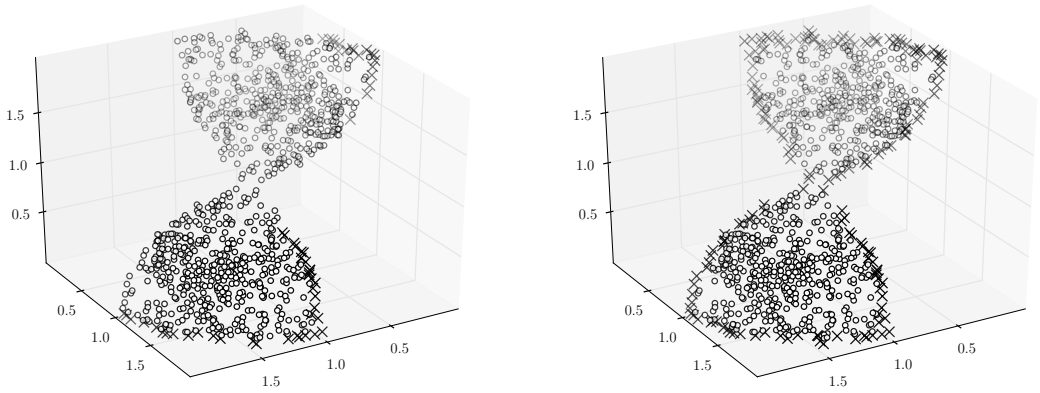
$$\hat{\mathbf{y}}_i = \mathbf{y}_i/(\mathbf{y}_i \cdot \mathbf{n}). \tag{4.10}$$

Importantly, we note that a mutually non-dominating population remains mutually non-dominating following projection onto the simplex.

Having projected the population onto the simplex we must find an initial direction on which to base the rotations of the individuals. We observe that the population now comprises a single "direction", in the principal component sense. If we were to project the simplex projections of the individuals onto the $M-1$ principal components, 1 in this case, we would be able to simply find the edge individuals by examining which are non-dominated, given sufficient rotations to expose both end-points.

This idea can be extended into the 3-criterion domain of the hourglass population; consider the example shown in Figure 4.5(a). There, the hourglass population has been projected onto the simplex, and from there onto the first two $(M-1)$ principal components. If we find the non-dominated individuals- we find some of the individuals around the edge. Those individuals are marked with a cross in the figure. A possible rotation of the hourglass is shown in Figure 4.6(b). In that version of the population, different individuals are non-dominated, and thus more of the edge has been revealed.

In order to discover the entire edge, we use an iterative procedure in which the principal component population $\hat{\mathbf{Y}}$ is rotated by a random amount repeatedly. To obtain a random rotation matrix we begin by generating a $(M-1)\times(M-1)$-dimensional matrix of Gaussian random values. From this, we take the QR-decomposition, which yields two matrices. One of these matrices, which we denote by $\mathbf{Q}$, is a rotation matrix with which we can apply a



(a) 8 rotations

(b) 256 rotations

Figure 4.6.: The edge of the hourglass as identified by the rotated dominance method. Clearly (for this population) 4 rotations are insufficient, however 256 rotations has resulted in a near-complete set of edge individuals.

random rotation to the members of $\hat{\mathbf{Y}}$ so that each $\hat{\mathbf{y}}_i$ is rotated to the position $\hat{\mathbf{y}}'_i$:

$$\hat{\mathbf{y}}'_i = \mathbf{Q}\hat{\mathbf{y}}^T. \tag{4.11}$$

We then find the non-dominated members of the rotated projection $\hat{\mathbf{Y}}'$ and then rotate the population again. Hence, we define the set of edge individuals as being those individuals for which there is at least one rotation matrix $\mathbf{Q}$ that causes the individuals planar projection to be mutually non-dominating:

$$\{\ \mathbf{y} \in \mathbf{Y} \mid \exists \mathbf{Q}(\ \hat{\mathbf{y}}' = \mathbf{Q}\hat{\mathbf{y}}^T \wedge\ \mathtt{doms}_{\hat{\mathbf{Y}}'}(\hat{\mathbf{y}}') = 0\ )\ \}. \tag{4.12}$$

The rotated dominance method is demonstrated for the hourglass population in Figure 4.6. We demonstrate the technique twice, once with 4 rotations and once with 256. While it is clear that the 4 rotation instance does not fully reveal the entire edge, the 256 rotation version does. We note that the process of rotating the population and finding the non-dominated individuals is relatively cheap and therefore the entire process can be quickly run for a large number of rotations in order to ensure that the whole edge is included. Comparing the hourglass edge in Figure 4.6(b) with that shown in Figure 4.4 we can see that the rotation method has improved upon the edge discovered by the attainment surface method. The method has discovered the entire edge of the population, rather than missing extensive regions of the convex portion of the population. Also, it finds fewer interior individuals than the attainment surface approach, and as such it provides a more exact impression of the population edge. That said, the quality of the edge identified by this method is dependent on the fidelity of the projection onto the $M-1$ principal components. For this reason, we cannot state that the corner individuals (as defined by Singh et al. [2011]) are incorporated as we did for the attainment surface method, since

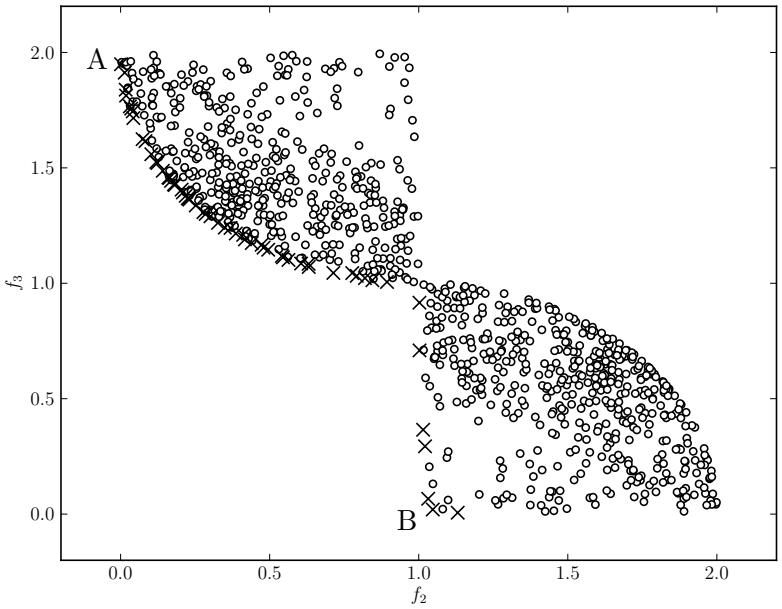


Figure 4.7.: The hourglass population projected onto criteria f_2 and f_3. The edge between points A and B in Figure 4.2 marked with crosses.

it is possible that they will not be on the edge of the population projection. If this is the case, it is possible for the corners to be excluded from the edge identified by this approach.

As with the attainment surface method we applied this method to the task of identifying the edge of the 9-criterion radar population. Unsurprisingly, since this method also relies on dominance in a high-dimensional space, it also identifies all of the individuals as being on the edge.

4.4. Criterion Subset Edge Identification

In the work by Singh et al. [2011] on identifying the corners of a mutually non-dominating population, the authors considered the dominance relations in k-criterion subsets. Returning to the 2-criterion population introduced at the beginning of this chapter, we note that the same intuition can be applied to identifying the edge of such a population. Recall, that the two individuals that minimise the criteria are those which form the edge in a 2-criterion population (shown with crosses in Figure 4.1). Thus, we can say that one of the individuals (for example, the one which minimises the first criterion f_1) is non-dominated in a k-criterion subset comprising f_1 alone.

This approach can be used to identify the individuals on the edge of the hourglass. To do this we consider a k-criterion version of the population, where $k = (M - 1)$. If, for example, we discard f_1, the remaining criteria form the projection shown in Figure 4.7.

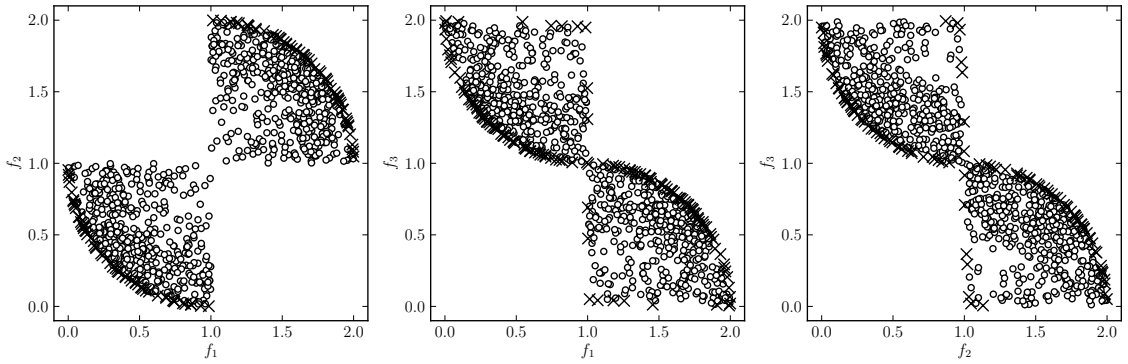


Figure 4.8.: The criterion subset method applied to the hourglass population. In each 2-criterion subset, the maximising and minimising individuals are found and added to the edge set E.

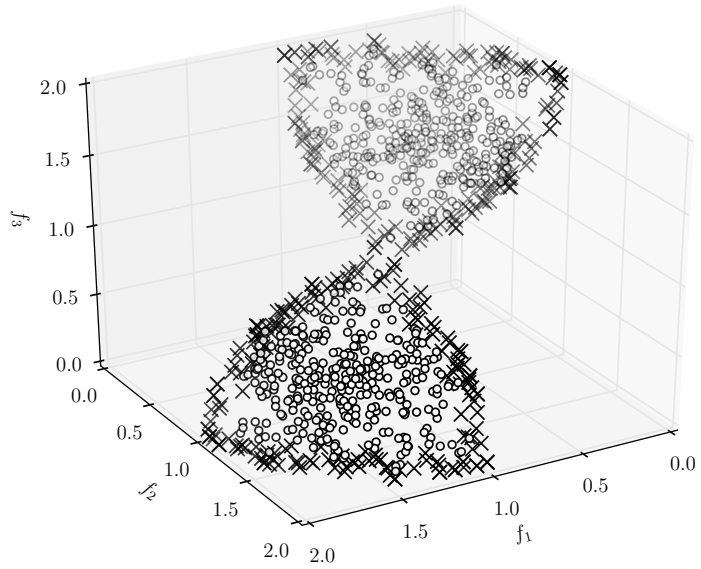


Figure 4.9.: The hourglass population. Individuals identified by the criterion subset method as residing on the edge of the population have been marked with crosses.

Referring to the labels in Figure 4.2 we can see that the individuals on edge between the corners A and B are those which are non-dominated (with respect to minimisation) in this projection. Finding the non-dominated (in the maximisation sense) individuals identifies those which are on the edge C—D. In fact, by examining different projections of the individuals $\mathbf{y}^\kappa \in \mathbf{Y}^\kappa$, where $\mathbf{y}^\kappa$ is the individual $\mathbf{y}$ as represented by the criterion subset κ, we can recover the entire edge. The full process for identifying the edge of the hourglass is shown in Figure 4.8. The three panels show the population projected into the $M = 3$ possible criterion subsets ($\{f_2, f_3\}, \{f_1, f_3\}$ and $\{f_1, f_2\}$). Each of the individuals marked with a cross lies on the edge. Once each of the three 2-criterion subsets has been considered, the edge comprises each of the individuals that were on the edge in any of the subsets. This set therefore comprises the entire edge of the 3-criterion population, and they are shown in Figure 4.9; the individuals on the edge are again shown with crosses.

Size of k	Number of individuals on edge
1	51
2	583
3	1842
4	1997
5	2000
6	2000
7	2000
8	2000

Table 4.1.: The number of individuals which lie on the edge of the population of 2000 solutions from the radar archive for criterion subsets k of size $1, \ldots, (M-1)$.

From this we can define the set of edge individuals; we introduce a new function $\texttt{domsmax}(\mathbf{y}, \mathbf{Y})$ which returns the individuals in the population $\mathbf{Y}$ that dominate the individual $\mathbf{y}$ when the criteria are to be *maximised*, noting that `doms` is defined in terms of minimisation, and write the set of edge individuals as follows:

$$\{\ \mathbf{y} \mid \kappa(\ \mathbf{y}^\kappa\ \in\ \texttt{doms}(\mathbf{y}^\kappa, \mathbf{Y}^\kappa) \cup \texttt{domsmax}(\mathbf{y}^\kappa, \mathbf{Y}^\kappa))\ \} \tag{4.13}$$

As we have observed above, it is important that corner individuals are present in the edge. According to the definition provided by Singh et al. [2011] an individual forms a corner of the population if it is the only non-dominated individual according to some k-criterion subset. Given that, as long as at least one of the criteria in that k-criterion subset is present in a other criterion subsets, that individual will be non-dominated and will be on the edge according to this definition.

Unlike the previous two methods, this method can be applied to a many-criterion population such as the radar population and we demonstrate its application to finding the edge of a population comprising 2000 samples from the solution set to the 9-objective radar waveform problem [Hughes, 2007a]. Table 4.1 presents the number of times that an individual is found to be on the edge.[2] Clearly, the number of individuals quickly rises with the dimensionality of the k-criterion subset and beyond 3-criterion subsets all of the individuals are on the edge. As such, we propose a slight modification; since a large proportion of the individuals in the population will be present in at least one criterion subset we consider the frequency that individuals occur. An individual which is frequently found to be non-dominated in a criterion subset is more likely to be a part of the edge.

Figure 4.10 illustrates the 2000 solutions from the radar population by reducing their dimensionality with dominance distance MDS as presented in Chapter 3. Unfortunately, the density of the points in this visualisation makes identifying regions that are part of the edge difficult. To ameliorate this, we modify the visualisation slightly, which is shown in

[2]Some of the objectives contain tied values, hence the total number of maximising and minimising individuals found with criterion subsets of size 1 is not $2 \times M$.

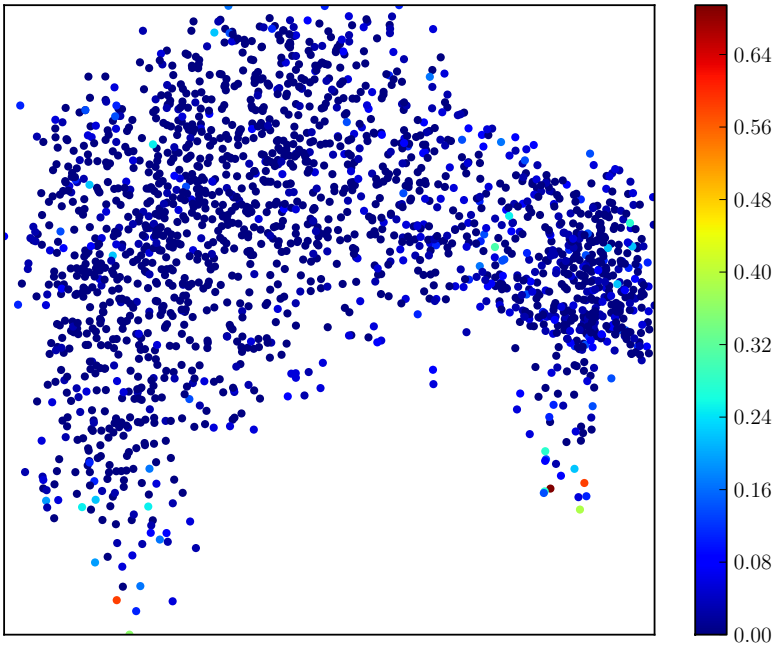
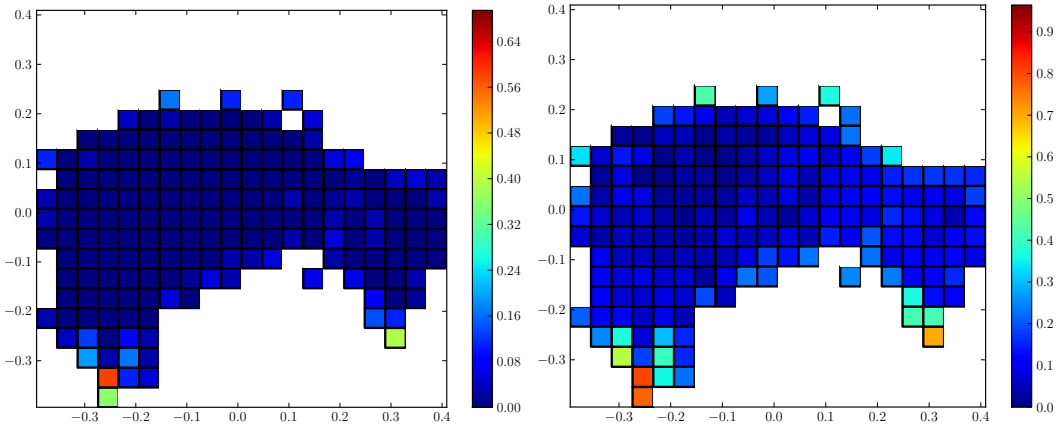


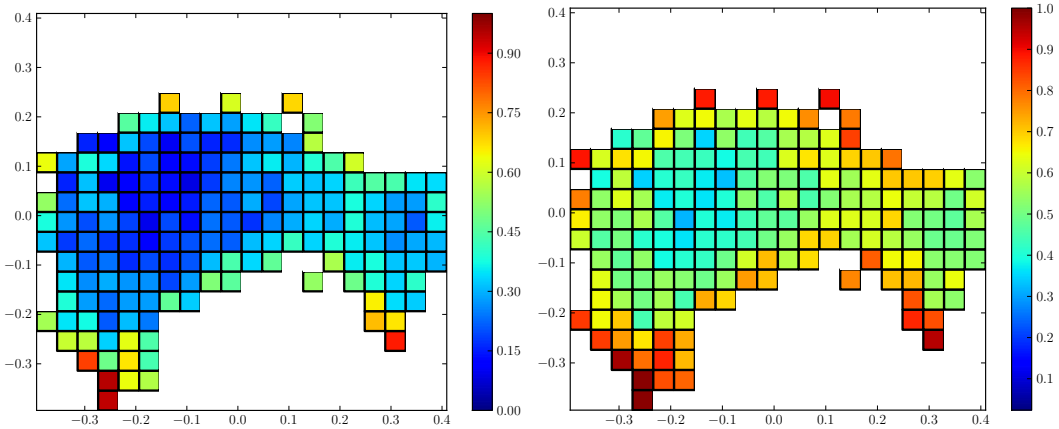
Figure 4.10.: A dominance distance MDS embedding of the 2000 radar solutions. The individuals are coloured according to the frequency that they occur a 2-criterion subset of the population. The density of the points in the visualisation makes identifying particular regions as being on the edge difficult.

Figure 4.11. In order to enhance the clarity of the presentation we place a grid of cells over embedding space. Each cell is coloured according to the median edge frequency within it such that a red individual is present in a large number of criterion subsets. Since we know how many possible 2-criterion subsets there are, $\binom{M}{k}$, we divide by this number to normalise the frequency

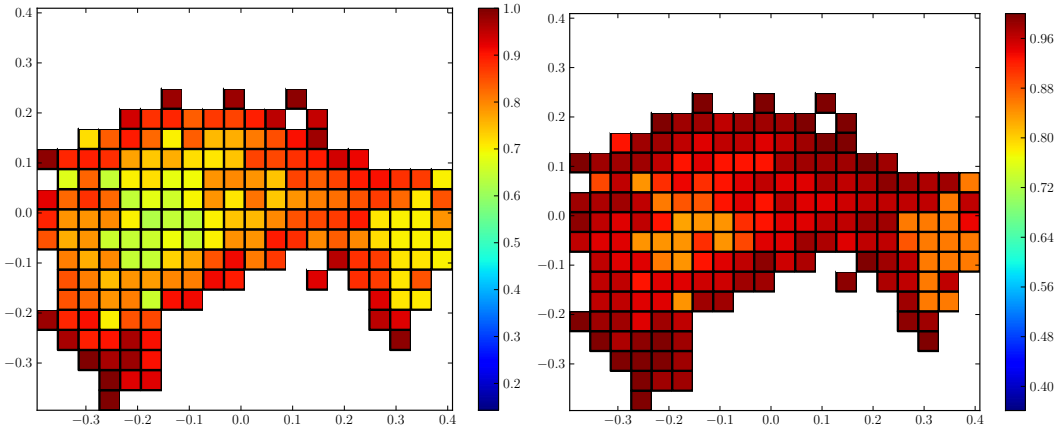
Results are shown for $k = 4, 5, 6$ and 7. We can observe that the cells around the edge of the embedding are generally coloured with lighter shades than those in the centre. This means that the MDS projection has placed those individuals that are more frequently found in D-criterion subsets on the edge of the embedding, as we would hope. The lightest colours are at the ends of the "crescent" shape. An additional feature of interest is the appearance of an edge that bisects the population from top to bottom towards the right-hand third of the embedding. It is perhaps less intuitive to expect an edge in this position, however upon investigation the reason becomes clear. The criteria in the radar population divide into three groups. One group (f_1, f_3, f_5 and f_7) relate to the range at which a radar can discern targets. The second group (f_2, f_4, f_6 and f_8) relates to the velocity at which the radar can be moving and still discern targets, while the final group (f_9) contains a single criterion which minimises the transmission time of the radar waveform. Figure 4.12 presents the MDS embedding of the 2000 radar individuals; the criterion that a given individual best optimises has been marked with a symbol. If an individual best optimises a range criterion, that individual is shown with a plus symbol. If it best optimises a velocity criterion, then it is shown with a circle, and if it best optimises



(a) $k = 2$ (b) $k = 3$



(c) $k = 4$ (d) $k = 5$



(e) $k = 6$ (f) $k = 7$

Figure 4.11.: Cellular visualisations of the radar population embedded into 2 dimensions with dominance distance MDS. The shade of each cell indicates the median number of times that the individuals within that cell are on the edge in a k-criterion subset. We show results for $k = 2, \ldots, 7$, which show that in general individuals on the edge of the population are arranged around the edge of the embedding, with the exception of those which bisect the population toward the right-hand side of the plot.

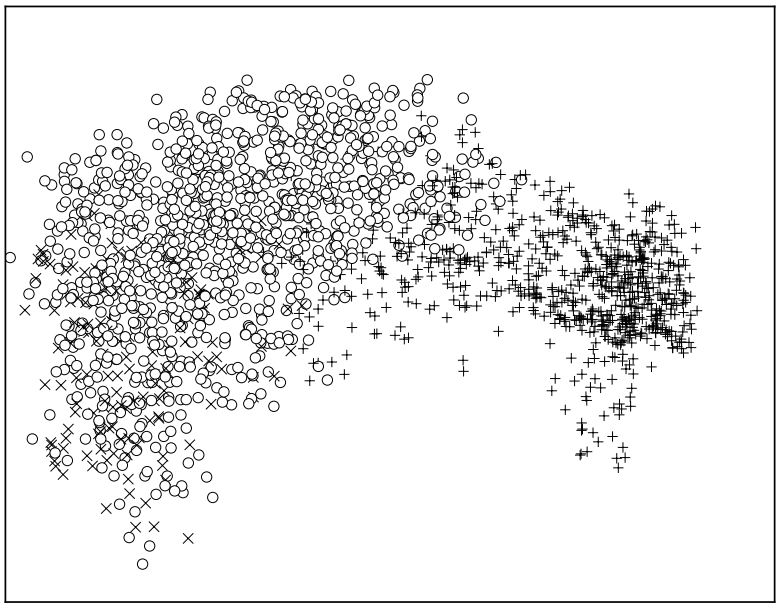


Figure 4.12.: The MDS embedding of the radar individuals in which the type of criterion best optimised by an individual is represented by its symbol. individuals which best optimise a range criterion are shown by a plus symbol; velocity criteria are shown by circles and the transmission time criterion by crosses. There is a distinct boundary between those individuals which best optimise range criteria and those which best optimise velocity criteria or transmission time. This boundary is in the same position as the edge shown in Figure 4.11.

the waveform transmission time it is shown with a cross. Although there is some degree of overlap, the individuals have been embedded in such a way that they are organised by criterion class. The range individuals are mostly located in the right-hand third of the embedding and do not generally overlap the velocity criteria. The boundary between the two is in the same position as the edge shown in Figure 4.11. The separation between the velocity and transmission time individuals is much less clear. These two types of criteria are well correlated, as we will discuss further in Chapter 5.

4.5. Conclusion

Identifying landmark individuals in a population allows us to explore and understand its structure. In this chapter we have investigated techniques for defining which of the individuals in a mutually non-dominating population are on the edge. We have provided three definitions for such individuals. The first used the attainment surface to identify the individuals on the edge, and it was shown to be better suited to finding the edge of a concave population than a convex one. The second projected the population onto the simplex and then rotated a principal component embedding of the individuals to find those which were non-dominated. The final method used 2-criterion subsets of the individuals to

find non-dominating individuals; in addition to finding the edge of an artificial 3-criterion population this method was shown to be capable of identifying the edge of a population of individuals to a real-world many-criterion optimisation problem. This in turn revealed useful information about the layout of the individuals in a dominance distance MDS visualisation.

This chapter has demonstrated methods that work well in the multi-criterion domain, and as shown here we have begun to investigate the extension of these methods into the many-criterion domain. Given the prevalence of many-criterion populations, future work in this area will address the potential to extend the attainment surface and rotation methods to cope with many-criterion populations, as well as the search for other techniques for characterising individuals on the edge of a mutually non-dominating population.

5. Seriation of Heatmaps

5.1. Introduction

A common approach to visualising high-dimensional data is to use a heatmap (e.g., Eisen et al. [1998]; Wilkinson and Friendly [2009]). Heatmaps represent values as relative temperatures in a grid, where the temperature of a particular cell is shown with colour. Several recent works have used heatmaps for visualising solution sets to multi- and many-objective problems; to visualise such a population, rows are used to represent individuals and columns to represent criteria. Pryke et al. [2006] presented early work in this area, visualising solution sets to an instance of the test problem DTLZ2 [Deb et al., 2002] as well as a real world bi-objective mineralogy problem. They observed that as well as being applicable to any number of solutions or objectives heatmaps provide a means to visualise parameter space solutions alongside their objective space counterparts. Nazemi et al. [2008] used a heatmap to visualise solutions to a 5-objective parameter calibration problem for a rainfall-runoff model and Biswas et al. [2009] visualised the results of optimising a 3-objective traffic management problem with a heatmap to demonstrate the trade-off between objectives. A heatmap was subsequently incorporated as a decision making tool in an interactive multi-objective particle swarm optimisation algorithm [Hettenhausen et al., 2010]. Most recently, Kiesling et al. [2011] discuss the use of heatmap visualisation within a decision support system to enable the selection of a portfolio based on multiple criteria.

An example of a heatmap is shown in Figure 5.1 in which the individuals in the GUG09 dataset are visualised. Recall that each of the $N = 113$ individuals $\mathbf{y} \in \{\mathbf{y}_i\}_{i=1}^{N}$ describes the performance of a UK university on eight performance indicators; a university achieves the best possible performance on these indicators by minimising its score on each of them. The universities are ordered arbitrarily, alphabetically in this case. Rather than visualise the objective values y_{im} themselves, we rank the individuals on each criterion to obtain r_{im}, the rank of the ith individual on the mth criterion. We denote the vector of ranks for an individual $\mathbf{y}_i$ by $\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{iM})$. This ranking has two principal benefits. First, the values to be displayed by assigning a colour should be on a common scale, as noted by Pryke et al. [2006]. Ranking the individuals brings them onto the same range, $1 \leq r_{im} \leq N$, without altering the dominance relations between individuals, so that $\mathbf{y}_i \prec \mathbf{y}_j$ if and only if $\mathbf{r}_i \prec \mathbf{r}_j$. Second, providing there are no tied ranks, each ranked value occurs exactly

Some of the material in this chapter has been published as Walker et al. [2010a, 2012b,a].

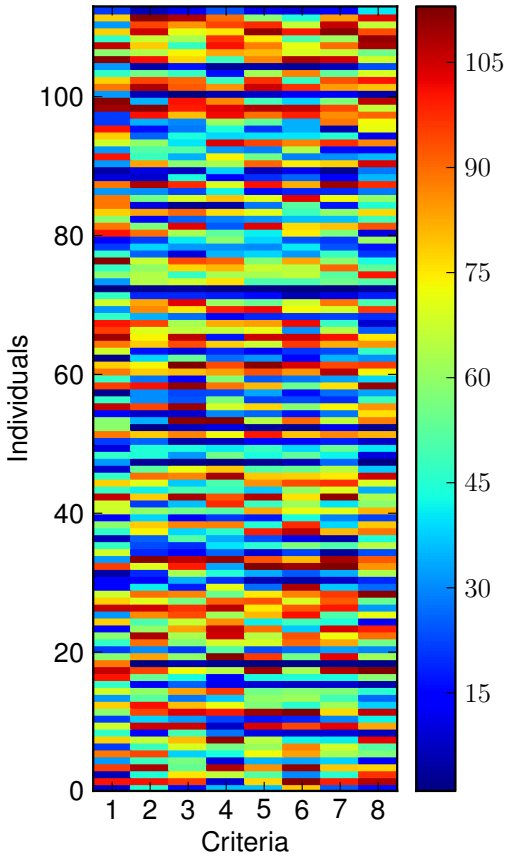


Figure 5.1.: A heatmap of the GUG09 population. Each row represents a university in the data and each column is a criterion. The value indicated by the colour of a cell is the rank of an individual on a criterion.

M times (once for each criterion), so each colour in the heatmap occurs M times, thus using the full range of colours equally. This is equivalent to histogram equalisation (e.g., [Gonzalez and Woods, 2007]) for each criterion and avoids the problem with linear scaling of objectives (i.e., that detail can be lost because a few large values force the majority of the heatmap to be displayed in cooler colours). If ties are present, the tied individuals are assigned the average of their ranks had a distinct ordinal rank been assigned to each individual. Using ranks has the disadvantage that the ranks must be recomputed if the population changes. However, the cost of recomputation is small and in this thesis we confine our deliberations to static populations.

Heatmaps are a particularly useful method for visualising many-criterion populations because they can be used to illustrate the trade-off between criteria, providing important information to a decision maker. It is not possible to see the trade-off between the criteria in the GUG09 population shown in Figure 5.1, however we demonstrate methods for exposing the conflict between criteria later in this chapter. Their scalability, both in terms of individuals and criteria, means that they can visualise large populations of individuals whose performance is described by a large set of criteria. In addition, this information

is available without having to compress or discard criteria, meaning that no information is lost in the visualisation process and the original data is recoverable. Techniques mentioned in previous chapters, such as MDS, do not offer this feature since they compress the dimensionality of the population discarding potentially useful information.

Whilst heatmaps can convey useful information, we identify two problems: the presence of one or two large criterion values in the data means that the full range of colours may not be used. We have already discussed an approach which addresses this by visualising the individuals in a population in rank coordinates. More seriously, the arbitrary ordering of the individuals and criteria in a heatmap hampers its interpretability. In Figure 5.1 the individuals and criteria are unordered and the heatmap is generally unclear because of the cluttered colours, making it difficult to compare individuals. Methods for reordering a heatmap to enhance the clarity of the visualisation have been employed in the literature [Pryke et al., 2006; Nazemi et al., 2008], which are dicussed later in this chapter. *Seriation* is a method which has been used to make heatmaps easier to interpret [Wilkinson and Friendly, 2009] and in this chapter we apply *spectral seriation* to many-criteria populations in order to enhance their clarity. We show how the ranking of individual criteria prior to seriation allows for the use of rank-based correlation measures, such as Spearman's footrule and Kendall's τ metric, and extend the work of Pryke et al. [2006] by seriating both parameter and objective spaces together. We formulate this as a multi-objective problem in which we aim to place individuals together that are similar in both parameter and objective space. We solve this problem by producing a set of solutions which approximate the trade-off between the two objectives, and use a MOEA to refine the resulting permutations.

5.2. Seriation of Heatmaps

The goal of *seriation* is to construct a permutation over the individuals in a population such that similar individuals are placed close together, and dissimilar individuals far apart. Seriation has a long history with early uses in archaeology to establish a chronological ordering of artefacts. An early example of its use was by Robinson [1951] who use seriation to construct a permutation over a collection of pottery-based artefacts. Pottery can be classified by types, and the prevalence of these types varies over time. Since the type of pottery in an artefact can be quantified, a similarity was defined in terms of the relative pottery type found in pairs of artefacts. This is based on the notion that two artefacts from a similar chronological period will be of a similar type while those from different periods will not. The resulting matrix of similarities was then reordered in order to ensure that a pattern of increasing similarity toward the diagonal entry of a row, and decreasing similarity from the diagonal to the right-hand edge of the row, is followed as closely as possible. This produces a chronological ordering of artefacts given that artefacts produced in similar periods will have similar features and therefore appear closer in the ordering than those with dissimilar features, which are placed far apart. Seriation has also been

used in sociology for grouping similar people together [Forsyth and Katz, 1946]. There, the measure of similarity between the individuals participating in the sociological study relates to their attitude to the other participants, either a positive attitude, a negative attitude or an ambivalent attitude. The rows and columns of their similarity matrix are then reordered to produce *sub-groups* of individuals with largely positive attitudes toward the other members of the sub-group so that a permutation of the participants, in which those with similar attitudes are close, is produced. An important advance was made by Atkins et al. [1998], who introduced a spectral method for finding an approximate solution to the seriation problem; we discuss this method in detail shortly. For an extensive historical review of seriation applications see e.g. Liiv [2010].

Since the adoption of heatmaps as a visualisation tool in the many-objective optimisation literature, efforts have been made to improve their clarity. To our knowledge, the first study of the efficacy of heatmaps at visualising many-objective populations was presented by Pryke et al. [2006]. That work illustrated the objective vectors of a population of solutions to a many-objective problem without requiring the compression of the objective vectors and losing information about the original objective values. In addition, the visualisation they presented also integrated the parameter space component of a solution into the heatmap. A monotonic normalisation of parameter and objective values was used to place all parameters and objectives on the same scale. It is observed in this work that the resulting heatmap requires reordering in order to be of use to a decision maker, and the authors employ hierarchical clustering to clarify the presentation of solutions. Specifically, they use single-linkage clustering to cluster solutions (rows) based on the Euclidean distance between their normalised objective values. Having done this, the visualisation is further improved by reordering the columns. Similar parameters and objectives are placed close together with a second single-linkage clustering. This has the effect of placing parameters in amongst objectives, which we feel is undesirable because it makes the observation of the trade-off between criteria, one of the useful features of a heatmap visualisation, more difficult. Another case in which a many-criterion heatmap was reordered was presented by Nazemi et al. [2008]. They used a heatmap to visualise the objective vectors corresponding to solutions to a many-objective problem and reordered the heatmap based on the simple strategy of ordering the objective vectors according to their value on the first criterion. This is a sensible scheme if their criteria are either well correlated or anti-correlated. In either case, the objective values and trade-off between criteria will be well illustrated. If, however, there is little correlation between criteria, then the reordering will be ineffective.

In this chapter we use spectral seriation to reorder the set of many-criterion individuals in the GUG09 and access point populations, as well as non-dominated solution archives to many-objective optimisation problems. In a many-objective optimisation problem, an individual comprises a parameter space and objective space component, and we apply seriation in both spaces; the aim is to place similar individuals, criteria and parameters close together in the heatmap, thus visualising trends and exceptions from the trends

present in the population.

5.2.1. Spectral Seriation of Many-objective Populations

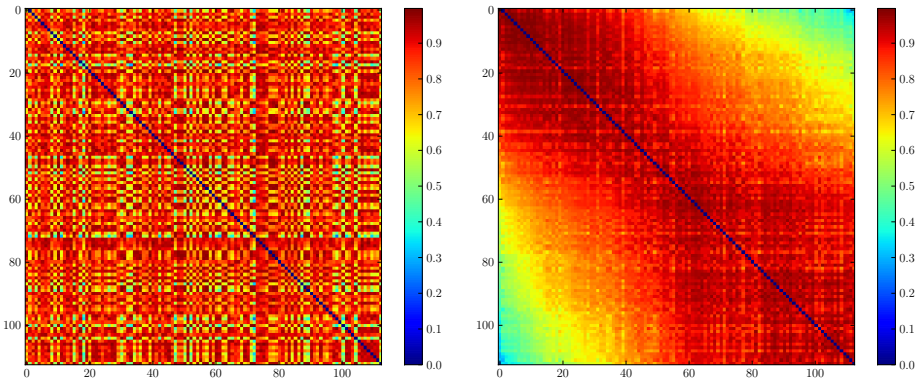
Seriation aims to find an ordering of individuals such that similar individuals are placed close together and dissimilar individuals far apart. Given a permutation $\boldsymbol{\pi}$ over the individuals and a symmetric similarity matrix $\mathbf{A}$ in which $A_{ij} = A_{ji}$ is the similarity between individuals $\mathbf{y}_i$ and $\mathbf{y}_j$, we seek to minimise

$$g_{\mathbf{y}}(\boldsymbol{\pi}) = \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} \left(\pi_i - \pi_j\right)^2 . \tag{5.1}$$

The objective function $g_{\mathbf{y}}$ is minimised when similar individuals are placed close together, and dissimilar individuals far apart. In this formulation the problem is NP-hard; there are $N!/2$ possible permutations of individuals and it is too expensive to evaluate the quality of all of them for even relatively small values of N. To resolve this difficulty, Atkins et al. [1998] propose a spectral method in which the first step is to relax the permutation to a continuous variable $\mathbf{v}$ and instead minimise:

$$h(\mathbf{v}) = \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} \left(v_i - v_j\right)^2 . \tag{5.2}$$

This relaxed definition is subject to two constraints. Firstly, to ensure that adding a constant to all v_i does not introduce multiple equivalent solutions to the seriation problem, $\sum_i v_i = 0$ must be satisfied. Assume that c is a constant value added to the v_i. Then, given Equation 5.1, $g_{\mathbf{y}}(\boldsymbol{\pi}) = \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} \left((\pi_i + c) - (\pi_j - c)\right)^2$. Since the two additional constants cancel, the permutation with the added c is equivalent to the permutation without it, hence a second (equivalent) solution to the seriation problem has been found. The second constraint avoids the trivial solution in which all $v_n = 0$ by requiring the satisfaction of $\sum_i v_i^2 = 1$. The solution to this constrained problem can be found with linear algebra. The problem is rewritten as $h(\mathbf{v}) = \mathbf{v}^T \mathbf{L} \mathbf{v}$, where $\mathbf{L}$ is the *graph Laplacian* of $\mathbf{A}$ defined as $\mathbf{L} = \mathbf{A} - \mathbf{D}$, and $\mathbf{D}$ is the degree matrix, a diagonal matrix whose elements are $D_{ii} = \sum_j A_{ij}$ [Fiedler, 1973; Mohar, 1991; Chung, 1997; von Luxburg, 2007]. The graph Laplacian is a positive semidefinite matrix; and the smallest eigenvalue is 0 with a corresponding eigenvector proportional to a vector of ones. A discrete permutation is recovered from the *Fiedler vector* [Fiedler, 1973], the eigenvector corresponding to the first non-zero eigenvalue of $\mathbf{L}$, by ordering the individuals such that the individuals with the nth smallest value in the Fiedler vector occupies the nth position in the permutation. At first sight locating the Fiedler vector requires a full eigendecomposition of $\mathbf{L}$, however, the Fiedler vector may be identified as the eigenvector corresponding to the largest eigenvalue of the complementary graph Laplacian matrix, which may be found by the power method [Kaveh and Rahimi Bondarabady, 2000]. In any case for the applications addressed here



(a) Original individual similarity (b) Seriated individual similarity

Figure 5.2.: The similarity matrix **A** and permuted similarity matrix for the universities in the GUG09 data during the seriation process. The right-hand matrix is a permutation of the left-hand similarity matrix according to the ordering produced by seriating **A**. Seriating collects the dark red (very similar) pairs of universities so that they lie along the diagonal of the matrix and pushes the blue (very dissimilar) pairs of universities to the edges.

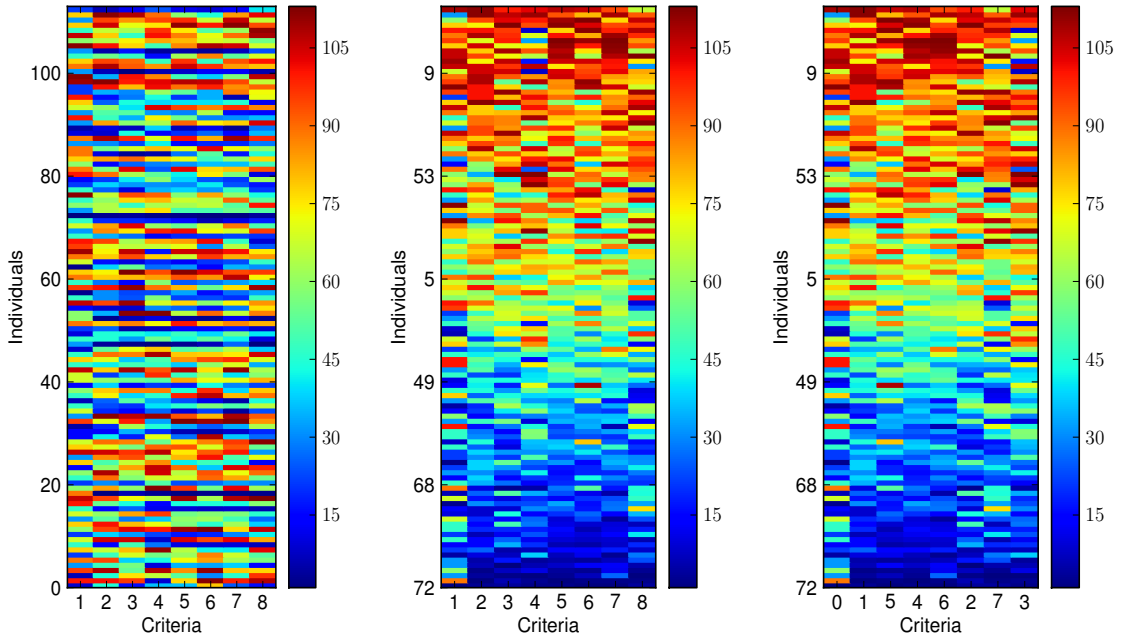
the matrix decomposition is computationally inexpensive (at worst $O(N^3)$) and very much faster than exhaustive search whose $O(N!)$ computational complexity renders it infeasible for $N \gtrapprox 10$.

This method requires a symmetric $N \times N$ similarity matrix **A**. In order to compute such a similarity we use the Euclidean distance between individuals. Prior to the construction of the similarity matrix, the criterion values are normalised with the rank-based procedure described above. The similarity A_{ij} between the rank vectors $\mathbf{r}_i$ and $\mathbf{r}_j$ corresponding to individuals $\mathbf{y}_i$ and $\mathbf{y}_j$ is defined as:

$$A_{ij} = 1 - \frac{1}{M(N-1)^2} \sum_{m=1}^{M} (r_{im} - r_{jm})^2 \,. \tag{5.3}$$

Since the greatest difference in ranks, which occurs when two criteria are anti-correlated, is $N-1$, we can guarantee that $\sum_{m=1}^{M}(r_{im} - r_{jm})^2 < M(N-1)^2$, and so $0 < A_{ij} < 1$ for all pairs i, j.

Figure 5.2 illustrates the similarity matrices for the individuals in the GUG09 population before and after seriation. Red indicates that two individuals are highly similar, and blue that they are very different, according to the similarity measure defined in Equation 5.3. Clearly, in the seriated similarity matrix shown in Figure 5.2(b), the individuals have been ordered so that similar individuals are close together: the red pairs of individuals, those which are most similar, are placed near to the diagonal and therefore close to the individuals to which they are similar. Blue, dissimilar, pairs of individuals are placed towards the edges. The heatmap corresponding to this seriation is shown in Figure 5.3.



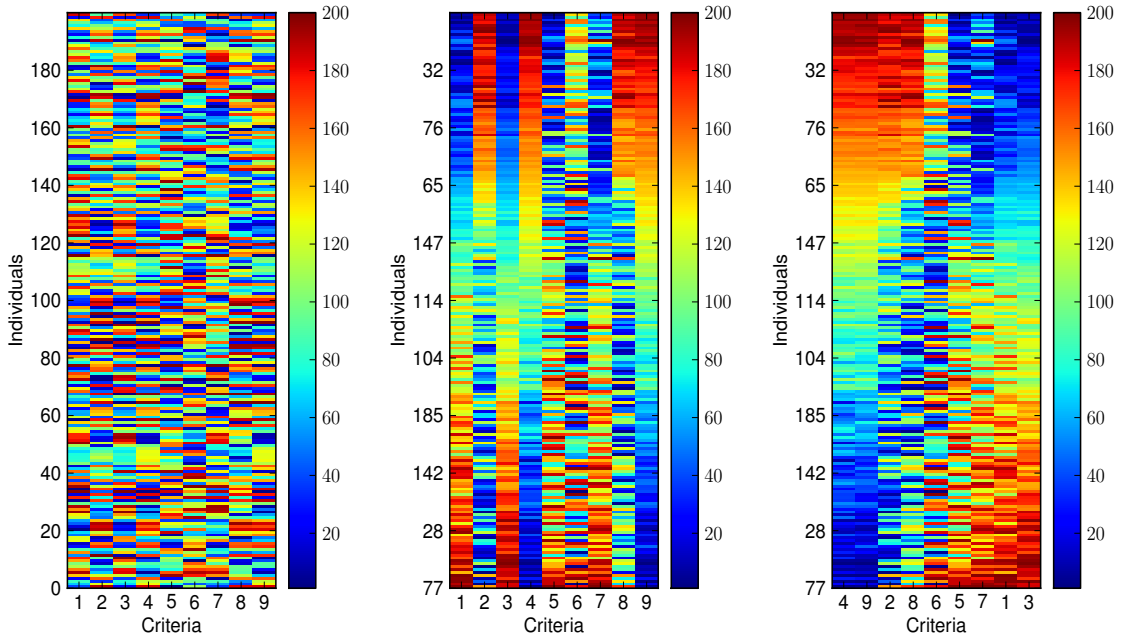
(a) Unseriated (b) Seriated Individuals (c) Seriated Individuals & Criteria

Figure 5.3.: Seriation of the GUG09 data. The left-hand panel shows the unseriated data; in the central panel, the universities (rows) have been seriated, and in the right-hand panel this ordering has been seriated with respect to the KPIs (columns).

Figure 5.3(a) repeats the visualisation of the GUG09 population shown in Figure 5.1, in which the universities have been ordered alphabetically by name, to facilitate comparison with Figure 5.3(b) in which the universities have been seriated. The clarity of the heatmap has been greatly enhanced and it is now possible to begin to observe correlation between criteria. For example, criterion 4 appears to be well correlated with criterion 5, but we could not have observed this from the visualisation in Figure 5.3a. In Figure 5.3c the criteria have also been seriated, and we discuss this visualisation shortly.

In addition to visualising the standard many-criterion populations we have discussed so far, an important aspect of this chapter is the visualisation of solutions to many-objective problems, specifically individuals with both parameter space and objective space components. In addition to comprising parameter and objective spaces, these populations are also mutually non-dominating as opposed to the GUG09 population which includes dominated individuals; recall from Chapter 3 that the GUG09 population sorts into 6 Pareto shells.

We now revisit an example population drawn from the many-objective optimisation literature. Recall from previous chapters, that [Hughes, 2003] optimised a set of waveforms for transmission by a Pulsed Doppler Radar to measure the velocity and distance of a target simultaneously. Here we visualise 200 solutions from the 12 PRI (Pulse Repetition Inter-

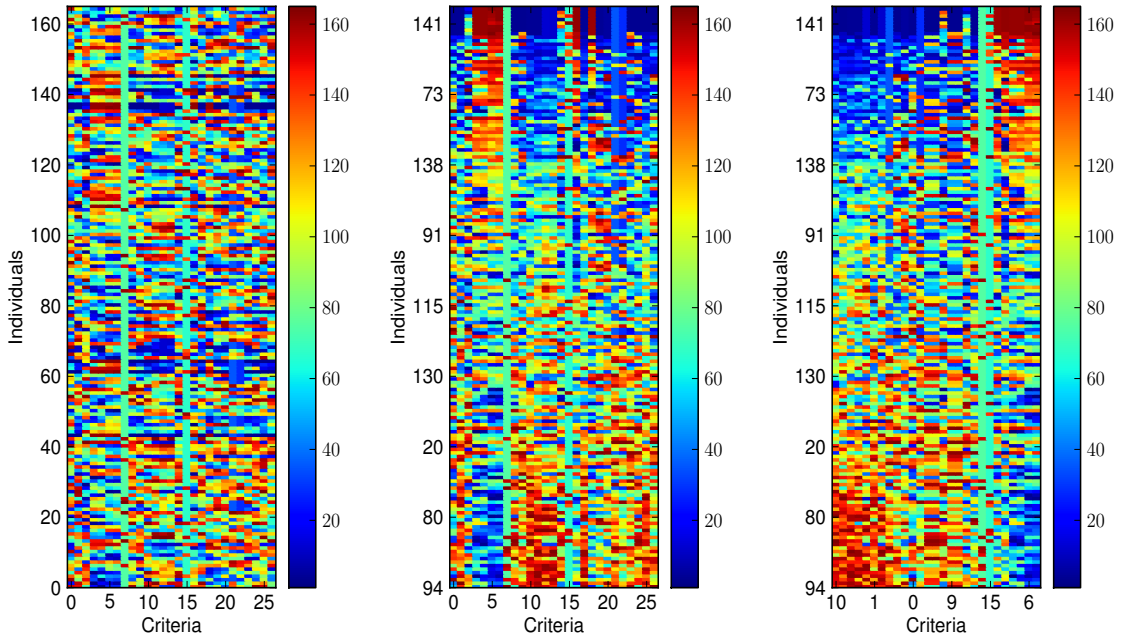


(a) Unseriated (b) Seriated Individuals (c) Seriated Individuals & Objectives

Figure 5.4.: Seriation of the radar waveform data. The left-hand panel shows the unseriated data; the central panel has been seriated with respect to solutions, and the right-hand panel with respect to solutions and objective.

val) archive published online by Hughes [2007b]. Figure 5.4 illustrates the radar archive with heatmaps. The left-hand panel shows the solutions in their original, arbitrary, order in which each solution, $\mathbf{r}_i$ in rank coordinates, comprises a row of the heatmap. The colour-scale extends between 1 and 200 because with 200 solutions the ranks lie in this range. The central panel shows the same archive after the solutions have been seriated. As was the case with the GUG09 example, shown in Figure 5.3, similar solutions have clearly been clustered together aiding the interpretation of the visualisation. As with the GUG09 visualisations in Figure 5.3, Figure 5.4(c) illustrates the radar population after the criteria have been seriated; these results are discussed in the following section.

The final example of this section shows the telecommunications population, introduced in Chapter 3. Recall that this population comprises 165 individuals, each of which is described by 27 KPIs; each individual represents a wireless access point in a mobile telephone network, and the KPIs are used by the network operator to assess how well each access point is performing for the purposes of scheduling maintenance. Figure 5.5 shows the telecommunications population, both as an unseriated heatmap in Figure 5.5(a) and as a heatmap in which the individuals have been seriated in Figure 5.5(b). Clear in both of these visualisations is the fact that there are a large number of tied values in criteria 8 and 16; more precisely, each of these values is 0, indicating missing data. If k is the rank of the first individual for which a value is present then the missing data is assigned the



(a) Unseriated (b) Seriated Individuals (c) Seriated Individuals & Objectives

Figure 5.5.: Seriation of the telecommunications data. The left-hand panel shows the unseriated data; the central panel has been seriated with respect to solutions, and the right-hand panel with respect to solutions and objective.

average of the first $k-1$ ranks, $\frac{1}{k-1}\sum k-1$. The improvement in the heatmap in Figure 5.5(b) over that shown in Figure 5.5(a) is, again, clear to see. The seriation according to KPIs shown in Figure 5.5(c) is discussed shortly.

5.2.2. Seriation of Criteria

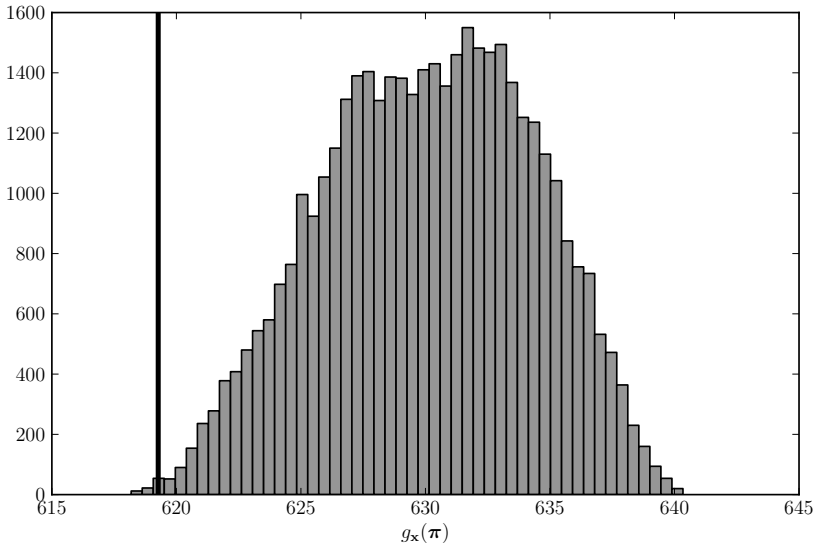
From observing Figure 5.3b we can see that having seriated the individuals in the population, the clarity of the heatmap could be further enhanced by seriating the criteria too, the result of which is shown in Figures 5.3(c), 5.4(c) and 5.5(c). We define the similarity matrix $\mathbf{S}$, which has dimensions $M \times M$:

$$S_{mn} = 1 - \frac{1}{N(N-1)^2} \sum_{i=1}^{N} (r_{im} - r_{in})^2 . \tag{5.4}$$

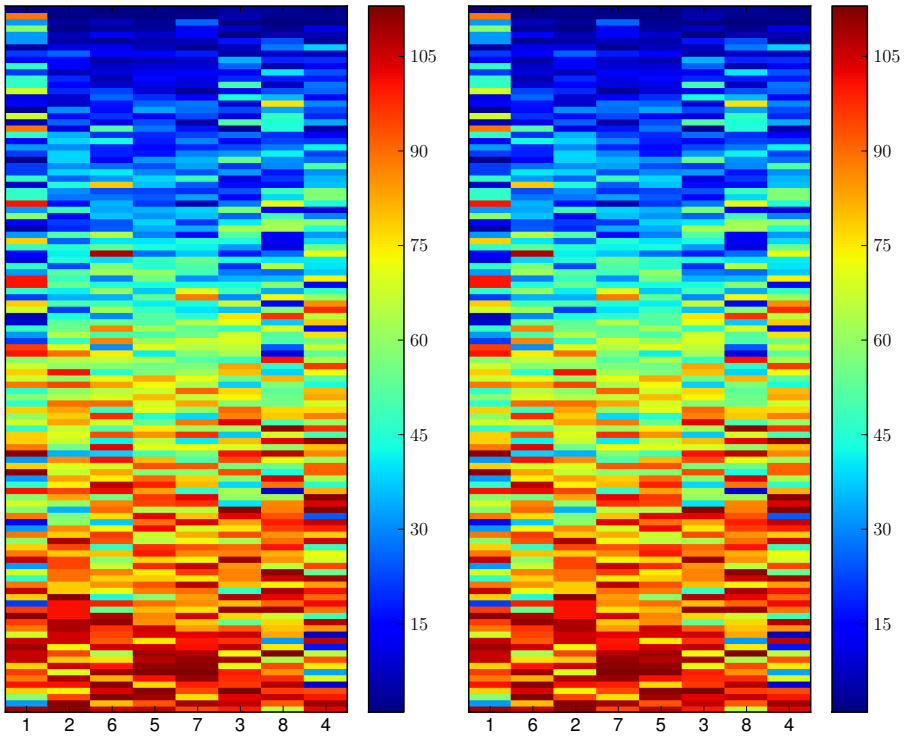
The rest of the procedure continues as before, substituting $\mathbf{S}$ for $\mathbf{A}$. The right-hand panel of Figure 5.3 shows the results of seriating the heatmap in the central panel with respect to criteria. As can be seen, similar criteria have been placed next to each other; in particular objectives 4 and 5 which as we observed earlier are visually well-correlated, have remained adjacent to one and other.

Figure 5.4 presents a visualisation of the radar data which is arrived at by following the same procedure used to produce Figure 5.3; individuals are seriated before criteria are seriated too. Figure 5.4(c) shows the result of seriating the individuals, as in Figure 5.4(b) *and then* the criteria. As can be seen, similar criteria have been placed next to each other, grouping all of the range-based objectives (objectives 1, 3, 5 and 7) on one side and the velocity-based objectives (2, 4, 6 and 8) on the other. Interestingly objective 9, which measures the transmission and decoding time, is placed with the velocity-based objectives and is clearly well correlated with objectives 2 and 4. Also it is clear that objective 6 (minimum transmission time before the schedule cannot be decoded) is least well correlated with either group, with small values occurring in combination with large and small range and velocity objective values. We emphasise that neither of these observations about the character of the non-dominated set as a whole could have been made from the original heatmap (Figure 5.4(a)). A similar improvement in heatmap clarity is observed in Figure 5.5(c) for the telecommunications population, in which similar performance indicators have been grouped together.

In the case of the individual seriation the size of the population, 113 universities in the case of the GUG09 population, prohibits qualitative analysis of the seriated permutation; there are 113!/2 potential permutations and it is simply not feasible to examine them all in order to see how close to the optimal permutation our approximation is. For the criterion seriation, however, the permutation only comprises 8 elements and it is possible to exhaustively evaluate the quality of each permutation so that we can see where in the distribution of permutation qualities the approximation lies. Figure 5.6 illustrates the distribution of permutation qualities for all 8! possible permutations of the GUG09 criteria. The quality of a permutation is evaluated by permuting the similarity matrix according to a given ordering and calculating the quality according to the objective function defined by Equation 5.1. The permutation produced with spectral seriation of the criteria is highlighted by a black line, and is clearly among the best, residing as it does in the tail of the distribution with the best permutations. In fact, of the 40320 permutations of 8 criteria, according to the definition of quality in Equation 5.1, the one identified by the seriation is the 50th best, placing it in the top 0.12% of all permutations. We reiterate that spectral seriation is not guaranteed to find the optimal ordering according to the objective function, but rather an approximation to it. Clearly, in this case, the seriation has succeeded in finding a good approximation to the optimal permutation. The visualisation produced using spectral seriation to order the criteria is shown in Figure 5.6(b) for comparison against that of the permutation which provides the highest quality (shown in 5.6(c)), according to Equation 5.1 in conjunction with the similarity defined in Equation 5.4. As can be seen, the two figures are very similar. The only differences are that the ordering of criteria 2 and 6 is reversed in the optimal ordering, as is the ordering of criteria 5 and 7. Otherwise the approximated permutation is the same as the optimal one.



(a) Distribution of seriation qualities (black line indicates spectral seriation quality)



(b) Spectral Seriation

(c) Optimal Seriation

Figure 5.6.: The distribution of all permutation qualities for the GUG09 criteria evaluated against Equation 5.1 using the criterion similarity measure in Equation 5.4. The vertical line indicates the position of the approximation identified by seriating the similarity matrix; as can be seen, the approximation is not the best possible permutation but is in the extreme tail of the distribution toward the ideal ordering. (b) shows the heatmap visualisation produced using spectral seriation to reorder the criteria, while (c) shows the ordering with the highest ordering according to Equation 5.1. Both are very similar.

The similarity measures defined so far have computed similarity between individuals and objectives using the Euclidean distance. The example presented in this section have shown that using Euclidean distance produces a satisfactory permutation. In the coming sections, we investigate methods for seriating individuals and criteria based on the rank of an individual.

5.3. Seriating Criteria with Rank Information

As previously discussed, it is useful to put all criteria onto the same scale when visualising individuals with a heatmap. As we discussed in the introduction to this chapter, since the dominance relation is only concerned with rank-order among criteria, a natural way to achieve this is to rank each individual criterion so that each lies on the range $[1, N]$.

Having ranked the individuals in this fashion, we can consider the problem of seriating criteria as the problem of placing similar criteria close together. Since ranking according to each criterion produces a set of M permutations, the distances between those criteria that are similar should be small. We therefore consider methods for measuring the distance between permutations and used them to compute a similarity between criteria. We evaluate Spearman's footrule [Spearman, 1904, 1906; Diaconis and Graham, 1977] and Kendall's τ metric [Kendall, 1938]. As defined in Chapter 2, Spearman's footrule δ^{SF} is computed with:

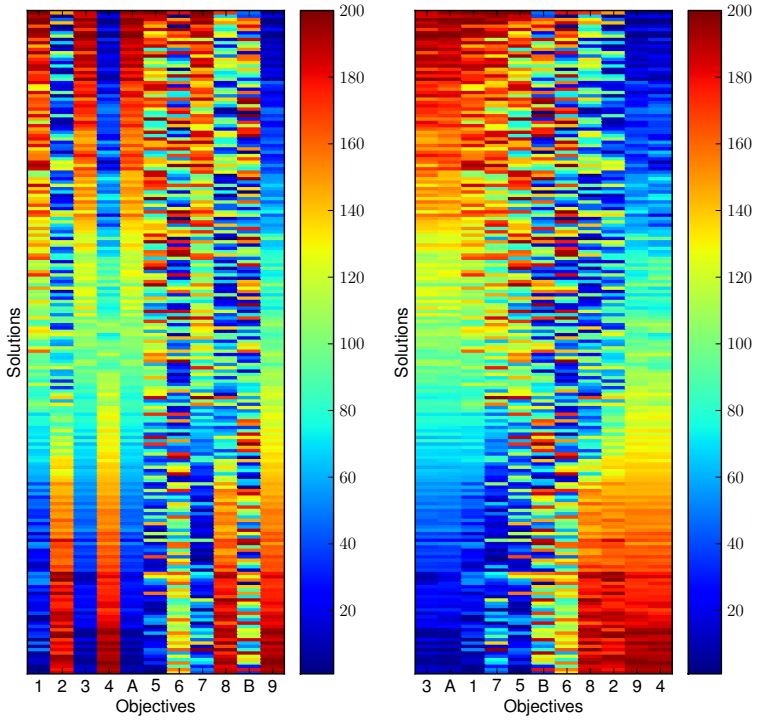
$$\delta^{\mathrm{SF}}_{mn} = \sum_{i=1}^{N} |r_{im} - r_{in}|, \tag{5.5}$$

where r_{im} is the rank of the ith individual on the mth criterion, and Kendall's τ metric δ^{KT} is computed with:

$$\delta^{\mathrm{KT}}_{mn} = \sum_{ij} \tau_{ij}(\boldsymbol{\pi}_m, \boldsymbol{\pi}_n), \tag{5.6}$$

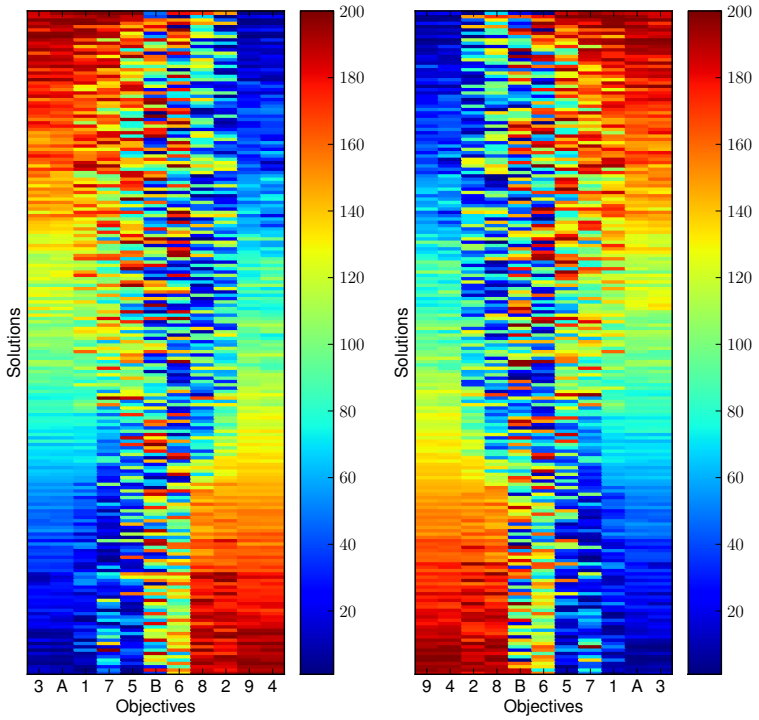
where $\tau_{ij}(\boldsymbol{\pi}_m, \boldsymbol{\pi}_n) = 1$ if the order of the ranks provided for the ith and jth element of the original data is different in the permutations $\boldsymbol{\pi}_m$ and $\boldsymbol{\pi}_n$, and 0 otherwise.

To demonstrate the use of permutation metrics for computing similarity for seriation, we seriated the criteria in a variant of the radar data. The new dataset contained all of the original 9 objectives, with two additional objectives. The first additional objective, which we denote by y_{iA}, was produced by averaging two existing, well correlated, objectives (objectives 1 and 3); thus $y_{iA} = (y_{i1} + y_{i3})/2$. The second is an objective drawn entirely from uniform random samples (N samples from a uniform distribution between 0 and 1); this objective is therefore expected to be uncorrelated with the other objectives, and as such we would expect that the process of seriation will move it away from the groups of well correlated objectives within the heatmap. The results of seriating the new dataset, measuring similarity with Euclidean distance, Spearman's footrule and Kendall's τ metric,



(a) Original population

(b) Euclidean distance



(c) Spearman's footrule

(d) Tau metric

Figure 5.7.: Criterion seriation by rank-based metrics. This version of the radar data has been modified to include an objective produced by averaging objectives 1 and 3 (objective A) and an objective entirely comprised of uniform random noise (objective B).

	S_{EUC}	$S_{\text{KE}\tau}$	S_{SPF}
$\boldsymbol{\pi}_{\text{EUC}}$	$\mathbf{7.9038 \times 10^2}$	5.2483×10^6	$\mathbf{2.6398 \times 10^6}$
$\boldsymbol{\pi}_{\text{KE}\tau}$	7.9053×10^2	$\mathbf{5.2477 \times 10^6}$	2.6445×10^6
$\boldsymbol{\pi}_{\text{SPF}}$	$\mathbf{7.9038 \times 10^2}$	5.2483×10^6	$\mathbf{2.6398 \times 10^6}$

Table 5.1.: Comparing the permutations produced by seriating by Euclidean distance, Kendall's τ and Spearman's footrule, against the set of all possible permutations of the 9 objectives. The value in each cell represents the quality of a given permutation according to a given metric. A value highlighted in bold indicates that it is the best possible quality when evaluated under that metric. In each case, the spectral seriation has identified the best possible permutation by the metric in use.

are shown in Figure 5.7. As can be seen, in all cases the well correlated objective (objective A) is placed next to one or both of the objectives upon which it is based. In addition, the random objective (objective B) is placed in the middle of the heatmap, between the two groups of well correlated objectives. In general, the permutations produced by seriation are very similar across the three metrics. In the case of Spearman's footrule and Kendall's τ metric this is perhaps unsurprising given the inequality provided by Diaconis and Graham [1977], $\delta^{\text{KT}} \leq \delta^{\text{SF}} \leq 2\delta^{\text{KT}}$. The Euclidean distance ordering is the same as that produced by Spearman's footrule and in the case of Kendall's τ, the ordering of objectives 4 and 9 have been reversed from that of the other two metrics. We note that while the ordering under Kendall's τ is the reverse of those produced with Euclidean distance and Spearman's footrule, the permutations are equivalent. Since all three metrics offer similar performance in this case, we suggest that Spearman's footrule should be used to seriate objectives because of its simplicity and speed of calculation.

Returning to the original 9-objective radar archive, it is feasible to evaluate all of the 9! permutations of objectives in order to examine their fidelity under each of the three metrics using Equation 5.1 in conjunction with the similarity matrix defined in Equation 5.4. Table 5.1 presents the results of this analysis. Each row of the table represents the fidelity of the permutation which was found by seriating with a similarity matrix computed with one of the metrics; for example, $\boldsymbol{\pi}_{\text{EUC}}$ is the permutation found by seriating using the Euclidean distance (Equation 5.3). We then consider the quality of this permutation under the other metrics too; each column represents one of the similarity measures (Euclidean distance, Kendall's τ and Spearman's footrule — S_{EUC}, $S_{\text{KE}\tau}$ and S_{SPF} respectively) which is reordered by the permutation such that the quality can be evaluated with Equation 5.1. A quality which is highlighted in bold indicates that the permutation is the best found in exhaustive search for a given metric. Reassuringly, the best permutation according to the Euclidean distance is that found by seriating the Euclidean distance-based similarity matrix, and as such it is highlighted in the Euclidean distance column. This is also the case both for Spearman's footrule and Kendall's τ, indicating that in all three cases the

"best" permutation has been found. However, as the spectral method is an approximation this will not always be the case, especially in the event that there are more objectives; indeed, it was not the case for the GUG09 results presented earlier in this chapter. Interestingly, the best quality under both the Euclidean distance and Spearman's footrule are the same, indicating that the permutations produced are either the same or the reverse of one another.

In general, because of its simplicity and ease of calculation, we prefer to calculate similarity between criteria with Spearman's footrule. That said, the difference in complexity of calculation between Spearman's footrule and Kendall's τ is small, and as we have observed here they produce similar results.

5.4. Seriating Individuals with Rank Information

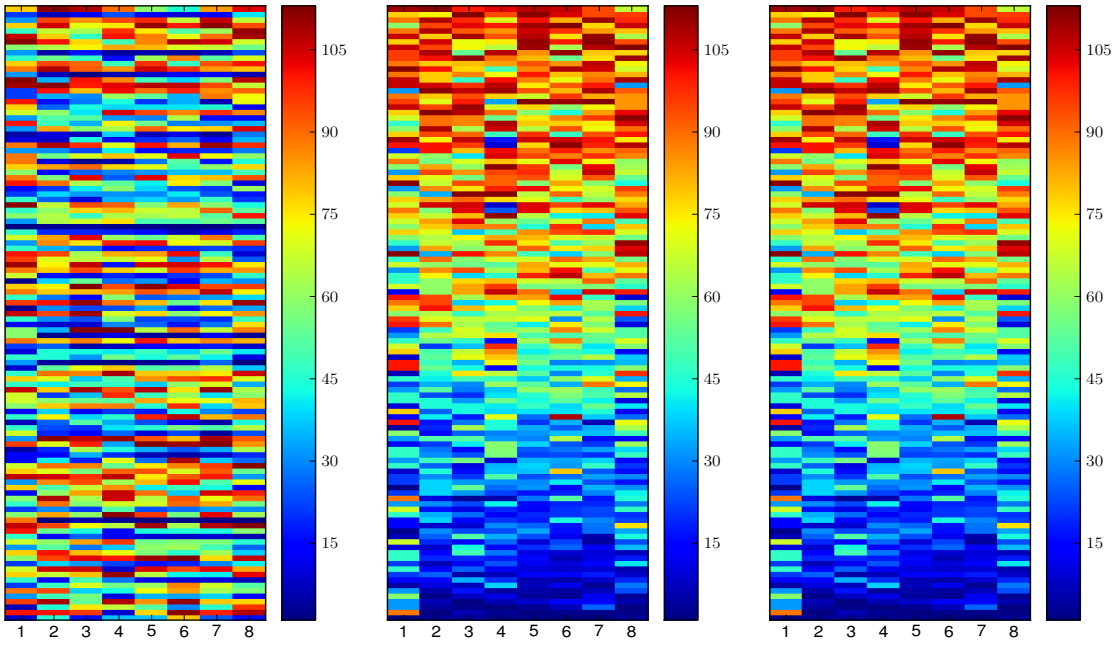
We first consider alternative rank-based methods of computing individual similarity, specifically using the rank of two individuals to determine how similar they are. To do this, we revisit the power index from Chapter 3, although the other ranking methods shown in that chapter, such as average rank, could also be used; the power index is used because it produces secondary information about the strength of an individual.

To compute the power index, we first construct a generalised tournament matrix (GTM) based on the probability of dominance; as described in Chapter 3, the probability that one individual dominates another is the likelihood that it is better on a randomly chosen criterion. The power index is then found by identifying the eigenvector $\mathbf{u}$ corresponding to the principal eigenvalue, so that the individual with the nth largest value in $\mathbf{u}$ is assigned rank n.

In order to use the power index as a measure of similarity between individuals we compute the power of each individual and then use the difference in power between each pair of individuals:

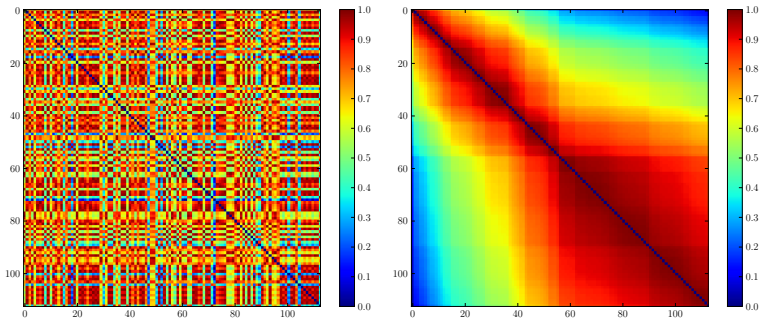
$$S_{ij} = \max_n u_n - \min_n u_n - |u_i - u_j|. \tag{5.7}$$

An alternative to this is to consider the ranked power index, such that we rank the vector $\mathbf{u}$ and compute the difference between ranks instead. The result of these methods are shown in Figures 5.8, 5.10 and 5.11. The GUG09 heatmaps (Figure 5.8) are clearer, and by examining the similarity matrices we can see that similar individuals have indeed been grouped together along the diagonal in what appears to be a very good seriation. The improvement in the clarity of the radar heatmap (Figure 5.10) is less than we have observed in previous examples; a likely reason for this is that ordering by Euclidean distance between values groups individuals together with similar objective values. The power index, however, can consider two individuals with very different objective values



(a) Original (b) Power index (values) (c) Power index (ranks)

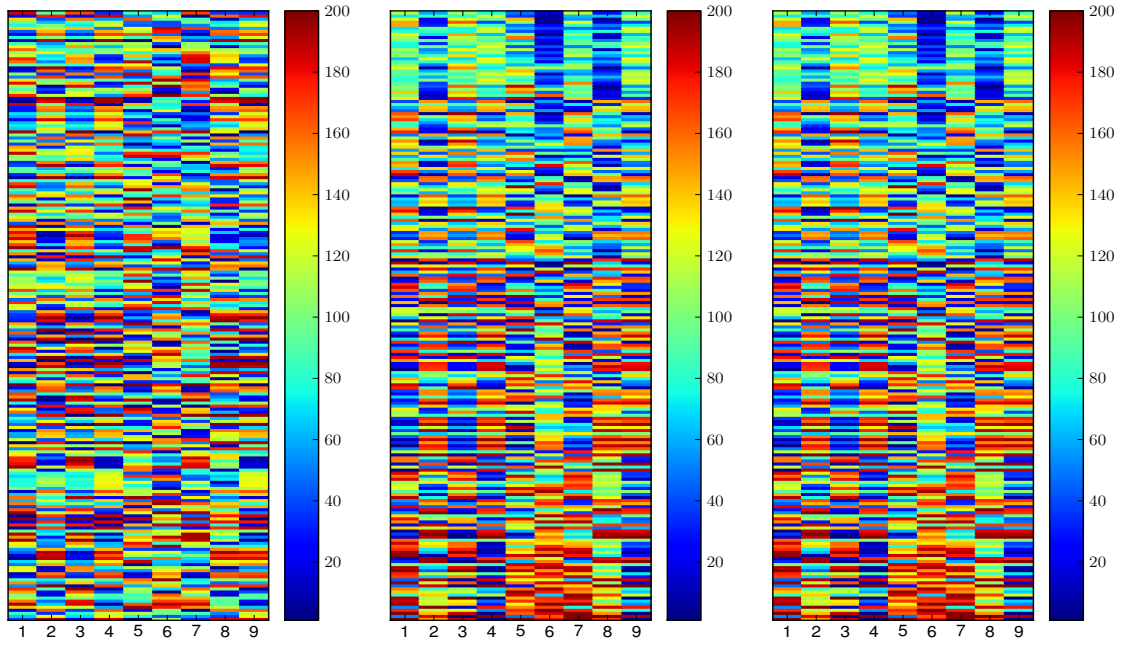
Figure 5.8.: Seriation of universities in the GUG09 dataset using the power index as a similarity measure. Similarity in the central panel was determined using the raw power index values, whilst in the right-hand panel the power index values were ranked.



(a) Original population (values) (b) Seriated population (values)

Figure 5.9.: Before and after similarity matrices for the process of seriating the GUG09 population using the power index as a similarity measure; similarity has been determined using the power index values. By inspection, the seriation appears to have found a very good ordering, grouping the red very tightly to the diagonal of the matrix.

similar if they are both deemed to be powerful in the context of the population. Such individuals, whilst having similar power, may not have similar objective values. This will result in placing very different colours next to each other in the heatmap, reducing the clarity of the final visualisation. This effect is less noticeable in the GUG09 population heatmaps shown in Figure 5.8. A likely explanation for this effect is the presence of Pareto

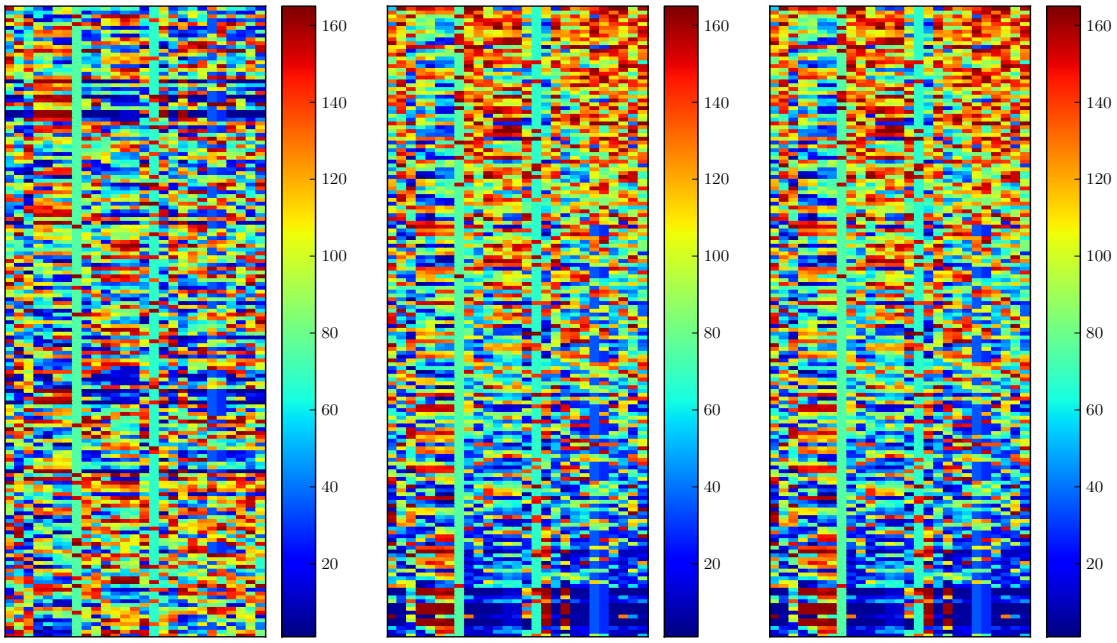


(a) Original (b) Power index (values) (c) Power index (ranks)

Figure 5.10.: Seriation of solutions in the radar archive using the power index as a similarity measure. Similarity in the central panel is based on the raw power index values, whilst in the right-hand panel it is based on ranked power index values.

shells within the population. The Pareto optimal individuals are generally superior on a range of criteria, while those individuals in the last Pareto shell generally exhibit poor performance on several criteria. As such, in a heatmap, the row corresponding to a Pareto optimal individual will generally be a cool colour representing poor ranks, and the row corresponding to a heavily dominated individual will comprise warm colours, representing good ranks. Since the greatest difference in ranks will be between those individuals in the first Pareto shell and those in the last, the most superior individuals will be placed away from the most inferior ones, which gathers together the cool colours at one end of the heatmap and warm colours at the other. In the case of a mutually non-dominating population, powerful individuals will be superior on different criteria, causing less clear separation between the colours and a more confusing visualisation, as shown in Figures 5.10 and 5.11. The visualising the telecommunications population (Figure 5.11) provides a similar result; while there is a noticeable improvement in quality, the improvement is not as great as in previous sections. That said, from examining the permuted similarity matrices for the GUG09 population (Figure 5.8), shown in Figure 5.9, it is clear that the spectral seriation has produced a permutation that places similar individuals close together as it should. This highlights the necessity of choosing the similarity measure used to compare individuals carefully in order to seriate them for visualisation with a heatmap.

In order to assure ourselves that the seriation process is finding suitable permutations



(a) Original (b) Power index (values) (c) Power index (ranks)

Figure 5.11.: Seriation of solutions in the telecommunications population using the power index as a similarity measure. As with the examples in Figures 5.8 and 5.10, similarity in the central panel is based on the raw power index values, whilst in the right-hand panel it is based on ranked power index values.

according to the similarity measure used, we evaluate the qualities of the permutations in a similar fashion to that used in Table 5.1. There we evaluated the quality of permutation produced by different metrics used to seriate criteria. Three metrics were used, resulting in three similarity matrices and three seriation permutations. Each similarity matrix was permuted by each seriated permutation, and the quality assessed using Equation 5.1. We repeat this procedure here, producing a similarity matrix for the individuals using Euclidean distance and the power index rank method described in this section; we use the power index *value* rather than rank for comparison and provide results for the three populations (GUG09, radar and telecommunications). Unfortunately, the additional step of highlighting when seriating produces the optimal ordering is impossible here as all three populations contain far too many individuals to enumerate all possible permutations. The results shown in Table 5.2 shows that permuting a similarity matrix by the ordering produced by its spectral seriation (i.e., the quality of $\mathbf{S}_{\text{EUC}}$ permuted by $\boldsymbol{\pi}_{\text{EUC}}$) produces the superior seriation quality. While this is far from a conclusive proof that the seriation is producing the best possible approximation to the optimal ordering, it does serve as a reassurance that the permutation produced is of reasonable quality.

In the last two sections we have shown how various methods can be used to reorder heatmaps of objective vectors, both in terms of the solution and objective orderings. In the next section we extend the work of Pryke et al. [2006] and produce a seriated heatmap

Population	Similarity Measure	$\boldsymbol{\pi}_{\text{EUC}}$	$\boldsymbol{\pi}_{\text{RK}}$
GUG09	$\mathbf{S}_{\text{EUC}}$	1.8593×10^7	1.8727×10^7
	$\mathbf{S}_{\text{RK}}$	1.3816×10^9	1.3564×10^9
Radar	$\mathbf{S}_{\text{EUC}}$	1.8278×10^8	2.1672×10^8
	$\mathbf{S}_{\text{RK}}$	2.9990×10^9	2.4833×10^9
Telecommunications	$\mathbf{S}_{\text{EUC}}$	9.4766×10^7	9.7116×10^7
	$\mathbf{S}_{\text{RK}}$	1.0331×10^9	8.6488×10^8

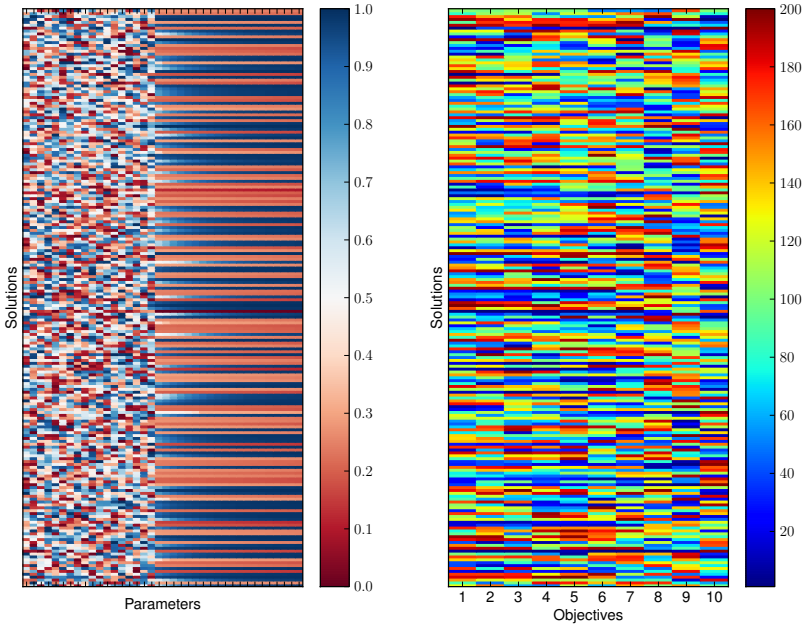
Table 5.2.: Comparing the permutation quality produced with Euclidean similarity ($\mathbf{S}_{\text{EUC}}$) and rank-based similarity using power index values ($\mathbf{S}_{\text{RK}}$). Permutations were produced by seriating both similarity matrices. Each similarity matrix was permuted by each seriated permutation to evaluate the quality of the seriation using Equation 5.1. In each case, the permutation of the similarity matrix used to produce the ordering (e.g., the quality of $\mathbf{S}_{\text{EUC}}$ permuted by $\boldsymbol{\pi}_{\text{EUC}}$) produces the superior quality, as would be expected.

visualisation which also incorporates information about parameter space.

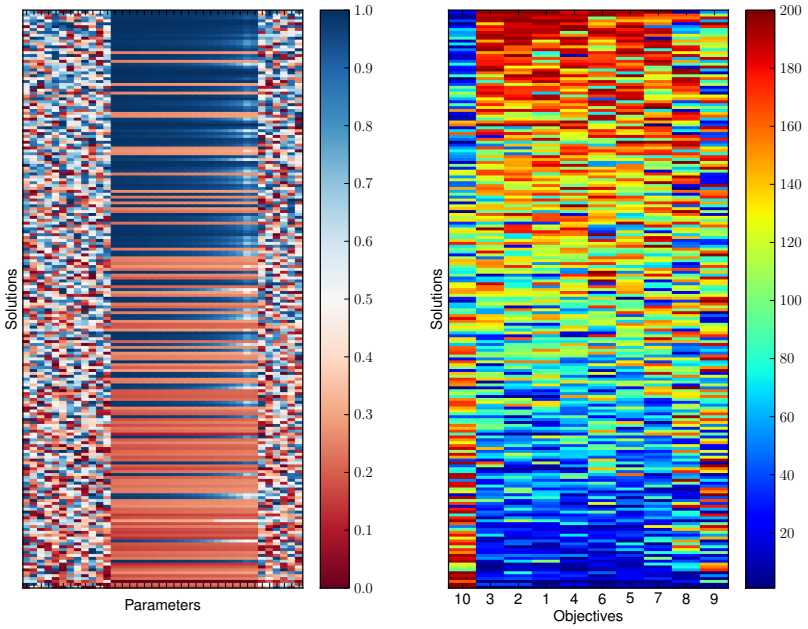
5.5. Joint Seriation of Many-objective Solutions

So far in this chapter we have shown how spectral seriation can be used to reorder objective space heatmaps to enhance their intepretability. In the case of populations of solutions to many-objective optimisation problems, it is often of interest to view the parameter space solutions $\mathbf{x}_i$ alongside their objective space counterparts [Pryke et al., 2006] and in this section we show how to simultaneously optimise the parameter space and objective space views. We assume that the parameters are real valued $x_i \in \mathbb{R}^P$, which allows meaningful distances between parameters to be calculated.

A straightforward way of jointly seriating parameter and objective space is shown in Figure 5.12 for a population of solutions to the WFG8 test problem [Huband et al., 2005]. The population comprises 200 solutions to a 10-objective instance of the problem, where the number of parameters is 38. The solutions were sampled from the known Pareto optimal set and perturbed with the addition of a small amount of Gaussian random noise. Here, the objective vectors $\mathbf{y}_i$ have been seriated in objective space with respect to objectives (reordering the columns) and then objective vectors (reordering the rows) which yields the heatmap shown in the lower right-hand panel of Figure 5.12. It is important to note that the same result would have been achieved by seriating first with respect to individuals (rows) and then criteria (columns) since the order of seriation does not affect the final outcome. Having done this, solutions $\mathbf{x}_i$ in parameter space were reordered to match



(a) Before seriation



(b) After seriation

Figure 5.12.: Seriation of both objective and parameter spaces for the WFG8 test problem. The upper panels show heatmaps of parameter space (left) and objective space (right). The lower panels show the result of first, seriating the objectives, followed by the solutions according to their objective space similarity (bottom right). The resulting solution ordering is then applied to the solutions in parameter space, and the parameters themselves are seriated to yield the bottom left heatmap. As can be seen, seriating parameters has kept the distance parameters (those on the left-hand side of the parameter space heatmap in (a)) together.

the order of the objective vectors in the seriated objective space (that is, the rows of the bottom left-hand panel), and finally the parameters were seriated to place similar parameters close to each other. In a similar manner to objective space seriation, the order of parameters was seriated by using spectral seriation to approximately minimise

$$g_{\mathbf{x}}(\boldsymbol{\pi}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \Lambda_{ij}(\pi_i - \pi_j)^2. \tag{5.8}$$

The objective function $g_{\mathbf{x}}(\boldsymbol{\pi})$ requires a parameter space measure of the similarity of individuals. Since ranking individuals in parameter space is meaningless we measure the parameter space similarity of individuals $\mathbf{x}_i$ and $\mathbf{x}_j$ in using their correlation or the well-known cosine similarity:

$$\Lambda_{ij} = \frac{\sum_p^P x_{ip}x_{jp}}{\sqrt{\sum_p^P x_{ip}^2 x_{jp}^2}}. \tag{5.9}$$

which is the cosine of the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$. We also provide an example later using the negative mean difference of solutions

$$\Lambda_{ij} = -\left|\frac{1}{P}\sum_{p=1}^{P}(x_{ip} - x_{jp})\right| \tag{5.10}$$

which has the effect of placing individuals with parameters of the same magnitude together. However, particular optimisation problems may suggest alternative measures of parameter space similarity to those used here.

As Figure 5.12 shows, the resulting ordering of parameter space individuals (induced by the objective space ordering of individuals) has indeed placed those solutions with larger parameter values together at the top of the heatmap. The seriation has revealed a clear correlation of solutions that optimise objective 10 well with larger parameter values, while the remainder of the objectives are best optimised by small parameter values. Clearly, however, reordering the parameter space solutions by the objective space seriation has not induced the same improvement in clarity as we have previously demonstrated in objective space; in a perfect reordering, *all* of the large-parameter solutions would reside in the top half of the heatmap and all of the small-parameter solutions would reside in the bottom. We discuss an approach to resolving this later in this section. The seriation of parameters has, however, produced a good result. As is common for many test problems, the parameters of WFG8 are grouped into two types. In this instance of the problem, 18 of the parameters are *position* parameters, which control the region of the Pareto front on which the solution lies. The remaining 20 are *distance* parameters, controlling the distance of the solution from the true Pareto front. The 20 distance parameters are well correlated (since the solutions are very close to the Pareto front their distance parameter values are nearly equal), and are therefore grouped together in the centre of the heatmap. Since the solutions were sampled uniformly from across the Pareto front, the position parameters are uncorrelated, and have been placed at the edges of the heatmap. Later in this section

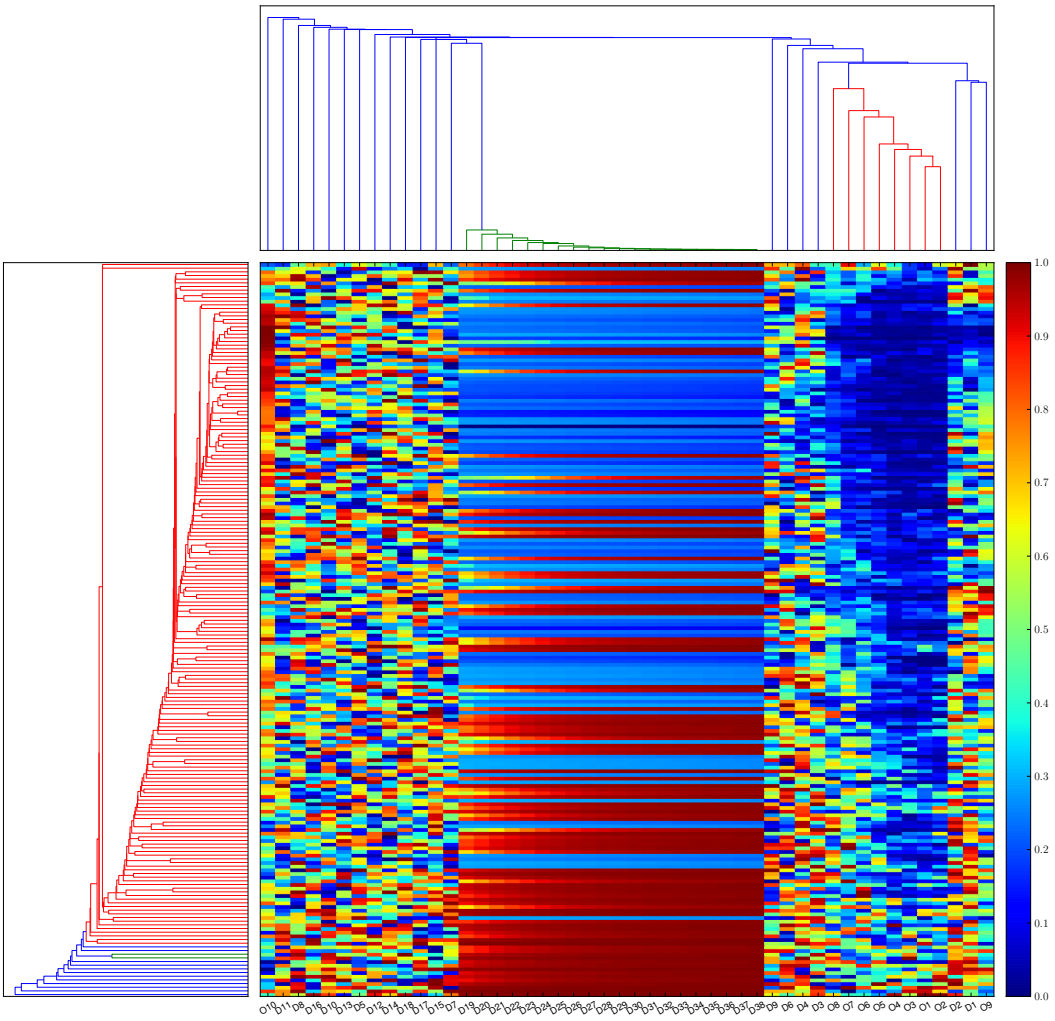


Figure 5.13.: A heatmap of the WFG8 parameters and objectives, reordered with hierarchical clustering as proposed by Pryke et al. [2006]. As in the other heatmaps presented in this chapter, a row represents a solution and a column is either an objective or a parameter. Solutions were clustered with single linkage clustering based on the Euclidean distance between normalised objective values, and objectives and parameters are clustered together. Dendrograms show the clustering for solutions (on the left) and objectives and parameters (on the top).

we show a further example with solutions to the radar data.

Figure 5.13 presents the same data, this time using the visualisation method presented by Pryke et al. [2006] Here, unlike the method we propose, the objectives and parameters are visualised with a single heatmap, and the columns of the heatmap are clustered so that both objectives and parameters are reordered together. Solutions are clustered with single-linkage clustering based on the Euclidean distance between their normalised objective values. Similarly, objectives and parameters are clustered together. As in Figure 5.12, the distance parameters have been gathered together, however the position parameters and objectives are intermixed. This makes observation of the trade-off between objectives

more difficult and as such we prefer a visualisation that keeps the two spaces separate. Additionally, the need to cluster both spaces together precludes the possibility of using ranks to normalise the objectives to fully use the colour scale, since they must be on the same range as the parameters that cannot themselves be ranked.

While the example in Figure 5.12 was seriated according to objectives space solutions, after which parameter space solutions were reordered to match the objective space seriation, it could justifiably have been done the other way around, seriating parameter space individuals and using the resulting permutation to reorder the objective vectors. However, there is clearly a trade-off between the quality of the solution orderings in parameter and objective spaces. The simultaneous clustering method [Pryke et al., 2006] obscures the trade-off, giving unknown relative weights to parameter and objective spaces. Here we therefore seek to simultaneously optimise the ordering of individuals in both spaces using a bi-objective evolutionary algorithm to locate an approximation of the Pareto front formed by the objective functions $g_{\mathbf{y}}(\boldsymbol{\pi})$ (Equation 5.1) and $g_{\mathbf{x}}(\boldsymbol{\pi})$ (Equation 5.8). A solution $\boldsymbol{\pi}$ to this multi-objective problem is a permutation by which individuals and criterion vectors are reordered in the heatmaps. Each permutation contains N elements, each element corresponding to a member of the population to be visualised, determining its position within the heatmap.

We use a basic multi-objective evolutionary algorithm to explore the trade-off between the quality of ordering in the two spaces, $g_{\mathbf{x}}$ and $g_{\mathbf{y}}$. Specifically, the algorithm is a $(\mu + \lambda)$–evolution strategy with a passive elite archive. A population $\Pi = \{\boldsymbol{\pi}_i\}_{i=1}^{P}$ is maintained and at each generation is used to produce an offspring population Π' of permutations which are evaluated under the two objective functions. Those μ solutions which are found to produce the most harmonious ordering of individuals and criterion vectors are used to replace Π as the parent population for the next generation.

The process is outlined in Algorithm 4. The inputs to the algorithm are the individual and criterion vectors to be ordered, as well as the similarity matrices which describe the two spaces. Its first step is to generate a seeded population by constructing a set of convex combinations of the similarity matrices. To do this we define a parameter η which determines the proportion of each matrix that is used to produce the combination; $\eta \in [0, 1]$, and the resulting combined similarity matrix $\mathbf{S}_\eta$ is a positive semidefinite matrix constructed as follows:

$$\mathbf{S}_\eta = \eta\mathbf{A} + (1 - \eta)\boldsymbol{\Lambda}. \tag{5.11}$$

In this formulation the example in Figure 5.12, where the populations $\mathbf{X}$ and $\mathbf{Y}$ shown in the top row are seriated favouring criterion space $\mathbf{A}$, $\eta = 1$, and the influence of the parameter space similarity matrix $\boldsymbol{\Lambda}$ is eliminated completely. If $\eta = 0$, we remove the influence of the criterion space similarity matrix and focus exclusively on similarity in parameter space. To generate a seeded population Π of permutations we generate $\mu = 100$ combination matrices by generating μ similarity matrices $\mathbf{S}_\eta$ for μ uniform samples

Algorithm 4 Multi-objective Optimisation of Joint Seriation

Require: Parameter space similarity matrix, $\mathbf{\Lambda}$; objective space similarity matrix, $\mathbf{A}$.

1: $E := \emptyset$ *Initialise an empty archive*
2: $\eta := 0$
3: **for** $i := 1, \ldots, \mu$ **do**
4: $\quad \mathbf{S}_i := \eta\mathbf{A} + (1-\eta)\mathbf{\Lambda}$ *Combine parameter and objective space similarity matrices*
5: $\quad \boldsymbol{\pi}_i := \text{seriate}(\mathbf{S}_i)$ *Seriate the combined similarity matrix*
6: $\quad \mathbf{f}_i := (g_{\mathbf{x}}(\mathbf{\Lambda}, \boldsymbol{\pi}_i), g_{\mathbf{y}}(\mathbf{A}, \boldsymbol{\pi}_i))$ *Evaluate the solution*
7: $\quad E := \text{update}(E, \boldsymbol{\pi}_i, \mathbf{f}_i)$ *Update the archive*
8: $\quad \eta := \eta + (1/\mu - 1)$
9: **end for**
10: **while** stopping condition not met **do**
11: $\quad$ **for** $i := 1, \ldots, \mu$ **do**
12: $\quad\quad \boldsymbol{\pi}'_i := \boldsymbol{\pi}_i$ *Copy the current solution*
13: $\quad\quad \boldsymbol{\pi}'_i := \text{mutate}(\boldsymbol{\pi}'_i)$ *Mutate the child solution*
14: $\quad\quad \mathbf{f}'_i := (f_1(\mathbf{\Lambda}, \boldsymbol{\pi}'_i), f_2(\mathbf{A}, \boldsymbol{\pi}'_i))$ *Evaluate the solution*
15: $\quad\quad E := \text{update}(E, \boldsymbol{\pi}'_i, \mathbf{f}'_i)$ *Update the archive*
16: $\quad$ **end for**
17: $\quad \{(\boldsymbol{\pi}, \mathbf{f})\} := \text{select}(\{(\boldsymbol{\pi}, \mathbf{f})\} \cup \{(\boldsymbol{\pi}', \mathbf{f}')\})$ *Pareto sorting selection*
18: **end while**

$\eta \in [0, 1]$. The objective vectors corresponding to the permutations produced by seriating these combined similarity matrices are shown for the WFG8 population and the radar data as insets on the main panel of Figures 5.14 and 5.15. As can be seen, the result is a set of permutations whose corresponding objective vectors lie between those for the $\eta = 0$ case (denoted by a circle) and $\eta = 1$ case (denoted by a cross). The procedure by which the solution sets Π were produced is outlined in Algorithm 4 on lines 3—9. Each of the non-dominated solutions from the initial seeded population is added to an elite archive E; this retains a copy of each solution which is non-dominated with respect to the other solutions in the archive at a given time during the optimisation. If at a later time a solution is found which dominates an archived solution, the dominated solution is removed from the archive in favour of the dominating solution. If the solution is not dominated by any member of the archive, it joins the archive.

Having generated an initial population and initialised the elite archive, the evolutionary process begins (Line 10). At the start of every generation, each solution is copied to produce a child solution which is then mutated to change the permutation that it represents Line (13). These child solutions are evaluated under the two objective functions (Line 14) and if they are not dominated by any members of the archive they are added to it (Line 15). At the end of each generation, the best μ solutions are identified from the union of the parent and child populations and are retained to become the new parent population for the next generation. To identify the top μ solutions, the union of the two populations, $\Pi \cup \Pi'$, is ranked using the Pareto sorting technique described in Chapter 3 (Line 17). When a stopping criterion is met, the evolutionary process stops and returns the final archive of non-dominated solutions.

Caution is required when mutating a permutation-based solution since a solution is constrained to contain each element exactly once. A traditional mutation or crossover scheme would not guarantee to obey this constraint so we employ operators based on the transposition of elements within the permutation [Eiben and Smith, 2003]. One method used is *block transposition*, in which a block of ξ elements are swapped with a second ξ-element block elsewhere in the permutation. A second method used is *shuffle transposition*, where a block is selected in the permutation and its constituent elements randomly reshuffled. We employed a mutation strategy in which either one of these operators was used independently, or both in combination. In the combination case, a pair of blocks of elements were transposed before a further block of elements were shuffled, or vice versa. The choice of operators, and the order in which the operators in a stacked mutation were performed, was determined based on a uniform random draw. 50% of the time both operators were used, either a shuffle followed by a transposition or vice versa, with even likelihood. 25% of mutations are purely based on a transposition, and the remaining 25% are purely based on a shuffle. In all cases, the block size is a random integer in the range $(0, \lfloor N/10 \rfloor)$.

Figure 5.14 shows the combined non-dominated permutations from 10 runs for the 200 WFG8 solutions and objective vectors used previously. As the main panel shows, the solutions after optimisation are very close to the initial solutions found by Equation 5.11.

The inset panel shows the objective vectors resulting from the seriations of $\mathbf{S}_\eta$ used to initialise the evolutionary population, together with the $g_{\mathbf{x}}(\boldsymbol{\pi})$ and $g_{\mathbf{y}}(\boldsymbol{\pi})$ corresponding to 200 randomly chosen permutations. Clearly the initialisation using $\mathbf{S}_\eta$ provides a very good approximation to the optimal trade-off between $g_{\mathbf{x}}(\boldsymbol{\pi})$ and $g_{\mathbf{y}}(\boldsymbol{\pi})$ and although the MOEA has improved the front slightly, its main effect, in this case, has been to remove dominated members of the seeded initialisation set, and to fill in the gaps in the initialisation set. Whilst it is useful for a decision maker to have full Pareto front approximation on which to base their selection of operating point, the MOEA has failed to uncover much beyond the original initialisation set, although of course the dominated solutions corresponding to low objective space qualities $g_{\mathbf{y}}(\boldsymbol{\pi})$ have been eliminated. The heatmaps on the right-hand side of the main panel show the seriations produced by solutions along the Pareto front approximation. The top heatmap represents the solution highlighted at the top of the approximated front and is the best ordering with respect to parameter space. The bottom heatmap is the solution highlighted at the bottom of the approximate front, namely the best objective space ordering found, and is essentially the visualisation shown in the bottom row of Figure 5.12. The middle heatmap represents a compromise between the objective space and parameter space coherence. There is a clear reduction in seriation quality when parameter space is ordered in terms of objective space similarity and vice versa. However, it may be worth accepting this compromise in order to be able to view the two spaces together.

We also demonstrate the result of optimising the joint seriation of parameters and objective

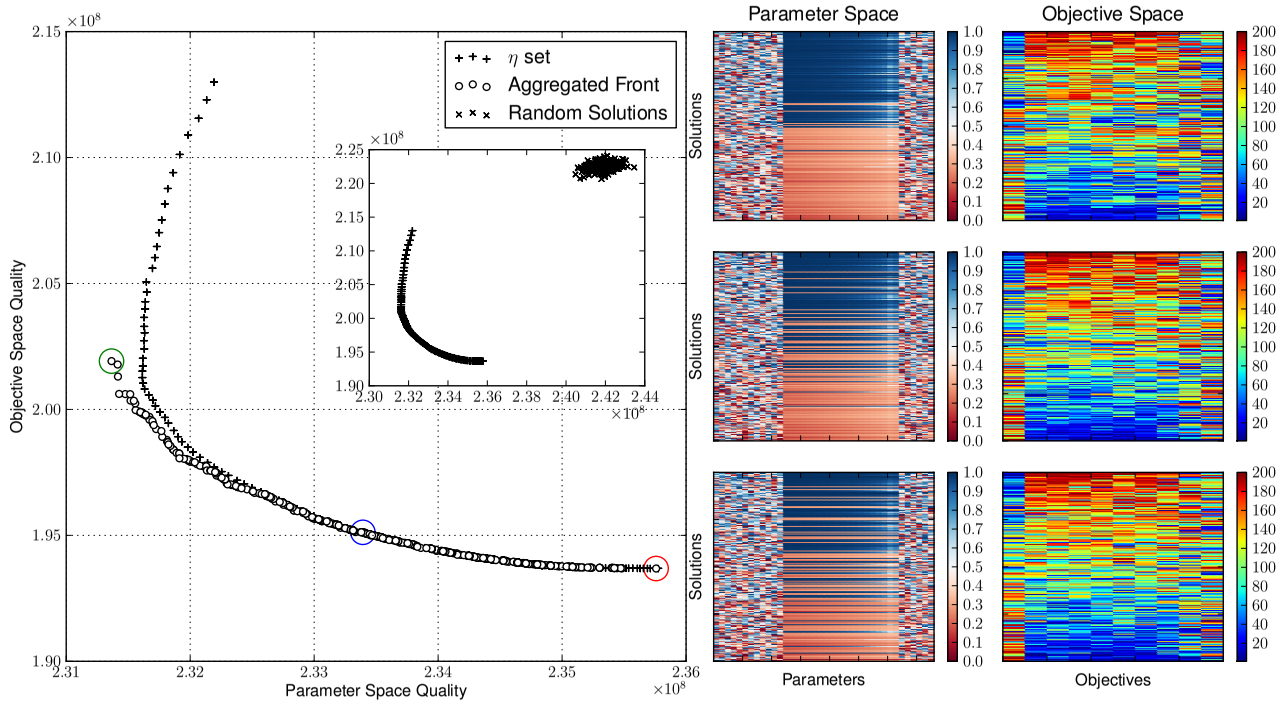


Figure 5.14.: Optimised trade-off between seriation quality in parameter space $g_{\mathbf{x}}(\boldsymbol{\pi})$ and objective space $g_{\mathbf{y}}(\boldsymbol{\pi})$ for the WFG8 solution set. The main panel shows the initial solutions from seriation using the similarity matrix $\mathbf{S}_\eta$ (5.11) as crosses (also shown in the insert) and the combined estimated Pareto front from 10 runs of the evolutionary optimiser. The heatmaps on the right-hand side of the figure correspond to optimised solutions marked on the main panel by large circles. The top and bottom visualisations are the solution orderings found by the MOEA which best optimise the parameter space ordering and objective space ordering respectively. The central heatmaps show a solution towards the centre of the Pareto front amount. Objectives and parameters have also been seriated independently. Parameter space seriation has grouped parameters $\mathbf{x}$ roughly into x_p controlling distance from the true Pareto front and x_p controlling angular location on the front.

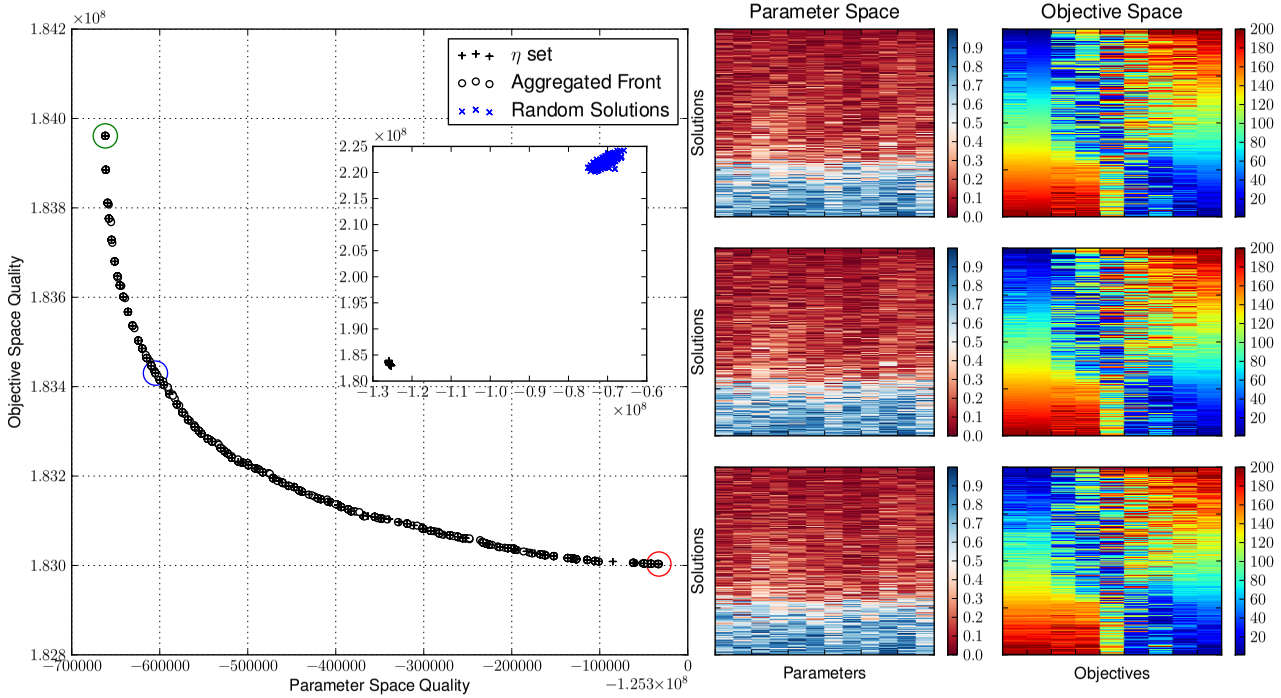


Figure 5.15.: Optimised trade-off between seriation quality in parameter space $g_{\mathbf{x}}(\boldsymbol{\pi})$ and objective space $g_{\mathbf{y}}(\boldsymbol{\pi})$ for the radar data. The main panel shows the initial solutions from seriation using the similarity matrix $\mathbf{S}_\eta$ (5.11) as crosses (also shown in the insert) and the combined estimated Pareto front from 10 runs of the evolutionary optimiser. The heatmaps on the right-hand side of the figure correspond to optimised solutions marked on the main panel by large circles. The top and bottom visualisations are the solution orderings found by the MOEA which best optimise the parameter space ordering and objective space ordering respectively. The central heatmaps show a solution towards the centre of the Pareto front amount. Objectives and parameters have also been seriated independently.

vectors for the radar data. Figure 5.15 presents the results of this optimisation, and was produced by following the same procedure as that used to produce Figure 5.14 – this time utilising the negative mean difference as the parameter space solution similarity (Equation 5.10). As before, the MOEA was seeded with 100 permutations obtained from seriating $\mathbf{S}_\eta$ for linearly spaced η and the non-dominated permutations from 10 runs are shown. Like the WFG8 population, the MOEA has found permutations that have only a marginal improvement over the initialisation seriations of $\mathbf{S}_\eta$. For this particular problem, however, it is clear to see that seriating solutions in objective space also leads to a good ordering of solutions in parameter space (and vice versa) and that either seriation is difficult to improve on. This indicates a strong correlation between parameters and solutions, providing useful information to the problem owner and we emphasise that this radar problem is a real problem rather than a synthetic test problem. We also draw attention to the marked improvement over random permutations when using spectral seriation with any of the $\mathbf{S}_\eta$.

It is not always possible to achieve an ordering that simultaneously groups like solutions in parameter space and objective space. Nonetheless the two examples illustrate that a seriation which compromises between parameter space and objective space grouping quality can be a helpful visualisation, particularly as it allows the investigator to assess objectives and parameters together.

Although on a limited number of examples, these results indicate that seriation of linear combinations of $\mathbf{A}$ and $\mathbf{\Lambda}$ provides very good approximations to the optimal orderings, which display an almost linear trade-off between parameter and objective space quality. There may be examples when an MOEA can improve significantly upon the convex combination defined in (5.11). Here we have used the cosine similarity and negative mean difference for measuring the proximity of parameter vectors, however, the choice of similarity is less clear than for objective space, where conversion to ranks simplifies the choice. Other similarity measures may be more useful for particular problems, particularly for categorical parameters for which there is no natural ordering.

5.6. Conclusions

Heatmaps provide a solution to the problem of visualising a many-criterion population without incurring information loss through dimension reduction, however the often arbitrary ordering of individuals and criteria can lead to a confusing visualisation. Seriation provides a means of resolving this issue, as has been demonstrated in this chapter.

A variety of metrics have been applied to determining the similarity between individuals and criteria. Euclidean distance operates well in both cases; ranking each objective and then using a permutation comparison metric such as Spearman's footrule or Kendall's τ metric allows the criteria to be reordered as similar criteria will produce similar rankings

over the population. Performance-based measures such as the power index do not effectively compare individuals since solutions can demonstrate similar performance whilst residing in a different region of the non-dominated set and having very different criterion values; the resulting ordering does not enhance the heatmap.

When visualising solutions to a multi-objective problem it can also be useful to visualise the parameter space component of a solution along with the corresponding objective vectors. We have shown that ordering solutions in parameter space by a permutation obtained by seriating objective vectors can enhance the parameter space heatmap, and vice versa, but that a trade-off exists between the two. Optimising this trade-off with a convex optimisation procedure generated a permutation which enhanced both spaces, although results were better for some populations, a set of WFG8 solutions for example, than for others, a set of solutions optimising radar waveform design.

In this chapter, the focus has been on visualising a many-criterion population in such a way that all of the criterion values are incorporated into the visualisation. In the next chapter we change tack and consider methods for identifying which criteria can be safely discarded without causing too much damage to the rank of the individuals in a population.

6. Rank-based Dimension Reduction

6.1. Introduction

So far, this thesis has examined methods for visualising a population of many-criterion individuals. These visualisations have been based on the structural information provided by the pairwise dominance relations between individuals. One of the visualisations shown in Chapter 3 used Pareto sorting to arrange the individuals in a population into a graph. Another sought to preserve dominance distances in a MDS embedding of a population. As has previously discussed, dominance itself does not scale well to differentiate between many-criterion individuals because the individuals are predominantly mutually non-dominating and hence are incomparable. While the dominance distance MDS method was shown to be robust in high-dimensional spaces, we found the method based on Pareto sorting to be badly affected by this lack of discrimination. It is therefore desirable to identify any redundant criteria, criteria which do not contribute to the structure of a population, so that they can be discarded. In this chapter we investigate methods for *criterion selection*.

The criterion selection problem is analogous to feature selection in pattern recognition [e.g. Bishop, 2007]. The general aim is to reduce the dimensionality of a dataset while preserving the characteristics of the original data. In Chapter 3, we used MDS to reduce the dimensionality of a many-criterion population. There, we defined a metric to operate in criterion space that computes distances between individuals in terms of their pairwise dominance relationships with the other individuals in the population. We used MDS to compress the dimensionality so that individuals that have similar dominance relations with other members of the population are placed close together in the new, low-dimensional, coordinate space. We provided illustrations on a variety of populations that demonstrated the dominance distance MDS approach producing a faithful representation of the population and its structure, however it did so at the expense of retaining the original criteria. Since the decision maker, who must use the visualisation to interpret the population, is familiar with the original criteria, it is helpful to present them with a visualisation in terms of those criteria. Since criterion selection reduces the dimensionality of a population by retaining the most important criteria and discarding the rest, we now turn our attention to those techniques.

Some of the material in this chapter has been published as Walker et al. [2011].

Quantifying the characteristics of a population can be done in a variety of ways. In feature selection, dimension reduction is often done to find the most informative features on which to base classification or regression. Feature selection methods therefore select the features that minimise classification or regression errors [Bishop, 2007]. In terms of the multi-objective optimisation literature, various approaches have been taken. These include selecting the objectives that capture the largest proportion of variance present in a population [Deb and Saxena, 2005] and minimising the number of pairwise dominance relationships that change as a result of omitted objectives [Brockhoff and Zitzler, 2007a]. In this work we strive to minimise the change to an individual's average rank caused by discarding criteria. We incorporate rank-based criterion selection into three algorithms: a greedy backward criterion selection algorithm; a single-objective hill-climber; and a multi-objective formulation of the criterion selection problem. We illustrate the effect of removing criteria with these methods on variants of the GUG09 population that we have used in earlier chapters, as well as the mobile telephone access point performance data.

As in previous chapters, since the terms *criterion* and *objective* are used interchangeably in the multi-objective literature, it is important to note that in this work we use *criterion* to refer to a feature of the population, some of which we intend to discard, and *objective* to refer to that which is minimised as part of the algorithm performing the criterion selection.

6.2. Criterion Selection

Most varieties of dimension reduction strive to minimise *information loss* as quantified by some measure. A common approach in pattern recognition, where the aim is to predict values, class labels in classification and real values in regression, is to identify the features which lead to the best prediction of the *target values*. As such, a small subset of features are chosen that maximise the predictive accuracy (for example, measured in terms of the correct number of classifications) of a pattern recognition tool.

In the evolutionary optimisation literature, one of the earliest studies on discarding redundant criteria was by Deb and Saxena [2005] who presented a method utilising Principal Component Analysis (PCA), a well known method for feature *extraction*. PCA aims to construct new features which retain as much variance as possible from the original features. In their study, Deb and Saxena use PCA to identify which criteria are responsible for the variance in the search population so that *they* can be retained, rather than using it to create new criteria. To do this, they use PCA to find the vector of eigenvalues and matrix of eigenvectors, $\boldsymbol{\lambda}$ and $\mathbf{U}$, of the correlation matrix of the set of criteria. Let U_{nm}, the mth element of eigenvector n, represent the contribution of the mth criterion to that eigenvector. The eigenvalue, λ_n, which provides the largest proportion of variance within the population, is identified. Then, two criteria, those representing $\arg\max_m U_{nm}$ and $\arg\min_m U_{nm}$ are selected to begin with. Subsequent eigenvectors are examined in the

order determined by the magnitude of their corresponding eigenvalues, starting with the largest and decreasing. As each eigenvector is considered, a cumulative sum of the proportion of the total variance within the population that has been retained is kept; Deb and Saxena [2005] suggest a threshold of 0.95. Once this threshold has been reached the part of the criterion selection process ends. Which criteria are selected at each stage depends on the structure and range of the values in the eigenvector. Having determined an initial criterion subset by means of eigenvector analysis, the correlation matrix is used to identify any remaining redundancies in the criteria.

Brockhoff and Zitzler [2007a] use the change in weak dominance relationships between individuals to identify criteria between which there is redundancy. By modelling individuals as the nodes in a directed graph, where an edge indicates that one individual weakly dominates another, it is possible to observe how many edges are changed (added or removed) with the removal of a criterion. An indication of the damage done to the structure of a population is given by determining the amount by which individuals must be shifted in order to ensure that individuals which previously dominated other individuals still do so in a low-dimensional space. This idea draws on ϵ-dominance and also bears similarity to the rank stability work presented in Chapter 3 of this thesis. The removal of a more redundant criterion will result in a smaller shift than if a conflicting criterion was removed. Brockhoff and Zitzler [2007a] propose two algorithms. One of them removes criteria until all but k criteria have been discarded, which they call k-EMOSS ("MOSS" is the *minimum objective subset problem*). The other, called δ-MOSS, removes criteria until a certain amount of damage has been done to the structure of the population.

López Jaimes et al. [2008] propose a method in which criteria are clustered into neighbourhoods of a given size q according to the degree of conflict between them, measured in terms of the correlation coefficient. The "most compact" neighbourhood is then found and all but the criterion at the centre of the neighbourhood are discarded. By discarding criteria in this fashion the amount of conflict between the remaining criteria is maximised. As with Brockhoff and Zitzler [2007a], two algorithms are presented. Again, one of the variants requires the size of the eventual criterion subset to be specified while the other discards criteria until a certain amount of damage has been done to the population structure, in this case quantified by the degree of conflict which is lost.

Another of the recent techniques for reducing the dimensionality of a population of solutions to a many-objective problem has already been discussed in this thesis. Singh et al. [2011] presented a technique which uses an algorithm to find individuals in the corners of a Pareto front which are then used to determine how the removal of criteria affects the dominance relationships in a similar manner to that shown by Brockhoff and Zitzler [2007a]. They calculate the proportion of non-dominated solutions after a criterion is removed to that of the full criterion set; if the proportion is high then the number of non-dominated individuals is similar with or without the criterion in question and it can be discarded

without damaging the structure of the population.

6.2.1. Rank-based Criterion Selection

To use ranks as a measure of criterion subset quality we return again to the two metrics used in Chapter 5 for identifying similar criteria, Spearman's footrule [Spearman, 1904, 1906] and Kendall's τ metric [Kendall, 1938]. As we outlined earlier, Spearman's footrule is the absolute difference between two rankings and Kendall's τ is a count of the number of occasions on which the order of a pair of individuals in two rankings is reversed. Both are proper metrics, assuming that there are no ties. It is possible to modify both so that they are metrics in the presence of ties, however since this work does not specifically require the use of a proper metric we use both measures in their original formulation. In order to guide the search for a suitable criterion subset we use these two measures to compare the ranking resulting from computing the average rank of a population with respect to the criteria in the criterion subset and that of the original, full-criterion population. Intuitively, since the distance between a permutation and itself is 0, we seek a criterion subset that minimises the difference between the two rankings.

Clearly, the identification of the set of criteria which best preserve the average rank of the individuals in a population is a combinatoric optimisation problem which we could solve by examining each possible criterion subset. Unfortunately, as the number of criteria used to describe a population becomes large, an exhaustive approach is infeasible. As a result, in the coming sections we investigate the efficacy of this approach by incorporating it into three methods: a greedy algorithm, a hill climber and a multi-objective formulation of the criterion selection problem. We apply the methods to variants of the GUG09 population, as well as the population of individuals describing the performance of $N = 165$ wireless access points in a mobile telephone network according to $M = 27$ criteria.

6.3. Greedy Criterion Selection

The first algorithm we investigate is a greedy procedure for backward criterion selection. Greedy backward selection is often used as a baseline method in feature selection problems [Bishop, 2007]. It is an iterative procedure which begins with the full criterion set and at each step removes the feature or criterion which minimises the loss of information contained within the data. The greedy backward criterion selection process is described in Algorithm 5. The process begins by producing a vector of ranks $\mathbf{r}$ for the full set of criteria (Line 1). We use average rank for this, however any of the rank aggregation methods discussed in Chapter 3 could be used in its place. At every iteration, each remaining criterion is removed from the population in turn (Line 6), and the resulting population is ranked

Algorithm 5 Greedy Criterion Selection

Require: $\mathbf{Y}$, the initial population consisting of M criteria; Q, the dimensionality to reduce to.

1: $\mathbf{r} := \texttt{rank}(\mathbf{Y})$ *Rank the original population (e.g., average rank)*
2: $\mathbf{Z} := \mathbf{Y}$
3: $q := M$
4: **while** $q > Q$ **do**
5: **for** $m = 1, \ldots, q$ **do**
6: $\mathbf{Z}' := \{\mathbf{Z}(:, l)\}, \forall l (l \neq m)$ *Remove the mth criterion*
7: $\mathbf{r}' := \texttt{rank}(\mathbf{Z}')$ *Rank the new population (e.g., average rank)*
8: $s_m := \delta(\mathbf{r}, \mathbf{r}')$ *Evaluate the quality of the current criterion subset*
9: **end for**
10: $n := \arg\max_m \{s_m\}$ *Find the criterion with the largest distance*
11: $\mathbf{Z} := \{\mathbf{Z}(:, l)\}, \forall l (l \neq n)$
12: $q := q - 1$
13: **end while**

(Line 7); we denote this vector of ranks as $\mathbf{r}'$. These ranks are then compared to those of the original population with the metric denoted by $\delta(\mathbf{r}, \mathbf{r}')$, which is either Spearman's footrule or Kendall's τ metric. At the end of the iteration (Line 11) the criterion whose removal yielded the subset furthest from the original set is removed from consideration, and the next iteration begins.

This is a simple criterion selection method, requiring $\sum_{m=Q}^{M} m = \frac{M}{2}(M+1) - \frac{1}{2}Q(Q+1)$ criterion subset evaluations to arrive at a criterion subset of size Q compared to $\binom{M}{Q}$ exhaustive evaluations. It does, however, only search a narrow region of the possible search space, and so will not necessarily locate the best possible criterion subset. This is because once a criterion has been deleted it is not reconsidered and its contribution to the structure of the data is permanently lost.

6.3.1. Illustration

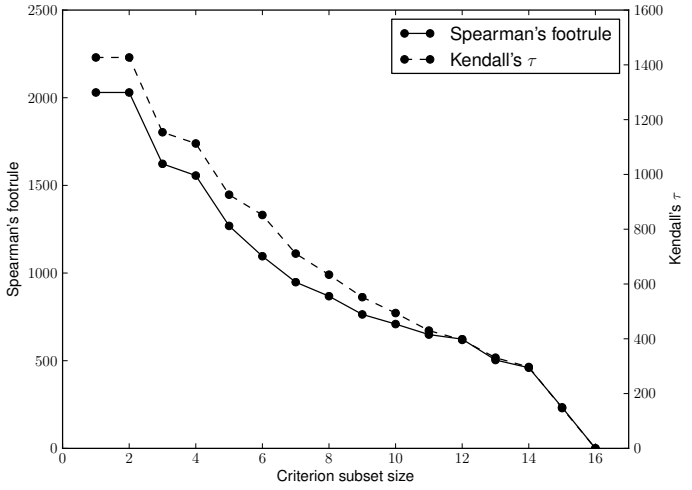
We now present an illustration of the use of Spearman's footrule and Kendall's τ metric in the greedy backward criterion selection algorithm. To begin with, we present two demonstrations of the algorithm using variants of the GUG09 population that we have used as an example in previous chapters. The modified populations contain all $N = 113$ universities along with additional criteria. One of the datasets was constructed by adding a duplicated set of criteria to the original data. Hence, the new dataset consists of 2×8 criteria, and the $(m+8)$th criterion is identical to the mth criterion, of which it is a copy. The second dataset consists of the original data and two artificial criteria, which are well correlated with criteria in the original data. As was done to modify the radar population in Chapter 5, these were created by identifying two pairs of well correlated criteria and averaging each pair to form the new criterion. Criteria 4 and 6 were averaged to produce

Q	Duplicated Criteria		Averaged Criteria	
	SF	KT	SF	KT
15	1	1	-	-
14	9	9	-	-
13	3	6	-	-
12	11	14	-	-
11	6	7	-	-
10	14	15	-	-
9	2	2	8	1
8	10	10	4	10
7	7	5	9	6
6	15	13	3	5
5	5	3	2	7
4	13	11	6	2
3	8	8	5	3
2	16	16	7	9

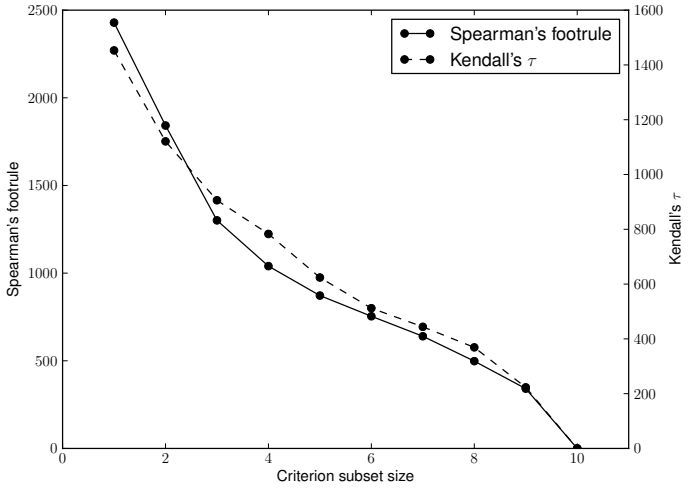
Table 6.1.: The order in which criteria are removed by the greedy backward algorithm for each metric on the two augmented GUG09 datasets. Columns labelled SF and KT show which criterion is removed at each stage for the corresponding metric. In neither case have the duplicated criteria been removed before the original criteria on which they are based, and as such more information is being lost in the criterion selection process.

criterion 9, and criterion 1 and 5 to produce criterion 10. Since the criteria in the GUG09 data are on different scales, they were normalised by converting them to rank coordinates before averaging. The new criteria were then ranked along with the rest of the data.

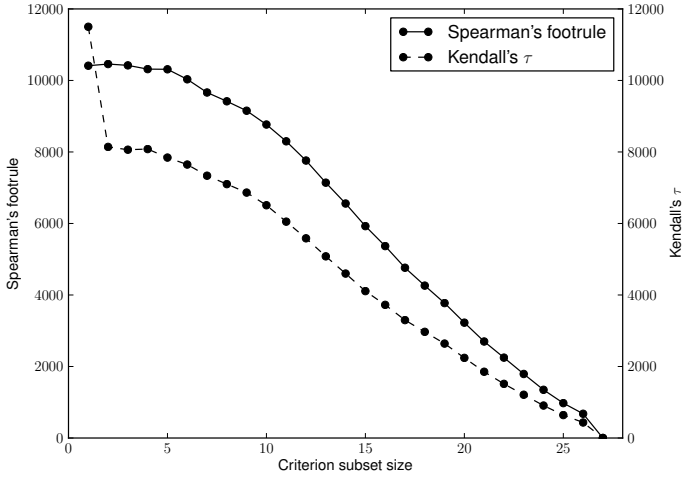
Figure 6.1 shows the distance $\delta(\mathbf{r}, \mathbf{r}')$ between the original population ranking and that of the selected subset as criteria are removed. Two cases are shown. In one, the criterion subset is identified using Spearman's footrule, while in the other the criterion subset is found with Kendall's τ. The plots show the number of remaining criteria Q along with the corresponding information loss quantified by the measure in use (either Kendall's τ metric and Spearman's footrule). Note that these measures are on different scales on the ordinate axes. Figure 6.1(a) shows the results for the duplicated data, in which both measures have produced similar trends, noting their different scales. As may be expected, the greatest distance is when $Q = 2$, since the majority of the criteria have been removed losing most of the original structural information. Table 6.1 illustrates the order in which criteria were removed. For the duplicated criteria, in the case of both Spearman's footrule (SF) and Kendall's τ (KT) the removal of a duplicate criterion is followed immediately by the removal of the criterion on which it was based because the average rank of a candidate criterion subset is compared with the average rank of the original population (comprising 16 criteria). A preferred result would be for the set of duplicate criteria to be removed, leaving only the original criteria; this would preserve all of the information present in the data, whereas some information is lost in the sequence shown here. Likewise, the two synthetic criteria would ideally be removed first; as shown in Table 6.1, that is not the



(a) Duplicated criteria



(b) Averaged criteria



(c) AP data

Figure 6.1.: The information loss changes for criterion subsets of increasing size. Note that although both Kendall's τ metric and Spearman's footrule follow a similar trend for all three datasets, they have different ordinates.

case. The order is different when the difference in ranks is measured with Kendall's τ; we recall that the two measures are well correlated but not identical [Diaconis and Graham, 1977].

Figure 6.1(b) presents analogous results for the averaged data. In common with the duplicated data, the two measures yield very similar results. As with the duplicated data, the worst case occurs when $Q = 2$, and from examining Table 6.1, the algorithm mostly identifies and removes the redundant criteria (the two averaged criteria), although we note that the instance of the method using Spearman's footrule does not remove the second of the redundant criteria, criterion 10. It might be expected that once the artificially created criteria have been removed from the two datasets that the sequence in which the remaining eight original GUG09 KPIs are removed would be the same. By examining Table 6.1 this is clearly not the case. This is because the comparison is made between the current criterion subset and the original, full, criterion set. Since the average ranks for these two datasets are different, and from those of the original GUG09 data, so is the sequence in which the criteria are removed.

Finally, we demonstrate the method on a larger population, the population of individuals describing the performance of wireless access points in a mobile telephone network. This population comprises 27 criteria, and it is clearly desirable to reduce these criteria to a more manageable amount. The results shown in Figure 6.1(c) illustrate the effect of applying the greedy method, and show that again a degree of similarity between Spearman's footrule and Kendall's τ is present. The sequence in which criteria are removed by the two measures are similar, although in some places there is some disagreement. For example, having removed the first nine criteria Spearman's footrule removes criteria 8 and 24 while Kendall's τ removes 24 and then 8.

6.4. Hill Climber Criterion Selection

Various single-objective algorithms have been used for feature selection [e.g. Oh et al., 2004; Jarmulak and Craw, 1999]. They explore different combinations of criteria and, unlike greedy backwards feature selection, allow for the reintroduction of criteria that have previously been discarded. This results in a more thorough search of the space than was possible with the greedy backward selection algorithm.

We employ a basic hill-climber, which is outlined in Algorithm 6. A solution $\boldsymbol{\theta}$ is represented as a M-bit string, where $\theta_m = 1$ indicates that the mth criterion is selected and $\theta_m = 0$ that it is not. An initial solution is generated by selecting Q criteria at random (Line 2). Each iteration begins by mutating the current solution (Line 6). This is done by deselecting one of the criteria in the bit string and selecting one of the currently deactivated ones. Initialising the solution and mutating it in this way ensures that the

Algorithm 6 Hill-climber Criterion Selection

Require: $\mathbf{Y}$, the initial population consisting of M criteria; Q, the dimensionality to reduce to.

1: $\mathbf{r} := \texttt{rank}(\mathbf{Y})$
2: $\boldsymbol{\theta} := \texttt{initialise}(M, Q)$
3: $\mathbf{Z} := \mathbf{Y}(:, \boldsymbol{\theta})$ — Initially selected criteria
4: $\phi := \delta(\mathbf{r}, \texttt{rank}(\mathbf{Z}))$ — Evaluate initial solutions
5: **repeat**
6: $\quad\boldsymbol{\theta}' := \texttt{mutate}(\boldsymbol{\theta})$
7: $\quad\mathbf{Z}' := \mathbf{Y}(:, \boldsymbol{\theta}')$ — Select active criteria
8: $\quad\phi' := \delta(\mathbf{r}, \texttt{rank}(\mathbf{Z}'))$ — Evaluate quality of current subset
9: $\quad$**if** $\phi' < \phi$ **then**
10: $\quad\quad\boldsymbol{\theta} := \boldsymbol{\theta}'$
11: $\quad\quad\phi := \phi'$
12: $\quad$**end if**
13: **until** stopping condition met (200 iterations)

constraint

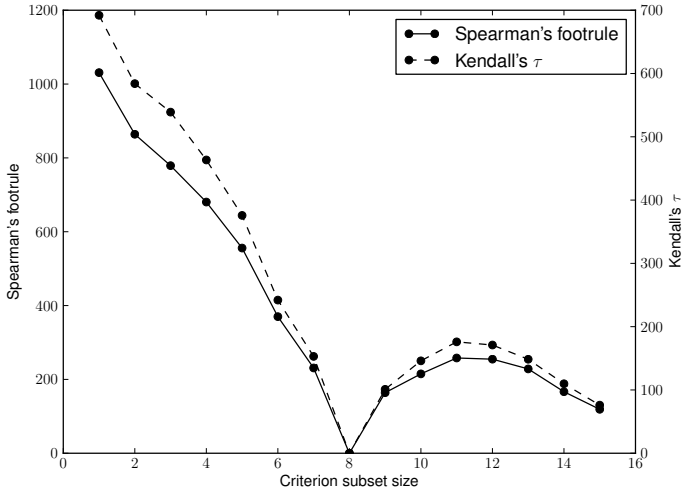
$$\sum_{m=1}^{M} \theta_m = Q \tag{6.1}$$

is satisfied, and that any subset generated contains the correct number of criteria.

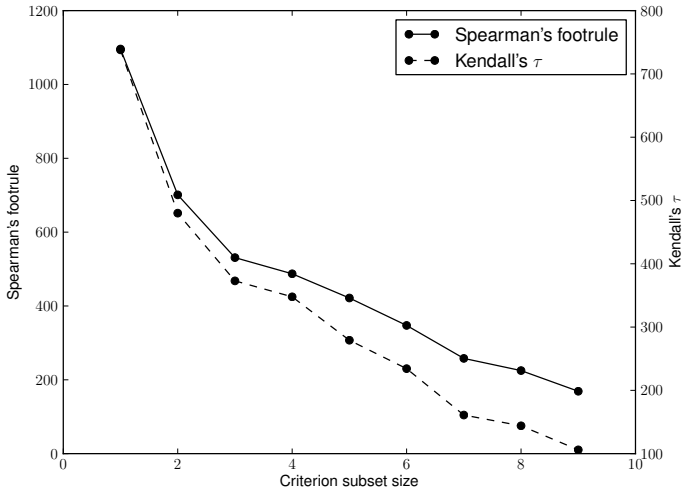
As in the greedy algorithm, the quality of a solution is evaluated by ranking the population with respect to the currently selected criteria and computing the distance between that ranking and the ranking of the full-criterion set using either Spearman's footrule or Kendall's τ (Line 8). If $\delta(\mathbf{r}, \mathbf{r}')$ is less than the current best solution, then the current best is replaced by the new solution (Line 10).

Experimental results for the hill-climber on the augmented datasets are shown in Figure 6.2 and can be compared to Figure 6.1. The algorithm was run 20 times for each $Q = 2$ to $M - 1$. The hill-climber has clearly outperformed the greedy method in terms of finding a criterion subset which preserves the original ranking well under the Spearman's footrule and Kendall's τ. We note that in the case of $Q = 8$ for the modified GUG09 population with duplicated criteria the distance between the permutations is 0. This indicates that exactly one copy of each of the original criteria remains in the criterion subset and the average rank is therefore the same as the original population. As the hill-climber is able to search a wider portion of the search space, it has identified subsets whose average ranks are closer to that of the original populations; this is evident when comparing the smaller ordinate ranges of this experiment with those of Figure 6.1.

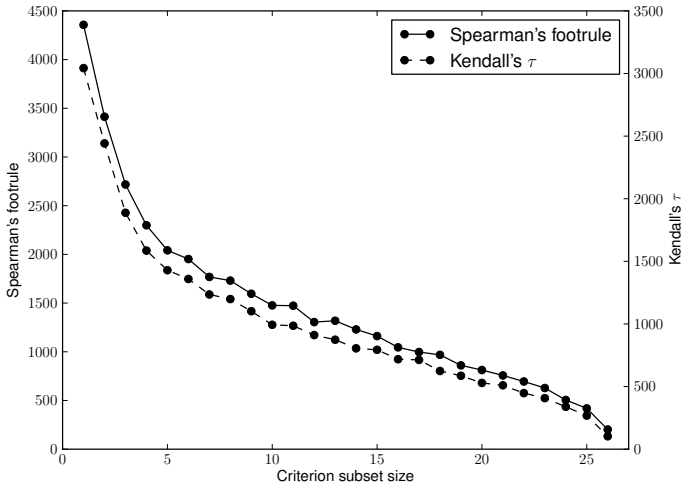
Figure 6.3 shows how the change in distance between permutations changes during the execution of the algorithm for $Q = 4$. Both the τ metric and Spearman's footrule have converged to the optimal criterion subset in 200 iterations, and no further progress is made. The process is therefore quite cheap to run as few iterations are required to arrive



(a) Duplicated criteria

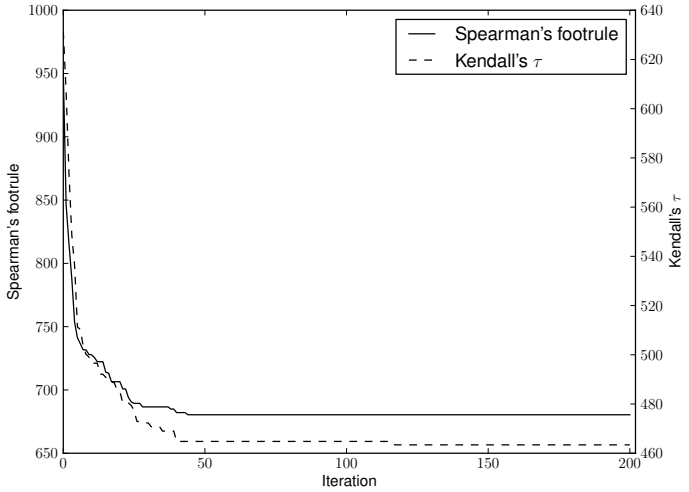


(b) Averaged criteria

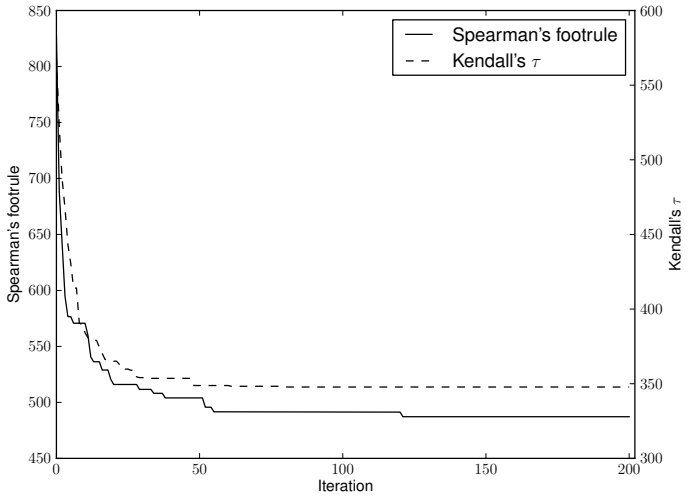


(c) AP data

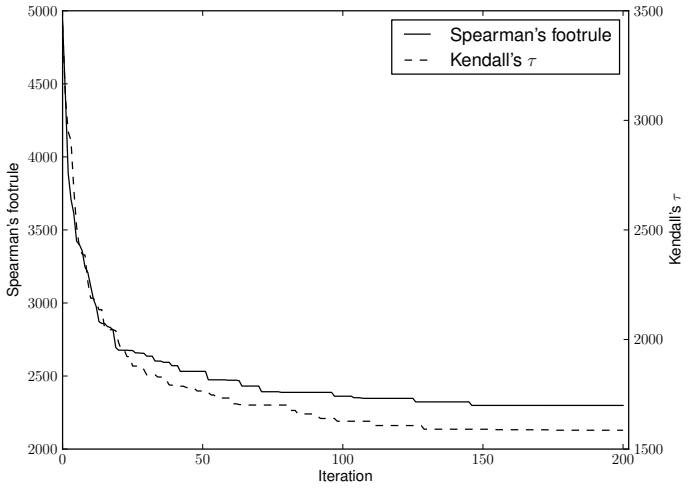
Figure 6.2.: The results of criterion selection with a hill-climber for increasing values of Q. Whilst the trends are similar to those of the greedy algorithm, the error loss is substantially lower.



(a) Duplicated criteria



(b) Averaged criteria



(c) AP data

Figure 6.3.: Hill-climber convergence results for $Q = 4$, showing how the algorithm is converging towards the optimal criterion subset containing four criteria. The algorithm has converged by 200 iterations.

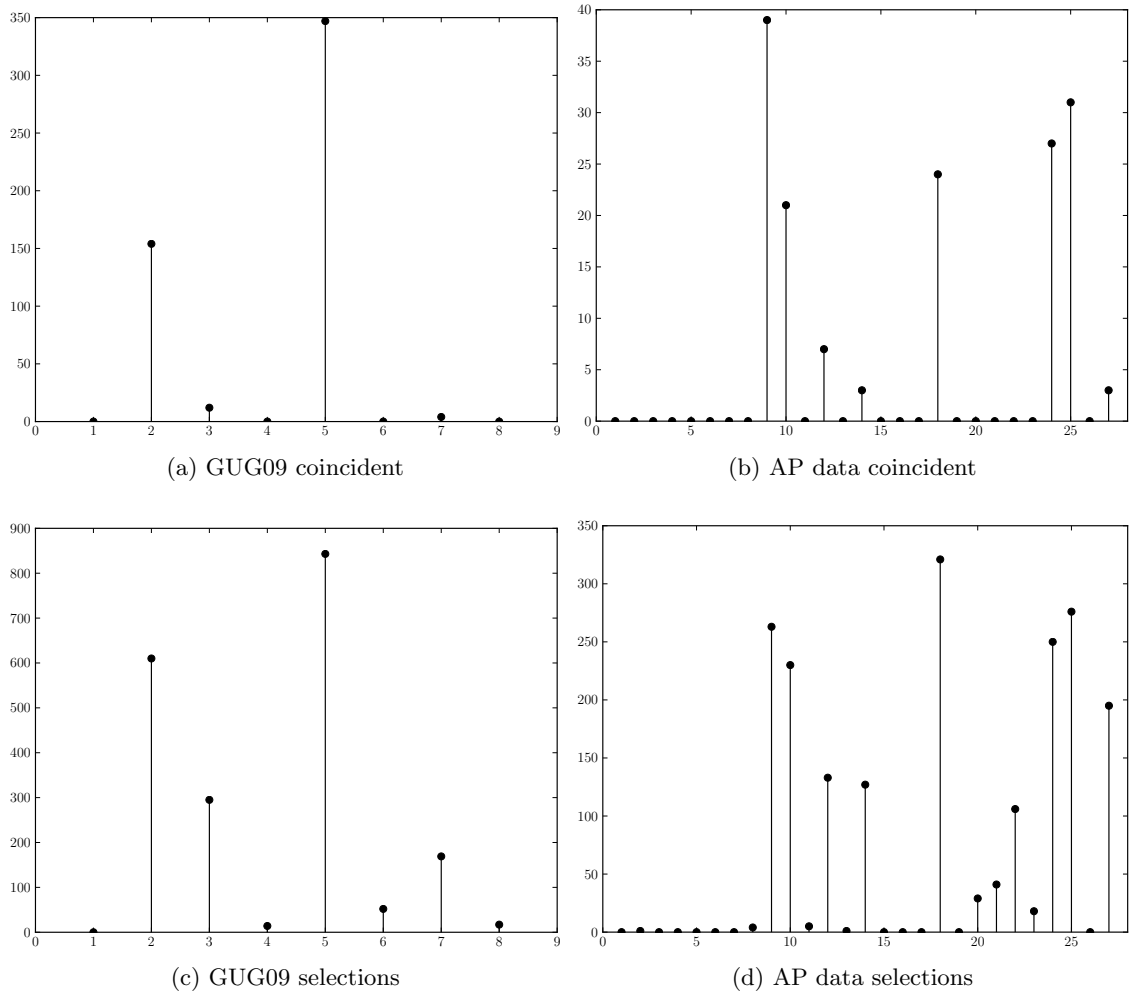
at a sensible solution, although it does not guarantee to result in an optimal criterion subset. Exhaustive search for a four-criterion subset of the duplicated GUG09 data would require $\binom{16}{4} = 1820$ function evaluations, and for the AP data this would be $\binom{27}{4} = 17550$.

Since the greedy algorithm is deterministic it is possible to examine which criteria remain in the final Q-dimensional criterion subset. The hill climber is a stochastic process, so it is not possible to evaluate the results in the same way. To resolve this, we consider an approach taken by Li et al. [2002] in which they investigate which of 2000 possible features are the most relevant for classifying cancer cases. Instead of running the hill climber once, the population is randomly split into two equal halves and the hill climber run on each split. This is repeated for some number of splits at the end of which we identify how often each criterion was one of the Q selected criteria at the end of the optimisation *in both splits*. The criteria which most often appear in both sets are then selected. An alternative method is to simply count the number of times each criterion is selected in either set. In addition to guarding against the stochastic nature of the hill-climber we note that this splitting procedure also reduces the chance that a criterion will appear unrealistically important because of a small number of individuals in the population.

The hill-climber was run on 500 splits of the both the original (8-criterion) GUG09 population and the AP data. The top row of Figure 6.4 illustrates the coincident selections for populations. In the case of the GUG09 population, there are two criteria which are clearly the most important according to the hill-climber method (criteria 2 and 5, research quality and entry standards respectively). Five of the criteria in the AP data are also clearly more important to determining the rank of an individual than the others. From examining the bottom row of Figure 6.4 we can see that in general the same criteria are important, however some of the criteria that were not selected in both splits of the data were still selected independently. We note that while the examples shown here actually require more than the $\binom{M}{2}$ evaluations required for exhaustive search, the splitting procedure is more beneficial when used to find criterion subsets larger than those shown here for which the number of function evaluations required to do an exhaustive search becomes extremely large.

6.5. Multi-objective Criterion Selection

A drawback of employing a hill-climber such as the one described previously is that the user must decide *a priori* how to set the parameter Q, the maximum number of criteria to retain in the low-dimensional data. A plot in the style of Figure 6.2 can be produced by running the hill-climber for multiple settings of Q, but this requires multiple independent runs of the algorithm. An alternative is to relax the constraint governing the size of Q to an objective and employ a MOEA to simultaneously trade-off information loss and the number of remaining criteria.



(a) GUG09 coincident

(b) AP data coincident

(c) GUG09 selections

(d) AP data selections

Figure 6.4.: Plots showing the frequency that a criterion is selected in both splits of the population in the Li et al. [2002] procedure, applied to the GUG09 population ((a) and (c)) and the AP data ((b) and (d)). The top row shows the number of times that a criterion is selected in both splits of the population and the bottom row shows the number of times a criterion is selected in either.

Evolutionary multi-objective optimisation is a popular method for selecting criteria from a dataset, and has been used in a variety of applications including handwritten digit and letter recognition [Oliveira et al., 2002; Morita et al., 2003], diagnosing faults in industrial machinery [Emmanouilidis, 2002] and for improving the results of clustering in unsupervised learning tasks [Mierswa and Wurst, 2006]. A variety of objectives have been investigated. Usually, a pair of objectives includes some measure of the effect of removing the redundant criteria, as well as a count of the remaining criteria [Oliveira et al., 2002], however other approaches have optimised the receiver operating characteristic (ROC) curve of a classifier [Emmanouilidis, 2002] and where the aim is to produce an effective clustering of the data, the within- and between-class spread of the clusters has been considered [Mierswa and Wurst, 2006; Morita et al., 2003].

We define the following multi-objective problem. The solution $\boldsymbol{\theta}$, representing the retained

Algorithm 7 Multi-objective Dimension Reduction

Require: $\mathbf{Y}$, the initial population consisting of M criteria.

1: $\mathbf{r} := \texttt{rank}(\mathbf{Y})$
2: $\boldsymbol{\theta} := \texttt{initialise}(M)$ *Select a random number of criteria*
3: $\mathbf{Z} := \mathbf{Y}(:, \boldsymbol{\theta})$ *Initially selected criteria*
4: $\boldsymbol{\phi} := \mathbf{f}(\boldsymbol{\theta})$ *Evaluate initial selection*
5: $E := \{\boldsymbol{\theta}\}$ *Initialise archive*
6: **repeat**
7: $\quad \boldsymbol{\theta}' := \texttt{mutate}(\boldsymbol{\theta})$
8: $\quad \mathbf{Z}' := \mathbf{Y}(:, \boldsymbol{\theta}')$ *Select the active criteria*
9: $\quad \boldsymbol{\phi}' := \mathbf{f}(\boldsymbol{\theta}')$ *Evaluate the objective functions*
10: $\quad E := \texttt{update}(\boldsymbol{\theta}', \boldsymbol{\phi}')$
11: $\quad \boldsymbol{\theta}, \boldsymbol{\phi} := \text{select}(E)$ *Select a member of the archive at random*
12: **until** stopping condition met

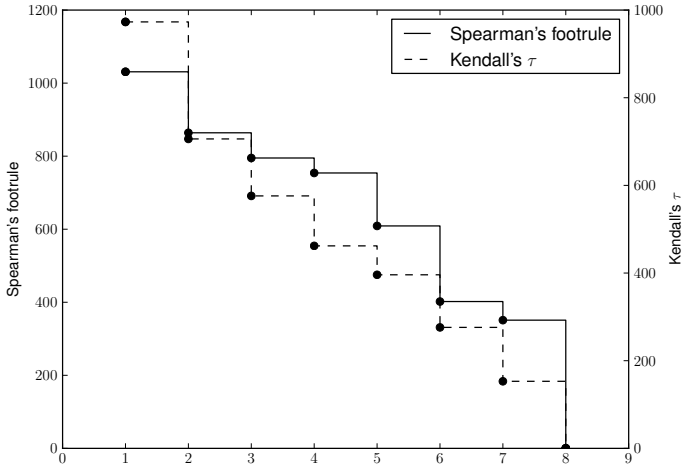
criteria, is again represented as a M-bit string, which now maps to a pair of objectives:

$$f_1(\boldsymbol{\theta}) = \delta(\mathbf{r}, \texttt{rank}(\mathbf{Z})) \tag{6.2}$$

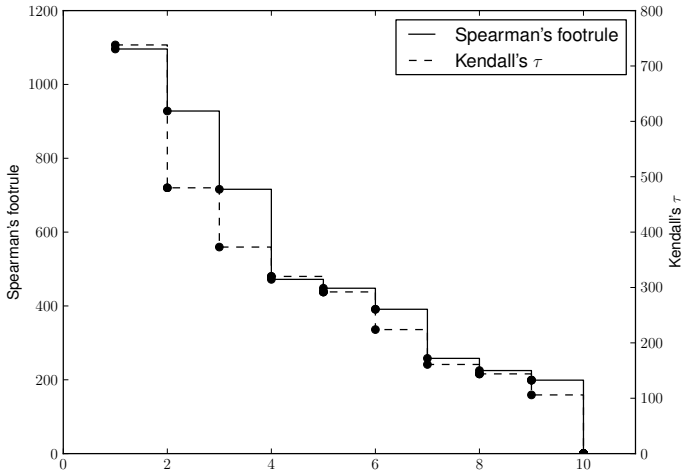
$$f_2(\boldsymbol{\theta}) = \sum_{m=1}^{M} \theta_m. \tag{6.3}$$

where $\mathbf{Z}$ in the first objective is a copy of the population $\mathbf{Y}$ described by the criteria selected in $\boldsymbol{\theta}$. The first objective is the distance between permutations as before while the second objective counts the number of remaining criteria. Algorithm 7 presents the multi-objective criterion selection algorithm. As with the single-objective hill-climber, the multi-objective version begins by initialising a random bit string (Line 2). As the dimensionality is now an objective rather than a constraint, this is done by selecting a random number of criteria. This solution is then evaluated against the two objectives, and the evolutionary process begins. A slightly different mutation operator is employed which flips bits with probability $1/M$ (Line 7). A archive E of solutions is maintained, which contains all of the currently non-dominated solutions found by the algorithm. Once the objective values of the new solution have been evaluated, it is compared to the solutions in the archive. Any solutions which it is found to dominate are removed, and if the new solution is not dominated by any solutions in the archive, it is added (Line 10). The child solution for the next generation is then chosen by selecting a member of the elite archive E at random (Line 11). This procedure continues until a stopping criterion is reached, and the decision maker is provided with the estimated Pareto front from which they must select a criterion subset. In the work reported here, the algorithm runs for 2000 generations which is ample to ensure convergence.

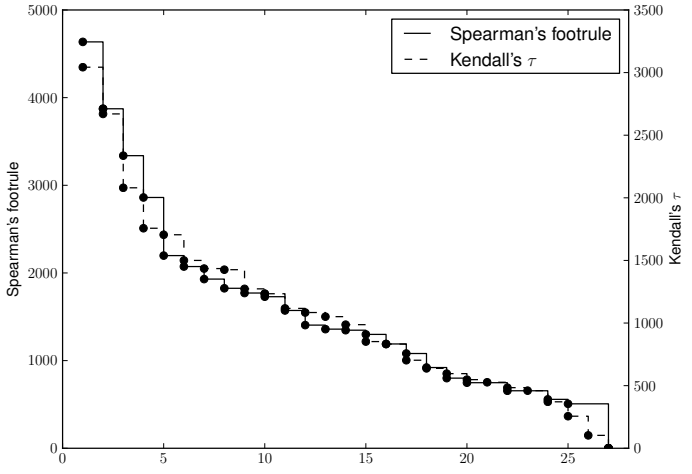
Figure 6.5 shows best attainment surfaces for the optimisation. Results for 10 independent runs of each metric have been merged to produce a single archive containing all of the non-dominated solutions found during the 10 runs. For brevity, the attainment surfaces have been presented on the same plot, however as with earlier figures these plots have separate ordinates corresponding to the metrics. It is therefore not possible to infer the



(a) Duplicated data



(b) Averaged data



(c) AP data

Figure 6.5.: The trade-off curve produced by optimising the number of criteria against the information loss as measured by one of the metrics. Note that since the metrics are on different scales, we cannot infer that one front dominates the other.

dominance relationship between solutions on the two attainment surfaces.

By using the approximately Pareto optimal criterion subsets, we can compare the solutions identified by the algorithm. In the duplicated GUG09 population, both Spearman's footrule and Kendall's τ identify the same criterion subset in most cases. In the case where $Q = 8$, where the information loss is minimised, the subset contains exactly one instance of each criterion, either the original criterion or its copy. None of the solutions for $Q > 8$ are included in the solution set for this multi-objective instance because they are dominated by the $Q = 8$ solution.

For the averaged GUG09 population, the smaller subsets prefer to include the additional criteria formed by averaging original criteria. We infer that this is because they were constructed from two of the original criteria, so that retaining a composite criterion preserves more information about the rank structure of the data than either individual criterion. As the size of the criterion subsets increase, there is sufficient influence from the original criteria, and the averaged criteria are no longer included; they are only present when all of the original criteria have been included. When $Q = 8$, the subset comprises all 8 original GUG09 criteria. The results shown in Figure 6.5(c) show that the MOEA has identified a similar trend to that of the repeated runs of the hill-climber for different values of Q. Although it is not very pronounced, we note that a knee has started to form around the criterion subset of size 5 in the estimated Pareto front.

Comparisons with Figure 6.2 show that distances which are at worst equivalent to those identified by running the hill-climber for fixed values Q repeatedly are identified by the MOEA. It is therefore unnecessary to repeat the hill-climber for each value of Q, which could be prohibitively expensive for a large dataset as a more complete view of the trade-off can be identified with a single run of the MOEA. That said, we note that there is less of a computational advantage in using the multi-objective version of the hill-climber in the populations we have presented here.

6.6. Conclusion

This chapter has introduced methods for revealing which criteria are most responsible for the structure in a many-criterion population. We introduced a method in which the structure of a population was quantified using the rank of individuals, which we sought to preserve; criteria were discarded so that the distance between the average rank of the full dataset and the selected criteria was minimised. We measured the distance between different rankings of the population using Spearman's footrule and Kendall's τ which are known to be well correlated and produced similar results. Unsurprisingly, a hill-climbing algorithm outperforms a greedy backwards search, while a multi-objective algorithm allows the full trade-off surface between fidelity of the selected criteria and the number of criteria

to be located in a single evolutionary optimisation.

We have demonstrated the efficacy of the algorithm at removing redundant criteria. Applying the criterion selection process to the GUG09 population without synthetically redundant criteria finds that the two most significant criteria, contributing the most to the overall structure, are research quality and entry standards. In addition to providing an understanding of the structure of the data, it also allows for a better visualisation; a visualisation of individuals in their Pareto shells is easier to interpret for a smaller number of criteria.

A logical extension to this work would be to incorporate the rank-based approach presented in this chapter into the selection operator of an evolutionary algorithm tasked with optimising a many-objective problem. It would also be useful to see how the proposed algorithms fare when dealing with populations of criteria larger than the 27-criterion AP data used as an illustration in this chapter.

7. Conclusion

7.1. Introduction

This thesis has presented methods for visualising and understanding many-criterion populations. This chapter summarises the main contributions in these two areas and gives pointers to possible future avenues of exploration.

7.2. Visualising Many-criterion Populations

A common approach to understanding datasets is to use a visualisation technique and the visualisation of high-dimensional data is known to be difficult; conventional visualisation tools such as scatter plots do not work for datasets of more than three dimensions. In this work we are interested in visualising many-criterion populations, and it is particularly important for such a visualisation to convey a notion of the relative quality of individuals in the population. We have developed techniques for presenting many-criterion populations using the dominance relations that characterise them as well as the fact that the individuals can be ranked according to the criteria.

In Chapter 3 we evaluated various methods for ranking many-criterion populations for the purpose of visualisation. We used the Pareto sorting technique commonly used for ranking populations of solutions in multi-objective evolutionary algorithms (MOEAs) as the basis for visualising the individuals as a graph; each individual is sorted into a Pareto shell, and dominance relations between individuals in adjacent shells are shown. To further enhance the visualisation we explored other methods for ranking many-criterion populations (average rank, outflow, average shell, stationary distribution and the power index), extending some of these methods into the multi-criterion domain with the introduction of a generalised tournament matrix (on which the ranking methods are based) which uses dominance to determine the outcome of tournaments. We note that future ranking methods based on a GTM will be applicable to multi-criterion populations as a result of this development. Each of the additional ranking methods we investigated was used to colour the nodes in the graph. This revealed additional information about the nature of the individuals in the population, such as identifying individuals who reside in a strong Pareto

shell by specialising on a single criterion as opposed to those which have a good score on the entire set of criteria. A shortcoming of the approach is its reliance on dominance for arranging the graph. In a high-dimensional space dominance is unable to distinguish between the quality of individuals. All of the population therefore sorts into a single Pareto shell and less structural information is revealed. That said, we have demonstrated that for some many-criterion populations (our main example population, the GUG09 population, comprises 8 criteria) it is possible to produce a useful visualisation. For that reason, and because of the current interest in criterion reduction reported in Chapter 6, we do not feel that such a shortcoming makes this an infeasible choice for visualising many-criterion populations. A future area of study is to incorporate this method into an interactive visualisation tool. We feel that the ability to observe how the dominance relationships between individuals change as different combinations of the criteria comprising a population are tested would be of value to a decision maker. Enabling the decision maker to change the ranking method on which the nodes are coloured might also be a useful feature. The computational complexity of the Pareto sorting, and the eigendecomposition required to rank the population with the power index means that, even for relatively large populations, the visualisation could be redrawn rapidly.

In investigating the character of a population of individuals it is natural to consider distances between pairs of individuals. We introduced a measure of distance that captures the essence of the dominance relations between individuals. This distance has an embedding in Euclidean space, and as such can be used to provide the basis for a visualisation. A common approach to visualising high-dimensional data is to use metric MDS; Euclidean distance is often used as the basis for the dimension reduction. We therefore used MDS and the dominance distance to embed the population in a low-dimensional space so that we could produce a scatter plot of the population in two dimensions. We demonstrated this method on a range of populations and showed that the resulting mapping captures the main trends in the power index or average rank of the individuals. Also, whereas the Pareto shell visualisation does not scale to large numbers of criteria, this method was shown to be suitable for visualising large populations; a useful visualisation of a 27-criterion population was shown. Both the dominance distance and the MDS embedding are computed in polynomial time, so it is feasible that this method could be used interactively. Showing how the arrangement of individuals change with different combinations of criteria will provide useful information about how the criteria of a population are related, which will be useful as populations with larger numbers of criteria occur.

MDS is a linear dimension reduction technique, however various nonlinear dimension reduction techniques have been developed. These methods, such as locally linear embedding [Roweis and Saul, 2000], isomap [Tenenbaum et al., 2000], Laplacian eigenmaps [Belkin and Niyogi, 2001] and maximum variance unfolding [Weinberger and Saul, 2006] have been shown to provide a more faithful representation of a dataset in some cases, and as such future work will investigate the use of these methods in concert with the dominance

distance for visualising many-criterion populations.

Another common visualisation technique is to present a dataset as a heatmap, representing criterion values as coloured cells in a grid. In a heatmap visualisation of a many-criterion population, each individual is represented as row and each criterion a column. A problem with such a visualisation is that the arbitrary ordering of individuals and criteria can make the heatmap difficult to interpret; we presented a spectral seriation method to enhance the clarity of a heatmap which was shown to be quite effective on a range of populations. Seriation uses a measure of similarity to place similar individuals and criteria close together in a permutation, which is used to reorder the heatmap. One of the strengths of this method is that a similarity measure can be chosen to suit any particular type of population; we demonstrated similarity measures which take advantage of the fact that the individual criteria in a population can be ranked. An additional advantage is that by presenting two heatmaps side by side, parameter space and objective space, in which solutions to a many-objective optimisation problem reside, can be visualised. We demonstrated the trade-off between preferring individual ordering according to parameter space or objective similarity and illustrated a simple technique for optimising the ordering with respect to both. As many-objective algorithms mature, we expect that the ability of this technique to illustrate large populations (in terms of the number of individuals and criteria) will prove particularly useful for visualising the large number of solutions needed to properly cover a many-objective Pareto front. As with the previous two visualisation methods, the computational complexity of this method means that it can be recomputed relatively rapidly. While we note that, as discussed in Chapter 2, research in the MCDM literature has found that heatmaps are less well suited to interactive manipulation than other methods (e.g., parallel coordinate plots), there is potential for interaction in allowing the user to further rearrange the seriated individuals and criteria. Such a system might also allow the user to seriate according to different similarity measures to evaluate the difference.

The visualisation work presented in this thesis has concentrated on static populations, such as a population of solutions in a single generation of an evolutionary algorithm, or league table data describing university performance for a particular year. This work could be extended to incorporate the temporal aspect of many-criterion populations. One possibility would be to develop methods for visualising the lineage of solutions by extending the 2-dimensional Pareto shell graph into a third dimension representing time. This would prove useful in the maintenance application described in Chapter 3 in that an engineer could see that not only is an individual exhibiting poor performance according to the KPIs on which they are evaluated, but they could also see how long they have been performing poorly to spot anomalies and trends more easily.

7.3. Understanding Many-criterion Populations

In addition to visualising many-criterion populations we have also investigated methods for exploring their structure. The structure of a many-criterion population can be characterised in terms of the rank of individuals as well as the pairwise dominance relations between individuals. The work presented in this thesis was particularly concerned with identifying landmark individuals within the population as well as identifying the most important criteria.

In Chapter 4 we presented three dominance-based candidate definitions for determining which individuals are on the edge of a mutually non-dominating population. Identifying which individuals are extreme on particular criteria provides a useful way of understanding the structure of a population. Of the three methods, the first defined the edge in terms of the attainment surface. While it was able to adequately find the edge of a concave population, the geometry of a convex population meant that few of the individuals were found. The second method defined the edge in terms of an embedding of the individuals in the plane which was then rotated in order to find those which were non-dominated in the embedding. This was able to find the edge of a multi-criterion population but it was not scalable to many-criterion populations because of its use of dominance in a relatively high-dimensional space. The final definition was in terms of criterion subsets. This method was able to identify the edge of both multi- and many-criterion populations because of the criterion subset approach taken, and was the most promising of the three candidate definitions. Additional work is warranted to investigate methods for finding the edge of a mutually non-dominating population, particularly where the individuals are described by a large number of criteria. As discussed in Chapter 4, we believe a useful aspect of this work might be its inclusion into the fitness assignment process of an evolutionary algorithm, to ensure that the extent a search population is expanded to properly cover the true Pareto front of a many-objective optimisation problem.

We have presented two approaches to understanding population structure with the rank of individuals. The first was an investigation of the structural information provided by ranking methods such as the power index. Through the visualisation methods described above we were able to reveal similarities between individuals, as well as identify outliers and fill in missing data. Some of the methods investigated, such as Pareto sorting and average rank, were drawn from the evolutionary multi-objective optimisation literature. The work presented in this thesis could be extended by applying novel methods, such as the power index and dominance distance, to solving multi- and many-objective optimisation problems. While it is likely that a selection operator based on the power index will result in premature convergence and poor diversity in the estimated Pareto front because of its similarity to average rank, which is known to cause the premature convergence of a search population, the dominance distance might be used as a niching method by removing solutions close to those which have already been selected.

Our second rank-based approach to revealing the structure within a many-criterion population involved using the change in an individual's rank to identify redundant criteria. We define a redundant criterion as one whose removal does not dramatically change the rank of the individuals in the population. We demonstrated different criterion selection approaches (a greedy algorithm, a stochastic hill-climber and a multi-objective evolutionary algorithm) and illustrated how the dimensionality can be reduced for the purposes of visualisation. We also provided a short discussion on its use in an evolutionary algorithm so that standard dominance-based selection methods can be used to solve a many-objective optimisation problem. An extension of this work would be to investigate such an approach, applying it to both test problems and real-world optimisation problems.

7.4. Summary

This thesis has investigated the visualisation and understanding of many-criterion populations, and we have provided a useful set of linear methods for understanding static populations. We believe that there is much interesting work ahead in terms of investigating nonlinear approaches, applying them to dynamic populations which incorporate a notion of time and developing tools to facilitate the use of these methods interactively.

A. Times Good University Guide 2009

The tables on the following pages reproduce the Times Good University Guide from 2009 [O'Leary, 2009], which has been a running example throughout this thesis.

University	NSS	Research quality	Student staff ratio	Services and facilities spend	Entry standards	Completion	Good honours	Graduate prospects	GUG Score
Oxford	0.840	6.200	11.600	2884.000	502.000	98.600	90.100	83.900	1.000
Cambridge	-	6.500	12.200	2299.000	518.000	97.900	85.400	88.400	2.000
Imperial	0.760	5.800	10.400	3218.000	473.000	96.000	69.100	89.300	3.000
LSE	0.740	6.300	12.600	1562.000	469.000	96.900	75.200	87.700	4.000
St Andrews	0.820	5.300	12.600	1162.000	446.000	94.800	83.900	73.700	5.000
Warwick	0.760	5.600	13.600	1881.000	448.000	96.700	79.400	74.900	6.000
UCL	0.760	5.500	9.100	1702.000	434.000	94.300	75.100	81.500	7.000
Durham	0.780	5.200	15.400	1375.000	447.000	96.400	78.800	75.900	8.000
York	0.770	5.500	13.100	1313.000	423.000	95.200	74.700	70.500	9.000
Bristol	0.750	5.200	14.700	1535.000	430.000	95.800	78.400	81.500	10.000
King's	0.770	4.700	11.900	1696.000	406.000	93.200	72.100	80.400	11.000
Loughborough	0.830	4.300	17.100	1293.000	361.000	94.000	67.400	73.200	12.000
Exeter	0.810	4.700	16.800	1183.000	381.000	94.800	79.800	68.500	13.000
Leicester	0.830	4.500	14.500	1329.000	360.000	92.900	69.000	72.300	14.000
Bath	0.750	5.200	16.600	1291.000	428.000	95.300	77.300	81.000	15.000
Nottingham	0.750	5.000	13.800	1390.000	403.000	96.200	75.700	76.000	16.000
Southampton	0.790	5.400	16.300	1479.000	389.000	90.700	74.800	71.800	17.000
Edinburgh	0.730	5.000	13.300	1294.000	430.000	92.200	79.900	74.900	18.000
Lancaster	0.780	5.400	12.700	1227.000	375.000	92.500	68.800	60.900	19.000
Newcastle	0.750	4.400	14.900	1481.000	394.000	92.300	71.100	75.300	20.000
Glasgow	0.780	4.300	13.400	1373.000	396.000	85.500	68.400	75.000	21.000
Sheffield	0.760	4.500	14.500	1140.000	403.000	92.400	71.900	73.700	22.000
East Anglia	0.820	5.000	17.900	1127.000	359.000	91.200	67.500	63.000	23.000

SOAS	0.710	5.300	10.500	1746.000	368.000	84.400	73.400	68.000	24.000
Birmingham	0.770	4.300	15.300	1342.000	402.000	92.400	68.400	70.400	25.000
Aberdeen	0.800	4.000	13.500	1168.000	350.000	78.800	69.700	76.200	26.000
Manchester	0.730	5.100	12.800	1378.000	412.000	92.300	70.500	67.400	27.000
Aston	0.760	3.900	17.400	1607.000	351.000	88.400	65.800	76.400	28.000
Cardiff	0.770	4.500	14.300	1085.000	381.000	90.100	67.600	73.500	29.000
Royal Holloway	0.740	5.200	14.700	1244.000	361.000	88.200	66.400	66.900	30.000
Leeds	0.750	4.500	14.600	1014.000	385.000	92.100	71.800	67.300	31.000
Reading	0.780	4.900	16.200	1012.000	340.000	91.700	71.700	63.300	32.000
Queen's Belfast	0.770	4.300	15.300	1222.000	364.000	86.600	70.300	76.100	33.000
Liverpool	0.750	4.600	13.300	1013.000	377.000	90.000	67.600	70.700	34.000
Strathclyde	0.760	3.600	16.800	1219.000	367.000	81.800	74.600	76.400	35.000
Kent	0.790	4.000	16.300	1143.000	307.000	87.000	59.700	70.500	36.000
Queen Mary	0.750	4.700	12.800	1173.000	340.000	88.500	60.800	75.100	37.000
Sussex	0.690	5.100	15.000	1136.000	387.000	88.300	79.500	62.000	38.000
Surrey	0.720	4.700	16.300	1156.000	339.000	86.800	61.900	77.300	39.000
Stirling	-	3.900	15.400	964.000	288.000	89.900	67.900	71.800	40.000
Keele	0.750	4.000	15.300	1048.000	313.000	84.300	64.000	77.600	41.000
Essex	0.750	4.800	14.100	1149.000	309.000	83.400	60.400	62.800	42.000
Aberystwyth	0.810	4.000	19.000	997.000	296.000	89.800	61.800	56.200	43.000
Dundee	0.780	4.200	13.700	971.000	373.000	68.700	66.300	73.900	44.000
Hull	0.800	3.200	18.600	934.000	297.000	83.900	56.400	71.000	45.000
Goldsmiths	0.730	4.900	15.500	793.000	307.000	82.200	64.800	63.600	46.000
Heriot-Watt	0.750	4.200	16.600	1115.000	343.000	82.300	55.000	72.100	47.000

Swansea	0.770	4.400	15.100	1010.000	295.000	88.300	49.900	61.400	48.000
Bradford	0.760	3.400	15.000	1167.000	280.000	82.200	60.500	73.900	49.000
City	0.720	3.700	18.100	878.000	321.000	85.000	63.700	77.100	50.000
Bangor	0.770	3.900	18.400	1023.000	279.000	77.000	55.600	65.900	51.000
Brunel	0.690	3.100	16.300	1475.000	313.000	85.400	63.700	63.600	52.000
Ulster	0.750	2.400	16.400	1462.000	278.000	77.000	60.100	62.100	53.000
Robert Gordon	-	0.900	17.900	1070.000	321.000	79.800	58.800	77.600	54.000
Oxford Brookes	0.760	1.600	18.200	944.000	300.000	84.900	59.100	65.900	55.000
N'ham Trent	0.700	1.400	18.000	990.000	269.000	89.100	56.400	74.000	56.000
Bournemouth	0.720	0.700	19.500	967.000	287.000	85.200	57.000	70.800	57.000
Gloucestershire	0.730	1.700	17.200	1074.000	258.000	80.500	54.900	60.200	58.000
Chichester	0.780	1.700	19.600	733.000	253.000	88.000	46.700	55.500	59.000
Brighton	0.730	1.700	19.600	806.000	288.000	83.000	58.800	67.700	60.000
Portsmouth	0.750	1.700	19.400	917.000	272.000	81.400	49.300	65.400	61.000
Plymouth	0.740	1.600	16.400	1000.000	277.000	81.200	60.200	55.300	62.000
Central Lancs	0.750	0.900	17.500	984.000	271.000	76.900	50.900	64.400	63.000
Napier	-	0.800	17.900	915.000	257.000	73.300	60.800	77.900	64.000
West England	0.740	1.600	18.800	906.000	261.000	80.600	55.900	64.400	65.000
Winchester	0.750	1.500	17.900	796.000	267.000	84.500	57.400	49.700	66.000
Staffordshire	0.760	1.100	17.600	1000.000	251.000	79.800	52.500	53.800	67.000
Glasgow Cal	0.740	1.200	19.800	850.000	311.000	74.800	65.700	60.500	68.000
QM Edinburgh	-	1.500	20.200	831.000	286.000	77.700	58.500	69.800	69.000
Lampeter	0.740	4.400	25.000	480.000	274.000	76.900	56.100	58.800	70.000
B'ham City	0.710	0.800	15.800	1138.000	268.000	77.300	56.600	63.800	71.000

Bath Spa	0.720	1.500	19.500	563.000	274.000	85.700	64.700	55.200	72.000
Northumbria	0.730	1.100	21.000	1010.000	292.000	80.700	49.900	66.400	73.000
Coventry	0.760	1.000	20.200	1004.000	261.000	73.700	60.400	62.000	74.000
U. Arts	0.620	4.400	20.800	870.000	304.000	87.100	58.800	51.500	75.000
S'field Hallam	0.720	1.100	19.900	938.000	269.000	85.000	58.100	60.600	76.000
Glamorgan	0.750	1.300	15.900	1123.000	237.000	69.300	51.800	58.400	77.000
De Montfort	0.720	1.400	17.400	831.000	250.000	80.600	49.000	64.800	78.000
Hertfordshire	0.720	1.400	15.600	1087.000	256.000	75.900	47.700	62.100	79.000
Canterbury CC	0.740	1.100	17.800	559.000	247.000	83.500	47.000	63.200	80.000
Worcester	0.760	0.800	19.700	814.000	245.000	80.800	45.600	62.700	81.000
Sunderland	0.730	1.700	16.100	737.000	249.000	77.000	47.700	59.300	82.000
Salford	0.710	2.100	18.400	923.000	269.000	73.200	54.100	63.500	83.000
Northampton	0.760	1.000	19.000	786.000	225.000	80.000	55.200	56.300	84.000
UWIC Cardiff	0.720	0.800	19.900	1112.000	246.000	85.100	48.600	55.300	85.000
Roehampton	0.710	1.900	21.000	1308.000	224.000	79.500	52.100	56.900	86.000
Chester	0.740	0.500	17.900	734.000	274.000	76.100	54.300	57.600	87.000
Teesside	0.770	0.700	19.500	750.000	253.000	74.400	50.200	57.300	88.000
Bedfordshire	0.730	0.600	21.200	1685.000	220.000	75.400	47.000	59.600	89.000
Huddersfield	0.750	0.900	19.800	616.000	262.000	78.900	53.300	53.200	90.000
York St John	0.720	0.400	22.300	841.000	281.000	85.400	50.700	55.200	91.000
Manchester Met	0.710	1.200	20.200	937.000	272.000	75.200	51.300	59.400	92.000
Kingston	0.710	1.200	19.800	966.000	236.000	78.600	50.000	60.200	93.000
Liverpool JM	0.710	1.200	19.500	902.000	250.000	77.100	45.100	61.300	94.000
Derby	0.730	0.900	19.800	1054.000	267.000	77.000	47.800	49.500	95.000

UWCN	0.710	0.500	20.100	890.000	219.000	74.900	50.700	60.400	96.000
Soton Solent	0.710	0.500	20.500	980.000	229.000	78.200	43.100	55.700	97.000
Edge Hill	0.760	0.400	22.100	597.000	250.000	78.300	44.400	57.000	98.000
Cumbria	0.680	0.600	20.400	685.000	267.000	76.600	47.500	72.800	99.000
Abertay	-	0.700	20.300	1671.000	264.000	64.000	47.600	51.400	100.000
Leeds Met	0.680	0.800	23.600	984.000	264.000	82.700	53.600	59.400	101.000
Westminster	0.680	1.200	16.400	763.000	254.000	80.500	49.600	51.000	102.000
West Scotland	-	0.700	19.400	1099.000	255.000	68.700	46.100	62.300	103.000
Lincoln	0.700	0.700	23.700	842.000	261.000	83.900	52.900	50.300	104.000
Middlesex	0.670	1.300	22.100	1412.000	219.000	71.900	47.300	61.200	105.000
Anglia Ruskin	0.700	0.500	17.100	647.000	256.000	64.800	52.300	60.600	106.000
Wolverhampton	0.720	0.600	18.600	945.000	197.000	73.900	41.100	56.800	107.000
Bucks New	0.680	0.800	18.700	999.000	203.000	82.000	44.700	45.200	108.000
East London	0.700	1.400	20.600	1218.000	187.000	68.500	40.600	59.400	109.000
Greenwich	0.680	1.200	23.400	824.000	212.000	79.700	42.400	63.000	110.000
Bolton	0.740	0.800	21.000	772.000	203.000	63.300	52.600	51.800	111.000
Thames Valley	0.670	0.400	20.600	747.000	212.000	69.500	48.400	62.700	112.000
London S Bank	0.720	1.300	25.200	822.000	193.000	72.000	48.400	56.300	113.000

Bibliography

R. B. Agrawal and K. Deb. Simulated Binary Crossover for Continuous Search Space. Technical report, Indian Institute of Technology, 1994.

J. E. Atkins, E. G. Boman, and B. Hendrikson. A Spectral Algorithm for Seriation and the Consecutive Ones Problem. *SIAM Journal on Computing*, 28(1):297–310, 1998.

J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-based Many-objective Optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

P. Baty. Times Higher Education World University Rankings, September 2010.

M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, 2001.

R. Benayoun, B. Roy, and B. Sussman. Electre: Une méthode pour guider le choix en présence de points de vue multiple. *Direction Scientifique*, 1966.

M. Benito and R. Romera. Improving Quality Assessment of Composite Indicators in University Rankings: a Case Study of French and German Universities of Excellence. *Scientometrics*, 89(1):153–176, October 2011.

P. J. Bentley and J. P. Wakefield. Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. In *Soft Computing in Engineering Design and Manufacturing*, pages 231–240, 1998.

C. Berge. *The Theory of Graphs*. Methuen, London, 1962.

H-G. Beyer and H-P. Schwefel. Evolution Strategies – A Comprehensive Introduction. *Natural Computing: an international journal*, 1:3–52, May 2002.

C. Bishop, M. Svensén, and C. Williams. GTM: The Generative Topographic Mapping. *Neural Computation*, pages 215 – 235, 1998.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. ISBN 0387310738.

U. Biswas, U. Maulik, A. Mukhopadhyay, and M. Naskar. Multiobjective Evolutionary Approach to Cost-effective Traffic Grooming in Unidirectional SONET/WDM rings. *Photonic Network Communications*, 18:105–115, 2009.

L. Bradstreet, L. Barone, and L. While. Maximising Hypervolume for Selection in Multi-objective Evolutionary Algorithms. In *IEEE Congress on Evolutionary Computation*, pages 1744–1751, 2006.

J.P. Brans, Ph. Vincke, and B. Mareschal. How to select and how to rank projects: The promethee method. *European Journal of Operational Research*, 24:228–238, 1986.

D. Brockhoff and E. Zitzler. Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In K. H. Waldmann and U. M. Stocker, editors, *Operations Research Proceedings 2006*, pages 423–429. Springer, 2007a.

D. Brockhoff and E. Zitzler. Offline and Online Objective Reduction in Evolutionary Multiobjective Optimization Based on Objective Conflicts. TIK Report 269, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, April 2007b.

D. Brockhoff, T. Friedrich, and F. Neumann. Analyzing Hypervolume Indicator Based Algorithms. In G. Rudolph et al., editors, *Conference on Parallel Problem Solving From Nature (PPSN X)*, volume 5199 of *LNCS*, pages 651–660. Springer, 2008.

F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

W. S. Cleveland. *Elements of Graphing Data*. AT & T Bell Laboratories, Murray Hill, NJ, 1994.

C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 0387332545.

D. Corne and J. Knowles. Techniques for Highly Multiobjective Optimisation: Some Nondominated Points are Better than Others. In *Genetic and Evolutionary Computation Conference*, pages 773–780, London, UK, 2007.

Piotr Czyzak and Adrezej Jaszkiewicz. Pareto simulated annealinga metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, 1998.

D. Lowe and M. E. Tipping. NeuroScale: Novel Topographic Feature Extraction using RBF Networks. In *NIPS*, pages 543–549, 1996.

I. Das and J. E. Dennis. A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems. *Structural and Multidisciplinary Optimization*, 14:63–69, 1997.

L. Davis. Applying Adaptive Algorithms to Epistatic Domains. In *IJCAI*, pages 162–164, 1985.

M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, second edition, 2000.

K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.

K. Deb and M. Goyal. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics*, 26:30–45, 1996.

K. Deb and D. Saxena. On Finding Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems. Technical Report 2005011, Indian Institute of Technology, 2005.

K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2001.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Proceedings of IEEE Congress on Evolutionary Computation*, volume 1, pages 825–830, May 2002.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977.

F. di Pierro, S. T. Khu, and D. Savic. An Investigation on Preference Ordering Ranking Scheme in Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 11(1):17–45, 2007.

P. Diaconis and R. L. Graham. Spearman's Footrule as a Measure of Disarray. *Journal of the Royal Statistical Society, Series B*, 39(2):262–268, 1977.

J. Ding and J. Qiu. An Approach to Improve the Indicator Weights of Scientific and Technological Competitiveness Evaluation of Chinese Universities. *Scientometrics*, 86 (2):285–297, February 2011.

M. D'Ocagane. *Coordonnées paralléles et axiales: Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles*. Gauthier-Villars, reprinted by Kessinger Publishing, 1885.

N. Drechsler, R. Drechsler, and B. Becker. Multi-objective optimisation based on relation favour. *Lecture Notes in Computer Science*, 1993:154–168, 2001.

A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computation*. Springer, 2003.

M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster Analysis and Display of Genome-wide Expression Patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863–14868, December 1998.

C. Emmanouilidis. Evolutionary Multi-Objective Feature Selection and ROC Analysis With Application to Industrial Machinery Fault Diagnosis. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design, Optimisation and Control*, 2002.

M. Erghott. *Multicriteria Optimization*. Springer, 2 edition, 2005.

P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32(1):211–227, 2010.

R. M. Everson and J. E. Fieldsend. Multi-class ROC Analysis from a Multi-objective Optimisation Perspective. *Pattern Recognition Letters*, 27:531–556, 2006.

R. M. Everson, J. E. Fieldsend, and S. Singh. Full Elite Sets for Multi-Objective Optimisation. In I.C. Parmee, editor, *Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, volume 5, pages 343–354, University of Exeter, Devon, UK, April 2002. Springer-Verlag.

R. M. Everson, D. J. Walker, and J. E. Fieldsend. Edges of Mutually Non-dominating Sets. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, GECCO '13, pages 1–8, 2013.

M. Farina and P. Amato. On the Optimal Solution Definition for Many-criteria Optimization Problems. In *2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings*, pages 233–238, 2002.

M. Farina and P. Amato. Fuzzy Optimality and Evolutionary Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization*, volume 2632, pages 72–73. Springer Berlin / Heidelberg, 2003.

N. Fenton and W. Wang. Risk and confidence analysis for fuzzy multicriteria decision making. *Knowledge-based Systems*, 19:430–437, 2006.

M. Fiedler. Algebraic Connectivity of Graphs. *Czechoslovak Mathematical Journal*, 23 (98):298–305, 1973.

J. E. Fieldsend. *Novel Algorithms for Multi-objective Search and their Application in Multi-objective Evolutionary Neural Network Training*. PhD thesis, University of Exeter, Exeter, UK, 2003.

J. E. Fieldsend and R. M. Everson. Visualisation of Multi-class ROC Surfaces. In *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning*, pages 49–56, 2005.

J. E. Fieldsend, R. M. Everson, and S. Singh. Using Unconstrained Elite Archives for Multiobjective Optimization. *Evolutionary Computation, IEEE Transactions on*, 7(3): 305 – 323, june 2003.

M. Fleischer. The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics. In *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 519–533. Springer, 2003.

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. Morgan Kauffman, 1993.

V. Grunert da Fonseca, C. M. Fonseca, and A. O. Hall. Inferential performance assessment of stochastic optimisers and the attainment function. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, EMO '01, pages 213–225, 2001.

E. Forsyth and L. Katz. A Matrix Approach to the Analysis of Sociometric Data: Preliminary Report. *Sociometry*, 9(4):340–347, 1946.

F. Franceschini, M. Galetto, and D. Maisano. *Management by Measurement: Designing Key Indicators and Performance Measurement Systems*. Springer, 2007. ISBN 978-3-540-73211-2.

M. Garza-Fabre, G. Toscano Pulido, and C. A. Coello Coello. Ranking methods for many-objective optimization. In *Proceedings of the 8th Mexican International Conference on Artificial Intelligence*, MICAI '09, pages 633–645, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-05257-6.

M. Garza-Fabre, G. Toscano-Pulido, and C. A. Coello Coello. Two Novel Approaches for Many-objective Optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 4480–4487, July 2010.

F. Glover and C. McMillan. The General Employee Scheduling Problem: an Integration of MS and AI. *Comput. Oper. Res.*, 13:563–573, May 1986.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989. ISBN 0201157675.

D. E. Goldberg and R. Lingle. Alleles, Loci and the Travelling Salesman Problem. In J. Grefenstette, editor, *Proceedings of an International Conference on Genetic Algorithms and their Applications*, pages 154–159, 1985.

H. Goldstein and D. J. Spiegelhalter. League Tables and Their Limitations: Statistical Issues in Comparisons of Institutional Performance. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 159(3):pp. 385–443, 1996.

G. H. Golub and C. F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press, 3rd edition, 1996. ISBN 0801854148.

R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice Hall, 3 edition, 2007. ISBN 013168728X.

J. C. Gower. Euclidean Distance Geometry. *Math. Sci.*, 1:1–14, 1985.

GraphViz, April 2012. URL `http://www.graphviz.org/`.

G. Grinstein, M. Trutschl, and U. Cvek. High Dimensional Visualisations. In *Proceedings Visual Data Mining Workshop, KDD 2001*. ACM, 2001.

D. Henriet. The Copeland Choice Function an Axiomatic Characterization. *Social Choice and Welfare*, 2:49–63, 1985. ISSN 0176-1714.

J. Hettenhausen, A. Lewis, and S. Mostaghim. Interactive Multi-objective Particle Swarm Optimization with Heatmap-visualization-based User Interface. *Engineering Optimization*, 42:119–139, 2010.

D. S. Hochbaum. Ranking Sports Teams and the Inverse Equal Paths Problem. In *Proceedings of the Second international conference on Internet and Network Economics*, WINE'06, pages 307–318, Berlin, Heidelberg, 2006. Springer-Verlag.

J. H. Holland. *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Untelligence*. University of Michigan Press, 1975. ISBN 9780472084609.

J. Horn, N. Nafpliotis, and D. E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 82–87, 1994.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. CUP, 1990. ISBN 0521386322.

S. Huband, L. Barone, L. While, and P. Hingston. A Scalable Multi-objective Test Problem Toolkit. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 280–295, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *Evolutionary Computation, IEEE Transactions on*, 10(5):477–506, 2006.

E. J. Hughes. Multiple Single Objective Pareto Sampling. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 4, pages 2678–2684 Vol.4, dec. 2003.

E. J. Hughes. Evolutionary Many-objective Optimisation: Many Once or One Many? In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 222–227, 2005.

E. J. Hughes. Radar Waveform Optimisation as a Many-objective Application Benchmark. In *Proceedings of the 4th international conference on Evolutionary multi-criterion optimization*, EMO'07, pages 700–714, Berlin, Heidelberg, 2007a. Springer-Verlag. ISBN 978-3-540-70927-5.

E. J. Hughes. Best Known Non-dominated Results of Radar Waveform Optimisation. `code.evanhughes.org`, 2007b.

A. Inselberg. N-Dimensional Coordinates. In *Picture Data Description & Management, IEEE PAMI*, page 136, 1980.

H. Ishibuchi and T. Murata. Multi-objective genetic local search algorithm. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 119–124, 1996.

H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary Many-objective Optimization: A short Review. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2419–2426, June 2008.

J. Jarmulak and S. Craw. Genetic Algorithms for Feature Selection and Weighting. In *In Proceedings of the IJCAI99 workshop on Automating the Construction of Case Based Reasoners*, pages 28–33, 1999.

M. Jelokhani-Niaraki and J. Malczewski. A user-centred multicriteria spatial decision analysis method for participatory decision making: An ontology-based approach. In *Proceedings of the Global Geospatial Conference 2012*, 2012.

I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

L. V. Kantorovich. Mathematical Methods of Organizing and Planning Production. *Management Science*, 6(4):pp. 366–422, 1969.

A. Kaveh and H.A. Rahimi Bondarabady. Finite Element Mesh Decomposition Using Complementary Laplacian Matrix. *Commun. Numer. Meth. Engng*, 16(379-389), 2000.

J. P. Keener. The Perron-Frobenius Theorem and the Ranking of Football Teams. *SIAM Rev.*, 35(1):80–93, 1993. ISSN 0036-1445.

M. G. Kendall. A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81–93, 1938.

M. G. Kendall. Further contributions to the theory of paired comparisons. *Biometrics*, 11:43–62, 1955.

E. Kiesling, J. Gettinger, C. Stummer, and R. Vetschera. An Experimental Comparison of Two Interactive Visualization Methods for Multicriteria Portfolio Selection. In Ahti Salo, Jeffrey Keisler, and Alec Morton, editors, *Portfolio Decision Analysis*, volume 162, pages 187–209. Springer New York, 2011. ISBN 978-1-4419-9943-6.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

J. Knowles. ParEGO: a Hybrid Algorithm with On-line Landscape Approximation for Expensive Multiobjective Optimization Problems. *Evolutionary Computation, IEEE Transactions on*, 10(1):50–66, 2006.

J. Knowles and D. Corne. The Pareto Archived Evolution Strategy: a New Baseline Algorithm for Pareto Multiobjective Optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 1, 1999.

J. Knowles and D. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8:149–172, June 2000.

D. E. Knuth. *The Art of Computer Programming. Volume 4A: Combinatorial Algorithms Part 1.* Addison-Wesley Publishing, 2011.

T. Kohonen. *Self-organising Maps.* Springer, 1995.

M. Koppen and K. Yoshida. Visualization of Pareto-Sets in Evolutionary Multi-Objective Optimization. In *Proceedings of the 7th International Conference on Hybrid Intelligent Systems*, pages 156–161, Washington, DC, USA, 2007. IEEE Computer Society.

P. Korhonen and J. Wallenius. Introduction to Multiobjective Optimization: Interactive Approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowiński, editors, *Visualization in the Multiple Objective Decision-Making Framework*, pages 195–212. Springer, 2008.

Pekka Korhonen and Jyrki Wallenius. A pareto race. *Naval Research Logistics (NRL)*, 35 (6):615–623, 1988.

Pekka J. Korhonen and G.Y. Yu. Quadratic pareto race. Working papers, International Institute for Applied Systems Analysis, 1997.

J. B. Kruskal. Multidimensional Scaling by Optimizing Goodness-of-Fit to a Nonmetric Hypothesis. *Psychometrica*, 29:1–27, 1964.

M. Laumanns, E. Zitzler, and L. Thiele. A Unified Model for Multi-objective Evolutionary Algorithms with Elitism. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 46–53 vol.1, 2000.

P. J. Lewi and J. Van Hoof. Multicriteria decision making using pareto optimality and promethee preference ranking. *Chemometrics and Intelligent Laboratory Systems*, 16: 139–144, 1992.

M. Li, J. Zheng, K. Li, Q. Yuan, and R. Shen. Enhancing Diversity for Average Ranking Method in Evolutionary Many-objective Optimization. In *Proceedings of the 11th international conference on Parallel problem solving from nature: Part I*, PPSN'10, pages 647–656, Berlin, Heidelberg, 2010a. Springer-Verlag. ISBN 3-642-15843-9, 978-3-642-15843-8.

M. Li, J. Zheng, R. Shen, K. Li, and Q. Yuan. A Grid-based Fitness Strategy for Evolutionary Many-objective Optimization. In *Proceedings of the 12th annual conference*

on Genetic and evolutionary computation, GECCO '10, pages 463–470. ACM, 2010b. ISBN 978-1-4503-0072-8.

Y. Li, C. Campbell, and M. Tipping. Bayesian Automatic Relevance Determination Algorithms for Classifying Gene Expression Data. *Bioinformatics*, 18:1332–1339, 2002.

I. Liiv. Seriation and Matrix Reordering Methods: An Historical Overview. *Stat. Anal. Data Min.*, 3:70–91, April 2010. ISSN 1932-1864.

Amy Poh Ai Ling, Mohamad Nasir Saludin, and Masao Mukaidono. Deriving consensus rankings via multicriteria decision making methodology. *Business Strategy Series*, 13: 3–12, 2012.

A. López Jaimes, C. A. Coello Coello, and D. Chakraborty. Objective Reduction Using a Feature Selection Technique. In *Proceedings of the Genetic and evolutionary computation*, GECCO '08, 2008.

A. V. Lotov. Methodology and application of pareto frontier visualization in dss for the design of water quality improvement strategies in multi-regional river basins. In *Proceedings of the MULINO Conference on Policies and tools for the sustainable water management in the EU*, 2002.

A. V. Lotov and K. Miettinen. Introduction to Multiobjective Optimization: Interactive Approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowiński, editors, *Visualizing the Pareto Frontier*, pages 213–243. Springer, 2008.

A.V. Lotov, V.A. Bushenkov, and G.K. Kamenev. *Interactive Decision Maps*, volume 89 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

D. Lowe and M. E. Tipping. Feed-Forward Neural Networks and Topographic Mappings for Exploratory Data Analysis. *Neural Computing and Applications*, 4(2):83–95, 1996.

T. Magoc, M. Ceberio, and F. Modave. Using preference constraints to solve multi-criteria decision making problems. *Reliable Computing*, 15(3):218–229, 2011.

K. Maneeratana, K. Boonlong, and N. Chaiyaratana. Compressed-objective Genetic Algorithm. In *Proceedings of the 9th international conference on Parallel Problem Solving from Nature*, PPSN'06, pages 473–482, Berlin, Heidelberg, 2006. Springer-Verlag.

K. McClymont, D. Walker, E. Keedwell, R. Everson, Fieldsend J., D. Savic, and M. Randall-Smith. Novel Methods for Ranking District Metered Areas for Water Distribution Network Maintenance Scheduling. In *CCWI 2011*, Exeter, 2011.

W. L. Meisel. *Tradeoff Decision in Multiple Criteria Decision Making*. University of South Carolina Press, 1973.

I. Mierswa and M. Wurst. Information preserving multi-objective feature selection for unsupervised learning. In *Proceedings of the 8th annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 1545–1552. ACM, 2006.

K. Miettinen. *Nonlinear multiobjective optimization*. International Series in Operations Research and Management Science Series. Kluwer Academic Publishers, 1999. ISBN 9780792382782.

K. Miettinen. Introduction to Multiobjective Optimization: Noninteractive Approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 1–26. Springer, 2008.

K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to Multiobjective Optimization: Interactive Approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 27–59. Springer, 2008.

B. Mohar. The Laplacian Spectrum of Graphs. In *Graph Theory, Combinatorics, and Applications*, pages 871–898. Wiley, 1991.

J.W. Moon and N.J. Pullman. On Generalized Tournament Matrices. *SIAM Review*, 12 (3):384–399, 1970.

M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Unsupervised Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Word Recognition. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, ICDAR '03, pages 666–670. IEEE Computer Society, 2003.

I. T. Nabney. *NETLAB: Algorithms for Pattern Recognition*. Oxford University Press, 2004.

A. Nazemi, A. H. Chan, and X. Yao. Selecting Representative Parameters of Rainfall Runoff Models Using Multi-objective Calibration Results and a Fuzzy Clustering Algorithm. In *BHS 10th National Hydrology Symposium*, 2008.

J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 1965.

S. Obayashi. Pareto Solutions of Multipoint Design of Supersonic Wings using Evolutionary Algorithms. In I.C. Parmee, editor, *Adaptive Computing in Design and Manufacture V*, pages 3–15, London, 2002. Springer-Verlag.

I. Oh, J. Lee, and B. Moon. Hybrid Genetic Algorithms for Feature Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1424–1437, 2004.

J. O'Leary. *The Times Good University Guide 2009*. Times Books, 2009.

L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Digit Recognition. *International Conference on Pattern Recognition*, 1:10568–10571, 2002.

L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, Stanford University, 1998.

A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap Visualization of Population Based Multi Objective Algorithms. In *Proceedings of Evolutionary Multicriterion Optimization*, pages 361–375, 2006.

I. Rechenberg. *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, 1973.

C. Rinner. A geographic visualization approach to multi-criteria evaluation of urban quality of life. In *Proceedings of GIScience*, 2003.

W. S. Robinson. A Method for Chronologically Ordering Archaeological Deposits. *American Antiquity*, 16(4):293–301, April 1951.

S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

J. Salmi and A. Saroyan. League Tables as Policy Instruments: The Political Economy of Accountability in Tertiary Education. In *Higher Education in the World 2007*. Palgrave Macmillan, 2006.

J. W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, 18(5):401–409, 1969.

I. J. Schoenberg. Remarks to Maurice Frechet article Sur la définition axiomatique d'une classe d'espaces distanciés vectoriellement applicable sur l'espace de Hilbert. *Annals of Mathematics*, 38:724–732, 1935.

H. P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels Evolutionsstrategie*. Birkhäuser Verlag, Basel, 1994.

H. K. Singh, A. Isaacs, and T. Ray. A Pareto Corner Search Evolutionary Algorithm and Dimensionality Reduction in Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 15(4):539–556, 2011. ISSN 1089-778X.

G. Slutzki and O. Volij. Ranking Participants in Generalized Tournaments. *International Journal of Game Theory*, 33(2):255–270, 06 2005.

K. I. Smith, R. M. Everson, J. E. Fieldsend, C. Murphy, and R. Misra. Dominance-Based Multiobjective Simulated Annealing. *IEEE Transactions on Evolutionary Computation*, 12(3):323–342, 2008. ISSN 1089-778X.

C. Spearman. The Proof and Measurement of Association Between Two Things. *American Journal of Psychology*, 15:72–101, 1904.

C. Spearman. "Footrule" for Measuring Correlation. *British Journal of Psychology*, 2: 89–108, 1906.

N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

R. Storn and K. Price. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.

P. Subbaraj, R. Rengaraj, and S. Salivahanan. Enhancement of Self-adaptive Real-coded Genetic Algorithm Using Taguchi Method for Economic Dispatch Problem. *Applied Soft Computing*, 11(1):83 – 92, 2011. ISSN 1568-4946.

S. Tarkkanen, K. Miettinen, and J. Hakanen. Interactive multiobjective optimization – a new application area for visual analytics. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, 2009.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

Praveen Thokala. Multiple criteria decision analysis for health technology assessment. Technical report, School of Health and Related Research, University of Sheffield, 2011.

E. Triantaphyllou, B. Shu, S. Nieto Sanchez, and T. Ray. Multi-criteria decision making: An operations research approach. *Encyclopedia of Electrical and Electronics Enginering*, 15:175–186, 1998.

T. Tušar and B. Filipič. Visualizing 4D approximation sets of multiobjective optimizers with prosections. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 737–744, New York, NY, USA, 2011. ACM.

A. Usher and M. Savino. A World of Difference: A Global Survey of University League Tables, 2006.

J. J. Valdés and A. J. Barton. Visualizing High Dimensional Objective Spaces for Multi-objective Optimization: A Virtual Reality Approach. In *IEEE Congress on Evolutionary Computation*, pages 4199–4206, 2007.

R. van den Brink and R. P. Gilles. The Outflow Ranking Method for Weighted Directed Graphs. *European Journal of Operational Research*, 193:484–491, 2009.

D. A. Van Veldhuizen and G. B. Lamont. Evolutionary Computation and Convergence to a Pareto Front. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1998 Conference*, University of Wisconsin, Madison, Wisconsin, USA, jul 1998. Stanford University Bookstore.

R. Vetschera, J. Gettinger, E. Kiesling, and Christian Stummer. Visualization methods for multi-criteria portfolio selection: An empirical study. In *Proceedings of the 15th IFIP WG 8.3 International Conference on Decision Support Systems (DSS 2010)*, pages 1–9, 2010.

U. von Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4): 395–416, December 2007. ISSN 0960-3174.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualisation and Ordering of Many-objective Populations. In *2010 IEEE World Congress on Computational Intelligence (WCCI2010)*, pages 3664–3671, CCIB, Barcelona, Spain, July 2010a. IEEE Service Center.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Ordering Multi-objective Populations with the Power Index. In *2010 Postgraduate Conference for Computing: Applications and Theory (PCCAT2010)*, pages 3664–3671, Exeter, Devon, UK, June 2010b. University of Exeter.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Rank-based Dimension Reduction for Many-criteria Populations. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 1–2, New York, NY, USA, 2011. ACM.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualising Mutually Non-dominating Solution Sets in Many-objective Optimisation. *Transactions on Evolutionary Computation (in press)*, 2012a.

D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualising many-objective populations. In *Proceedings of the 3rd Workshop on Visualisation in Genetic and Evolutionary Computation (VizGEC 2012)*, 2012b.

A. R. Webb. *Statistical Pattern Recognition, 2nd Edition.* John Wiley & Sons, October 2002.

T. Wei. *The Algebraic Foundation of Ranking Theory.* PhD thesis, Cambridge University, Cambridge, UK, 1952.

K. Q. Weinberger and L. K. Saul. Unsupervised Learning of Image Manifolds by Semidefinite Programming. *Int. J. Comput. Vision*, 70(1):77–90, October 2006.

L. While, P. Hingston, L. Barone, and S. Huband. A Faster Algorithm for Calculating Hypervolume. *Evolutionary Computation, IEEE Transactions on*, 10(1):29–38, 2006.

L. While, L. Bradstreet, and L. Barone. A Fast Way of Calculating Exact Hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, 2012.

R. L. While. A New Analysis of the LebMeasure Algorithm for Calculating Hypervolume. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005, Proceedings*, volume 3410 of *Lecture Notes in Computer Science*, pages 326–340. Springer, 2005.

L. D. Whitley, T. Starkweather, and D. Fuquay. Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 133–140, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

L. Wilkinson and M. Friendly. The History of the Cluster Heat Map. *The American Statistician*, 63(2):179–0184, 2009.

X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999.

K Paul Yoon and Ching-Lai Hwang. *Multiple attribute decision making: an introduction*. Number 104. SAGE Publications, Incorporated, 1995.

T. Yoshikawa, D. Yamashiro, and T. Furuhashi. A Proposal of Visualization of Multiobjective Pareto Solutions - Development of Mining Technique for Solutions. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)*, 2007.

E. K. Zavadskas, A. Zakarevicius, and J. Antucheviciene. Evaluation of ranking accuracy in multi-criteria decisions. *Informatica*, 17(4):601–618, 2006.

Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, dec. 2007.

K. Zhao, B. Liu, T.M. Tirpak, and A. Schaller. Detecting Patterns of Change Using Enhanced Parallel Coordinates Visualization. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 747–750, nov. 2003.

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.

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

E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2001.