

Visualising Multi-objective Populations with Treemaps

David J. Walker
College of Engineering, Mathematics and Physical Sciences
University of Exeter, UK
D.J.Walker@exeter.ac.uk

ABSTRACT

Visualising populations of solutions is an important aspect of evolutionary computation (EC), allowing an algorithm user to evaluate the performance of an algorithm and a decision maker to understand the solution set from which they must choose an operating solution. We present a novel approach to visualising multi-objective data, employing *treemaps* to display both solutions and objectives. We define a simple approach to constructing a tree that can represent a multi-objective population in terms of dominance, and illustrate several ways in which it can be used. Examples are provided that reveal characteristics of objective space, as well as combining information about the parameter space component of the population. The paper concludes with a discussion about the further potential of treemaps within EC.

CCS Concepts

•Human-centered computing → Treemaps; •Applied computing → Multi-criterion optimization and decision-making;

Keywords

Treemaps; multi-objective optimisation; visualisation.

1. INTRODUCTION

Multi-objective evolutionary algorithms (MOEAs) abound, however there is still work to be done in designing useful visualisations that relate the parameter and objective space components of solutions. Visualisation within evolutionary computation often focuses on the objective space component, as solution quality is often the basis for the selection of a final operating solution. That said, parameter space also provides useful insight into the characteristics of the evolved solutions. As such, it is important to present a visualisation that illustrates the features of both spaces.

Some existing work has examined this area. Heatmaps have been used to visualise many-objective solutions. Methods presented by [16, 10, 21] showed how heatmaps can be reordered in order to present a population of solutions in terms of parameters as well as objectives. [14] described the visualisation of solutions using

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '15, July 11 - 15, 2015, Madrid, Spain
© 2015 ACM. ISBN 978-1-4503-3488-4/15/07...\$15.00
DOI: <http://dx.doi.org/10.1145/2739482.2768445>

dimension reduction; they incorporated distances from parameter space to ensure that individuals with similar parameters are close together in the visualisation when the isomap [19] dimension reduction procedure is applied.

In this paper we use *treemaps* [13] to visualise multi-objective populations. A treemap is a 2-dimensional visualisation technique in which space is assigned to the nodes in a tree according to the “value” of each node; a high-value node will receive more space in the visualisation than a low-value node. Despite the prevalence of tree-like structures within evolutionary computation, we are unaware of any instance in which a treemap has been used to visualise data arising within evolutionary algorithms. In this work, we present an investigation into the efficacy of treemaps for visualising the search population of an EA tasked with optimising multi-objective benchmark problems. To facilitate this visualisation, we present a simple method for casting a population of multi-objective individuals as a tree.

This paper is structured as follows. Following a brief outline of some relevant background material in Section 2 we present examples of treemaps used to visualise solutions in Section 3 and objectives in Section 4. Some concluding remarks and pointers to future and ongoing work are made in Section 5.

2. BACKGROUND

2.1 Population Visualisation

Beyond the simple case of 3-objective visualisation, methods generally fall into one of two classes. In the first, the dimensionality of the population is reduced using either feature selection (in which redundant objectives are identified and discarded) or feature extraction, where a technique such as principal component analysis or multidimensional scaling projects the objective vectors into a low-dimensional space. In both cases, the population can then be visualised using a standard two or three dimensional method. In the second class, a visualisation is constructed in terms of the full set of objectives. It is this class to which the work presented herein belongs. Visualising a population in terms of all of its objectives is advantageous, because employing dimension reduction inherently results in a loss of information. It does, however, require thought as to how the full set of objectives can be represented.

A basic visualisation is the pairwise coordinate plot, which presents a series of visualisations, each showing two objectives. The difficulty with this, in addition to the potential combinatoric explosion, is that relationships involving three or more of the objectives will be lost. Parallel coordinate plots [12] are useful for illustrating the exact objective values, but can become cluttered and difficult to read. Rearrangement of objective ordering can ameliorate this to some extent [3]. One possibility presented by [22] cast the

population as a weighted directed graph constructed in terms of dominance relationships between the objective vectors, and added additional information through the use of many-objective ranking schemes. Heatmaps have been employed, with work on arranging solutions and objectives to enhance the clarity of the visualisation including agglomerative clustering [16] and seriation [21].

As well as considering visualisation of (relatively small) many-objective populations, we also seek to enhance the visualisation with parameter space information. Some existing work has been done in this area, for example [14] in which the isomap algorithm was used to project individuals into a 2-dimensional space. This requires the computation of geodesic distances, which was done with parameter space distances using solution neighbourhoods obtained in objective space.

2.2 Treemaps and Multi-objective Trees

Treemaps are a 2-dimensional space filling algorithm used to visualise hierarchical data [17]. They have been used widely to represent a plethora of data, and are particularly useful for revealing clusters within data sets. They are a natural choice for visualisation within evolutionary computation, as data often arises that can be represented as such a hierarchy (an example being solutions to a genetic program or the structure of a population of dominating solutions).

Treemaps can be constructed in a range of ways, but approaches generally involve subdividing a space (often a rectangle, although not necessarily) so that the amount of space allocated to a node indicates its importance. [20] provides a good review with examples of possible alternatives to the standard space filling approach. Exactly how the space is divided depends on the algorithm chosen. A popular example is the *squarify* algorithm proposed by [2]. In this work we use a simple recursive algorithm so that the ordering of individuals imposed by the dominance relation is retained.

The multi-objective optimisation literature contains earlier attempts at representing populations of multi-objective solutions in such a structure. Trees are a desirable data structure in which to store multi-objective solutions because of the efficiency with which they can be searched. An example of their use is [8] in which *dominated trees* and *non-dominated trees* are proposed as efficient structures for archiving solutions. Work to improve the efficiency of MOEAs presented in [15] resulted in a data structure called *non-dominated tree*. Trees were also used in a recent endeavour to improve the visualisation of many-objective populations. [3] presented an approach that used *aggregation trees* in order to identify similar objectives so that parallel coordinate plots can be reordered to enhance their clarity. We present an alternative approach to representing problem objectives with a tree later in this paper.

3. SOLUTION VISUALISATION

3.1 Dominance-based Solution Trees

One of the novelties introduced in this work is an algorithm for constructing a dominance-based tree with which to represent objective vectors. We present a scheme in which a tree is constructed based on a combination of the well-known non-dominated sorting procedure [18] and the dominance distance [21].

The procedure begins by inserting a new node which will form the root of the tree. The construction of this node, which we denote by \mathbf{n}^r can be arbitrary, however we constrain that it must dominate the entire population. In this work we use the global best method [9], whereby an objective vector formed of the best value obtained

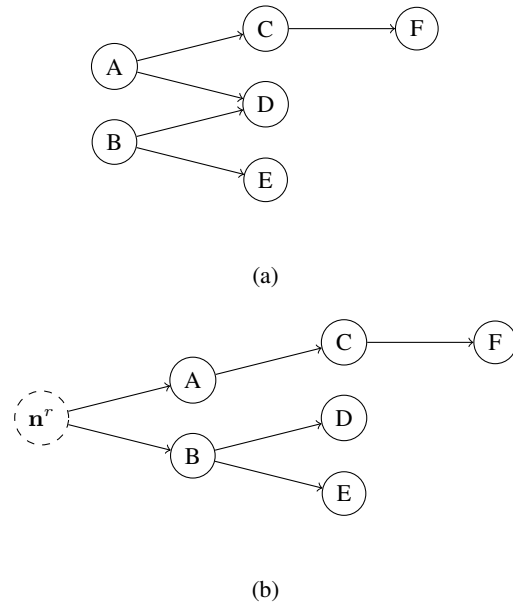


Figure 1: The process by which a dominance tree is constructed. In (a) a Pareto sorting of the solutions is obtained. Node D has two parent nodes (A and B). This tie is broken by examining the relevant dominance distances and retaining the closest parent. In (b) the relationships have been updated so that each node has exactly one parent node, and the pseudo root node has been inserted.

by any solution on each objective is defined:

$$\mathbf{n}^r = \left(\min_i (y_{i1}), \min_i (y_{i2}), \dots, \min_i (y_{iM}) \right) \quad (1)$$

The next step of the tree construction process is to apply non-dominated sorting to the population. The individuals in the first Pareto shell can then be connected by edges to \mathbf{n}^r . The ordering resulting from non-dominated sorting can be represented as a directed graph (which formed the basis of the many-objective visualisation scheme presented in [22]). The nodes of the graph are the population's solutions, which are connected by edges indicating dominance relationships. Considering only those dominance relationships between nodes in adjacent shells, it is common for a node to be dominated by multiple objective vectors. As we require a tree, some of these relationships must be pruned.

In order to identify which dominating nodes should be removed from the graph we employ the dominance distance [21]. This is a proper metric used for computing distances between objective vectors in terms of their dominance relationships. Two objective vectors \mathbf{y}_i and \mathbf{y}_j are "close" in terms of the dominance distance if they share dominance relationships with the rest of the population; if they have few relationships in common with the other objective vectors then they are distant. Here, we identify which is the closest dominating node, and that node is retained as the parent node in the tree while the other dominating dominance relationships are discarded. Retaining the closest dominating node is intended to retain the most important information about population structure. The process of visualising the population is outlined in Figure 1.

3.2 Constructing the Treemap

A plethora of algorithms exist for arranging a treemap. In this paper we use a simple layout algorithm that allocates space to a node in the tree based on the "size" of that node. For the moment, consider a node's size to be a count of itself and its descendants.

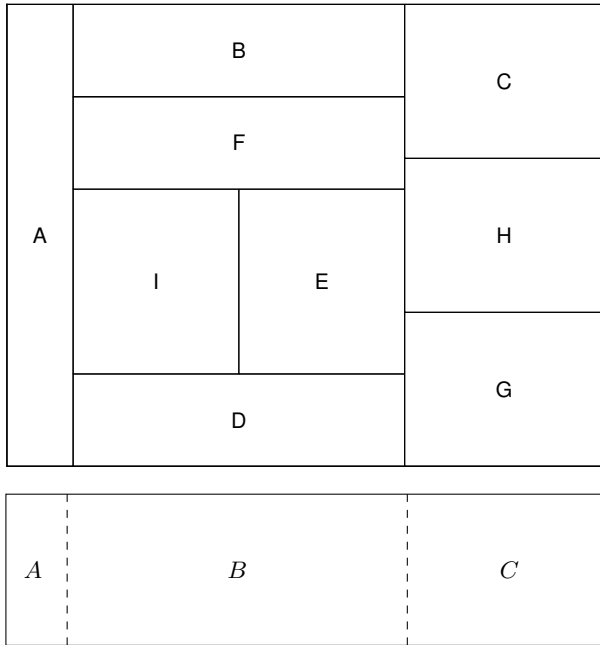
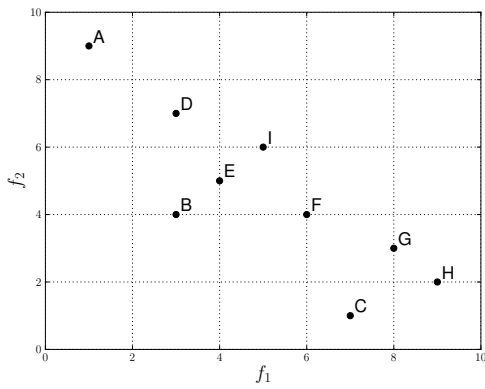


Figure 2: A simple 2-objective population of objective vectors. The top panel shows the objective space, while the middle panel illustrates the corresponding treemap. The bottom panel shows the amount of space assigned to the Pareto optimal individuals A , B and C .

We first demonstrate the operation of the visualisation using a simple 2-objective example population (shown in Figure 2). Of the three Pareto optimal individuals (A , B and C), B dominates the most individuals, so it has the biggest proportion of the treemap. A dominates no other individuals, so it has the smallest proportion of the space. This is shown in the bottom panel of Figure 2.

The treemap is constructed by adding an individual's child nodes recursively. The direction in which nodes are added changes with each layer of recursion. Thus, assuming that the current nodes are being arranged horizontally, child nodes will be placed vertically, grandchildren horizontally, and so on. Child nodes are added in the same fashion as the Pareto optimal solutions, by dividing up the region of the treemap assigned to their parent (thus the region B in Figure 2 is divided between individuals D , E and F according to their respective node sizes. Since individual E dominates I , while D and F dominate nothing, E is allocated more space; some of this space is then used to include I in the visualisation.

In addition to defining node size in terms of the number of individuals an individual dominates, we also scale the node in terms of its Pareto shell. This has the effect of increasing the size of the

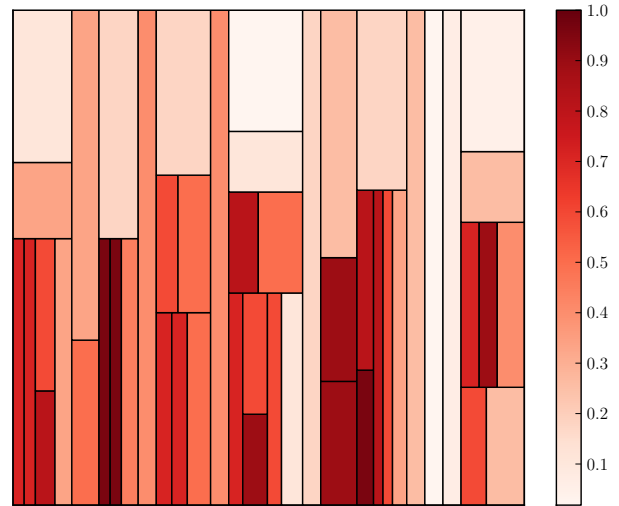
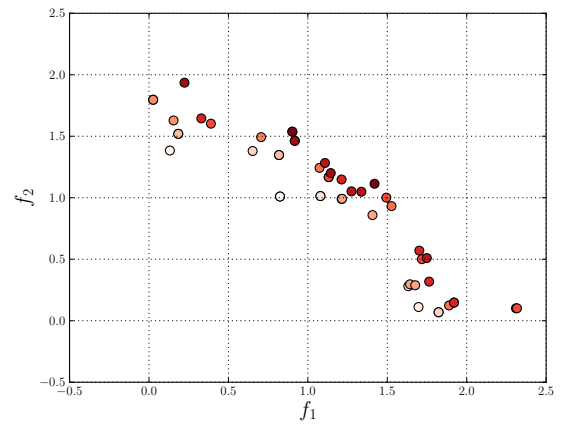


Figure 3: DTLZ2 sample solutions. The top panel shows the objective vectors corresponding to the treemap in the bottom panel. Solutions are coloured according to average rank.

preferable, more highly ranked individuals, and reducing the size of the more poorly ranked ones.

Figure 3 illustrates a treemap visualisation of a population of solutions to the test problem DTLZ2 [5], a well-known multi-objective test problem designed to scale to many-objective instances. A population of $N = 50$ solutions $\{\mathbf{x}_i\}_{i=1}^N$ were sampled from the feasible space ($x_p \in (0, 1), \forall p \in P$); each solution contains P parameters). The top panel shows the objective space corresponding to these solutions for comparison. In both plots, the solutions are coloured according to their *average rank* [1]. A low average rank (and light colour) indicates a good solution and a high average rank (and dark red colour) indicates poorer solutions. The average ranks have been normalised to the range (0,1). The placement of the nodes in the treemap is arbitrary, in so far as no attempt has been made to organise them beyond defining the layout as described above. From examining this treemap, those solutions with higher quality (the lighter coloured nodes) are easily identified by examining the size of the nodes. Dominated individuals are given less relevance by the smaller nodes with which they are represented. We note that there are individuals in the treemap that appear to be Pareto optimal without dominating any other individuals in the population, yet from observing the objective space visualisation we can see that there are no such individuals in the population. This is because the relationships with the solutions they dominate

have been pruned during the construction of the tree representing the population.

Figure 4 shows three more treemaps for solutions to DTLZ2. The top panel is the same population as is shown in Figure 3. This time, however, the treemap has been arranged in terms of the solutions' average rank. This causes the fitter solutions to be clustered in the lower-left hand corner of the treemap, enhancing its clarity. The nodes can be arranged in other ways, for example according to the values of a specific objective (as is demonstrated later).

The second two examples are for three (centre) and five objective populations of 50 solutions to DTLZ2, generated in the same fashion as the earlier example. This illustrates that a treemap can be used to provide a simple indication of population quality with a single visualisation for populations of three or more objectives. When using traditional visualisations, such as scatter plots, it is often necessary to produce multiple views of the data in order to gain a proper appreciation for its structure. That said, as the number of objectives increases the amount of relative quality that can be discerned from the treemap decreases. This is an artefact of the use of dominance to construct the tree; as is well known, as the number of objectives increase the ability of dominance to discriminate between solution quality is impaired. As a result, for a mutually non-dominating population (as would likely occur when visualising a population comprising a large number of objectives), the treemap would display a single row of vertical nodes. The tree would contain no child nodes. This effect is visible in the 5-objective DTLZ2 case, where the right-hand side of the treemap largely consists of childless nodes.

3.3 Visualising Population Diversity

In addition to generating high quality solutions, it is also necessary for a MOEA to maintain population diversity and properly cover the full extent of the Pareto front. We present two approaches for visualising the diversity of a population using treemaps. The first is based on the crowding distance operator used within NSGA-II [4]. Therein, solutions are selected that maximise the diversity of the next generation's search population.

Figure 5 illustrates a treemap highlighting the diversity of a sample population. A set of 2-objective individuals are arranged along the plane, such that the population sorts into two Pareto shells. Those individuals at the edge of the population are close together, and those at the centre are more sparse. The treemap highlights this by colouring the central nodes in darker blue and the edge nodes in a lighter colour, indicating that their crowding distance is smaller.

This is further demonstrated in Figure 6, which shows two populations of solutions to DTLZ2. The solutions were generated by sampling solutions from the Pareto front, then adding noise to the parameters controlling the solutions' distance from the true front; this introduces dominated solutions to the population. In the left-hand panel the solutions have been arranged so that they lie at the edges of the Pareto front, resulting in a large space between two clusters of individuals. The solutions on the edge of these clusters have a large crowding distance. The treemap has been coloured according to crowding distance (large distances are shown in blue) and the discontinuity in the population is clearly visible by the group of dark blue individuals in the centre of the visualisation. The right-hand panel visualises a population with no such discontinuity, and the node colouring is more uniform and light. Such information is useful in evaluating the performance of an optimiser, as it would inform the algorithm user that the space has not been properly explored.

As well as examining population diversity in terms of objective space, we can also incorporate information from parameter space.

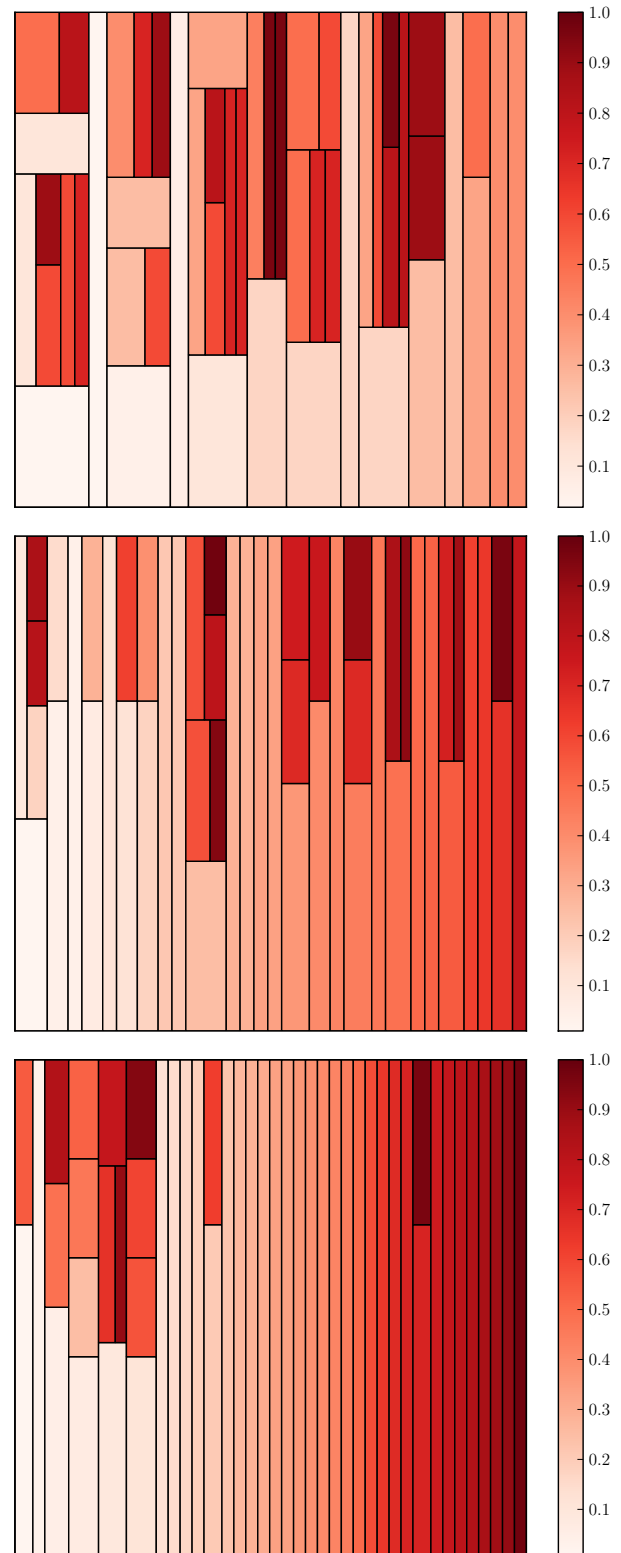


Figure 4: DTLZ2 sample solutions; the populations shown are for $M = 2$ objectives (top panel), $M = 3$ objectives (centre panel) and $M = 5$ objective (bottom panel). The elements of the treemap are ordered according to average rank, with the best average rank appearing in the lower left-hand corner of the treemap.

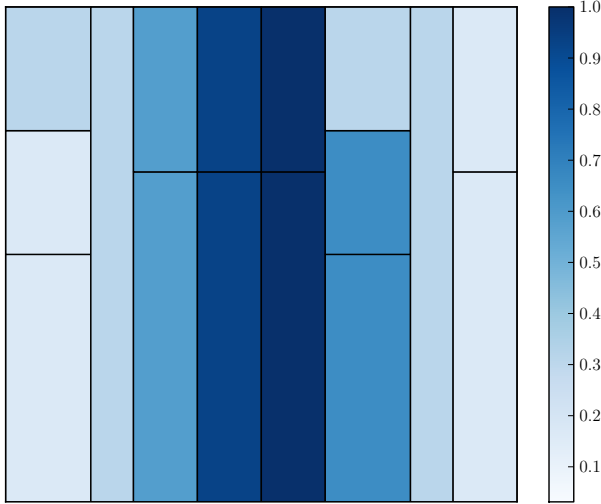
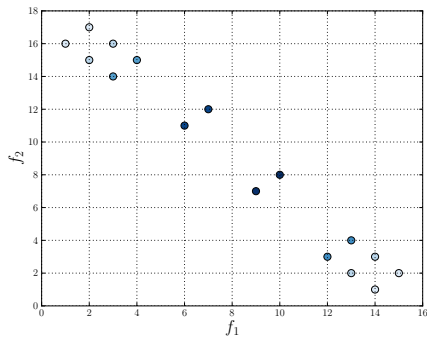


Figure 5: Visualising population diversity using a treemap. Those individuals in the centre of the population with a larger crowding distance are shown in the centre of the treemap. Individuals are ordered according to their objective value for the first objective.

Figure 7 illustrates this; this example takes the DTLZ2 population with poor diversity from Figure 6 and modifies it by moving some of the solutions to the middle of the population. In the visualisations, colour indicates the Euclidean distance to a solution’s nearest neighbour, such that dark orange indicates a greater distance in parameter space. The dark orange individuals lying in the centre of objective space here indicate a discontinuity in parameter space. Should an optimisation problem feature a nonlinear mapping from parameter to objective spaces, revealing information of this type is extremely useful. These examples have demonstrated the use of two distance measures (crowding distance and Euclidean distance) however these could be replaced with arbitrary measures in order to convey the type of information required for a specific task or type of population.

4. OBJECTIVE VISUALISATION

In addition to visualising solutions, visualising objectives can also provide useful insight into the structure of an optimisation problem. This has received less attention in the literature, with a recent example including [7] which uses Spearman’s footrule in concert with multidimensional scaling to project the objectives into a two-dimensional space for visualisation.

In order to visualise objectives with a treemap we must first represent them with a tree. Existing work has examined this issue, with a recent study [3] using a *aggregation trees* to identify harmony and conflict between objectives for the purposes of enhancing

many-objective visualisations. Here we use a simple agglomerative approach based on the characteristic that each of the objectives can be placed in rank order by ranking the population M times, in terms of the M objectives.

We begin by converting the population to *rank coordinates*, whereby r_{im} is the rank of the i -th objective vector on the m -th objective. Spearman’s footrule [6] is a metric for comparing such orderings, and is defined as follows:

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

In the first step in constructing a tree we calculate Spearman’s footrule for each pair of objectives in the population. Tree construction is then a matter of identifying the two closest objectives, according to Spearman’s footrule, and selecting one as the parent node and one as the child node. This choice is made using the footrule distances between the two closest objectives and the other objectives in the population. Given two objectives m and n with the smallest footrule of any in the population, we examine the next nearest footrule p for each of the objectives. The objective with the largest p is the child, and the smallest is the parent node in the tree. Once a tree has been assigned as a child node, it is removed from the population for the purposes of further constructing the tree.

We demonstrate this procedure on two populations of 200 solutions. The first is a population of 10-objective DTLZ2 solutions; DTLZ2 is designed in such a way that there is little or no correlation in the objectives. The other is a population of solutions to a real-world 9-objective radar waveform design problem [11], in which four objectives correspond to the range at which objects can be detected using a candidate waveform, four correspond to the velocity at which they can be detected, and the final objective minimises waveform transmission time (this objective is known to be well correlated with the velocity objectives). Objectives within these classes are well correlated.

Figure 8 presents treemaps that visualise the objectives in the three problems examined. The top panel shows the DTLZ2 population; as can be seen, there is little structure to the objectives, since all but one of the objectives are associated with the root node. The utility of this approach can best be observed in the real-world test case, shown in the bottom panel. There, the objectives have been arranged such that the correct correlation between their objectives can be seen. The range objectives are children of the root node, and the velocity objectives, as well as the correlated transmission time objective (objective 9) are grouped together. In the case of a real world optimisation problem for which correlation information was unavailable, as is often the case, this would reveal important information about the nature of the problem.

5. CONCLUSIONS AND FUTURE WORK

This work has introduced the treemap to multi-objective population visualisation within evolutionary algorithms. We have demonstrated the efficacy of an algorithm for representing a population of multi-objective solutions as a tree, before visualising the resulting trees with treemaps. The technique was demonstrated on several multi-objective populations, and as well as using them to convey characteristics of objective space we incorporated parameter space characteristics to extend their potential applications. The result is a framework into which other quality measures and metrics can be incorporated to suit a specific application. As well as visualising solutions, we demonstrated the use of treemaps for revealing information about a population’s objectives.

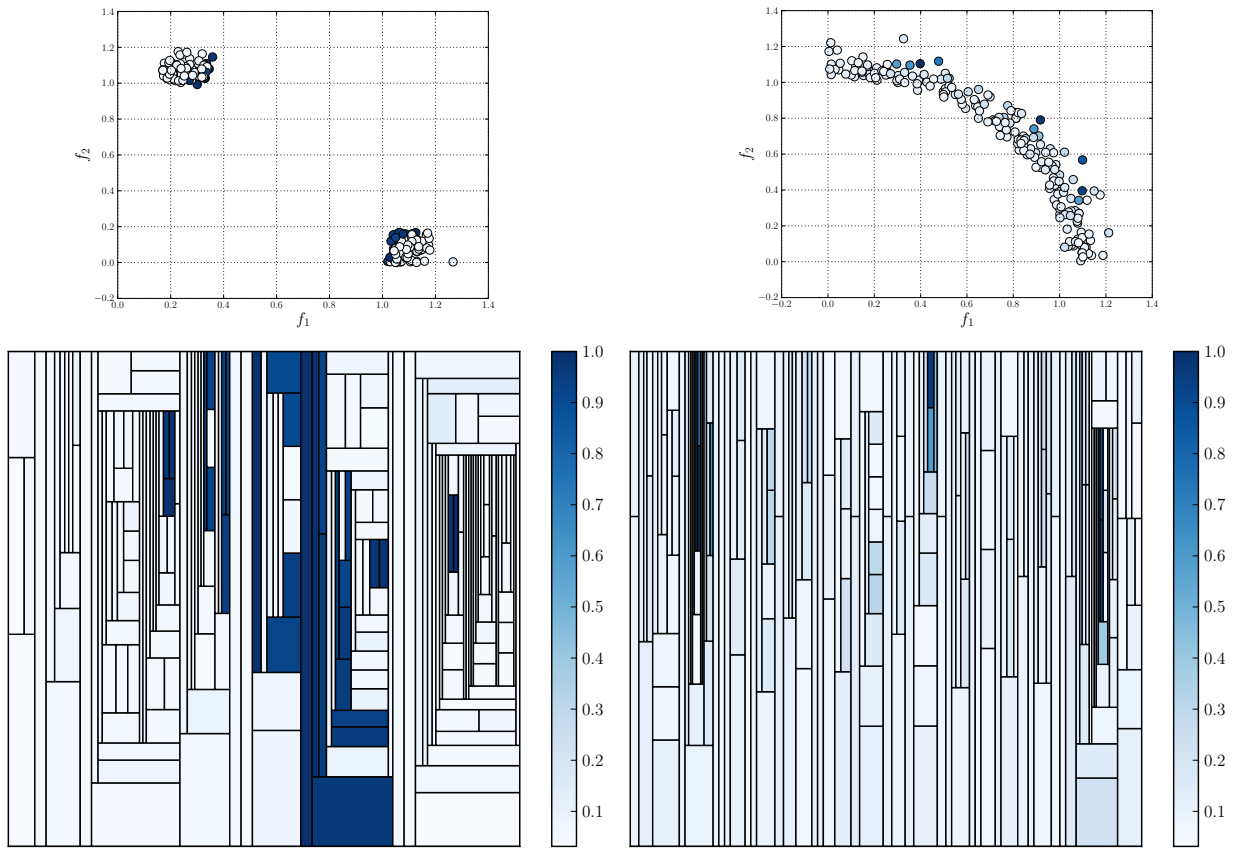


Figure 6: Two examples of visualising population diversity with treemaps coloured according to crowding distance for samples drawn from the Pareto front of DTLZ2. The left-hand treemap clearly shows the discontinuity in the population. In the right-hand figure, there is no such characteristic, so the nodes are more uniformly coloured.

Beyond the applications demonstrated herein, we have several aspects of future work to explore. A shortcoming of this work is in the visualisation of many-objective populations. Such populations abound in optimisation, and in order to be widely applicable the methods demonstrated here need to scale better, so that they can represent such populations. This will require the investigation of alternative layout algorithms, which is ongoing. Trees appear elsewhere in evolutionary computation, notably when used as a solution representation in genetic programming. We are currently extending this work to incorporate the visualisation of such solutions.

An issue with the visualisations shown throughout this paper is that it is difficult to directly observe the dominance relationships between individuals. While this is important, and is an area of future investigation, we do not feel that this significantly detracts from the potential of treemap visualisations, as various other visualisation methods also suffer from this disadvantage (heatmaps and parallel coordinate plots are good examples, as are any of the feature extraction methods). A potential approach to addressing this issue would be to incorporate the ability to select an individual and highlight its dominating individuals in an interactive setting. We intend to investigate approaches for illustrating dominance relationships in a static visualisation too.

6. REFERENCES

- [1] 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.
- [2] Mark Bruls, Kees Huizing, and Jarke van Wijk. Squarified treemaps. In *In Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42. Press, 1999.
- [3] Alan R.R. de Freitas, Peter J. Fleming, and Frederico G. Guimaraes. Aggregation trees for visualization and dimension reduction in many-objective optimization. *Information Sciences*, 298:288–314, 2015.
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, Apr 2002.
- [5] 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.
- [6] 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.
- [7] R. M. Everson, D. J. Walker, and J. E. Fieldsend. League Tables: Construction and Visualisation from Multiple Key

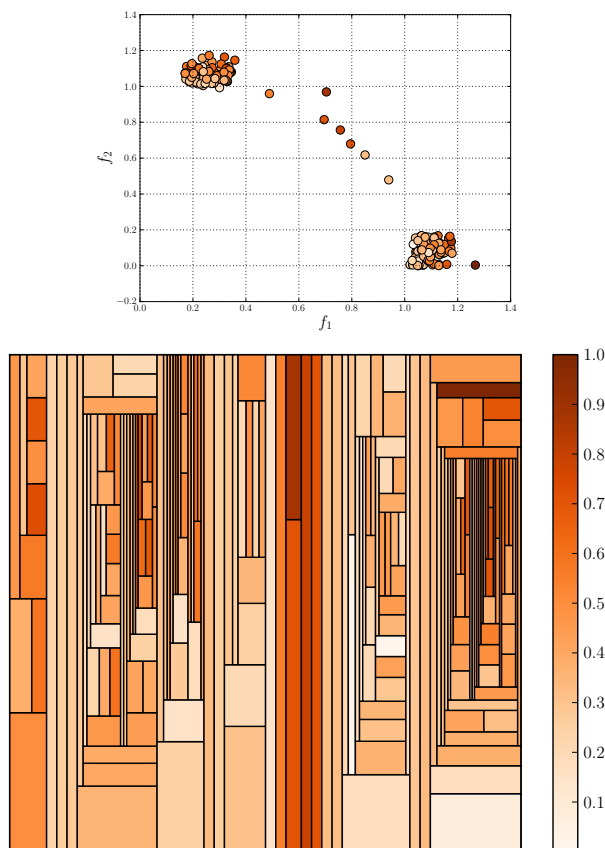


Figure 7: A treemap in which colour illustrates solution diversity in parameter space.

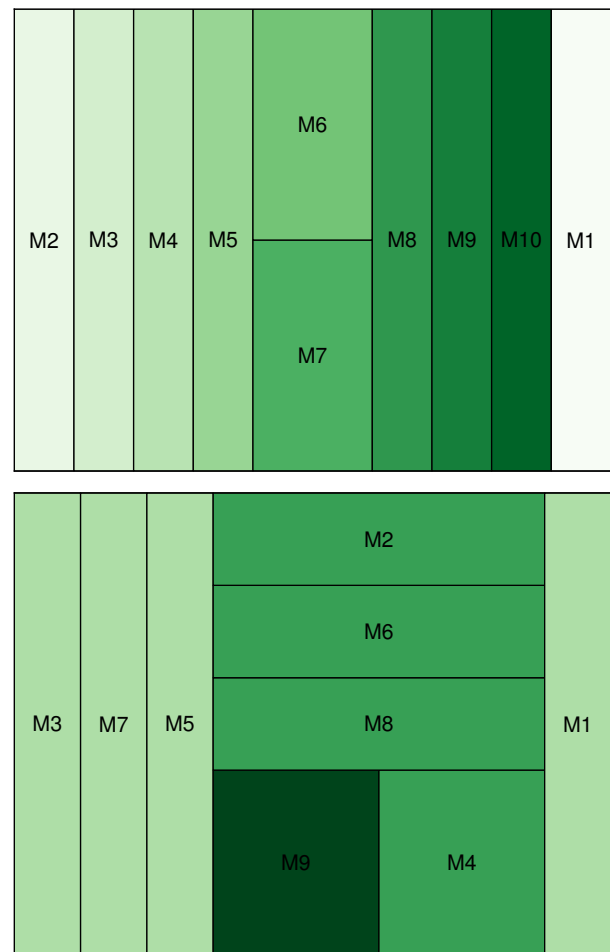


Figure 8: Visualising objectives with treemaps. The top panel visualises 10 objectives to DTLZ2, while the bottom panel illustrates the 9 objectives to the radar waveform problem. Correlated objectives are grouped together in the radar case.

Performance Indicators. Technical report, The University of Exeter, 2012.

- [8] J. E. Fieldsend, R. M. Everson, and S. Singh. Using unconstrained elite archives for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 7(3):305–323, June 2003.
- [9] M. Garza-Fabre, G. Toscano-Pulido, and C.A.C. Coelho. Two Novel Approaches for Many-objective Optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 4480–4487, July 2010.
- [10] 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.
- [11] 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, 2007. Springer-Verlag.
- [12] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*. Springer, 2009.
- [13] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization. In *Proceedings of the 2nd International IEEE Visualization Conference*, pages 284–291, 1991.
- [14] F. Kudo and T. Yoshikawa. Knowledge extraction in multi-objective optimization problem based on visualization of pareto solutions. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1–6, June 2012.
- [15] Joao Batista Mendes and Joao Antonio de Vasconcelos. Using an adaptation of a binary search tree to improve the nsga-ii nondominated sorting procedure. In *Simulated Evolution and Learning*, pages 558–562. 2010.
- [16] A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap visualization of population based multi objective algorithms. In *Evolutionary Multi-Criterion Optimization (EMO 2006)*, pages 361–375, 2006.
- [17] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, January 1992.
- [18] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 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] R. Vliegen, J.J. van Wijk, and E.-J. van der Linden. Visualizing business data with generalized treemaps.

Visualization and Computer Graphics, IEEE Transactions on, 12(5):789–796, Sept 2006.

- [21] David J. Walker, Richard M. Everson, and Jonathan E. Fieldsend. Visualising mutually non-dominating solution sets in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(2), April 2013.
- [22] D.J. Walker, R.M. Everson, and J.E. Fieldsend. Ordering and visualising many-objective populations. In *2010 Congress on Evolutionary Computation*, pages 3664–3671, 2010.