Using Software Visualization Technology to Help Evolutionary Algorithm Users Validate their Solutions

Trevor D. Collins*
Knowledge Media Institute
The Open University
Walton Hall
Milton Keynes MK7 6AA, UK

Abstract

Evolutionary Computation (EC) offers a variety of robust search techniques, each of which operate by means of an evolutionary-based search algorithm. In order to assess the quality of any solutions found, the user must understand the search path taken to discover them. Software Visualization is recommended as a useful technique for aiding such understanding. This paper gives an introduction to Software Visualization, reviews five existing techniques for presenting EC output data and presents a new technique suitable for visualizing EC search space. To conclude a summary table of all six visualization techniques, including comments and criticisms of each is given and the selection of appropriate visualization techniques is discussed. By presenting EC output data in a format that is easy to understand software visualization makes EC validation much simpler for the user.

1 Introduction

This paper explains how *Software Visualization* (SV) may be applied to support peoples' understanding of their EC systems. It is only through understanding an algorithm's evolutionary-based search that the user can access the quality of the solution/s found.

Evolutionary algorithms output data in a common form, where each individual solution is described as a chromosome and an associated fitness rating. Chromosome data is of a type known as " profile data", i.e. the data is made up of a series of cases $(1, 2, \ldots, n)$ whose profiles vary over a range of variables $(X_1, X_2, \ldots, X_p)$, see figure 1. The associated univariate fitness data set contains the same number of cases $f_1, f_2, \ldots, f_n$. The term "population data" will be used here to refer to both of these data sets.

$$\begin{array}{cc} \textit{Variables}\ X_1\ X_2 \ldots\ X_p & \textit{Cases} \\ X = \begin{bmatrix} X_{11}\ X_{12} \ldots X_{1p} \\ X_{21}\ X_{22} \ldots X_{2p} \\ \vdots \\ X_{n1}\ X_{n2} \ldots X_{np} \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ \vdots \\ n \end{array} \end{array}$$

Figure 1: A data matrix used to describe EC chromosome data. Where X refers to a population containing n chromosomes, each of length p.

Software visualization is "the use of the crafts of typography, graphic design, animation and cinematography with modern human-computer interaction technology to facilitate both the human understanding and effective use of computer software" (Price, Baecker & Small, 1993), and has been applied to many different aspects of software development including design, debugging and education.

Brad Myers (1990) divided SV into three categories: code, data and algorithm visualization. *Code visualization* helps the user to understand the construction and execution of their code, an example being the Transparent Prolog Machine that uses and/or trees to visualize Prolog programs (Eisenstadt & Brayshaw, 1987). *Data visualization* presents data structures in an easy to understand format. For example the Incense system allows the programmer to design and use pictorial representations for the display of their data structures (Myers, 1983). Finally, *algorithm visualization* provides an abstract view of the user's pro-

*email - t.d.collins@open.ac.uk

gram as a series of algorithm steps. The Brown University Algorithm Simulator and Animator (Balsa) is an early example of algorithm visualization in which the user can construct custom animations for illustrating or learning about algorithms (Brown & Sedgewick, 1985).

Effective software development relies on the programmer having a clear understanding of their software at all three levels; code, data and algorithm. However, for the end user who has merely applied a system and wishes to ensure the quality of the solution found, it is the search path taken to discover that solution that is of primary interest. By visualizing the population data for each generation, the user can see the explored sample points in the problem search space and look for any unconsidered regions.

The remainder of this paper is broken down into five sections. Section two presents two traditional techniques for describing EC search: population data matrices and fitness versus generation number graphs. Section three describes three recently recommended techniques that are not, as yet, widely used i.e. allele versus loci frequency matrices, multidimensional scaling, and Sammon mapping. The fourth section presents a new technique described here as "Search Space Matrices" (SSM). The fifth section draws these techniques together in a summary table and identifies the associated pros and cons of each. Finally the paper concludes with a discussion of these techniques and how they may be applied.

2 Traditional EC Visualization

Even within the relatively short lifetime of evolutionary computing (according to (Goldberg, Deb & Thierens, 1992) 30 to 40 years) there has developed a traditional format for presenting information about EC search. The data is either displayed in a data matrix of chromosome and fitness values, or on a line graph of fitness variation over a number of generations. This section describes these traditional techniques and examines why additional visualizations are necessary.

2.1 Population Data Matrices

Population data matrices present the entire data set in its *raw* text form. A data matrix illustrating a population data set in order of decreasing fitness is the most simple and most complete way of showing the data. However, (Bertin, 1981 p. 32) recommends that no more than 10,000 elements ($x \times y$) be shown in any data matrix. This constrains the maximum population size to $(10,000 \times \textit{chromosome string length})$.

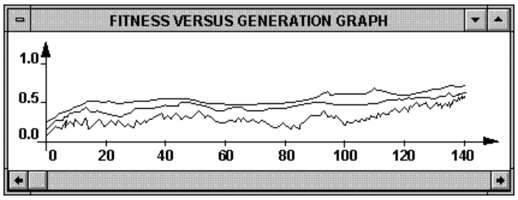

Figure 2: A fitness (y) vs generation number (x) graph showing how the best (top) average (middle) and worst (bottom) fitness ratings vary during evolution.

Even for smaller populations where this is applicable comparing values across generations and identifying the contribution of emerging schemata to the fitness ratings can be difficult:

> *"A data matrix usually contains too much information to absorb at once. The differences between the various rows and columns as well as the interactions between them, are difficult to determine merely by looking at the matrix."* (Du Toit, Steyn & Stumpf, 1986, p. 105)

It is for this reason that multidimensional scaling (discussed in section 3.2) is often recommended to illustrate the distribution of the cases within a data set.

2.2 Fitness vs Generation Number Graph

Fitness versus generation number graphs are used to illustrate the variation in fitness during evolution. Harvey and Thompson (1996) use a three dimensional graph to illustrate the fitness trace of a fitness ordered population over a number of generations (y = fitness, x = fitness ordered position, z = generation number).

A more common form of fitness versus generation graph is a two dimensional graph in which some aspect of the population's fitness data is shown against the generation number. Examples include the "online" and "offline" fitness ratings (i.e. the mean fitness rating and mean current best fitness rating across all generations, see De Jong, 1980) and the best, average and/or worst fitness rating per generation.

By plotting a two dimensional line graph of the best, average and/or worst fitness rating per generation, the user can gain immediate insight into how the fitness ratings within each population vary during evolution, see figure 2. Although graphs such as these present a compelling view of fitness data they present no information about the search path itself. For example, the number of solutions being considered, or their di-

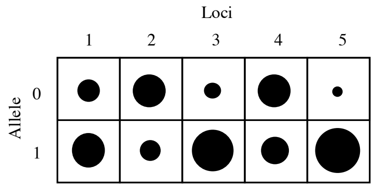

Figure 3: An allele versus loci matrix of a five bit binary data set the frequency of each allele is indicated by the relative size of the circle.

versity cannot be extracted from this view. Therefore, alternative or additional views must be considered if the user is to gain a complete understanding of a solution's evolution.

3 Recent EC Visualization Techniques

As a reaction to the shortcomings of the above *traditional* techniques, alternative forms of EC data visualization have started to appear. The following section describes three such techniques and examines the additional information available from each.

3.1 Allele versus Loci Frequency Matrices

Rather than attempting to illustrate the complete population data set, an image matrix showing the frequency of allele (i.e. the chromosome values) at the various loci (i.e. the chromosome positions) can be used to illustrate the distribution of allele within a population (see figure 3) (Collins, 1993) (Routen & Collins, 1993). By viewing several allele versus loci frequency matrices of subsequent generations the user can see how the allele-loci distributions vary during evolution. Although the matrix does illustrate the distribution of alleles it presents no information about the current fitness ratings and no information about the chromosomes' allele configurations.

3.2 Multidimensional scaling

As mentioned in section 2.1 above, multidimensional scaling is one technique for representing high dimensional data in two or three dimensions suitable for human observation. An example of where this has been applied to represent EC data can be found in (Harvey & Thompson, 1996).

The two techniques for scaling multidimensional profile data recommended by Du Toit, et al. (1986) are examined here, i.e. *biplots* (section 3.2.1) and *principal component analysis* (PCA, section 3.2.2), and comments on their application given (section 3.2.3). Both of these use the eigenvalues and eigenvectors of the data set to transform a set of p correlated variables to a new set of p uncorrelated variables, from which a scatterplot can be drawn to illustrate the distribution of the data. These uncorrelated variables are referred to as *factors* in biplots and *principal components* in PCA. The formation of clusters around areas containing fit solutions would be expected during evolution.

3.2.1 Biplots

For biplots an assumption known as the rank two approximation is made i.e. the first two factors illustrate enough about the distribution that the remaining factors can be ignored (Du Toit, et al., 1986). Hence a two dimensional scatterplot of the first two factors is sufficient to describe the distribution. This assumption generally holds if the first two eigenvalues $(l_1 + l_2)$ are large in relation to the remaining ones $(l_3 + l_4 + \ldots + l_p)$. The corresponding "goodness of fit" is described as:

$$\frac{l_1 + l_2}{l_3 + l_4 + \ldots + l_p}$$

The biplot factors for a profile data set X as described in the introduction, are generated in two steps; first the eigenvectors and eigenvalues are used to create $X_{(2)}$, and then the set of factors Y is produced using the elements in $X_{(2)}$ and the mean values for each variable in X, i.e:

$$X_{(2)} = \begin{bmatrix} U_{11}l_1^k & U_{12}l_2^k \\ U_{21}l_1^k & U_{22}l_2^k \\ \vdots & \\ U_{n1}l_1^k & U_{n2}l_2^k \end{bmatrix} \begin{bmatrix} V_{11}l_1^{1-k} & V_{21}l_1^{1-k} & \ldots V_{p1}l_1^{1-k} \\ V_{12}l_2^{1-k} & V_{22}l_2^{1-k} & \ldots V_{p2}l_2^{1-k} \end{bmatrix}$$

$$Y = \begin{bmatrix} X_{11} - \overline{X}_1 & X_{12} - \overline{X}_2 \ldots X_{1p} - \overline{X}_p \\ X_{21} - \overline{X}_1 & X_{22} - \overline{X}_2 \ldots X_{2p} - \overline{X}_p \\ & \vdots \\ X_{n1} - \overline{X}_1 & X_{n2} - \overline{X}_2 \ldots X_{np} - \overline{X}_p \end{bmatrix}$$

Where; $U_1, U_2, \ldots, U_n$ are the left eigenvectors of X, $V_1, V_2, \ldots, V_n$ are the right eigenvectors of X, $\overline{X}_n$ is the mean value of variable X_n, and $0 \leq k \leq 1$, whereby $k = 1/2$ if a biplot is made for rows and columns jointly; 0 if for columns only; and 1 if for rows only.

The resulting biplot is a scatterplot of the first two factors, i.e. the first two columns of Y. Note, this view is based on the rank two approximation and the validity of that assumption should be checked for each population before its biplot is used.

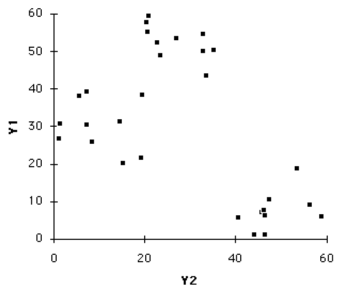

Figure 4: An example of a PCA Scatterplot showing the chromosome distribution in a single population.

3.2.2 Principal Component Analysis

In principal component analysis the p measurements $X_1, X_2, \ldots, X_p$ for each case go through a linear transformation to form the principal components $Y_1, Y_2, \ldots, Y_p$. In these cases the first principal component explains the largest percentage of variation in the original multivariate data set, the second explains the second largest variation and so on:

$$
\begin{aligned}
&Y_1 = a_{11}X_1 + a_{21}X_2 + \ldots + a_{p1}X_p \\
&Y_2 = a_{12}X_1 + a_{22}X_2 + \ldots + a_{p2}X_p \\
&\vdots \\
&Y_p
\end{aligned}
$$

Where a_{ij} refers to the coefficients of the data's covariance matrix and indicate the relative importance of each of the original p variables. The expressive power of a principal component scatterplot is defined by the percentage of variation for each of the two principal components being shown. The percentage of variation is calculated using the eigenvalues $l_1, l_2, \ldots l_p$, for principal component i this is given as:

$$\frac{l_i}{l_1 + l_2 + \ldots + l_p} \times 100\%$$

Two-dimensional scaling of either variables or cases can then be carried out, in order to illustrate the distribution of variables and cases respectively (see (Du Toit, et al., 1986) or (Chatfield & Collins, 1980) for further information). In the case of EC population data, scaling of cases (i.e. chromosomes) would indicate separable chromosome distributions (see figure 4, no advantage can be foreseen from the scaling of variables). It is the formation of these separable chromosome distributions over successive generations that can illustrate the emergence of multiple solutions.

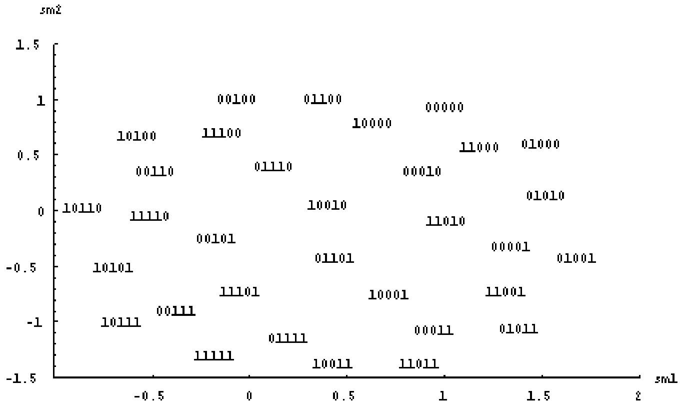

Figure 5: An example of the Sammon mapped search space for a five bit binary string representation, taken from (Dybowski, Collins & Weller, 1996).

3.2.3 Comments on Scaling Methods

As the above multidimensional scaling techniques illustrate the distribution of each population in turn, there is no consistent spatial relationship between the plotted points of subsequent generations.

This problem can be appeased by a technique known as "brushing". This enables the user to select ("brush") one or more cases (i.e. points) on a distribution scatterplot and the same case's will then be highlighted in any associated plots using the same brush colour. Hence the user can mark cases of interest (i.e. suspected solutions) and see how they are distributed in subsequent generations.

Another useful extension for these techniques is to use the point size in the scatterplots to illustrate the chromosomes' fitness ratings. A large dot indicating a highly fit individual and a small dot indicating a relatively unfit individual.

3.3 Sammon Mapping

An alternative to multidimensional scaling is to construct a 2 or 3 dimensional representation of the high-dimensional search space and highlight the sample points (i.e. chromosomes) found in each generation.

Sammon mapping (SM) is a non-linear technique for mapping high dimensional data to a lower number of dimensions, typically 2 or 3 (Sammon, 1969). This method starts with a random configuration of points in the low dimensional space intended to represent the given points in high dimensional space. The interpoint distances between each of these points in high dimensional and low dimensional space are compared and

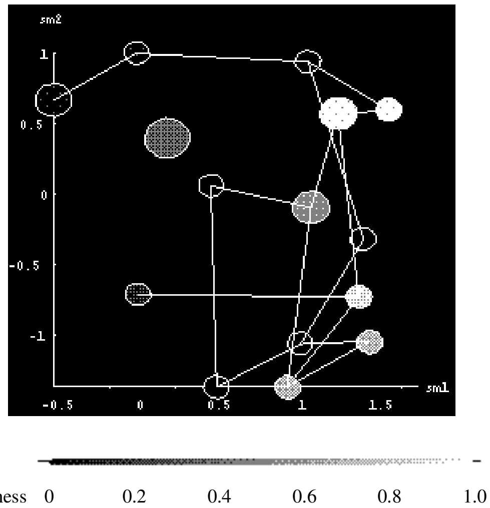

Figure 6: A scatterplot illustrating the sample points (chromosomes) in a Sammon mapped 2D search space. Grey scale is used to show fitness, circle size to show frequency and Hamming one neighbours are joined with a white line, taken from (Dybowski, et al., 1996).

the resulting error used to correct the position of the points in the low dimensional space. This error evaluation and correction process is then repeated until the error converges to an acceptable minimum (figure 5).

This method has been applied to GA search space visualization by (Dybowski, Collins & Weller, 1996). The computational complexity of constructing an accurate Sammon map is quadratic with respect to the number of points. This makes the construction of real-world EC search spaces a large (virtually impractical) computational overhead. However, once constructed these mappings can be stored as look-up tables, making the retrieval of chromosome coordinates a trivial task. Scatterplots of successive generations can be used to show the EC's search in a structured 2 or 3D representation of the problem space.

Like the scatterplots described in the previous section, SM scatterplots can also use point size to illustrate the chromosomes' fitness ratings. However unlike the distribution-based scatterplots, Sammon maps represent the entire search space. Therefore, the spatial relationship between points across all generations is constant and the need for brushing is removed.

An additional feature used to highlight separate solutions is to draw a line joining every chromosome with all its Hamming one neighbours in the current population (see figure 6 above) however, in some cases this can make the image too cluttered.

4 A New EC Visualization Technique

Search space matrices (SSM) have been developed by the author for constructing low dimensional representations of high dimensional data sets (Collins, 1996). This technique can be used to produce a unique linear mapping for an entire search space, thus making the comparison of spatial relationships between chromosomes possible over a number of generations with only linear computational complexity with respect to the number of points being shown.

Rather than scaling or mapping the population data to produce scatterplots, a matrix can be used to illustrate the entire search space. A new method proposed for doing this is to construct a matrix in which the Hamming distance between any two neighbouring points is minimised. This is achieved by constructing the matrix elements an allele at a time, dividing the matrix horizontally and vertically, and filling each sub-section with alternate allele. For example, a search space matrix for a four bit binary chromosome representation would be constructed as follows (see figure 7):

> 1st bit = 2 rows of 0s and 2 rows of 1s
> 2nd bit = 2 columns of 0s and 2 columns of 1s
> 3rd bit = 1 row of 0s, 1 row of 1s, 1 row of 0s and 1 one row of 1s
> 4th bit = 1 column of 0s, 1 column of 1s, 1 column of 0s and 1 column of 1s

However, rather than generating the entire search space the following direct linear equation can be used to translate a chromosome to its coordinate in r dimensions:

$$\begin{array}{c} \textit{chromosome coordinate} \\ \textit{in dimension } d \end{array} = \sum_{i=d}^{i \geq p} X_i \times W_i$$

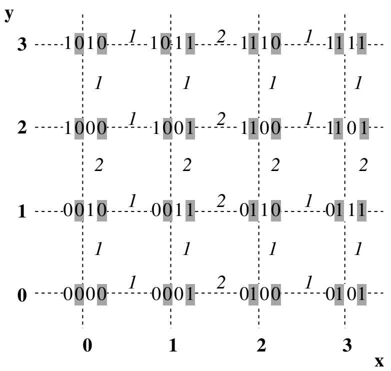

Figure 7: The construction of a complete search space matrix of a 4 bit binary dataset.

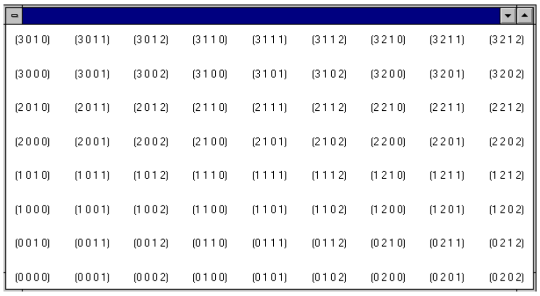

Figure 8: A matrix illustrating the complete search space for a chromosome data set with base numbers 4, 3, 2, and 3.

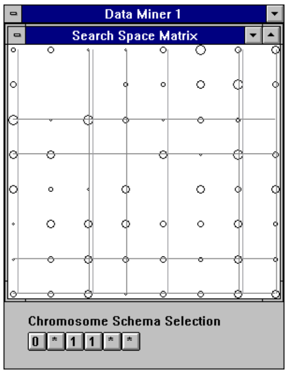

Figure 9: A search space matrix of a 6 bit binary data set illustrating how a schema selection dialog can be used to highlight the rows and columns on which chromosomes containing a selected schema may appear. Overlapping lines in the same plane are offset slightly so they remain visible.

Where; i is incremented by r. X is a chromosome with p variables. X_i is the position (indexed from the right at $i = 1$) of each allele (value) in the set of possible allele (loci alphabet). Each variable i has B_i different values, and the contributing weight of each variable $W_{1 \ldots r} = 1$ and $W_{r \ldots p} = W_{(i-r)} \times B_{(i-r)}$. In this way, not only can p bit binary strings be shown in r dimensions, but also any other form of categorical data (see figure 8).

The fact that a search space matrix is made up of alternate symbols across its rows and columns means that it can be used to emphasise the schema construction of the chromosomes it contains. Therefore, a dialog in which the user can identify a schema of interest may be used to highlight the rows and columns on which these schema occur.

Hence, for a six bit binary chromosome data set as shown in figure 9, the occurrence of chromosomes containing elements of the schema 0*11** could be highlighted with an additional twelve lines overlaid on the matrix, four for each of the three defined values (0,1,1). Chromosomes containing only one of these defined values would be shown as a point on a single line. Individuals containing two of the specified values would overlap two lines, and those containing three specified values would overlap three lines. An example chromosome (001100) containing all three specified values is shown at the intersection of the sixth row and third column from the bottom left corner in figure 9.

By using the point size to illustrate the chromosomes' fitness ratings, and enabling the user to highlight schema of interest, the user can seek out schema that emerge during evolution and directly examine their impact on the fitness ratings.

5 Summary

Although SV can support all aspects of software development it is the understanding of an EC search path i.e. EC output data, that has been of concern here. Table 1 summarises the six data visualization techniques presented above, along with their perceived pros and cons. Further general guidelines for visualizing information can be found in (Du Toit, et al., 1986), (Bertin, 1981), and (Tufte, 1990).

As can be seen in table 1 no single technique shows every aspect of the data, there is simply too much information for that to be possible. However, some techniques present information that others cannot and some present the same information in different ways.

The fitness versus generation number graph for example, consists of two linear spatial relationships; y position and fitness, and x position and time (i.e. generation number). This is the only technique that illustrates time as a linear spatial relationship. All others present population data from a number of generations either consecutively (a generation at a time), or concurrently (side by side).

Only Sammon mapping and search space matrices can show low dimensional representations of the search space. Although multidimensional scaling does present a distribution of points, that distribution is dependent on the points being considered and so comparisons of the spatial relationships across different scalings can not be made. Also a scaling of the entire space would not only be computationally expensive but also pointless. This is because the distribution of points within a complete population space will always be even and so the eigenvalues would all be equal making the prioritising of principal components impossible.

Table 1: A summary of data visualization techniques recommended for EC users.

TECHNIQUE	GRAPHIC	DATA	PROS	CONS
Population Data Matrices	Data Matrix	Entire population data set	-Shows complete data set, nothing is hidden	-Too much information to absorb
Fitness vs Generation Number Graphs	Graph	Best, average and/or worst fitness rating	-Clear comparison of fitness ratings -Applicable to all forms of EC	-No chromosome information
Allele vs Loci Frequency Matrices	Matrix	Allele freq. at each locus within a population	-Applicable to all forms of EC except genetic programming	-No chromosome structure (schema) or fitness information
Multi-dimensional Scaling	Scatterplot	Chromosomes' distribution and fitness	-Highlights separable subsets within a population	-No consistent spatial relationships across generations -Must verify rank two approximation or check the percentage of variation for each generation -Applicable to ECs with numerical allele only
Sammon Mapping	Scatterplot	Chromosomes and fitness	-Consistent spatial relationship across all generations	-Applicable to ECs with numerical allele only -Quadratic increase in computational complexity with respect to the number of points being generated
Search Space Matrices	Matrix	Chromosomes, schema and fitness	-Consistent spatial relationship across all generations -Linear mapping from chromosome to coordinate -Highlights chromosome schema structure	-Applicable to ECs with categorical allele only (i.e. no real numbers)

As well as fitness and search space visualization the user may wish to examine the make up of their chromosomes. The population data matrices, allele versus loci frequency matrices and search space matrices present information on the chromosomes' allele. The use of a data matrix to show selected population or chromosome data provides the most fine grained view of the data. Frequency matrices are a much more coarse grained view of population allele, and using a schema selection dialog in search space matrices allows the user to examine specific allele configurations and their contribution to fitness.

Selecting which technique or techniques to use is a decision the user must make based on the data they are interested in.

6 Discussion

Only visualizations suited to understanding EC search data have been considered here. Further forms of SV are also applicable to EC. Algorithm and code visualization to support EC comprehension and development, particularly the construction of problem-specific chromosome representations and evaluation functions, are important future applications for SV.

The representations used above to illustrate population data i.e. graphs, matrices and scatterplots, are based on the guidelines given by Bertin (1981). In order to construct effective visualizations of any form (data, software, scientific etc.) the designer must first examine the dimensions of the data they are interested in and then choose a graphic construction capable of displaying that information.

Clearly none of the above techniques shows every aspect of the population data, therefore a flexible data visualization tool incorporating these techniques and possibly other user-supplied, or problem specific, visualizations is recommended as a useful test tool. A prototype developed by the author is available from URL http://kmi.open.ac.uk/~trevor/Viz/tools/ feedback on this tool would be welcomed.

To conclude, it is hoped that the reader has gained an introduction to SV and how it may be applied to help EC users understand their search data. By presenting such visualizations either live during evolution, or post-mortem, the user can see the search path taken and make informed judgements about the quality of the solutions found. In the case of finding unsatisfactory solutions, having seen the areas of the search space that were examined the user can make an informed decision as to which parameter or operator changes they need to make.

Future Work

The author's future work involves the design and development of SV tools to support EC designers, particularly to help with the construction of new problem representations and evaluation functions. The author is also interested in evaluating the effectiveness of visualization techniques for EC and proposing SV design recommendations based on the results obtained. Further work on the search space matrix technique as a generic data mining tool is also planed.

Acknowledgements

The author would like to thank the Engineering and Physical Sciences Research Council for funding this work (EPSRC Studentship 94315065) and the members of the Open University's Knowledge Media Institute who helped comment on earlier drafts of this document, particularly John Domingue, Tony Hirst, Simon Masterton, Paul Mulholland and Stuart Watt.

References

Bertin, J. (1981). *Graphics and Graphic Information-Processing.* Berlin and New York: Walter de Gruyter.

Brown, M. H. & Sedgewick, R. (1985). Techniques for Algorithm Animation. *IEEE Software* January 1985. p. 28-39.

Chatfield, C. & Collins, A. J. (1980). *Introduction to Multivariate Analysis.* (1st ed.). Cambridge, UK: University Press.

Collins, T. (1993). *The Visualisation of Genetic Algorithms.* (MSc. Project Report, September 1993). De Monfort University, Leicester, UK.

Collins, T. D. (1996). *Genotypic-Space Mapping: Population Visualization for Genetic Algorithms* (Technical Report No. KMI-TR-39). KMI, The Open University, Milton Keynes, UK.

De Jong, K. (1980). Adaptive Systems Design. *IEEE Transactions on Systems, Man and Cybernetics.* p. 566-574.

Du Toit, S. H. C., Steyn, A. G. W. & Stumpf, R. H. (1986). *Graphical Exploratory Data Analysis.* (1st ed.). New York: Springer-Verlag Inc.

Dybowski, R., Collins, T. D. & Weller, P. R. (1996). Visualization of Binary String Convergence by Sammon Mapping. In L. J. Fogel, P. J. Angeline & T. Baeck (Ed.), *The Fifth Annual Conference on Evolutionary Programming (EP'96),* p. 377-383. San Diego, CA.: MIT Press.

Eisenstadt, M. & Brayshaw, M. (1987). *The Transparent Prolog Machine (TPM): An Execution Model and Graphical Debugger for Logic Programming* (Technical Report No. 21A). HCRL, The Open University, Milton Keynes, UK.

Goldberg, D. E., Deb, K. & Thierens, D. (1992). *Toward a Better Understanding of Mixing in Genetic Algorithms* (Technical Report No. 92009). IlliGAL, University of Illinois, Illinois, USA.

Harvey, I. & Thompson, A. (1996). Through the Labyrinth Evolution Finds a Way: A Silicon Ridge. In *The First International Conference on Evolvable Systems: From Biology to Hardware (ICES'96).*: Springer-Verlag.

Myers, B. A. (1983). Incense: A system for displaying data structures. *ACM SIGGRAPH '83 - Computer Graphics,* **17** (3), p. 115-125.

Myers, B. D. (1990). Taxonomies of Visual Programming and Program Visualisation. *Journal of Visual Languages and Computing,* **1** p. 97-123.

Price, B. A., Baecker, R. M. & Small, I. S. (1993). A Principled Taxonomy of Software Visualisation. *Journal of Visual Languages and Computing,* **4** p. 211-266.

Routen, T. & Collins, T. (1993). Visualisation of A.I. Techniques. In *COMPUGRAPH '93.* Portugal: ACM Press.

Sammon, J. W. (1969). A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers,* **C-18** (5), p. 401-408.

Tufte, E. R. (1990). *Envisioning Information.* Cheshire, Connecticut: Graphics Press.