Dynamic Visualisation of Many-Objective Populations

Jonathan E. Fieldsend

University of Exeter, Department of Computer Science, Exeter, UK
J.E.Fieldsend@exeter.ac.uk

Abstract

There has been an increase in research activity recently regarding the visualisation of many-objective populations. Two of the main drivers for this have been (i) to aid decision makers in comparing and selecting designs returned from a many-objective optimisation run, and (ii) to help in the selection of solutions in interactive optimisation. In both of these situations there is often a dynamic element – populations evolving over time change their relative relationships, and the quality comparison measure itself can be altered, redefining member relations. Here we illustrate how a number of existing visualisations from various domains may be applied to many-objective populations to aid the understanding of population relations using the d3 package. d3 is inherently dynamic, and will automatically respond to any changes in the base document underpinning the visualisation, allowing the visualisation package to 'bolt-on' to any other program that can produce or update the underlying file.

Keywords: Visualisation; Multi-objective optimisation

1. Introduction

Although evolutionary many-objective optimisation is a relatively young topic of research, only gaining a distinction from multi-objective optimisation in the early 2000s, many of the most popular data visualisation techniques used within it have a long history. The genesis of pairwise coordinate plots may be traced back to the 17th century work of Descartes, parallel coordinate plots were in use by at least the late 1800s (see e.g. Hewes and Gannett (1883)) and matrix heatmaps were in use by the 1950s (Sneath, 1957). Recent work has started developing approaches that are specifically tailored to visualising aspects of many-objective optimisation. These are typically concerned with conveying quality relationships between putative solutions derived from their D-dimensional objective space locations (such as Pareto-dominance) via a planar representation (e.g. (Köppen and Yoshida, 2007; Fieldsend and Everson, 2013)), and the relationships between objective space and decision space (see e.g. (Pryke et al, 2007; Walker et al, 2013)). Methods which have been developed more broadly by the data visualisation community may also be applied to aspects of many-objective optimisation, for instance visualising the search path of a genetic algorithm (Raper, 2012).

Here we limit ourselves to the consideration of approaches for visualising the putative solutions of a many-objective optimisation algorithm in objective space, both during and at the end of an optimisation run. We develop an open source JavaScript tool, which enables the dynamic visualisation of search populations, exploiting the recent d3 framework of Bostock et al. (2011), which has gained rapid popularity in a wide variety of data visualisation settings.

The tool allows both multiple exploratory visualisations of a single dataset, and dynamic regeneration of a visualisation based on user variation of the relationship measure, and in response to changes in the underlying document driving the visualisation (due to e.g. a population changing during an optimiser run). Currently seven distinct visualisation approaches are present in the toolkit, of which we provide examples of five here.

Unfortunately, the dynamic element of the data visualisation is less easy to convey in a static medium such as this, we therefore concern ourselves principally here with a brief discussion of some of the different visualisations available, which will be made accessible on https://github.com/fieldsend, and the use of one relationship measure to drive the visualisation.

2. Many-objective optimisation

A general multi-objective optimisation problem seeks to simultaneously maximise/minimise D objectives, where each objective depends upon a vector **x** of K parameters or design variables.

The parameters may also be subject to equality and inequality constraints that, for simplicity, we assume can be evaluated precisely. When faced with only a single objective an optimal solution is one that maximises/minimises the objective given the model constraints. However, when there is more than one objective under consideration, solutions may exist for which performance on one objective cannot be improved without reducing performance on at least one other. Such solutions are said to be Pareto optimal.

2.1. Pareto optimality

The notion of dominance may be used to make Pareto optimality clearer. Assuming, without loss of generality, that the goal is to minimise the objectives and there is no noise, a design vector is said to dominate another if it is equal or better on all criteria, and strictly better on at least one. A set of design vectors is said to be a mutually non-dominating set if no member of the set is dominated by any other member.

A solution to the optimisation problem is thus *Pareto optimal* if it is not dominated by any other feasible solution. The non-dominated set of all Pareto optimal solutions is known as the Pareto set.

Pareto dominance comparisons tend to drive the ranking and quality comparisons of solutions in multi-objective optimisers, however, when there are *many* objectives (typically this is deemed as being four or more), then the discriminatory ability of dominance breaks down. This is because increasingly designs are categorised as being mutually non-dominated (the proportion of objective space which is mutually non-dominated with a solution being $1 - \frac{1}{2^{D-1}}$). Because of this, a number of other quality measures have been proposed in the literature, which address the discriminatory shortcomings of Pareto dominance in higher dimensions. These include the favour relation (Dreschler et al., 2001), k-optimality (Di Pierro

et al., 2007), controlling dominance area of solutions (Sato et al., 2007), average ranking and sum of ratios (Bentley and Wakefield, 1997).

We focus on the Pareto comparison measure in the visualisations here, but note that the other approaches can also be embedded, although due to their complexity not all are amenable to the real-time calculations required for dynamic visualisation.

3. Visualisations

Rather than visualise optimiser runs in the illustrations here, we use a running example using data provided by the UK government under the open data initiative, available at https://data.gov.uk. This makes the visualisations reproducible, and uses data that the general reader may already have an inherent understanding of. Specifically we use data on football banning orders from the 2014-15 season, according to offences laid out in the Football Spectators Act 1989. There are 11 different offences for which statistics are available per club, across a range of leagues. Here we restrict the view to just the Premier League. The criteria are counts of banning orders in the season which fall under each of the following categories: violent disorder; public disorder; throwing missiles; racist and indecent chanting; pitch incursion; alcohol offences; ticket touting; possession of an offensive weapon; possession of pyrotechnics; breach of banning order; offences against the property. From a Pareto dominance point of view, we wish to minimise the number of individuals banned under each of these categories for each club. A club is therefore said to dominate another if it has the same or fewer banned individuals compared to another club on all categories (and strictly fewer on at least one).

3.1. Parallel coordinate plots

Parallel coordinate plots have been used extensively by both the multi-objective optimisation community and the wider research community (see e.g. Wegman (1990) for an early discussion of the technique). Figure 1 illustrates a parallel coordinate plot of the running example used in this paper. Although the visualisation is 'lossless' (there is no compression of the objective values), it can rapidly become quite cluttered (there are only 20 football teams in this example; in many datasets there are considerably more population elements to plot).

Each member is illustrated by a sequence of line segments connecting the element name (in this case on the left hand side), progressively through the values corresponding to each criteria (the coordinate plumb lines arranged left to right). The ordering of the criteria itself in such a plot can be an optimisation problem, however here we order the coordinates according to the original data source. As all criteria are to be minimised the lower values are at the bottom, and therefore a club with a sequence of connected line segments which are lower is better. Parallel coordinate plots do have the advantage that the raw data can be extracted fairly accurately directly from the visualisation – Figure 2 for instance shows the values corresponding to Arsenal football club, and reading off the values of the different coordinate

axis gives a good indication of the underlying data. This property is not evident in many alternative plotting approaches.
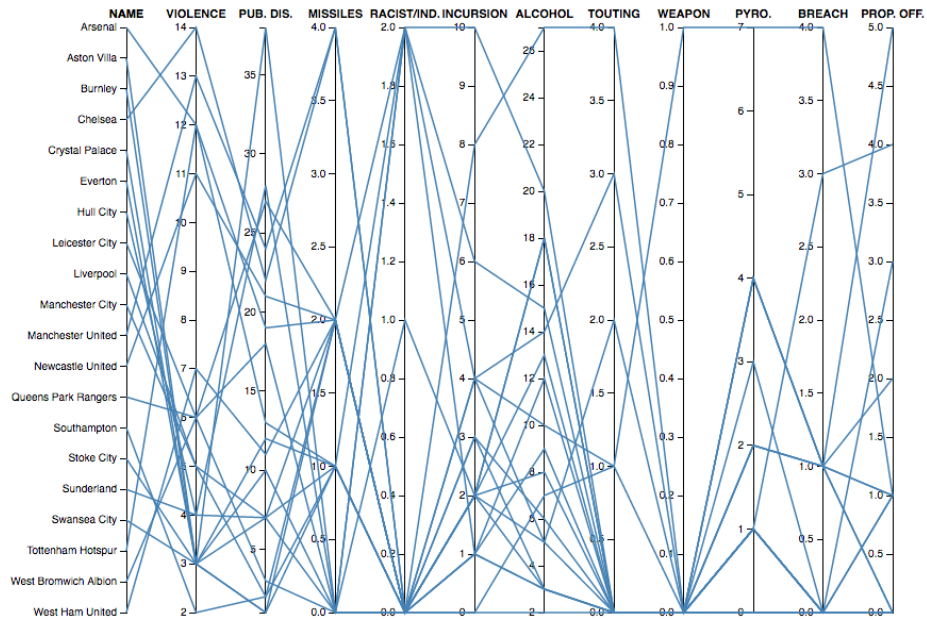

Figure 1 Parallel coordinate plot of the 11-dimensional football banning order data set. All team criteria values highlighted.
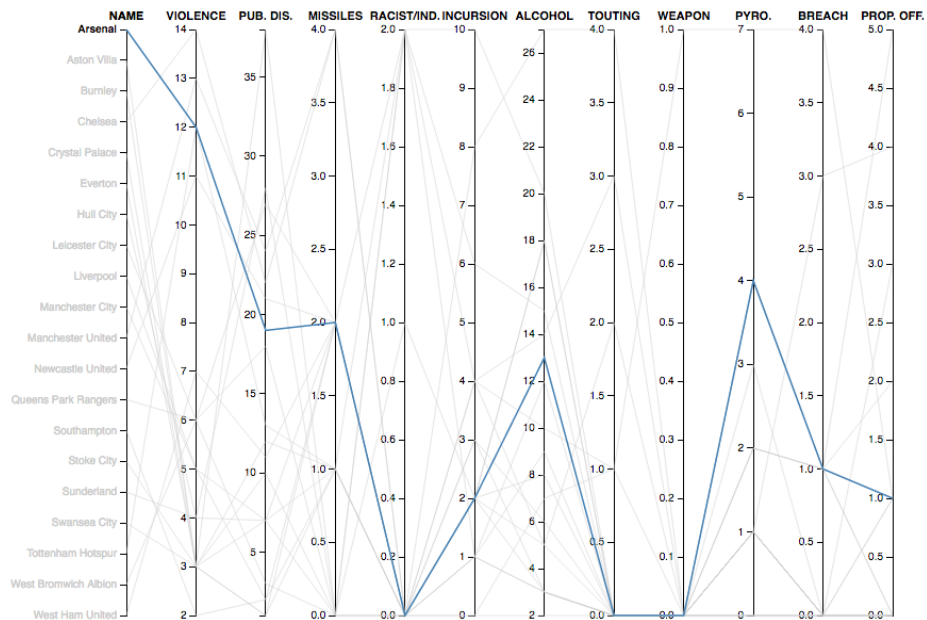

Figure 2 Parallel coordinate plot with a single individual highlighted.

3.2. Pairwise scatterplots

Pairwise scatterplots are another lossless representation, and can aid in detecting correlations between pairs, however they rapidly become unwieldy as the number of objectives increases, as amply illustrated in Figure 3.
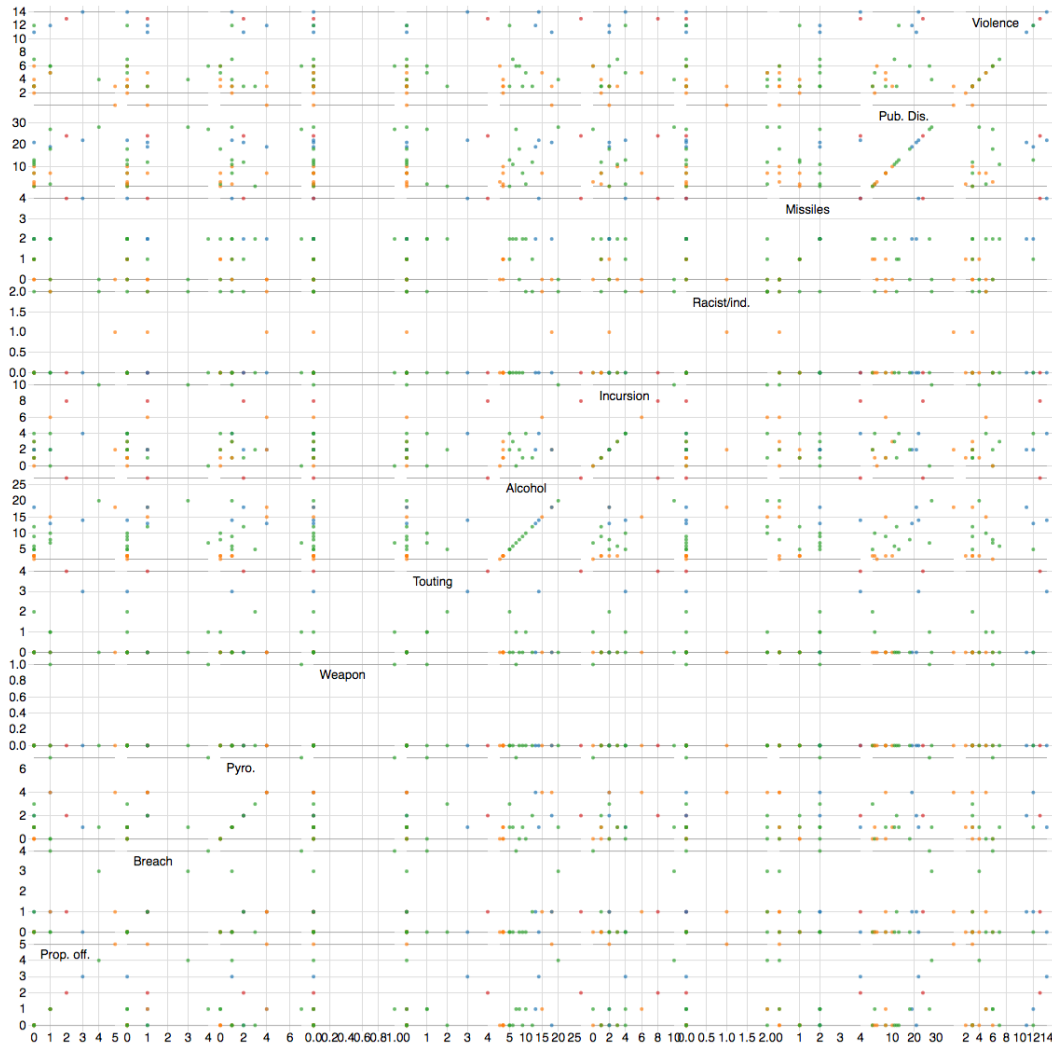

Figure 3 Pairwise scatterplots of the 11 football banning order criteria.

3.3. Hive plots

Hive plots (Krzywinski et al., 2011) are a relatively new visualisation technique, which attempts to bring order to network connectivity graphs. As far as the author is aware they have not been used previously to examine multi-objective data. A variant for displaying a multi/many-objective population is shown in Figure 4. Individuals (football clubs in this illustration) are represented as nodes, with edges between nodes representing domination relationships. The top vertical line of nodes is the non-dominated subset of the data (shell 0

when non-dominated sorting is used (Deb et al., 2001)). The members of the final shell are shown in the bottom left. All edges connecting to the top vertical line of nodes are therefore *outgoing* domination links, as they are dominating the other nodes they connect to. Conversely, all edges connecting to the final shell member are incoming domination links (as by definition the final shell extracted from non-dominated sorting contains elements which do not dominate any other in the population). The edges are coloured according to the dominating node (with each shell being coloured differently). This allows rapid determination regarding a link's dominating/dominated relationship to connecting nodes. The remaining intermediate shells are denoted in the lower right two lines of nodes. These lines are duplicated. The top line of nodes shows incoming domination links from shell 0, and domination links between these intermediate shells are shown via edges between these two duplicate lines. The lower line of nodes is connected to the final shell, showing the outgoing domination links to this last shell.
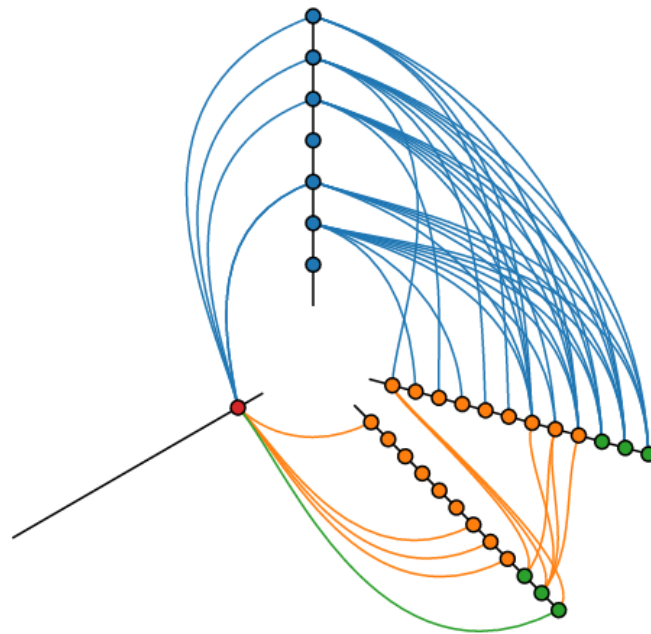

Figure 4 Hive plot illustration of the football banning order data. The blue nodes at the top are shell 0 members (the non-dominated set). The red node at the lower left is the final non-dominated shell (the worst performing shell), and the members of the two intermediate middle shells (coloured orange and green) are duplicated on the two lower right lines of nodes. Domination links are coloured according to the shell membership of the dominator.

3.4. Circle packing

Circle packing can usefully illustrate the number of solutions lying behind (dominated) by different elements of a population, and facilitate the dynamic 'drilling down' into data. Figure 5 illustrates the use of the circle packing plot for many-objective data. At the top level the shell 0 individuals are shown, with the number they dominate indicated by the size of the blue container they label. Note the two members in the centre of the plot which are in shell 0 but do not dominate any others (hence they are the smallest circles, coloured white, and are not containers). The same property of these two shell 0 individuals can be identified in Figure 4, where two shell 0 nodes are not linked to any others. The other subplots in Figure 5 show the progressive selection and expansion of the region behind particular individuals.
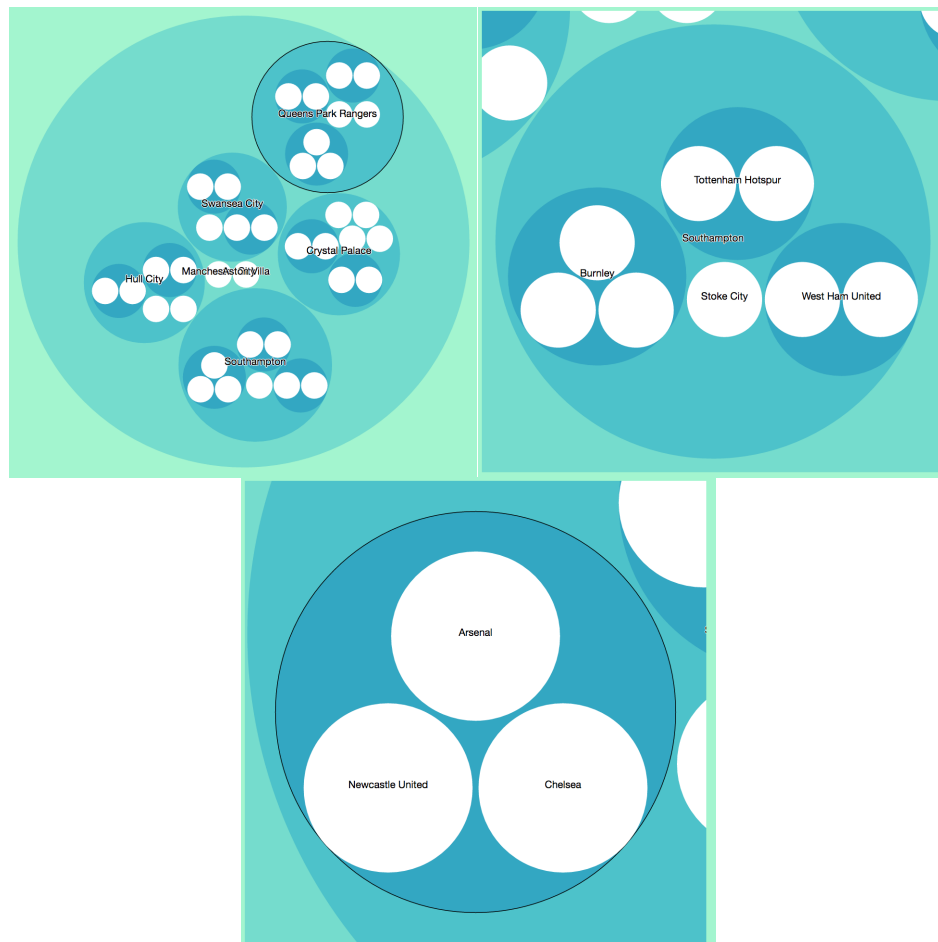

Figure 5 Circle packing illustration. Top-left: top-level view showing the seven members of shell 0, and the different groups of members that lie under (are dominated) by them. Top-right: drilling into the region dominated by Southampton. Bottom: drilling down further, into the region dominated by Burnley.

As with hive plots, the author is not aware of the use of circle packing visualisation within the multi-objective optimisation community (however, the related treemap has been used, e.g. Walker (2015)).

3.5. *Domination wheel*

The dependency wheel plot links items that are dependent on each other. The elements circle the outside and links plotted as lines or ribbons between elements that pass by the centre of the plot. It can readily be used to denote domination (dominating and dominated) relationships between individuals (although we are not aware of this graph being used in such a way previously). In the context of multi/many-objective population visualisation, we refer to such a graph as a *domination wheel*. An example is shown in Figure 6 on the running example.
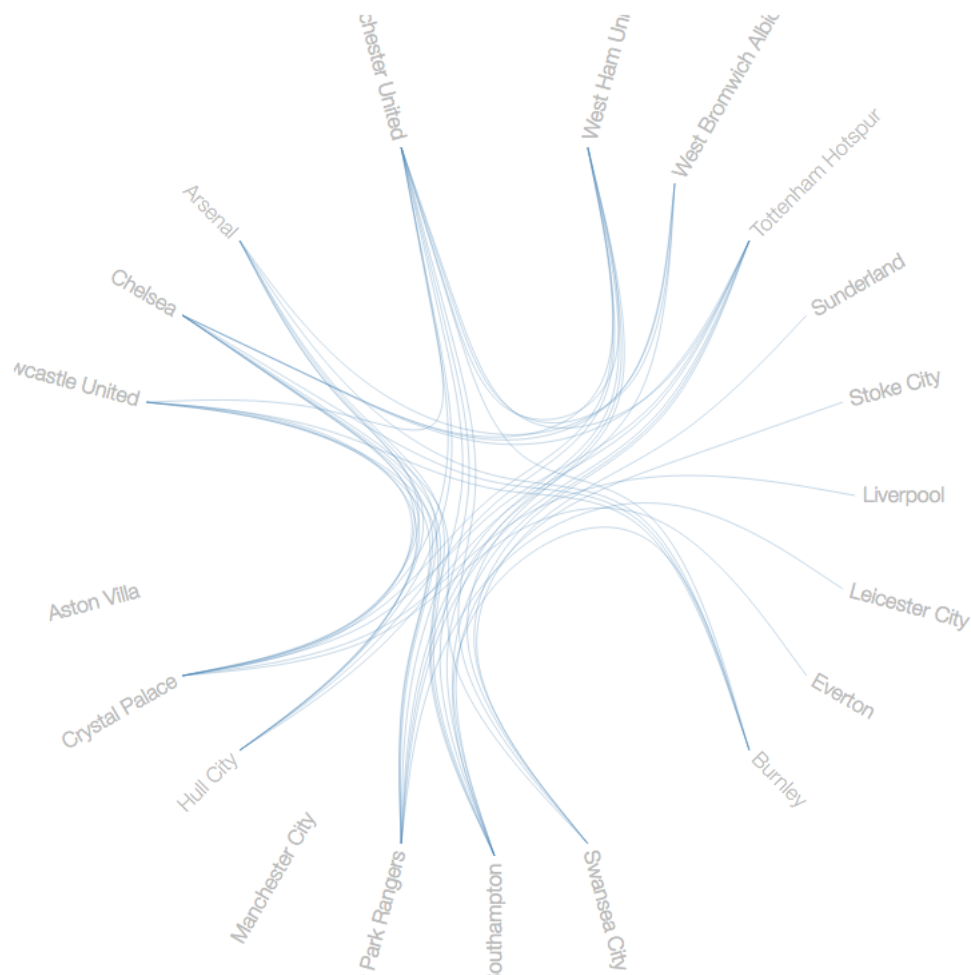

Figure 6 Domination wheel. Links between labelled clubs indicate there is a domination relationship. Clubs have been grouped by shell.

Figure 7 shows two different solutions (football clubs) being selected in each of the subplots. Red links denote connections to dominating clubs, and green links to dominated clubs. Unconnected clubs are therefore mutually non-dominating with the selected individual. Note the number of links and prevalence of a particular colour quickly gives a visual indication of the overall quality of a solution with respect to the other set members.
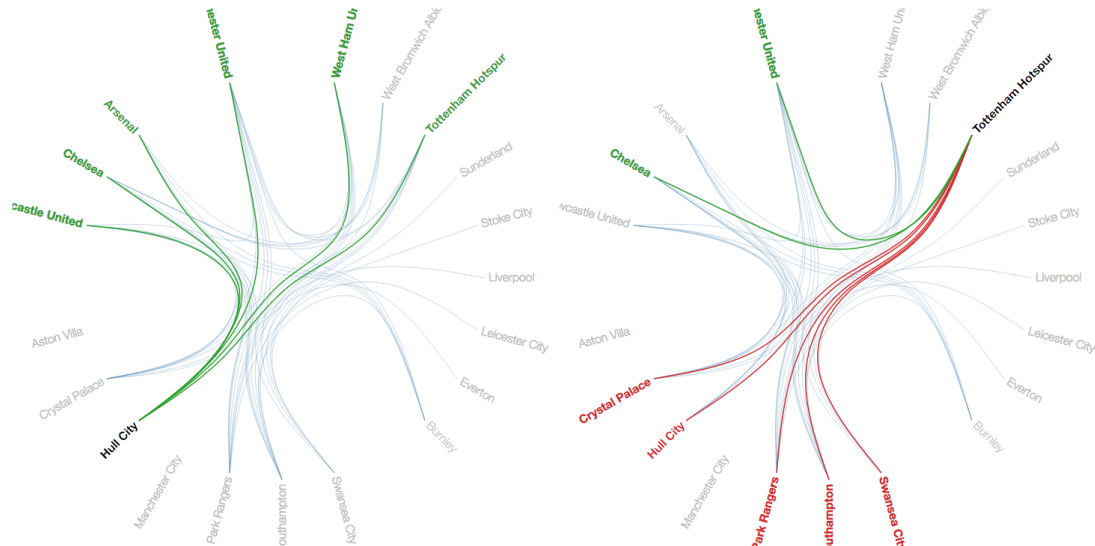

Figure 7 Two views of the domination wheel with different individuals selected. Left: A shell 0 solution is selected (Hull City), all links are green as only dominated solutions are connected to it. Right: A shell 1 solution is selected (Tottenham Hotspur). Red lines connect to the shell 0 members that dominate it, and green lines connect to the members of shell 2 and 3 it dominates.

4. Conclusion

A number of different visualisations of many-objective population data have been illustrated here, some which are already widely used by the multi- and many-objective optimisation community, but also a number which hitherto have had limited or no exposure in this domain. Many of them present useful structural information about the relationships between members of the population, and, although the dynamic nature of the visualisations is difficult to convey in this static medium, the visualisations are ideally suited to the dynamic optimisation environment (subject to the complexity of the relationship measure used).

5. Acknowledgements

References

Bentley, P. and Wakefield, J. (1997) Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. Soft Computing in Engineering Design and Manufacturing, 5:231–240

Bostock, M., Ogievetsky V. and Heer J. (2011), D³ Data-Driven Documents, IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2301-2309.

Deb, K., Agrawal S., Pratap, A. and Meyarivan T. (2000), A Fast Elistist Non-Dominated Srting Genetic Algorithm for Mutli-Objective Optimization: NSGA-II, In: Parallel Problem Solving from Nature, pp 849-858

Di Pierro, F., Khu, S.-T. and Savic, D. (2007), An Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization. IEEE Transactions on Evolutionary Computation, 11(1):17 –45

Dreschler, N., Dreschler, R. and Becker B. (2001), Multi-objective optimisation based on relation favour, In: Evolutionary Multi-criterion Optimization, pp 154-166

Fieldsend, J.E. and Everson, R.M. (2013), Visualising high-dimensional Pareto relationships in two-dimensional scatterplots, In: Evolutionary Multi-criterion Optimization, EMO 2013, pp 558-572

Hewes, F.W. and Gannett, H. (1883), Scribner's statistical atlas of the United States, showing by graphic methods their present condition and their political, social and industrial development, New York, C. Scribner's sons

Köppen. M. and Yoshida, K. (2007), Visualization of Pareto-sets in evolutionary multi-objective optimization, In: Proceedings of the 7th International Conference on Hybrid Intelligent Systems, pp 156-161

Krzywinski, M., Birol, I., Jones, S. J. M. and Marra, M. A. (2012), Hive plots—rational approach to visualizing networks, Briefings in Bioinformatics, 15(3): 627-644

Pryke, A., Mostaghim, S. and Nazemi, A. (2007), Heatmap Visualization of Population Based Multi Objective Algorithms, In: Evolutionary Multi-Criterion Optimization, EMO 2007, pp 361-378

Raper, S. (2012), Visualising the path of a genetic algorithm, http://www.coppelia.io/2012/04/visualising-the-path-of-a-genetic-algorithm/ Date accessed: 9/3/2016

Sato, H., Aguirre, H. and Tanaka, K. (2007). Controlling Dominance Area of Solutions in Multiobjective Evolutionary Algorithms and Performance Analysis on Multiobjective 0/1 Knapsack Problems. IPSJ Digital Courier, 3:703–718

Sneath, P.H.A. (1957), The Application of Computers to Taxonomy, Journal of General Microbiology, 17(1):201-226

Walker D.J. (2015), Visualising many-objective populations with treemaps , In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp 963-970

Walker D.J., Everson R.M. and Fieldsend J.E. (2013), Visualizing mutually nondominating solution sets in many-objective optimization, IEEE Transactions on Evolutionary Computation, 17(2):165-184

Wegman E.J. (1990), Hyperdimensional Data Analysis Using Parallel Coordinates, Journal of the American Statistical Association, 85(411):664-674