Multidimensional Scaling for Evolutionary Algorithms - Visualization of the Path through Search Space and Solution Space using SAMMON Mapping

Hartmut Pohlheim

DaimlerChrysler AG, Research and Technology
Alt-Moabit 96a, 10559 Berlin, Germany

Hartmut.Pohlheim@DaimlerChrysler.com

phone: +49 30 39982 456
fax: +49 30 39982 107

Picture This: How to see more than three dimensions

category: visualization gem

Old title of the gem:

Evolutionary Multidimensional Scaling -
Visualization of the Path through Search Space and Solution Space using SAMMON Mapping

Abstract

Multidimensional scaling as a technique for the presentation of high-dimensional data with standard visualization techniques is presented. The technique used is often known as SAMMON mapping. In this paper we explain the mathematical foundations of multidimensional scaling and its robust calculation.

We also demonstrate the use of this technique in the area of Evolutionary Algorithms. First, we present the visualization of the "path through the search space" of the best individuals during an optimization run. We then apply multidimensional scaling to the comparison of multiple runs regarding the variables of individuals and multi-criteria objective values ("path through the solution space").

Keywords: high-dimensional visualization, scaling, SAMMON mapping

1 Introduction

It is difficult to visualize the behavior of optimization algorithms for problems of more than three dimensions. Here we present a technique for visualizing multidimensional data – multidimensional scaling. This advanced technique is independent of the representation of the variables. The technique is applied to data produced by evolutionary algorithms - the visualization of variables of individuals from a population, and demonstrated for the visualization of multi-criteria objective values in multi-objective optimization. This technique is especially useful for the comparison of high-dimensional data from different runs.

2 Multidimensional Visualization

Most of the commonly used techniques for visualization are limited to representing data depending on one or two variables. This is due to the limitation of human vision to three dimensions. A number of techniques exist to extend visualization beyond this limitation, such as the use of color for the fourth dimension and time as the fifth dimension. However these are not commonly used and require practice by users. For problems incorporating higher numbers of dimensions, a method for visualizing arbitrarily high dimensions must be used.

In order to visualize multidimensional data, a method for transforming multidimensional data so that it has a lower number of dimensions is needed, preferably two or three. This transformation should provide a lower-dimensional picture where the dissimilarities between the data points of the multidimensional domain correspond with the dissimilarities of the lower-dimensional domain. These transformation methods are referred to as 'multidimensional scaling' ([3], [12]).

To measure the dissimilarity, the distance between pairs of data points is used. These distances can be genuine distances in the respective high-dimensional domain, for instance the Euclidean distance. If a genuine distance measure is not applicable, the dissimilarities can be defined by a substitute measure. The distances do not need to satisfy the triangle inequality ($d_{ik} \le d_{ij} + d_{jk}$, d: distance between data points); thus the term dissimilarity is used. In [12] examples of non-metric measures of categorical data are given (for instance, simple matching coefficient).

2.1 Calculation of low-dimensional representation

In order to find a low-dimensional representation, y_i, of a number of points in a high-dimensional domain, x_i, the distance between all the points in their respective domains is calculated.

$x_1,\ldots,x_n$: data points in $\Re^p$ δ_{ij} : distance between x_i und x_j
$y_1,\ldots,y_n$: data points in $\Re^q$, $p \ge q$ d_{ij} : distance between y_i und y_j

A configuration of low-dimensional image points, y_i, is then looked for, in which the distances d_{ij} between image points are as close as possible to the corresponding original distances δ_{ij}. Since it is usually not possible to find a configuration for which $d_{ij} = \delta_{ij}$ for all i and j, a quality criterion, J, for ranking different configurations, must be defined, see equations 1, 2 and 3.

In [4] three variants of the functions for evaluating the error during multidimensional scaling (sum of the square errors) are suggested (left column: quality criterion; right column: 1st derivation of the corresponding function):

quality criterion **derivation / gradient of quality criterion**

$$J_{ee} = \frac{1}{\sum\limits_{i<j} \delta_{ij}^2} \sum_{i<j} \left(d_{ij} - \delta_{ij}\right)^2 \qquad \nabla_{y_k} J_{ee} = \frac{2}{\sum\limits_{i<j} \delta_{ij}^2} \sum_{j \ne k} \left(\left(d_{kj} - \delta_{kj}\right) \frac{y_k - y_j}{d_{kj}} \right) \tag{1}$$

$$J_{ff} = \sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2 \qquad \nabla_{y_k} J_{ff} = 2 \sum_{j \ne k} \left(\frac{d_{kj} - \delta_{kj}}{\delta_{kj}^2} \cdot \frac{y_k - y_j}{d_{kj}} \right) \tag{2}$$

$$J_{ef} = \frac{1}{\sum\limits_{i<j} \delta_{ij}} \sum_{i<j} \frac{\left(d_{ij} - \delta_{ij}\right)^2}{\delta_{ij}} \qquad \nabla_{y_k} J_{ef} = \frac{2}{\sum\limits_{i<j} \delta_{ij}} \sum_{j \ne k} \left(\frac{d_{kj} - \delta_{kj}}{\delta_{kj}} \cdot \frac{y_k - y_j}{d_{kj}} \right) \tag{3}$$

Criterion J_{ee} in equation 1 emphasizes the larger errors, irrespective of whether the distances δ_{ij} are large or small. Criterion J_{ff} in equation 2 highlights the larger relative errors, irrespective of whether the value of the error is high or low. Criterion J_{ef} in equation 3 provides a sensible compromise between the first two criteria by emphasizing the largest product made up of errors and relative errors.

An optimal configuration minimizes the criterion function. Since the derivations of the criteria can be calculated easily, see right-hand column of equations 1, 2 and 3, one of the many known gradient procedures for determining an optimal configuration can be used.

2.2 SAMMON mapping

One of the best known methods for multidimensional scaling is SAMMON mapping [13]. SAMMON used the criterion in equation 3 and a steepest descent method (diagonal Newton method) for optimization. However, this search method is not very robust and diverges without any special interaction [12].

We also used the criterion in equation 3. It provides a good compromise by emphasizing large absolute and large relative errors at the same time. Thus, we refer to the multidimensional scaling method used as **SAMMON** mapping. However, we employed different optimization methods.

2.3 Gradient search method

We used two different methods for optimization. First, a standard optimization method included in MATLAB was employed (BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure - [7]). Later, a more robust search method was employed, the RPROP algorithm [11]. The RPROP algorithm uses only the changes in the sign of the gradient for step size control. RPROP is widely used in the field of neural networks.

However, because of the local search the optimization often gets stuck in local optima. Thus, when using either of these methods, multiple runs must be performed using different initializations.

For each run of the gradient search method a starting configuration must be provided. Different methods for the calculation of these starting configurations exist:

- The starting configuration for the gradient search can be chosen randomly.
- Any low-dimensional configuration of the high-dimensional data points is used directly (for instance x_1 and x_2 of all data points, then x_3 and x_4 and so on. The selection of the dimensions can be carried out randomly or according to any more or less systematic scheme).
- The main eigenvectors of the high-dimensional configuration (as many as there are dimensions in the low-dimensional configuration) are used. They can be calculated using a principal component analysis (available in MATLAB through the function svd – singular value decomposition).

The first option produced the slowest optimizations. The last option was the best when looking for quick optimizations. However, when using a principal component analysis for the initialization the optimization got stuck in local optima very often. This did not prove to be a robust initialization method. Thus, we used the second option for the initialization of the starting configuration. In the end it proved not only simple, but was also the most reliable initialization method.

When running multiple optimizations for the low-dimensional data set we presented all the received mappings to the user on the screen. These results must be evaluated by the user by comparing the quality criterion J_{ef} (see equation 2) and the corresponding graphics. A low value for the quality criterion and similar graphics pointed to the "best" mapping for the current data set.

Both optimization algorithms (BFGS Quasi-Newton and RPROP) produced good results. The RPROP algorithm ran more slowly but produced slightly more consistent results. Thus, neither method had a clear advantage over the other. Because of its better availability in MATLAB we employed the BFGS Quasi-Newton method for all further experiments.

A full implementation of the methods presented and the accompanying utilities in a MATLAB toolbox is available at [10].

3 Visualization examples of multidimensional scaling

How can multidimensional scaling be used for visualizing evolutionary algorithms? Its application is of interest when the interrelationship between a large number of high-dimensional data points is to be illustrated. The following examples provide a variety of such cases:

- Representation of several individuals of a population:
  - individuals from one, several or all generations,
  - the best individuals from several or all generations ("path of the best individual from each generation through the search space"),
- Representation of multi-criteria objective function values (objective values are often called fitness values, however, in evolutionary algorithms, fitness values have a different meaning):
  - visualizing the multiple dimensions of the objective function ("path of the best individual from each generation through the solution space").
- Comparison of several optimization runs:
  - variables of the best individuals ("path through the search space"),
  - objective function values of the best individuals ("path through the solution space").

When the individuals of the population are visualized one can picture how the individuals are distributed within the search space, i.e. how the algorithm used surveys the search space. It also makes it possible to see whether the individuals are all in one area or if they are far apart from each other.

The representation of multi-criteria objective function values allows us to evaluate them visually. For up to three dimensions this is directly possible with the methods already presented in various publications. Multidimensional scaling lends itself to cases in which there are four criteria and more.

Several papers on the use of multidimensional scaling for the visualization of evolutionary algorithms have been published recently ([2], [5], [14] and [15]).

3.1 Variables of the Best Individual from All Generations

For this demonstration we use ROSENBROCK's function (often called the 'banana function'). In figure 1 a 3-D mesh plot of two dimensions and the corresponding objective values is shown. The curve-shaped (banana-shaped) valley can clearly be seen. To visualize the steepness of the valley borders we bounded all objective values to a maximum of 100 (for visualization only). This graph gives a good impression of the function's properties; but only for two dimensions.

When we look at the results of an optimization run (over 3000 generations), we can visualize the "path through the search space" of the best individual of every generation. Figure 2, left, shows a 3-D line plot of the best individual from all generations optimizing ROSENBROCK's function in 10 dimensions. It is not clear what the "path through the search space" really looks like.

Using multidimensional scaling we can transform this 10-dimensional data set into a 2-dimensional data set. These low-dimensional data were visualized employing a 2-D point plot, figure 2, right. The label at each point is the generation of the respective individual. We now get a much clearer picture. This plot clearly shows the curve-shaped characteristic of this function known from the 3-D mesh plots of two variables in figure 1. We are probably now able to see the curve-shaped form of this function in the 10-dimensional data set in figure 2, left, as well. But would we have recognized it without having seen the 2-D representation from the multidimensional scaling?

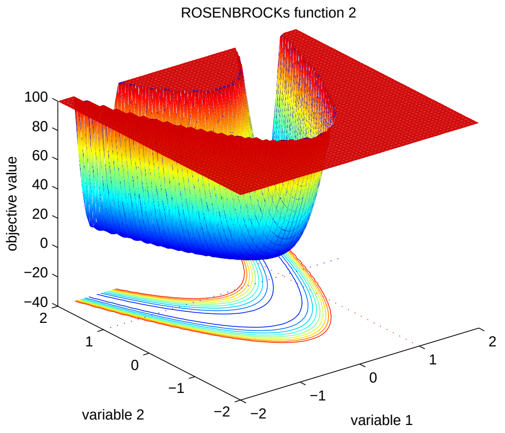

Figure 1: Mesh plot of ROSENBROCK's function (objective values bounded to maximum of 100), contour plot added, minimum of function marked by dotted lines

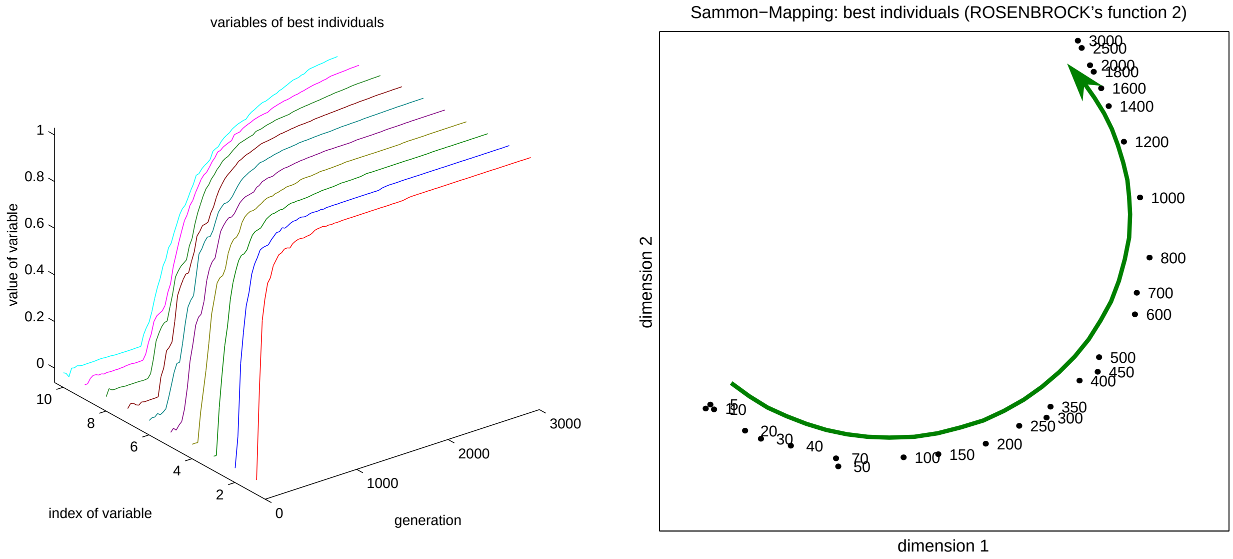

Figure 2: ROSENBROCK's function: variables of the best individuals of one optimization run; left: 3-D line plot of all 10 dimensions, right: 2-D point plot of low-dimensional data set calculated using multidimensional scaling (dimension 1 and 2 are the weighted linear combinations identified by the multidimensional scaling)

3.2 Multiple Objective Values of the Best Individual from Every Generation

The second example presents the multidimensional scaling of multi-criteria objective values. Here we employ a real-world example, the Chopper system. The objective function consisted of the simulation of up to nine different scenarios, where each scenario produced one objective value. The sum of all the separate objective values was used as the objective value for the evolutionary algorithm (for a full description of this system see [9]). It proved difficult to compare the course of two different runs. However, this problem was solved using multidimensional scaling.

In order to calculate the multidimensional scaling of two optimizations, the respective data from both Chopper optimization runs were combined to form one data set. This combined data set was then used for the multidimensional scaling procedure. It is important to input all the data for a comparison at the beginning of multidimensional scaling otherwise a comparison between different runs is only seldom possible.

Figure 3 shows the comparison of two optimization runs of the Chopper system. On the left hand side, one can see the best individuals from each generation (visualizing the “path through the search space”), on the right, the visualization of the multi-criteria objective values (visualizing the “path through the solution space”). The label for each point consists of the run number followed by the generation number.

The diagrams show that both runs start from the same area of the objective space or variable search space. However, during the run the objective values and the individuals both diverge to different areas, each along a line of their own. Without multidimensional scaling it would be very difficult to extract this information from the results.

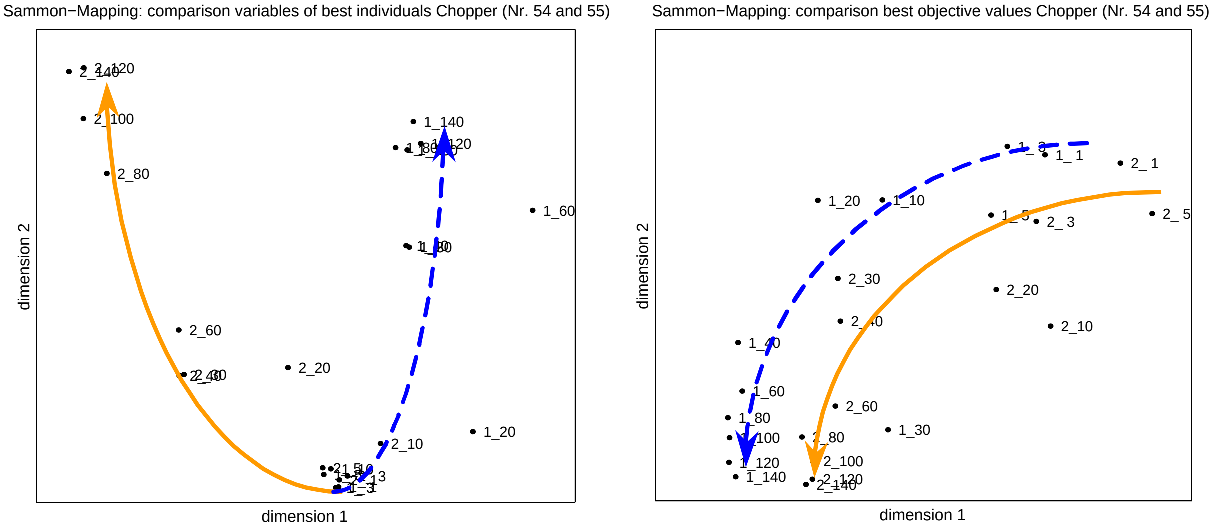

Figure 3: Comparison of two optimization runs of the Chopper application using multidimensional scaling; left: multi-criteria variables of individuals (“path through the search space”), right: multi-criteria objective values (“path through the solution space”)

By comparing different runs in this way, detailed information can be acquired about the “path of the individuals through the search space” or the relation of multi-criteria objective values (“path through the solution space”) to each other. Thus, multidimensional scaling establishes new possibilities for visualizing data, which the methods presented so far could not offer.

4 Concluding remarks

This paper presented multidimensional scaling techniques for the visualization of data produced by evolutionary algorithms. The examples presented showed the advantage of this technique. Information which is not available or very difficult to extract using other techniques can be derived using multidimensional scaling. This technique opens up a field for new visualization techniques not only applicable to the domain of evolutionary algorithms.

SAMMON mapping has the main advantage, that not only a straight 2-D projection or a principal component analysis is carried out. Instead, a low-dimensional configuration as similar as possible to the high-dimensional configuration is searched for. The quality criterion defines the error or dissimilarities between the configurations. Using gradient-based optimization methods configurations with a very small error can be searched for.

The technique has been used and proved informative (see [9]). An example implementation of all these and additional techniques can be found in [8] and [10]. Further examples of the use of multidimensional scaling for the visualization of the search space are demonstrated in [6].

References

[1] *Bird, J., Bullock, S., and Smith, T.*: Beyond Fitness: Visualising Evolution. Workshop at Artificial Life VIII: The 8th International Conference on the Simulation and Synthesis of Living Systems, University of New South Wales, Sydney, NSW, Australia, 2002. http://www.cogs.susx.ac.uk/users/toms/Visualization/Alife8Workshop.html

[2] *Collins, T. D.*: Genotypic-Space Mapping: Population Visualization for Genetic Algorithms. Technical Report, KMI-TR-39, Knowledge Media Institute, The Open University, Milton Keynes, UK, 1997.

[3] *Cox, T. F. and Cox, M. A. A.*: Multidimensional Scaling. London: Chapman & Hall, 1994.

[4] *Duda, R. O. and Hart, P. E.*: Pattern Classification and Scene Analysis. New York: John Wiley & Sons, 1973.

[5] *Dybowski, R., Collins, T. D. and Weller, P. D.*: Visualization of Binary String Convergence by Sammon Mapping. In *Fogel, D. B.*: Evolutionary Programming V, Proceedings of the Fifth Annual Conference on Evolutionary Programming. Cambridge, Massachusetts: MIT Press, 1996.

[6] *Gens, R. and Pohlheim. H.*: Mehrkriterielle Entscheidungsfindung - Optimiertes PARETO-Ranking für MATLAB. Technischer Bericht, 2001. (Multi-Criteria Ranking - Optimized PARETO-Ranking for MATLAB. in German, contains an extensive graphical description of often used multi-objective test function), http://pohlheim.com/publications.html.

[7] *Mathworks, The*: Matlab – User Guide. Natick, Mass.: The Mathworks, Inc., 1994-1999. http://www.mathworks.com/

[8] *Pohlheim, H.*: GEATbx - Genetic and Evolutionary Algorithm Toolbox for Matlab. http://www.geatbx.com/, 1994-2003.

[9] *Pohlheim, H.*: Evolutionäre Algorithmen - Verfahren, Operatoren, Hinweise aus der Praxis. Berlin, Heidelberg: Springer-Verlag, 1999. http://www.pohlheim.com/eavoh/

[10] *Pohlheim, H.*: Multi-dimensional Scaling by Sammon-Mapping – MATLAB toolbox and examples. http://www.pohlheim.com/matlab_main.html, 1998-2004.

[11] *Riedmiller, M. and Braun, H.*: A direct adaptive method for faster backpropagation learning: The RPROP algorithm. in H. Ruspini (editor), Proceedings of the IEEE International Conference on Neural Networks (ICNN), pp.586-591, 1993.

[12] *Ripley, B. D.*: Pattern Recognition and Neural Networks. Cambridge, GB: Cambridge University Press, 1996.

[13] *Sammon, J. W. jr.*: A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers, vol. C-18, no. 5, pp. 401-409, 1969.

[14] *Shine, W. and Eick, C.*: Visualizing the evolution of genetic algorithm search processes. In The Proceedings of the 1997 IEEE International Conference on Evolutionary Computation ICEC'97, pp. 367-372, 1997.

[15] *Spears; W.*: An overview of multidimensional visualization techniques. Workshop on Evolutionary Computation Visualization at GECCO'99 - Genetic and Evolutionary Computation Conference, San Francisco, CA: Morgan Kaufmann, pp. 104-105, 1999.