**31** Lyndon While, Phil Hingston, Luigi Barone, and Simon Huband. A Faster Algorithm for Calculating Hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1):29–38, February 2006.
**32** Gary G. Yen and Zhenan He. Performance Metric Ensemble for Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 18(1):131–144, February 2014.
**33** Saul Zapotecas-Martínez, Hernán E. Aguirre, Kiyoshi Tanaka, and Carlos A. Coello Coello. On the Low-Discrepancy Sequences and Their Use in MOEA/D for High-Dimensional Objective Spaces. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 2835–2842, May 2015.
**34** Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.
**35** Eckart Zitzler, Marco Laumanns, and Stefan Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. In Xavier Gandibleux, Marc Sevaux, Kenneth Sörensen, and Vincent T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, Lecture Notes in Economics and Mathematical Systems, pages 3 − 37, Berlin, 2004. Springer.
**36** Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.
**37** Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.

## 4.5 KaKaRaKe – User-Friendly Visualization for Multiobjective Optimization with High-Dimensional Objective Vectors

*Kerstin Dächert (Fraunhofer ITWM – Kaiserslautern, DE), Kathrin Klamroth (Universität Wuppertal, DE), Kaisa Miettinen (University of Jyväskylä, FI), and Ralph E. Steuer (University of Georgia, US)*



### 4.5.1 Introduction

Multiobjective optimization problems involve multiple conflicting objective functions to be optimized simultaneously. When solving such problems, we generate Pareto optimal solutions reflecting different trade-offs among the objectives. Typically, some type of preference information coming from a decision maker, a domain expert, is needed to identify the most preferred Pareto optimal solution as the final one.

By studying different Pareto optimal solutions and providing preference information, the decision maker can learn about interdependencies among the objectives. This task can be supported by visualizations. Visualizations can also be used to represent the progress of the solution process.

Surveys of various visualization techniques to represent a set of Pareto optimal objective vectors include [9, 12, 13, 18]. They discuss widely-used visualization techniques like parallel coordinate plots (sometimes known as value paths), spider web charts, petal diagrams, star coordinate plots, and glyphs. Examples of more recently proposed visualization methods for multiobjective optimization are heatmaps [6], knowCube [16], interactive decision maps [11], the prosection method [17], and 3d-radvis [7]. Visualization aspects in multiobjective optimization are also discussed in [5].

Questions involved in developing and applying visualizations include how to scale visualizations with increasing numbers of objectives; how to support the decision maker to gain understanding of the progress of the solution process; how to conduct "sensitivity analysis" by which the decision maker can understand the consequences of actions; can visualizations point to directions where small sacrifices in some objective can provide good improvement in some other objectives; how to find solutions with "robustness" properties; and how to communicate this to the decision maker. Overall, the decision maker should be able to affect the appearance of the visualization.

Our motivation is to develop visualizations to provide support to decision makers by identifying interesting aspects of Pareto optimal objective vectors as solution candidates and of the solution process itself. We aim at handling larger numbers of objectives (say between five and nine) in a comprehensible way without cognitively burdening the DM too much.

When using traditional visualization techniques, e.g., radar plots or parallel coordinates plots, the decision maker can conveniently grasp important interrelationships among the objectives when they are placed close to each other yet finding the best order in which to present the objectives may require some trial and error. Our goal is to assist the decision maker in identifying aspects of interest by pre-processing the set of solution candidates. We have two goals: to detect which objectives are strongly correlated or uncorrelated with one another, and to reduce the number of solutions presented when desirable by detecting similarities among them. The latter can be understood as finding good representative solutions.

One can detect correlations among objectives, for example, to reduce the computational burden by decreasing the number of objectives (see, e.g. [1, 2]). However, we are not aware of any such approach that is applied in a combined way to reduce the visual effort of the decision maker by clustering both objectives and solution candidates simultaneously.

We propose a new visualization technique by applying bi- or co-clustering to the set of Pareto optimal objective vectors. In this way, we can simultaneously visualize similarities and differences among objective functions and solution candidates. To communicate the information visually, we modify the idea of parallel coordinate plots so that the distances between correlated objective functions are shorter than otherwise and similar solutions are given the same color. Thanks to this kind of visualization, the decision maker can handle higher numbers of objectives and solution candidates and concentrate on aspects that are of greatest interest.

As for the following, in Section 4.5.2 we describe how data for a study like this can be obtained. In Section 4.5.3 we survey clustering techniques applicable to our needs, and in Section 4.5.4 the new visualization technique of this study is proposed. Finally, we conclude in Section 4.5.5.

### 4.5.2 Data generation

As mentioned above, we want to support the decision maker in handling larger numbers of objectives than is usually the case and by this we mean problems with 5 or more objectives. Of course, if one thinks long enough, one can probably imagine problems with almost any number of objectives, but because of the almost total lack of work in the area, we will in this paper only focus on problems with up to 9 objectives to get things started. However, there is a kind of "Catch-22" with problems in this area. On one hand, people are reluctant to attempt applications with many more than 5 objectives as there are essentially no tools for processing points from such high-dimensional Pareto fronts, and on the other hand, people have been slow to begin work on tools for processing high-dimensional solution vectors because of the lack of data from problems upon which to test such tools.

To get around this we describe two methods for randomly generating nondominated vectors in objective space. The first method is for the generation of Pareto optimal solutions as if coming from a problem in which all of the objectives have no especial correlations with one another. This method uses the random multiple objective linear program (MOLP) generator described in the documentation for Adbase [15] and then uses that code to solve the resulting MOLPs for all nondominated vertices of the problem's feasible region in the objective space. Data sets of any size for any number of objectives can be generated in this way by adjusting the parameters (numbers of objectives, constraints, variables) of the generator. Once a data set of Pareto solution vectors is generated, it is immaterial how it was generated for testing.

The second method is for the generation of Pareto solutions as if coming from a problem in which there are groups of objectives that have within group correlations with one another. For instance, consider a problem with 9 objectives such that 4 of the objectives are clustered in one group, 3 are in another, and the last two are in a third. In this method, the feasible region of a problem is generated in the same way as the first method, but instead of letting the MOLP generator generate the gradients of the objectives, the gradients of the objectives are randomly generated by a special routine that assures the clusters desired and their within group correlations. In this way, we can develop as many data sets of the types desired as needed. As for the example of this report, only the second method was necessary to generate data for it.

### 4.5.3 Clustering techniques

We now assume a given two-dimensional data set where each row corresponds to a Pareto optimal objective vector (solution candidate) and each column corresponds to an objective function. Our aim is to aggregate the given data matrix into solution-objective clusters according to the following rules.

1. Objectives are clustered if the values in any solution are “similar”.
2. Solutions are clustered if the values in any objective are “similar”.

“Similar” means as close as possible with respect to a specified distance, e.g., a Manhattan or Euclidean distance. The clustering is to be done simultaneously in both dimensions, rows and columns.

In the literature, there exist the concepts of biclustering and co-clustering which seem to denote the same, but are used in different communities and/or applications. A helpful survey is presented in [10].

Biclustering was first applied to bioinformatics, in particular to identify co-expressed genes under a subset of all conditions/samples, see [19] for a recent overview. A mathematical review of successful biclustering techniques is given in [3]. The authors particularly show that most of them are based on singular value decomposition (SVD). However, algorithms differ in the definition of biclusters. Some assume constant values in the data on rows within one cluster, some on columns, some on both. Others are more flexible and allow coherent values along rows and/or columns.

There are also “spectral” variants, i.e., “spectral biclustering” and “spectral co-clustering”, that make use of the underlying graph structure of the problem. Spectral algorithms are common in graph partitioning problems and refer to algorithms that compute eigenvalues, eigenvectors, and singular values to solve the underlying graph problem, see e.g., the lecture notes https://courses.cs.washington.edu/courses/cse521/16sp/521-lecture-11.pdf. There, we also find an illustrative example in which common clustering methods like $k$-means fail while a spectral algorithm detects a meaningful cluster. In [4], a spectral co-clustering algorithm is proposed for a text mining problem.

In the next section, we use a Python implementation of the biclustering algorithm of [8]. One should note that even though the terms bi- and co-clustering are often regarded as synonyms, they refer to different algorithms in the scikit-learn package employed. More details are given below.

### 4.5.4 Preliminary results

We present tentative numerical results for a data set containing 88 mutually nondominated objective vectors with nine objective functions. The data were generated by the second method described in Section 4.5.2, aiming at three objective clusters. The tests were implemented in Python 3.7 using the scikit-learn package for data clustering [14] and plotly.express to generate the parallel coordinate plots. The data were clustered w.r.t. objective vectors (rows) and objective functions (columns) using spectral biclustering [8]. This includes an automatic data normalization which was implemented as log-normalization, see the documentation on https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralBiclustering.html for further details.

The original, unclustered data are shown in a parallel coordinates plot in Figure 18a, while Figures 18b to 18d show the results obtained with spectral biclustering using log-normalization for different numbers of solution clusters ($S$) and objective clusters ($O$). Each solution cluster is identified by a specific color, and the objective clusters are distinguished by larger distances between the coordinate lines of the different clusters.

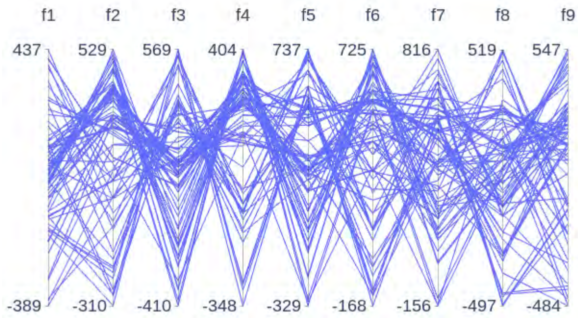
The visualizations nicely demonstrate that biclustering is a valuable tool to reveal tendencies among the solutions and correlations between the objective functions. For appropriately chosen values of $S$ and $O$, the three objective clusters that are present in the data are retrieved. However, we note that the outcome largely depends on the parameter settings, in particular on the choices of $S$ and $O$, but also on the employed normalization method.

In the future, we would like to use a measure to evaluate the quality of the visualization. Moreover, it would be interesting to automatically test different sizes of row and column clusters and present the "best" (with respect to a quality measure) result or results to the decision maker. Another open question is the way the columns are ordered. So far, we simply use the output of the biclustering algorithm. In the figures presented above we manually inserted gaps between the different clusters. By varying the sizes of these gaps, they could also serve as a source of information for the decision maker, e.g. by linking larger distances to a lower correlations between clusters. This and other technical issues like an automation are left for further improvements in the future.

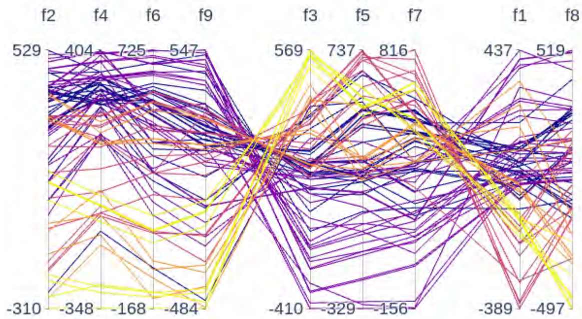
### 4.5.5 Conclusions

We have proposed a novel way of visualizing sets of Pareto optimal objective vectors by applying bi- or co-clustering and modifying parallel coordinate plots. Thanks to these visualizations, the decision maker can gain insight in the correlations among both objective functions and objective vectors simultaneously.

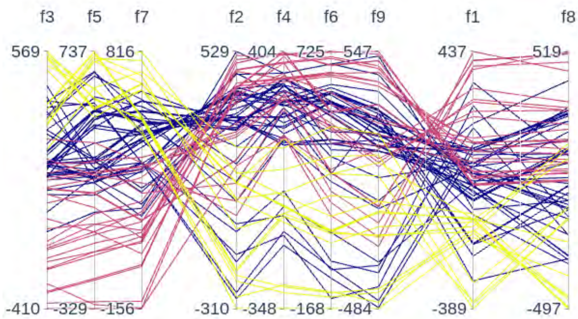
The novel visualizations can be applied to analyze any set of objective vectors. They can also be applied as a part of an interactive solution process. Our future research direction is to apply the findings and develop visualization assistance for solving multiobjective optimization problems with more than three objective functions.



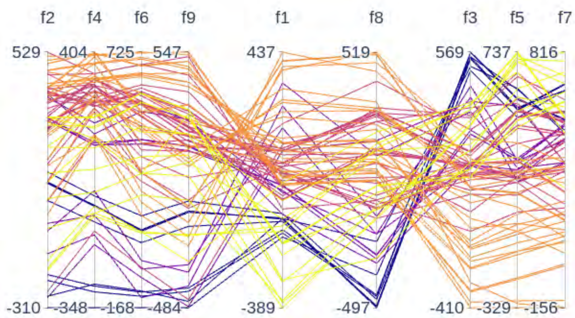

**(a)** Original data set




**(b)** BC with $S = 5$ and $O = 3$




**(c)** BC with $S = 3$ and $O = 4$




**(d)** BC with $S = 5$ and $O = 4$

**Figure 18** Spectral biclustering applied to a data set containing 88 nondominated points (a). Subfigures (b)-(d) show biclustering results for $S$ solution clusters and $O$ objective clusters.

## Acknowledgements

We would like to warmly thank Bhupinder Singh Saini for his valuable implementation support. This research is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, jyu.fi/demo) of the University of Jyvaskyla.

## References

1 D. Brockhoff and E. Zitzler. Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization. In T. P. Runarsson, H. G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature, PPSN IX, Proceedings*, pages 533–542, Berlin, Heidelberg, 2006. Springer.

2 D. Brockhoff and E. Zitzler. Dimensionality reduction in multiobjective optimization: The minimum objective subset problem. In K. H. Waldmann and U. M. Stocker, editors, *Operations Research Proceedings 2006*, pages 423–429, Berlin, Heidelberg, 2007. Springer.

3 S. Busygin, O. Prokopyev, and P. M. Pardalos. Biclustering in data mining. *Computers & Operations Research*, 35(9):2964–2987, 2008.

4 I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274, 2001.

5 C. M. Fonseca, C. A. Antunes, R. Lacour, K. Miettinen, P. M. Reed, and T. Tusar. Visualization in multiobjective optimization. In S. Greco, K. Klamroth, J. D. Knowles, and G. Rudolph, editors, *Understanding Complexity in Multiobjective Optimization, Report from Dagstuhl Seminar 15031*, pages 129–139, Dagstuhl, 2015.

6 J. Hettenhausen, A. Lewis, and S. Mostaghim. Interactive multi-objective particle swarm optimization with heatmap-visualization-based user interface. *Engineering Optimization*, 42(2):119–139, 2010.

7 A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb. 3d-radvis: Visualization of Pareto front in many-objective optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 736–745. IEEE, 2016.

8 Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13:703–716, 2003.

9 P. Korhonen and J. Wallenius. Visualization in the multiple objective decision-making framework. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 195–212. Springer, Berlin, 2008.

10 H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, 2009.

11 A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev. *Interactive Decision Maps: Approximation and Visualization of Pareto Frontier*. Kluwer Academic Publishers, Boston, 2004.

12 A. V. Lotov and K. Miettinen. Visualizing the Pareto frontier. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 213–243. Springer, Berlin, 2008.

13 K. Miettinen. Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum*, 36(1):3–37, 2014.

14 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

**15** R. E. Steuer. "ADBASE: A multiple objective linear programming solver for all efficient extreme points and unbounded efficient edges," Version 17.1, Terry College of Business, University of Georgia, Athens, Georgia, 2017.
**16** H. L. Trinkaus and T. Hanne. knowCube: A visual and interactive support for multicriteria decision making. *Computers and Operations Research*, 32(5):1289–1309, 2005.
**17** T. Tusar and B. Filipic. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 19(2):225–245, 2015.
**18** M. J. Woodruff, P. M. Reed, and T. W. Simpson. Many objective visual analytics: Rethinking the design of complex engineered systems. *Structural and Multidisciplinary Optimization*, 48(1):201–219, 2013.
**19** J. Xie, A. Ma, A. Fennell, Q. Ma, and J. Zhao. It is time to apply biclustering: A comprehensive review of biclustering applications in biological and biomedical data. *Briefings in Bioinformatics*, 20(4):1450–1465, 2018.

## 4.6 Supporting Problem Solving with Many Decision Makers in Multi-objective Optimization

*Michael Emmerich (Leiden University, NL), Jussi Hakanen (University of Jyväskylä, FI), Patrick Reed (Cornell University, US), Pradyumn Kumar Shukla (KIT – Karlsruher Institut fürTechnologie, DE), Jürgen Branke (University of Warwick, GB), Günter Rudolph (TU Dortmund, DE), Sanaz Mostaghim (Universität Magdeburg, DE), Heike Trautmann (Universität Münster, DE)*



### 4.6.1 Introduction

The problem of multiobjective optimization changes qualitatively as soon as many decision makers are involved. All problems that occur when a single decision maker is involved are inherited, but new problem aspects are added. Three main additional aspects are:

1. The different decision makers can differ on the constraints and objective functions that are relevant for the problem, and the way they are computed.
2. They may have different preferences for the different objective functions and this way the problem of fairness arises, that is the problem of considering different preferences in a balanced manner.
3. There is a possibility of negotiations and group dynamics that should be considered in designing decision making processes. Moreover, decision makers might form coalitions and there might be different types of (power) relations between decision makers and hidden objectives/agendas.

In this report we consider the somewhat ideal situation of a group of equal decision makers that are able and willing to express their preferences. In such cases computer systems can be used to find out solutions that are non-dominated with respect to all objectives considered by the decision makers and among them present solutions that achieve a high performance in fairness on the one side, and total gain in terms as being close in average to the decision makers' preferred solutions or reference points on the other side.