IMPROVING STRATEGY PARAMETERS OF EVOLUTIONARY COMPUTATIONS WITH INTERACTIVE COORDINATED VIEWS

Andreas Kerren
University of Kaiserslautern, Computer Science Department
P.O.Box 3049, D-67653 Kaiserslautern, Germany
kerren@acm.org

ABSTRACT
Evolutionary algorithms (EAs) use mechanisms inspired by biological evolution, e.g., natural selection, recombination, or mutation, that work on populations of solutions for a specific problem. These recurring processes produce a huge amount of time-varying data. In order to get a better insight into the progress of EAs, a Java-based visualization tool, called EAVis, was developed. The most important aims of EAVis are the selection, concentration, and abstraction of evolutionary data on different levels using a variety of visualization methods. Several coordinated views (2D and 3D) help the user to watch each generation step of the EA and to derive knowledge as well as better understanding of the underlying evolutionary computational models. Among other things, this is also important for a clever parameter setting to gain better performance values.

KEY WORDS
Information Visualization, Software Visualization, Evolutionary Algorithms, Coordinated Views, HCI

1 Introduction

Evolutionary computation (EC) imitates some basic principles of evolutionary processes by using a variety of evolutionary computational models. Such computational models are typically called *evolutionary algorithms* (EAs) which include evolutionary programming, evolution strategies, genetic algorithms (GA), and genetic programming. Basically, EAs correspond to population-based metaheuristic optimization algorithms. At this, they use a number of common evolutionary methods, e.g., *selection* ("survival of the fittest"), *recombination* ("crossover"), or *mutation* ("random changes"). EAs work on a population of solutions in which the initial population is randomly initialized in most cases. By repeated application of the evolutionary methods, a new generation of the population is produced and a so-called *fitness function* evaluates each new generation, see [1].

The complexity of evolutionary algorithms makes them difficult to understand. Detailed analysis is difficult due to the large amount of data that can be obtained in a single run. We have implemented a visualization tool that displays the data on different levels facilitating the understanding of the underlying process and the impact of strategy parameters which determine the behavior of an EA or define an EA in more detail. Examples are population size, mutation rates, or crossover rates. Our aim was to build an extendable tool for general use in visualization of EAs, i.e., not only for special applications.

In the remainder of this paper, we firstly give a short overview of related works in Section 2. Next, Section 3 defines the evolutionary computation of the Knapsack Problem that we use as running example throughout this paper. Section 4 describes the design of our visualization tool EAVis and presents some screenshot examples as well as important features. Some implementation aspects are illuminated in Section 5. Finally, our paper ends with Section 6 giving concluding remarks.

2 Related Works

So far, there is only few research in visualization of EAs. Most visualization approaches were published at the end of the nineties. At this time, the Evolutionary Computation Visualization Workshop 1999 was an important event to bring researchers of this area together. It was co-located with the GECCO '99 Conference and was organized by T. D. Collins. In his work, he tried to adopt popular Software Visualization techniques (see for example [2, 3]) in order to illustrate the search behavior of evolutionary algorithms. One result was the development of Search Space Matrices: a visualization technique that allows a linear mapping for an entire search space [4]. His paper compares this technique with older visualization approaches recommended for EC users, such as Allele Versus Loci Frequency Matrices [5]. Another visualization tool of Collins, called GONZO [6], supports three different views: a Fitness Versus Time Graph, a Search Space Visualization (shows the GA's chromosomes as a point set in a 2D representation of the search space), and a Fine-grained Chromosome view (displays the values of selected chromosomes and their fitness ratings). GONZO was developed using the HENSON framework [7].

The VIS tool is discussed by Wu et al. [8]. Their aim is to facilitate the examination and analysis of GA runs. VIS supports the representation and examination of individuals, populations, and entire runs. Several "visualization formats" for displaying individuals are offered, e.g., as a string of characters, as a series of multicolor stripes (sim-

Proceedings of the 6th IASTED International Conference on Visualization, Imaging, and Image Processing, VIIP 2006, pages 88-93, Palma de Mallorca, Spain, 2006.

ilar to a bar code), and as collection of defined building blocks. These formats also include related details, such as parents of each individual or the possibility to watch individuals from a single population, etc.

A more line plot-like visualization of adaptive evolutionary phenomena is introduced by Bedau et al. [9]. So-called Activity Wave Diagrams show evolutionary activities of alleles on the basis of three different evolving systems, e.g., a model of traders selling and buying securities in a financial market using an evolving set of market-forecasting rules.

Pohlheim presents a set of standard techniques for different data types and time frames of EAs, see [10]. These techniques are based on data types, e.g., variables of best individual of every generation, and can be displayed with the help of standard visualization tools (plotter or diagram viewer). Additionally, multidimensional scaling as method for presenting high-dimensional data is discussed.

It is interesting that only few methods or techniques from the area of Information Visualization (InfoVis) are used to develop new visualizations of the time-varying data produced by EAs. Of course, this kind of visualization can be subsumed by Software Visualization. That is obvious but the vast amount of data also leads to problems which are tried to solve by the InfoVis community. Spears provides in his short paper [11] a nice brief overview of multidimensional visualization techniques for visualizing EAs including the use of color, glyphs, or parallel coordinates [12].

3 Application: 0/1-Knapsack Problem

As a simple application example, we have chosen the 0/1-Knapsack Problem [13] that is a classic NP-hard, combinatorial optimization problem with a wide range of applications. It may be formulated as follows: We have given N items i of different values $c_i > 0$ (costs) and volume $w_i > 0$ (weight), and a knapsack of fixed volume C (capacity). The task is to find the most valuable selection of items that fit in the knapsack. Each item must be put entirely in the knapsack or not included at all, i.e., objects cannot be broken up arbitrarily. Formally, we introduce N new variables x_i ($1 \leq i \leq N$) to encode the item selection ($x_i = 1$ if selected, $x_i = 0$ if not). We can describe the 0/1-Knapsack Problem as maximization problem:

$$\max \left\{ \sum_{i=1}^{N} c_i x_i \;\mid\; x_i \in \{0,1\} \wedge \sum_{i=1}^{N} w_i x_i \leq C \right\} \quad (1)$$

The original population P of our EA is initially created by random setting of the alleles, i.e. of the x_i variables. Here, we get a population size of $|P| = 100$ individuals (solutions) $s_j = (x_{ij}, \ldots, x_{Nj})$ with $1 \leq j \leq |P|$, $C = 250$, and $N = 30$. During the EA's run, all individuals are tested by a very simple fitness function F which evaluates their property to be a valid solution:

$$F(s_j) = \begin{cases} 0 & : \sum_{i=1}^{N} w_i x_{ij} > C \\ \sum_{i=1}^{N} c_i x_{ij} & : \sum_{i=1}^{N} w_i x_{ij} \leq C \end{cases} \quad (2)$$

Note that there are many other possibilities to define a fitness function. This procedure results in a ranking of the different individuals. The best individuals of the population are entitled to inherit their alleles to the next generation (selection). Mutation and inversion (a special kind of mutation) of alleles of randomly selected individuals also result in the forming of individuals in a new population. Individuals that are weak with regard to the fitness function are discarded: they are not entitled to spread their alleles in further generations. Actually, there is no break condition. The user can stop the evolutionary process by mouse-click and the result, the fittest individual of the population, is displayed. But we plan to extend EAVis by a possibility to enter any break condition. In the remainder of this paper we use this application as running example.

4 EAVis

EAVis provides an interface by which problem instances, e.g., the implementation of an EA that solve our 0/1-Knapsack Problem, can be specified (see Section 5). The tool takes the role of a process controller and looks for all relevant data for the visualization. In this way, it is also possible to control the EC process itself with the help of our tool. So, EAVis can be considered as on-line visualization system in contrast to post-mortem or off-line visualization systems, such as the VIS system [8]. The user can control the generation process online by using a VCR-metaphor, cp. [14] for more details. As first step, a new initial population is generated on the basis of the given specification. The user can estimate this population with the help of several coordinated views. If he/she is displeased with the result then the tool can generate another initial population by mouse-click. In case of an acceptable result, the user can start the evolutionary process.

4.1 Coordinated Views

Each produced generation can be monitored by a variety of coordinated views. Besides standard methods, like plotting mean fitness values of the entire generation versus fittest individual, the aim was to create novel views that will convey a more intuitive idea of the quality of the underlying strategy parameters. Moreover, our views should help to facilitate the non-trivial task of setting up the strategy parameters and other basic configurations, such as population size, termination conditions, etc. That is an ongoing work and we have not completely implemented all envisaged views yet. Additionally, the tool supports an easy possibility to embed an application-dependent Phenotype View, i.e., a visualization of the problem itself (cp. Section 4.2).

In order to display the permanent change of the population, several views have been created to facilitate the understanding of the underlying evolutionary processes. EAVis provides two different kinds of views: general views and tabbed views.

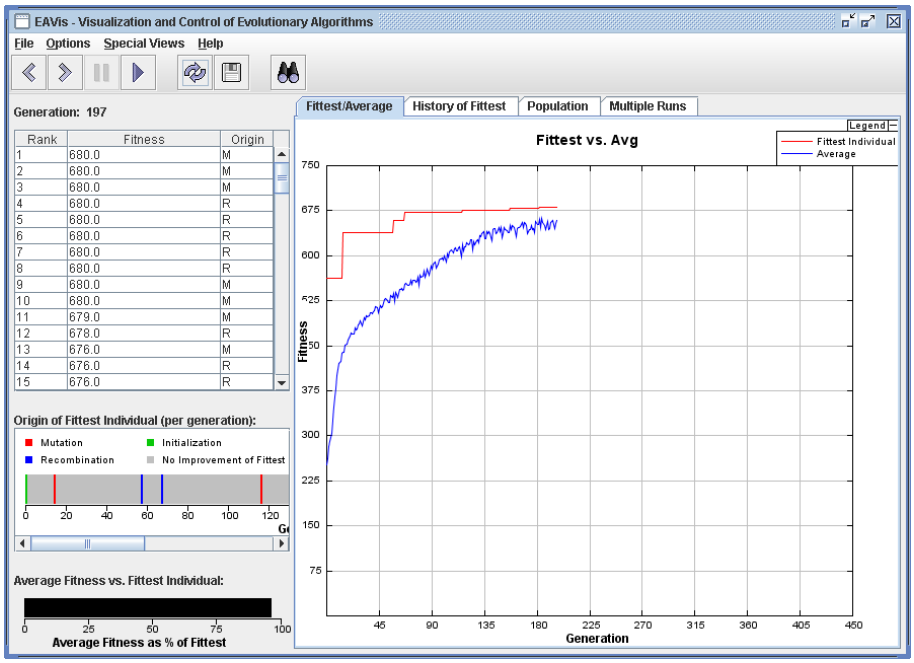


Figure 1. Screenshot of the EAVis tool (tab I opened)

4.1.1 General Views

Value of the general views is to inform the users about the overall state of the current evolutionary process. These views are shown at the left-hand side of Figure 1 and displayed at any time.

Ranking The Ranking view is basically a list which continuously displays all individuals in descending order of their fitness values, see Figure 1, top left-hand side. This view presents for each individual its rank, the last value of the fitness function F, and the evolutionary method (origin) which was used by the EA to create this individual. Thus, it can show whether there is a trend towards better fitness descended from a certain method of generating individuals. In our example, there are no unique identifiers (IDs) for the individuals. But there are applications in which IDs could be very useful, for example if the individuals have a more complex structure like neural networks etc. Another question is how IDs should be graphically represented (cp. the VIS system [8]). Currently, EAVis does not support the detailed graphical representation of individuals yet (our Allele subview is a first step in this direction, see Section 4.1.2).

Origin of Fittest Individual Receiving information about the origin of the fittest individual per generation is a very important issue for the understanding of the efficiency of the strategy parameters of the EA. If an improvement of the best fitness value occurs then this view draws vertical stripes into a horizontal generation scale (cp. Figure 1, centered on the left-hand side). Each stripe is colored to represent the origin of the present fittest individual, i.e, mutation (red), recombination (blue), or initialization (green).

Average Fitness vs. Fittest Individual This view is strongly connected with the Average/Fittest Plot view, see below. It symbolizes the relationship between the average fitness values and the best fitness value of the actual generation as a horizontal-bar diagram (cp. Figure 1, bottom left-hand side). If the user switches between the other views arranged on tabs then he can watch this view to receive a fast impression of the relative difference.

4.1.2 Tabbed Views

The user can easily switch between all views that are arranged on tabs, displayed at the right-hand side of Figure 1. These views give more detailed information. From the developer point of view, this concept supports any future extensions of EAVis with additional views.

Average/Fittest Plot The visualization of the fitness values of the best individual of each generation is one of the most used standard diagrams for the progress of an EA's run. In our example, we only have one fitness function because the considered 0/1-Knapsack Problem has one optimization variable namely the maximization of the costs. The Average/Fittest Plot view that is shown at the right-hand side of Figure 1 draws the fitness of the fittest individual (upper line plot) together with the average fitness value of all individuals (lower line plot) of each generation as line plots in one coordinate system.

To get a space-filling visualization, we decided that the coordinate system dynamically adapts to the current maximum generation number at the x-axis and to the highest fitness value at the y-axis. Note that the resulting plots are dependent on the actually used fitness function. Thus, it occurs that the average fitness value decreases temporary.

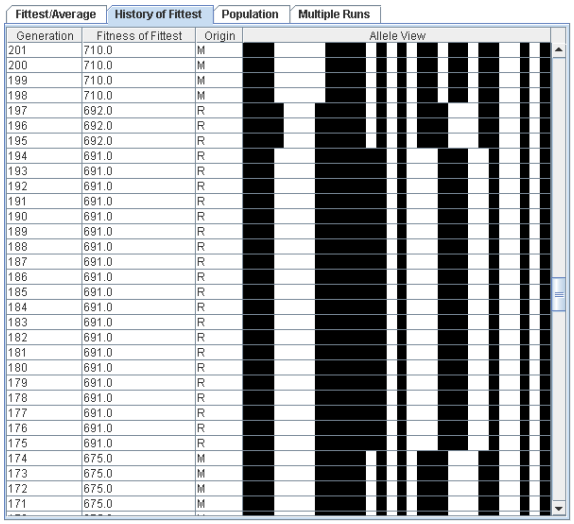
Within this view, it is also possible to highlight special events, for example if the fitness value increases by 10%. Then, the view fades in a small graphical symbol which can be clicked on. Subject to the event type additional information is displayed or the user is directed to another view.

History of the Fittest This view on the second tab is a table depicting the history of the fittest individuals. For each generation one individual with the best fitness is placed in an own row labeled with the generation number, fitness value, origin, and alleles (cp. Figure 2(a)).

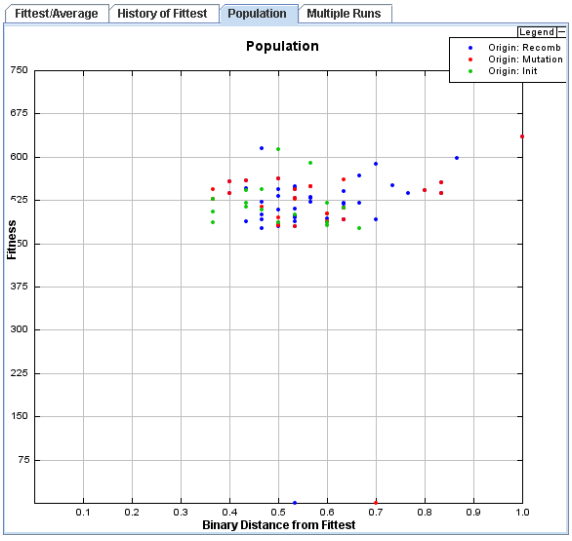
The Allele subview represents a history of the fittest individual of each generation. In our running example, the item selections for the knapsack are displayed as black and white "barcode" (white = 0, black = 1), similar to the display of real world gene sequences in molecular biology. Due to this representation, it is possible to see how and which alleles are passed from generation to generation or which alleles are subject to vast changes. Note, we merely have a b/w-coloring because the alleles, i.e., the x_i variables in Section 3, code binary information. Other application examples which use another genetic alphabet lead to more complex color coding schemes, for example schemes that are presented in the paper of Wu et al. [8]. Our tool supports such complex coding schemes too.

Population Sometimes it is important to see all individuals of a population of one generation. Individuals represent potential solutions to the problem that we want to solve. Thus, similarities between different individuals could be very interesting. The Population view (see Figure 2(b)) represents all individuals of the current population as colored dots. The color of these dots depends on the origin of the individual, i.e., whether it originated from a recombination (blue), mutation (red), or initialization (green). The positions of these dots are defined as follows: The fitness value of any individual is plotted on the y-axis; the x-axis represents the binary distance of the fittest individual (its computation is detailed in Section 5). As result, the dot representation of the fittest individual is located on the value 1.0 of the x-axis and the entire scatter plot describes the similarity of all solutions within one population shown by Figure 2(b). It follows that approximately 50% of the alleles of all individuals (dots) that are located near at the x-value 0.5 are identical with the alleles of the currently best individual. In this way, it is also possible to visualize subpopulations with similar genetic properties.

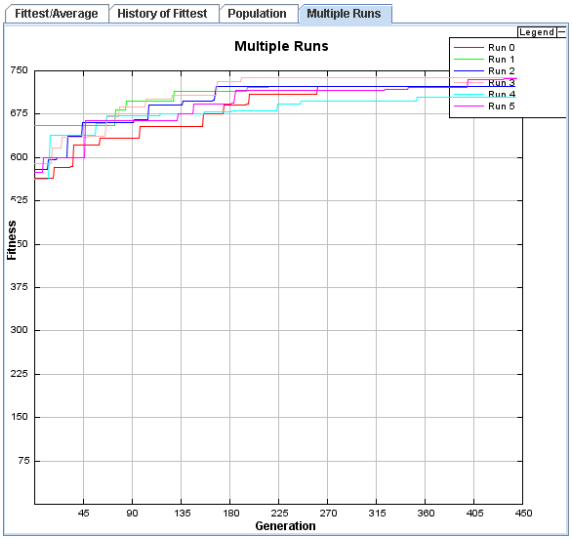
Multiple Runs EAVis offers the possibility to store all data produced in previous runs. The Multiple Runs view takes this information (of course based on the same input problem) and displays the fitness values of the best specific individual of each generation in a single coordinate system. This is only possible up to eight different runs in order to save storage. By this means, comparisons between different runs and strategy parameters are supported, shown in Figure 2(c). The screenshot illuminates six EA runs on the same input problem. They only differ in the initial population. Here, all runs have a very similar growth behavior.



(a) Tab II: History of Fittest



(b) Tab III: Population



(c) Tab IV: Multiple Runs

Figure 2. Three tabbed views

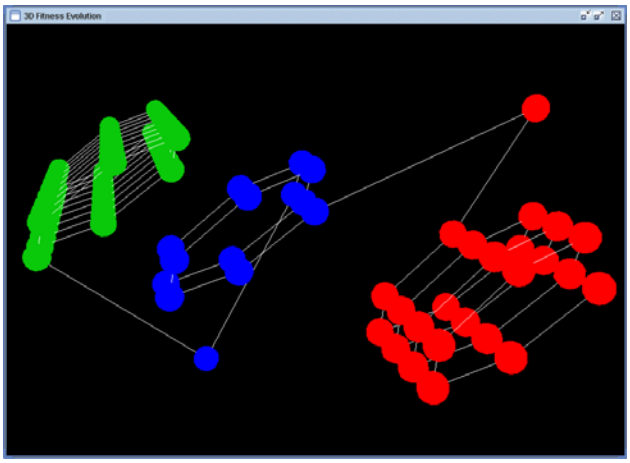


Figure 3. 3D Fitness Evolution

4.1.3 3D Fitness Evolution View

This 3D visualization is an alternative of the Origin of Fittest Individual view. In a 3D space, the fittest individual of each generation is displayed with the help of a sphere colored according to its origin (depending on whether the individual was created through mutation, recombination, or initialization). Within the space, the spheres are primarily arranged according to a helix. During the EA's run, the helix grows by adding new spheres till the fitness value of the fittest individual changes. Then, the helix is displaced to a new position and it can continue to grow, cp. Figure 3.

Note that this is a first step of the development of a novel 3D visualization for EAs. Our intent is to integrate more information into a 3D view, e.g., specific properties of the fittest individuals. Using a helix allows a space-filling design with the chance to add new features relatively easy. We have used an own separate window (no tab) for implementation reasons.

4.2 Features

EAVis offers a rich set of views which helps the user to get a very good overview and insight into the large amount of data produced by evolutionary algorithms. Using a predefined family of interfaces (cp. Section 5) the tool facilitates the integration of new implementations of any problems that are solved by evolutionary processes. This extensibility together with the possibility to embed problem-dependent phenotype views (see below) are important issues. They distinguish our tool from the most other visualization systems that we know. Further important features are briefly summarized in the following list:

- *Generation Control:* An important feature of EAVis is the ability to control the entire evolutionary process on-line with the help of a graphical user interface. When a user-defined EA is registered with the tool, the user can activate several running modes or start new runs via toolbar buttons.
- *Comparison of Multiple Runs:* Each run can be stopped at any time and archived by clicking on the floppy disk symbol in the toolbar. The Multiple Runs view can display this data for further investigation and comparison with other runs.
- *Phenotype:* There is also a support for making application specific visualizations and weaving them into the tool. Each EA that registers with our visualization tool can have one or more phenotype visualizations of individuals. These visualizations can be accessed through the menu bar *Special Views* of EAVis. This is specifically interesting for complex EAs, where certain properties cannot be depicted in an appropriate way by means of facilities provided by EAVis.

  In a previous work [15], we have visualized an evolutionary computation on the basis of neural networks. These neural networks should learn complex tasks, such as the wing stroke of a bird. A simulation of the wing stroke of the fittest individual was displayed in a separate phenotype view.
- *Parameter Adjustment:* Of course, the main aim of all visualizations is to facilitate the understanding of the underlying EA which allows to alter strategy parameters and to achieve a higher efficiency. Fully automatic adjustments of the strategy parameters are not implemented yet but this is a challenging feature that we plan to implement in future versions.

5 Implementation Aspects

EAVis was implemented in Java using the Java3D API for the development of the 3D Fitness Evolution view. As the aim of this tool is not only to provide the visualization of EAs but also to allow the complete control of the EA during its run, it was a necessity to support an interface between the EA to be visualized and the EAVis software itself. Therefore, a number of abstract interface classes was developed. Each implementation of a specific EA has to implement these abstract classes in order to meet the requirements that allow registration and handling the algorithm through the EAVis tool. In detail, there are three abstract Java classes that have to be derived from:

- *EAIndividual:* This class holds phenotype and genotype information of each individual.
- *EAPopulation:* It provides utilities to set up and manipulate a population of individuals (e.g., calculation of the next generation or initialization).
- *EAConfig:* Here, all algorithmic details are implemented that are subsequently used by the population in its evolution process.

The description of the Population view (see Section 4.1.2) poses the question how the binary distance between two individuals of our running example is computed.

In this view, the y-axis represents the fitness and the x-axis represents the binary distance of the individual's alleles from the alleles of the fittest individual and therefore providing an indication for similarity with the fittest individual. The binary distance between two individuals i and j is calculated with the following function:

$$\Delta_{\delta\gamma}(i,j) = \frac{a(i,j) + \delta e(i,j)}{a(i,j) + \delta e(i,j) + \gamma(b(i,j) + c(i,j))} \quad (3)$$

In this context, $a(i,j)$ returns the number of properties that both individuals have; $e(i,j)$ is the number of properties that neither has, and $b(i,j)$ respectively $c(i,j)$ return the number of properties that one has but the other does not and vice versa. Depending on the weight factors $\delta, \gamma \geq 0$, the function takes into account the absence of certain properties in both individuals or the existence of a certain property in one individual and the lack of this property in the other. The function $\Delta_{\delta\gamma}$ returns 1 if the two compared individuals are identical and 0 if there is no similarity between them.

6 Conclusion

In this paper, we have presented a visualization tool for evolutionary computations that can facilitate the understanding of evolutionary algorithms by using several coordinated views. It allows users to watch each generation step of the underlying EA from different viewpoints. Among other things, this is also important for a good setting of the strategy parameters to gain better performance values.

The development of EAVis is an ongoing work. In the future, we plan to extend and to improve the 3D Fitness Evolution view using InfoVis techniques. Furthermore, the on-line adaption of the strategy parameters is a challenging task with a high potential in practice.

From a user-centered view, EAVis was already observed as learning aid very positively and could also be used as lecture supplement: One of our students had embedded an EA into EAVis without any problems. His EA had computed a TSP tour for an optimization course (implemented without phenotype view). Future evaluations are planned to prove the efficiency of EAVis.

Acknowledgements: I would like to thank Thomas Egger for implementing the EAVis tool and for many constructive discussions.

References

[1] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing* (Springer, 2003).

[2] S. Diehl (Ed.), *Software Visualization* (volume 2269 of *LNCS State-of-the-Art Survey*, Springer, 2002).

[3] A. Kerren and J. T. Stasko, Algorithm Animation – Chapter Introduction, *Software Visualization*, volume 2269 of *LNCS State-of-the-Art Survey*, 1–15. Springer, 2002.

[4] T. D. Collins, Using Software Visualization Technology to Help Evolutionary Algorithm Users Validate Their Solutions, *Proceedings of the 7th International Conference on Genetic Algorithms (ICGA '97)*. Morgan Kaufmann, 1997, 307–314.

[5] T. Routen and T. D. Collins, Visualisation of A.I. Techniques, *Proceedings of the COMPUGRAPH '93*. ACM Press, 1993.

[6] T. D. Collins, Applying Software Visualization Technology to Support the Use of Evolutionary Algorithms, *Journal of Visual Languages and Computing*, 14(2), 2003, 123–150.

[7] T. D. Collins, Henson: A Visualization Framework for Genetic Algorithm Users, *Proceedings of the Congress on Evolutionary Computation (CEC '99)*. IEEE Press, 1999, 562–569.

[8] A. S. Wu, K. A. De Jong, D. S. Burke, J. J. Grefenstette, and C. L. Ramsey, Visual Analysis of Evolutionary Algorithms, *Proceedings of the Congress on Evolutionary Computation (CEC '99)*. IEEE Press, 1999, 1419–1425.

[9] M. A. Bedau, S. Joshi, and B. Lillie, Visualizing Waves of Evolutionary Activity of Alleles, T. D. Collins (Ed.), *Evolutionary Computation Visualization Workshop*, Orlando, Florida, USA, 1999, 96–98.

[10] H. Pohlheim, Visualization of Evolutionary Algorithms – Set of Standard Techniques and Multidimensional Visualization, W. Banzhaf (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99)*. Morgan Kaufmann, 1999, 533–540.

[11] W. M. Spears, An Overview of Multidimensional Visualization Techniques, T. D. Collins (Ed.), *Evolutionary Computation Visualization Workshop*, Orlando, Florida, USA, 1999.

[12] A. Inselberg, The Plane with Parallel Coordinates, *The Visual Computer*, 1(2), 1985, 69–92.

[13] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations* (John Wiley & Sons, 1990).

[14] A. Kerren and T. Egger, EAVis: A Visualization Tool for Evolutionary Algorithms, M. Erwig and A. Schürr (Eds.), *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC '05)*, Dallas, TX, USA, 2005, 299–301.

[15] A. Kerren and T. Egger, Poster: Visualization Tool for Evolutionary Algorithms, *Poster Session of the 1st ACM Conference on Software Visualization (SoftVis '03)*, 2003.