

# Visualising Many-objective Populations

David J. Walker<sup>\*</sup>  
Computer Science  
University of Exeter, UK  
D.J.Walker@ex.ac.uk

Jonathan E. Fieldsend  
Computer Science  
University of Exeter, UK  
J.E.Fieldsend@ex.ac.uk

Richard M. Everson  
Computer Science  
University of Exeter, UK  
R.M.Everson@ex.ac.uk

## ABSTRACT

Optimisation problems often comprise a large set of objectives, and visualising the set of solutions to a problem can help with understanding them, assisting a decision maker. If the set of objectives is larger than three, visualising solutions to the problem is a difficult task. Techniques for visualising high-dimensional data are often difficult to interpret. Conversely, discarding objectives so that the solutions can be visualised in two or three spatial dimensions results in a loss of potentially important information. We demonstrate four methods for visualising many-objective populations, two of which use the complete set of objectives to present solutions in a clear and intuitive fashion and two that compress the objectives of a population into two dimensions whilst minimising the information that is lost. All of the techniques are illustrated on populations of solutions to optimisation test problems.

## Categories and Subject Descriptors

H.5.0 [Information Interfaces and Presentation]: General.

## General Terms

Algorithms.

## Keywords

Visualisation, Sorting, Multi-objective optimisation.

## 1. INTRODUCTION

Scientific and engineering optimisation problems are often defined in terms of *objectives*, quantitative measures of solution performance. Commonly, problems are defined by a set of objectives that are in conflict with one and other, so that a solution that is good with respect to one objective is often a poor candidate with respect to another. Unlike *single-objective* problems that have a single optimal solution, a *multi-objective* problem comprising objectives that are in conflict with each other is generally characterised by a set of

<sup>\*</sup>David Walker was supported by an EPSRC CASE award with Motorola during the preparation of this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '12 Companion, July 7–11, 2012, Philadelphia, PA, USA.  
Copyright 2012 ACM 978-1-4503-1178-6/12/07 ...\$10.00.

optimal solutions that trade-off the problem objectives. The image of these solutions lie on the Pareto front. Formally, without loss of generality, a multi-objective optimisation problem is solved by finding a solution  $\mathbf{x}$  which minimises a set of objectives

$$\mathbf{y} = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})) \quad (1)$$

such that  $y_m = f_m(\mathbf{x})$ .

A widely used approach for solving multi-objective problems is to apply an *Evolutionary Algorithm* (EA) to generate a set of solutions that approximate the *Pareto front*.

Evolutionary Algorithms are iterative, population-based, algorithms, and visualising the change in the population of an EA can reveal useful information about how successfully the algorithm is exploring the search space. Furthermore, when the algorithm has generated a set of approximately Pareto optimal solutions, a decision maker is required to select the solution that will be implemented to solve the problem. A visualisation of the solution set that clearly illustrates the trade-off between objectives is a useful tool to assist the decision maker in their choice.

The relative quality of solutions to multi-objective problems can be determined with the dominance relation, and it is useful for a visualisation to convey information about the dominance relationships between solutions. A solution  $\mathbf{x}_i$  dominates  $\mathbf{x}_j$ , denoted  $\mathbf{x}_i \prec \mathbf{x}_j$ , if it is no worse than  $\mathbf{x}_j$  on any objective and superior on at least one:

$$\mathbf{x}_i \prec \mathbf{x}_j \iff \forall m (y_{im} \leq y_{jm}) \wedge \exists m (y_{im} < y_{jm}). \quad (2)$$

If no solution dominates  $\mathbf{x}_i$  then it is *non-dominated*. An EA can operate with two types of population. The first is the *search population*, which is used by the EA to generate new candidate solutions to the problem and can contain dominated solutions. The second, which is optional, is an *elite archive* of solutions which represent the current best set of solutions found by the algorithm at a given point in the algorithm's execution. Such an archive comprises exclusively non-dominated solutions. Thus, we seek visualisation methods that can visualise populations of dominated and non-dominated solutions.

A distinction is often drawn between multi-objective problems, comprising two or three objectives, and *many-objective* problems, comprising four or more. Visualisation is, in part, the basis for this distinction. Visualising populations of solutions to multi-objective problems is relatively straightforward. The objective vectors can be represented as a point on a set of two or three dimensional axes and their location in objective space relative to other solutions in the population provides an indication of the trade-off between objectives. Populations of solutions to many-objective problems are particularly difficult to visualise since most visualisation methods require some degree of dimension reduction and incur an undesirable loss of information. Therefore, we seek techniques that can

visualise a population of objective vectors corresponding to putative solutions to many-objective problems in which any information lost in the visualisation process is recoverable.

In this paper we illustrate four techniques that can be used to visualise many-objective populations. Two of the techniques visualise a population based on the full set of objectives, while the other two reduce the dimensionality of the population so that it can be visualised with a conventional scatter plot. We introduce two many-objective visualisation methods in Sections 2 and 3, and in Sections 4 and 5 we illustrate dimension reduction techniques to facilitate simple visualisations.

## 1.1 Illustration

Throughout the paper we refer to an example set of populations, all produced by using an Evolutionary Algorithm to solve a well-known multi-objective test problem. We present two instances of DTLZ2 [5], comprising three and ten objectives respectively. The solutions were generated with a basic multi-objective  $(\mu + \lambda)$ -Evolution Strategy (ES) where both  $\mu$  and  $\lambda$  were set to 50. The algorithm was run for 100 generations for each problem instance and the search population at each generation was saved. An elite archive of solutions was maintained throughout. At each generation the parent population was copied to produce  $\lambda$  child solutions that were perturbed with a single point Gaussian mutation of standard deviation  $\sigma = 0.1$ . Selection was based on Pareto sorting [18] of the combined population of parent and child objective vectors. If a shell contains more solutions than are necessary to fill the new population a subset of those solutions are chosen at random.

We illustrate the methods presented in this paper on two types of population. One is the search population used by the algorithm to evolve new solutions. Each search population contains 50, potentially dominated, solutions. The other is the elite archive of solutions resulting from an optimisation run. In the case of the 3-objective DTLZ2 problem instance, the elite archive contains 305 solutions while the 10-objective problem archive contains 1217 solutions.

## 2. VISUALISING WITH RANKS

The purpose of visualising a population of solutions is to inform the decision maker, who must choose the solution that will be implemented to solve the problem at hand. As such, a useful visualisation will graphically convey relative quality between solutions so that the best solutions in the population can be easily identified. A common way of comparing solution quality is to rank them in terms of their objective values. This is an approach which is often incorporated into the selection operator of an evolutionary algorithm, so that those solutions which best optimise the problem are retained as progenitors in the next generation.

Ranking a single-objective population is trivial; assuming, without loss of generality, that the objective is to be minimised, those solutions with the smallest values are given the best ranks. The potential for conflict between objectives in multi- and many-objective populations means that such a strategy is insufficient. One of the most well known schemes for ranking a multi-objective population is the *Pareto sorting* method used in the original NSGA algorithm [18]. Pareto sorting constructs a partial ordering of solutions by peeling away layers of non-dominated solutions and placing them into *Pareto shells*. Those solutions that are Pareto optimal with respect to the whole population are assigned to shell 1 and temporarily discarded from the population. This leaves a new subset of non-dominated solutions, which become shell 2. These solutions are then also discarded, and the process continues until the entire population has been assigned to a Pareto shell. Having done this,

we can consider the solutions in shell  $n$  to be notionally superior to those in shell  $n + 1$ ; each member of shell  $n + 1$  is dominated by at least one member shell  $n$ . Those in shell 1 are Pareto optimal.

Having constructed a partial ordering of the solutions we can use this as the basis for a visualisation of the population [20]. We define a directed graph in which a node represents a solution and a directed edge from solutions  $\mathbf{y}_i$  to  $\mathbf{y}_j$  indicates that  $\mathbf{y}_i \prec \mathbf{y}_j$ . Solutions are arranged into columns according to the Pareto shell in which they reside so that those on the left-hand side of the figure are the Pareto optimal solutions, and those on the right are the most dominated. Figure 1 illustrates such a visualisation of a population; colour should be disregarded for the moment. The examples shown are for the three-objective DTLZ2 problem and are of the search population used by the ES to generate an approximation to the Pareto front at three points during its execution. The left-hand panel shows the initial random population; the central panel shows the population after five generations; and the right-hand panel shows the population after 10 generations. The objective vectors corresponding to the solutions in the three populations have been ranked with Pareto sorting and arranged into columns indicating their Pareto shell. For clarity, only those dominance relationships between solutions in adjacent shells are shown. The result is a visualisation that begins to illustrate the structure of the population. Some solutions are clearly responsible for defining the immediately inferior shell as they dominate large numbers of solutions in that shell; those individuals are likely to be of interest to a decision maker. However, many of the solutions do not dominate anything in the next shell, and the visualisation as it is (uncoloured) does not provide any means of discriminating between them. Also, it is clear to see that as the algorithm identifies good solutions the population sorts into fewer Pareto shells. This issue is exacerbated for problems comprising larger number of objectives; the 10-objective problem used in this paper generally sorts into a single Pareto shell. Thus, all of the solutions in the population have the same rank.

To ameliorate this, we introduce a second ranking method. As mentioned previously, Pareto sorting is frequently used as a ranking tool in evolutionary selection operators, so the lack of discrimination it provides in many-objective populations has motivated considerable research to identify ranking techniques which can operate in a many-objective space. One of the simplest is *average rank* [2], and it has been shown to provide much better convergence to the true Pareto front of many-objective problems [4]. In order to compute the average rank  $\bar{r}_i$  of a solution, the population is ranked according to each individual objective in turn, producing a set of ranks  $\mathbf{r}_m$  where  $r_{im}$  is the rank of the  $i$ th solution on the  $m$ th objective. The solution with the best objective value for a given objective is ranked 1, and the worst is ranked  $N$ . These ranks are then averaged, to provide an overall score for the solution:

$$\bar{r}_i = \frac{1}{M} \sum_{m=1}^M r_{im}. \quad (3)$$

The colouring in Figure 1 indicates a solution's average rank. The solutions are coloured so that a blue solution has a very good average rank, while a red solution has a poor rank. By comparing the colouring of solutions with their placement in the figure according to Pareto sorting we can see some agreement between the two methods; those previously highlighted dominant solutions are generally coloured in blue, meaning that they are seen as high-quality solutions under both schemes. In some cases, however, there is disagreement between a solution's average rank and the Pareto shell into which it has been sorted. Examples of such solutions are the

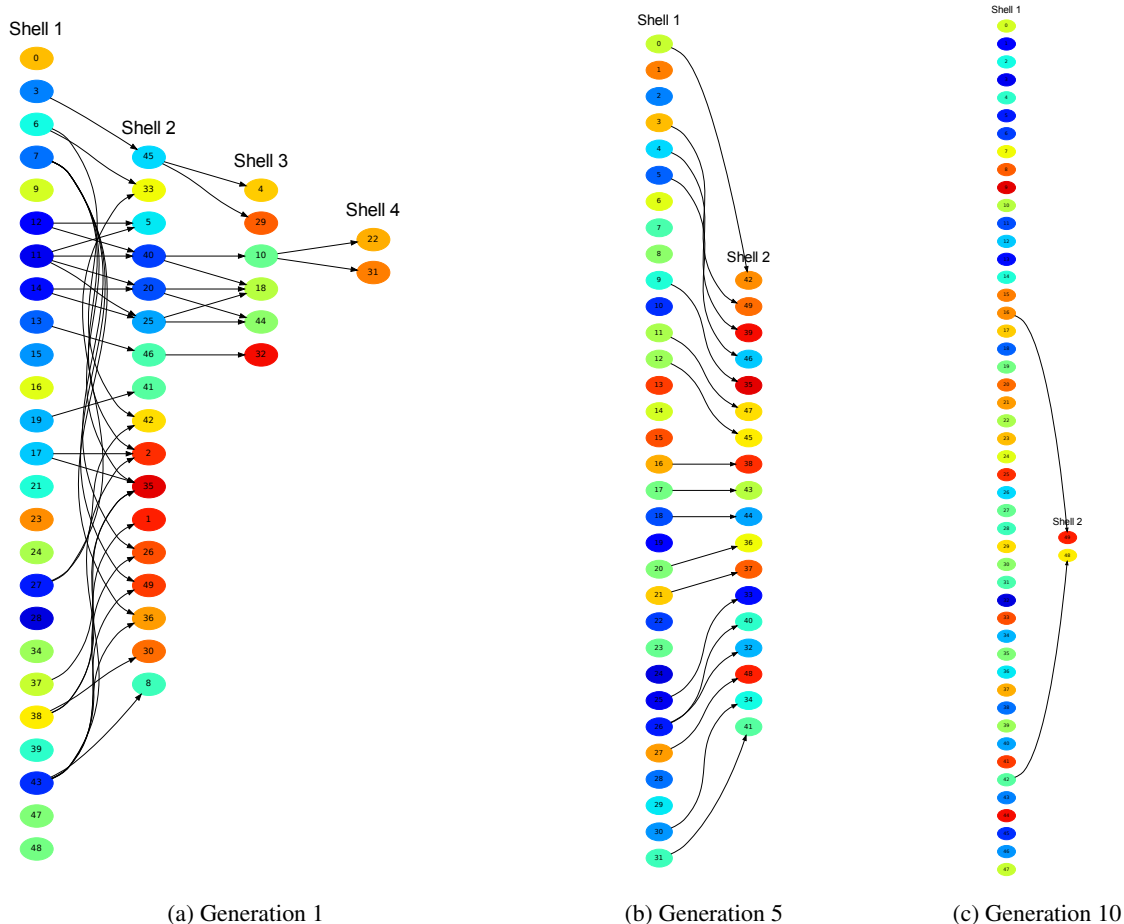


Figure 1: Pareto shell visualisations of the search population in the  $(\mu + \lambda)$ -ES evolving solutions to the 3-objective instance of DTLZ2. Solutions are sorted into their Pareto shells and coloured by average rank; a blue solution represents one with a good average rank while red indicates 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.

orange and yellow solutions in shell 1 (Figure 1(a)). These are solutions that, under Pareto sorting alone, would be deemed to be of high quality. With the additional information we can see that this is not the case. In fact, this indicates that a solution is performing very well on one objective, but poorly on the remainder. Such a solution is difficult to dominate, as it requires a better objective value on all objectives, including one which is already fairly optimal. It does not, however, guarantee a good average rank, since the remainder of the ranks  $r_{im}$  for that solution are poor; the overall average will be poor too.

This illustration shows the useful information that combining these two ranking methods can provide; indeed, it is possible to combine Pareto sorting with any many-objective ranking technique [20]. Average rank is used here because of its simplicity, as well as its ability to provide a more discriminative ordering of many-objective solutions. Unfortunately, however, this method is unsuitable for visualising a non-dominated population such as an archive since there are no dominated solutions and it forms a single Pareto shell. In the next section, we present a method which can visualise non-dominated populations, as well as populations containing dominated solutions.

### 3. SERIATION OF HEATMAPS

A common technique for visualising high-dimensional data is to produce a *heatmap* (e.g., [23, 6]). Heatmaps are well suited to the display of solutions to many-objective problems and have been used for this in several instances [15, 13, 21]. In such a visualisation, a row represents a solution and a column represents an objective. Objective values are assigned a colour by treating them as temperatures, such that a small value is assigned a cool colour and a large value a warm colour. Objectives must be normalised so that they are all on the same scale prior to visualisation. A drawback with visualising a population with a heatmap is that the arbitrary ordering of solutions and objectives can lead to a visualisation that is difficult to interpret. Ideally, it should be possible to infer patterns in the population from the visualisation, and given an arbitrary scattering of colour this is not possible.

Various approaches have been taken to enhance the clarity of a heatmap. Eisen *et al* [6] and Pryke *et al* [15] used hierarchical clustering techniques to arrange the heatmap by placing similar rows and columns close together so that similar colours are grouped. Nazemi *et al* [13] order the solutions in ascending order according to the first objective. We illustrate *spectral seriation* methods

for reordering a heatmap. Atkins *et al* [1] demonstrate a spectral method for seriation which can be used to place similar solutions close together based on a similarity measures  $\mathbf{S}$  in which  $S_{ij}$  represents similarity between objective vectors  $\mathbf{y}_i$  and  $\mathbf{y}_j$ . The seriation problem is solved by minimising

$$g(\boldsymbol{\pi}) = \sum_{i,j} S_{ij} (\pi_i - \pi_j)^2 \quad (4)$$

where  $\boldsymbol{\pi}$  is a permutation of the solutions. As the permutation is discrete the problem is NP hard. Atkins *et al* [1] suggest that the permutation can be relaxed to a continuous variable  $\mathbf{z}$  so that we instead minimise

$$h(\mathbf{z}) = \sum_{i,j} S_{ij} (z_i - z_j)^2 \quad (5)$$

subject to constraints  $\sum_n z_n = 0$  and  $\sum_n z_n^2 = 1$ . The solution to this problem can be found with linear algebra; the solution to the constrained problem can be written as  $\mathbf{z}^T \mathbf{Lz}$ , where  $\mathbf{L} = \mathbf{D} - \mathbf{S}$  is the graph Laplacian of  $\mathbf{S}$  and  $\mathbf{D}$  is the diagonal matrix  $D_{ii} = \sum_j S_{ij}$ . The permutation of solutions is obtained by ordering them according to their value in the *Fiedler vector*, the eigenvector corresponding to the smallest non-zero eigenvalue of  $\mathbf{L}$ , such that the  $p$ th smallest value in the Fiedler vector occupies position  $p$  in the permutation.

In order to obtain a measure of similarity, we consider the absolute distance between the ranks of pairwise individuals. Since it is necessary to normalise the objectives prior to constructing the heatmap, we convert the objective vectors to rank space by ranking each objective in turn, as with the calculation of average rank. This results in a set of *rank vectors* such that  $\mathbf{r}_i$  is the rank vector corresponding to an objective vector  $\mathbf{y}_i$ . Similarity between  $\mathbf{y}_i$  and  $\mathbf{y}_j$  is computed with:

$$S_{ij} = 1 - \frac{1}{M(N-1)^2} \sum_{m=1}^M (r_{im} - r_{jm})^2. \quad (6)$$

A heatmap of the reordered population can then be constructed. As well as permuting solutions, the seriation process outlined above can be used to permute objectives so that similar objectives are gathered together. To do this, we define a new similarity measure which produces a  $M \times M$  matrix as follows:

$$S_{mn} = 1 - \frac{1}{(N-1)^3} \sum_{i=1}^N (r_{im} - r_{in}). \quad (7)$$

Alternative similarity measures employ methods for differentiating between permutations such as Spearman's footrule or Kendall's  $\tau$  metric.

In order to provide a more complete illustration of a solution set, it is also useful to visualise solution parameters alongside objectives. The solutions themselves can be ordered by the same ordering generated by seriating objective vectors; objective vectors can be ordered by a seriation of parameter-space solutions, however a trade-off exists between finding a harmonious ordering in both spaces, and it is therefore necessary to assert a preference for one space over the other. The parameters can be seriated in much the same way as we have shown for objectives. The principle difference between seriating objectives and parameters is that objectives can be ranked and parameters cannot. In that case we propose using the cosine similarity between the  $p$ th and  $q$ th parameters.

$$S_{pq} = \frac{\sum_p x_{ip} x_{iq}}{\sqrt{\sum_p x_{ip}^2 \sum_p x_{iq}^2}}. \quad (8)$$

Figure 2 presents two sets of heatmaps, both of the 10-objective DTLZ2 archive. The top two panels show the unsorted archive, parameter space on the right and objective space on the left, and the bottom two panels show the same populations after seriation. To produce the bottom two heatmaps the objectives were first normalised by conversion to rank-coordinates. The objective vectors were then seriated, and both the parameter space solutions and objective vectors reordered by the resulting permutation. As can be seen, the objective with large values on most objectives have been collected at the top of the heatmap, heatmap and those with lower values, represented by blue, have been collected at the bottom. In addition, solutions with similar parameter values have been clustered together. Then, the parameters and objectives were seriated to place similar parameters and similar objectives close together. Some correlation between the objectives is apparent: most of the objectives are well correlated, however the trade-off between optimising objectives  $f_8$  and  $f_{10}$  against the rest of the objectives is clear to see. Such information was not visible in the unsorted version of the heatmap. Correlation between parameters is also clear to see, although the overall seriation of parameter space is less effective than that of objective space. If the solutions were seriated with respect to parameter space, the parameter space heatmap would be clearer but at the cost of reducing the quality of the objective space heatmap. Since the decision maker is interested in observing the trade-off between objectives we chose to prefer objective space.

## 4. VISUALISATION IN THE PLANE

Methods presented in the previous two sections have visualised many-objective populations with respect to the full set of objectives, however it can also be useful to visualise the population in a low-dimensional space. Existing techniques, such as *self organising maps* [10], *generalised tournament matrices* [3] and *Neuroscale* [12] have all been used to visualise solutions to many-objective problems [14, 7] however they are ignorant of the dominance structure between solutions and as a result it can be difficult to infer the trade-off between objectives using these methods. In this section we present a technique, visualisation in the plane, which takes advantage of the mutually non-dominating nature of an elite archive of solutions, and in the next section we compress the dimensionality of the solutions by ensuring that solutions that are similar in terms of dominance remain close together.

In order to visualise solutions in the plane we map  $M$ -objective solutions into a  $M$ -sided regular polygon. All objective values are assumed to be positive. The *simplex* is the portion of the (hyper-) plane that lies in the positive (hyper-) octant at distances  $\lambda_m$  from the origin, where  $\lambda_m > 0$  for all  $M$ , defined by  $\mathbf{n} \cdot \mathbf{y}$  where all  $y_m \geq 0$  and the elements of the unit normal vector  $\mathbf{n}$  are  $n_m = d/\lambda_m$ . We project an objective vector onto the simplex as follows:

$$\tilde{\mathbf{y}} = \mathbf{y}_i / (\mathbf{y}_i \cdot \mathbf{n}) \quad (9)$$

where

$$d^{-2} = \sum_{m=1}^M \lambda_m^2 \quad (10)$$

It is important to note at this point that any non-dominated set of objective vectors is still non-dominated once it has been projected onto the simplex.

The position of the objective vector on the simplex is determined with *barycentric coordinates*, known for triangles but which generalise to convex polygons and multi-dimensional simplices such as the one described above [9, 22]. Given position vectors of the  $m$ th

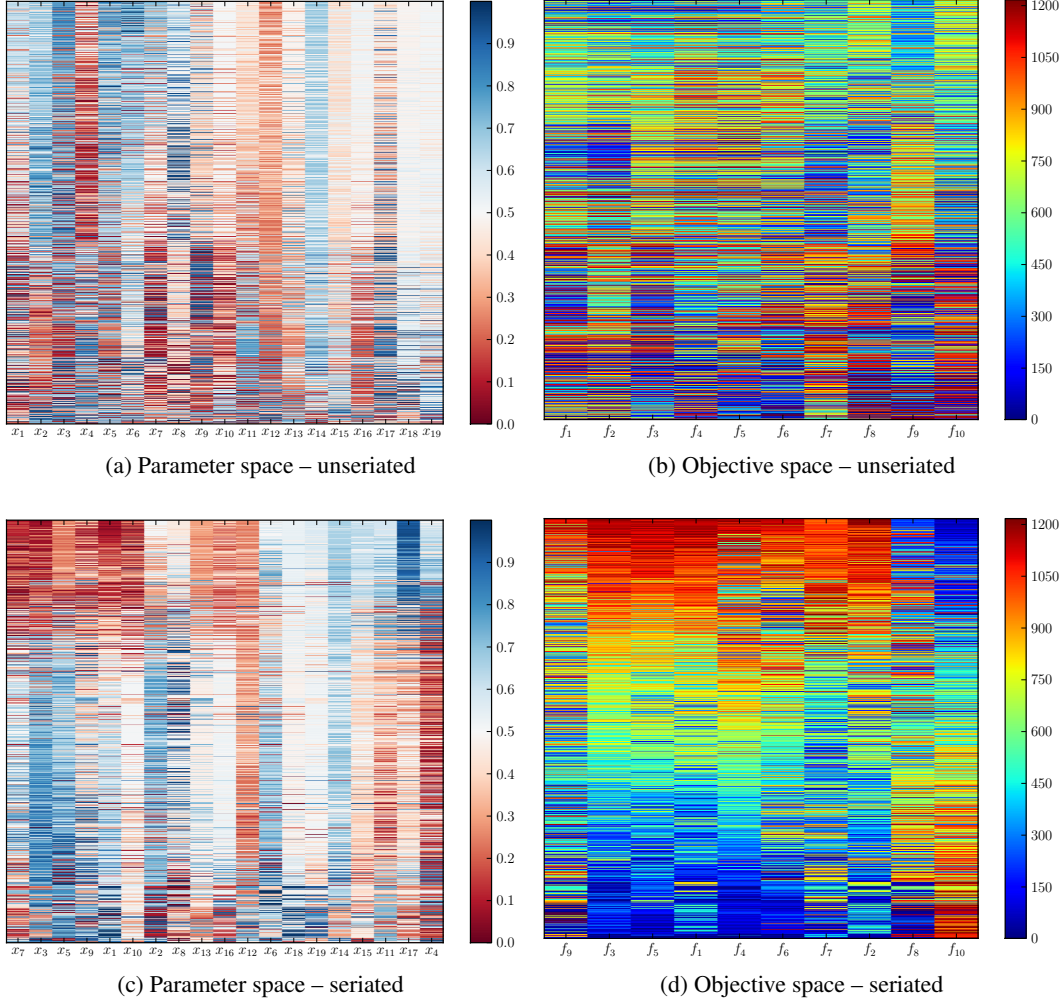


Figure 2: Heatmaps of both the solutions and objective vectors in the elite archive resulting from optimising the 10-objective DTLZ2 instance. The top row shows parameter space (left) and objective space (right) in which the solutions are ordered arbitrarily. Parameters and objectives are in their original ordering. The bottom row shows the result of permuting objective vectors with seriation and ordering the parameter space solutions by that permutation. Parameters and objectives have also been seriated. The seriation in objective space illustrates the trade-off between objectives  $f_8$  and  $f_{10}$  with the rest of the objectives. In parameter space, the seriation of parameters has also enhanced the clarity of the visualisation: a cluster of high-valued parameters is visible in the top-left hand corner of the heatmap, where similar solutions and parameters have been gathered together.

objective  $\lambda_m = (0, \dots, \lambda_m, \dots, 0)$  the barycentric coordinates  $\omega_m$  of an objective vector projection  $\tilde{\mathbf{y}}$  are defined as follows:

$$\tilde{\mathbf{y}} = \sum_{m=1}^M \omega_m \lambda_m \quad (11)$$

with constraints  $\omega_m > 0$  and  $\sum_m \omega_m = 1$ . The vector of the first  $M - 1$  barycentric coordinates corresponding to  $\tilde{\mathbf{y}}$ , noting that since the second constraint requires that the coordinates sum to 1 we can recover the  $M$ th coordinate, are found as follows:

$$\omega_{1:M-1} = \mathbf{B}^\dagger (\tilde{\mathbf{y}} - \lambda_m) \quad (12)$$

where  $\mathbf{B}$  is the  $M \times (M - 1)$  matrix whose  $m$ th column is  $\lambda_m - \lambda_M$  and  $\mathbf{B}^\dagger = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}$  is the pseudo-inverse of  $\mathbf{B}$ . The vertices of the  $M$ -sided polygon  $\mathbb{P}$  in the plane, centred on the origin, are defined as

$$\mathbf{v}_m = \begin{bmatrix} \cos(2\pi(m - 1)/M) \\ \sin(2\pi(m - 1)/M) \end{bmatrix}. \quad (13)$$

An  $M$ -dimensional point in the simplex  $\tilde{\mathbf{y}}$  is visualised by mapping it to the point  $\mathbf{z}$  in  $\mathbb{P}$  that has the same barycentric coordinates  $\omega$  as  $\tilde{\mathbf{y}}$ , namely

$$\mathbf{z} = \mathbf{V}\omega \quad (14)$$

where  $\mathbf{V}$  is the matrix whose columns are the vectors  $\mathbf{v}_m$ . The vertices of  $\mathbb{P}$  are identified with objectives to be minimised, thus an objective vector  $\mathbf{y}_i$  with a large  $y_{im}$  will be mapped close to the  $m$ th vertex  $\mathbf{v}_m$ .

Figure 3 illustrates a visualisation in the plane of the 10-objective DTLZ2 archive. Objective vectors have been coloured by their average rank, and the method has clearly placed solutions with a similar average rank close together. Clearly, as opposed to the seriated heatmaps of the previous section, information pertaining to the quality of a solution on each objective has been lost in the dimension reduction process, however objectives on which the solution is particularly poor can still be identified by observing which vertices it is positioned close to. The solutions with the best average

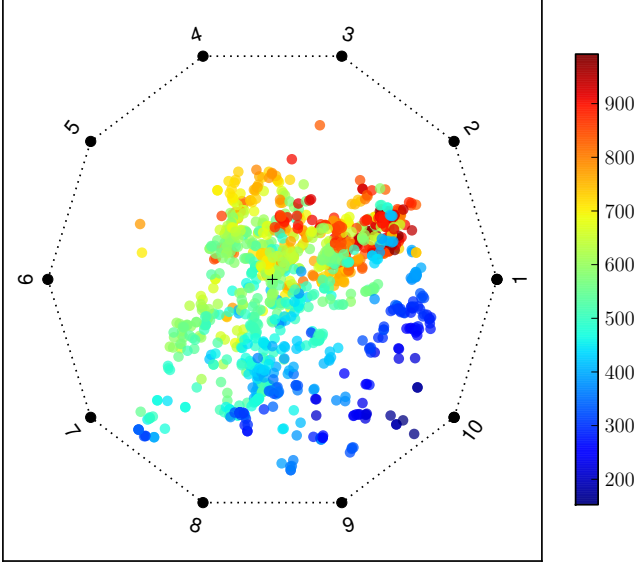


Figure 3: Planar visualisation of the 10-objective DTLZ2 archived population. Solutions are coloured by average rank and proximity of highly ranked solutions to objective  $f_{10}$  indicates that those solutions perform poorly on that objective; this corresponds with information inferred from the seriated heatmaps in Section 3.

rank are placed close to objective 10, which implies poor objective values for  $f_{10}$ . Referring back to the seriated heatmap in Figure 2, this is the objective on which solutions with the lowest ranks, and hence the best average rank, show the worst performance.

Despite the drawback of losing information, visualising objective vectors in the plane is a useful technique as it is easily scalable to large numbers of objectives and can be used with both convex and non-convex Pareto fronts; illustrations of non-convex Pareto fronts are omitted due to lack of space.

## 5. DOMINANCE SIMILARITY

Whilst the planar method demonstrated in the previous section takes account of the mutually non-dominated nature of solutions, it does not consider the dominance relationships between solutions. In this section we describe *dominance similarity* [21] and show how it can be used to visualise both search populations and elite archives.

### 5.1 Dominance Similarity

The intuition behind our definition of similarity lies with the notion that two solutions  $\mathbf{y}_i$  and  $\mathbf{y}_j$  are similar if they dominate the same solutions. Given a third solution  $\mathbf{y}_p$ , we can extend this idea by evaluating on how many objectives the solutions have the same dominance relationships with  $\mathbf{y}_p$ . The dominance relationships are the same if both  $y_{im}$  and  $y_{jm}$  are dominated by  $y_{pm}$ , both dominate  $y_{pm}$ , or both are equal to  $y_{pm}$ . The similarity of the two solutions is then computed by counting the number of times that the relationships are the same. Thus, we define *dominance similarity* between  $\mathbf{y}_i$  and  $\mathbf{y}_j$  with respect to  $\mathbf{y}_p$  as:

$$S(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p) = \frac{1}{M} \sum_{m=1}^M \left[ I((y_{pm} < y_{im}) = (y_{pm} < y_{jm})) + I((y_{pm} = y_{im}) \wedge (y_{pm} = y_{jm})) \right] \quad (15)$$

where  $I(p)$  is the indicator function that is 1 when the proposition  $p$  is true and 0 otherwise. The second term accounts for ties between solutions. The overall dominance similarity between two solutions is then computed by considering the other  $N - 2$  members of the population as the test particle  $\mathbf{y}_p$ :

$$S(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{N-2} \sum_{\substack{p=1 \\ p \notin \{i,j\}}}^N S(\mathbf{y}_i, \mathbf{y}_j; \mathbf{y}_p). \quad (16)$$

Clearly,  $\mathbf{S}$  is symmetric and  $0 \leq S(\mathbf{y}_i, \mathbf{y}_j) \leq 1$ , being equal to 1 when  $\mathbf{y}_i = \mathbf{y}_j$ . From this, we obtain a distance measure thus:

$$D(\mathbf{y}_i, \mathbf{y}_j) = 1 - S(\mathbf{y}_i, \mathbf{y}_j). \quad (17)$$

This measure is equivalent to the Hamming distance between binary strings; we define two binary strings,  $\mathbf{b}_i$  and  $\mathbf{b}_j$ , one for each solution. The  $m$ th element of  $\mathbf{b}_i$  is set to 1 if  $y_{im} < y_{pm}$ , and the distance between the two solutions is obtained by counting the number of positions they disagree on. Since the Hamming distance is a proper metric,  $D(\cdot, \cdot)$  is also a metric.

A method for quickly computing  $D$  is provided by considering that in the single-objective case the distance is just the difference in the ranks  $D(y_i, y_j) = |r_i - r_j|$ . Consequently

$$D(\mathbf{y}_k, \mathbf{y}_j) = \frac{1}{M} \sum_{m=1}^M |r_{km} - r_{jm}| \quad (18)$$

so the distance between individuals is measured by the average magnitude of the difference in their ranks on each objective (note, that this is not the magnitude of the difference of their *average rank*).

### 5.2 MDS and Isomap

Two common visualisation methods are to use *multi-dimensional scaling* (MDS) and *isomap* to reduce the dimensionality of the population by projecting onto a set of new coordinate axes and visualise the resulting low-dimensional population.

Having defined a metric  $D$ , obeying the triangle inequality, the matrix with elements  $D_{ij} \equiv D(\mathbf{y}_i, \mathbf{y}_j)$  is a Euclidean distance matrix [17, 8] and there exists a set of points  $\mathbf{z}_i \in \mathbb{R}^N$  separated by Euclidean distances  $\|\mathbf{z}_i - \mathbf{z}_j\| = D_{ij}$ .

If  $\mathbf{F} = \mathbf{Z}\mathbf{Z}^T$  is a decomposition of

$$\mathbf{F} = -\frac{1}{2} \left( \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) \mathbf{D} \left( \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) \quad (19)$$

then the rows of  $\mathbf{Z}$  are coordinates of the points that generate  $D$ . Metric MDS [11, 16] finds a spectral decomposition of  $\mathbf{F}$ , which is positive semi-definite, and projects the embedding onto the principal eigenvectors of  $\mathbf{F}$ , thus retaining the best linear approximation (in a least squares sense) to the full embedding.

Whilst linear methods, such as MDS, can produce a suitable visualisation of a population, in some cases the data exist on a non-linear manifold; in this event, a more appropriate method to use is *isomap* [19]. Isomap is a global method for reducing the dimensionality of high-dimensional data by preserving *geodesic* distances. It assumes that the data lie on a low-dimensional *manifold* and seeks to preserve pairwise distances along it, rather than preserving Euclidean distances in the original space since two solutions that are close in terms of Euclidean distance might be at opposite ends of the manifold. The process of reducing the dimensionality of data begins by computing a graph of nearest neighbours, either by placing edges between the nearest  $k$  neighbouring points or by identifying all of the neighbours within a certain distance and

placing edges between those. An edge is weighted with the Euclidean distance between the two points that it connects. Then, the shortest path between each pair of points in the data is computed, and this collection of distances forms the basis of dimension reduction using MDS.

Figure 4 presents MDS embeddings for the 10-objective DTLZ2 problem, showing both the final population and elite archive of solutions. The objective vectors in both populations have been coloured by average rank and, as was the case in the visualisation in the plane examples shown in the previous section, solutions with a similar average rank have been clustered together. Interestingly, this is also the case for the population containing dominated solutions, which the planar visualisation could not visualise without distorting the dominance relationships between solutions. In both cases, the solutions have been arranged by the MDS procedure so that the best ranked solutions are on the right-hand side of the embedding and the poorly ranked solutions are on the left, so that the  $x$ -axis of the embedding conveys the quality of the solutions.

Figure 5 presents similar results, but this time the solutions have been embedded into a low-dimensional space using isomap. The relationship between the  $x$ -axis and solution quality remains, and in general solutions with similar ranks have been placed near to each other. The embedding itself is different to that produced by MDS, implying that there is an underlying structure to the solutions that isomap has tried to preserve. Based on this, although both methods provide a clear embedding of a many-objective population, isomap provides additional information about the dominance structure of the population.

## 6. CONCLUSION

With the quantity of many-objective problems that are currently being tackled in the literature, the visualisation of many-objective solution sets is an important but challenging problem. Unlike multi-objective problems, it is not possible to plot the solutions on coordinate axes because a decision maker often cannot comprehend more than three spatial dimensions. In this paper we have presented four methods for visualising such many-objective populations. The first two represent the solutions based on the full set of objectives. One of them visualised solutions as a graph to illustrate the dominance relationships between them while the other used a spectral method to enhance heatmaps of the population. The other methods employ dimension reduction methods that project non-dominated solutions onto the plane and compress the solutions taking into account their dominance relationships with each other, respectively. These methods provide interesting ways of observing and interacting with the solutions in a population that should prove useful to researchers solving many-objective problems.

## 7. REFERENCES

- [1] Jonathan E. Atkins, Erik G. Boman, and Bruce Hendrikson. A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal on Computing*, 28(1):297–310, 1998.
- [2] Peter J. Bentley and Jonathan P. Wakefield. Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. In P.K. Chawdhry, R. Roy, and R.K. Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, pages 231–240. Springer-Verlag, 1998.
- [3] C. Bishop, M. Svensén, and C. Williams. GTM: The generative topographic mapping. *Neural Computation*, pages 215 – 235, 1998.
- [4] David Corne and Joshua 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.
- [5] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC’2002)*, volume 1, pages 825–830, Piscataway, New Jersey, May 2002. IEEE Service Center.
- [6] 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.
- [7] Jonathan E. Fieldsend and Richard 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.
- [8] J.C. Gower. Euclidean distance geometry. *Math. Sci.*, 1:1–14, 1985.
- [9] T. Ju, P. Liepa, and J. Warren. A general geometric construction of coordinates in a convex simplicial polytope. *Computer Aided Geometric Design*, 24(3):161–178, 2007.
- [10] T. Kohonen. *Self-organising Maps*. Springer, 1995.
- [11] J.B. Kruskal. Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrica*, 29:1–27, 1964.
- [12] David Lowe and Michael E. Tipping. Neuroscale: Novel topographic feature extraction using RBF networks. In *NIPS*, pages 543–549, 1996.
- [13] Alireza Nazemi, Andrew H. Chan, and Xin Yao. Selecting representative parameters of rainfall-runoff models using multi-objective calibration results and a fuzzy clustering algorithm. In *BHS 10th National Hydrology Symposium, Exeter*, 2008.
- [14] S Obayashi. Pareto solutions of multipoint design of supersonic wings using evolutionary algorithms. In *Adaptive Computing in Design and Manufacture V*, pages 3–15. Springer-Verlag, 2002.
- [15] Andy Pryke, Sanaz Mostaghim, and Alireza Nazemi. Heatmap visualization of population based multi objective algorithms. In *EMO*, pages 361–375, 2006.
- [16] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409, 1969.
- [17] 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.
- [18] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.
- [19] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.
- [20] David J. Walker, Richard M. Everson, and Jonathan E. Fieldsend. Visualisation and ordering of many-objective populations. In *IEEE Congress on Evolutionary Computation*, pages 3664–3671, 2010.

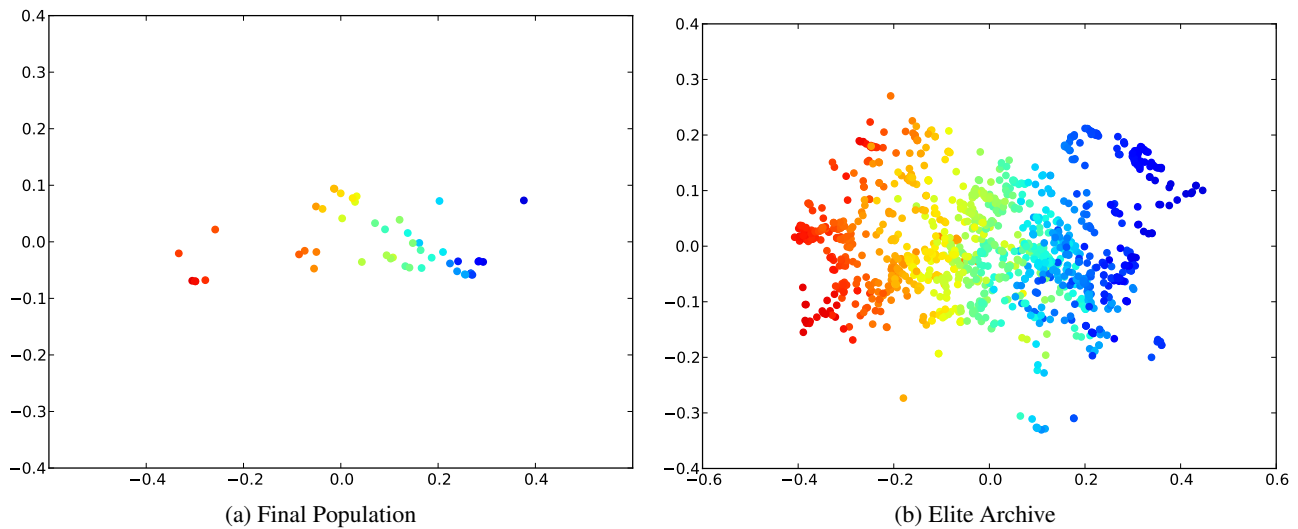


Figure 4: MDS embeddings of the 10-objective DTLZ2 populations. The left-hand plot displays an embedding of the search population in the final generation of the ES while the right-hand plot displays the final elite archive. Solutions are coloured by average rank and there is an agreement between the placement of the solutions in the  $x$ -axis and the average rank of a solution. Solutions with a similar average rank have been placed together.

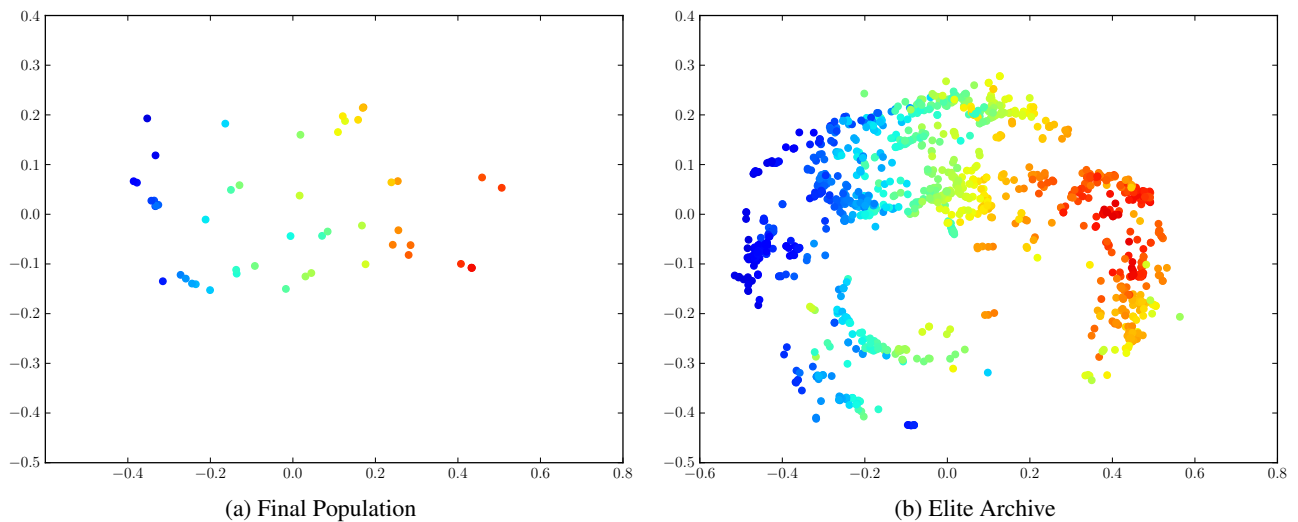


Figure 5: Isomap embeddings of the 10-objective DTLZ2 populations. The left-hand figure again shows the final search population, while the right-hand figure is again the final archive. As with the MDS figure, those solutions with a similar average rank have been placed close together, however the arrangement of projected solutions is different from the MDS. We infer that this is because the solutions lie on a low-dimensional manifold, and the isomap embedding attempts to preserve pairwise distances on this manifold.

- [21] David J. Walker, Richard M. Everson, and Jonathan E. Fieldsend. Visualisation and ordering of many-objective populations. *Submitted to Transactions on Evolutionary Computation*, 2012.
- [22] Joe Warren, Scott Schaefer, Anil N. Hirani, and Mathieu Desbrun. Barycentric coordinates for convex sets. *Advances in Computational Mathematics*, 27(3):319–338, October 2007.
- [23] Leland Wilkinson and Michael Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–0184, 2009.