EAVis: A Visualization Tool for Evolutionary Algorithms

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

Thomas Egger
Computer Science Department
Vienna University of Technology
Favoritenstraße 9-11, A-1040 Vienna, Austria
egger@pyramus.org

Abstract

Evolutionary algorithms (EAs) produce a vast amount of data by recurring processes, e.g., selection, recombination, or mutation, that work on populations of solutions for a specific problem. In order to get a better insight into the progress of EAs a Java-based visualization tool, called EAVis, was developed. Several coordinated views 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.

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. 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 evolutionary methods, a new generation of the population is produced and a *fitness function* evaluates each new generation [7].

The complexity of evolutionary algorithms make them difficult to understand. We have implemented a visualization tool, called *EAVis*, that displays the data on different levels facilitating the understanding of the underlying process and the impact of strategy parameters (for example population size or crossover rates) which determine the behavior of an EA.

The next Section 2 describes related works. In Section 3 the visualization design of our visualization tool EAVis is presented. Finally, Section 4 gives concluding remarks and discusses future work.

2. Related Works

There is only few research in visualization of EAs. Most visualization approaches were published at the end of the nineties. T. D. Collins tried to adopt popular Software Visualization techniques (see for example [3]) in order to illustrate the search behaviour of evolutionary algorithms with the help of his visualization tool GONZO that was developed using the HENSON framework [2]. 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. A more line plot-like visualization of adaptive evolutionary phenomena is introduced by Bedau et al. [1]. So-called Activity Wave Diagrams show evolutionary activities of alleles on the basis of three different evolving systems. Pohlheim presents a set of standard techniques for different data types and time frames of EAs, see [5]. Such standard 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.

It is interesting that only few methods or techniques from the area of Information Visualization (InfoVis) are used to develop new visualizations of data produced by EAs. It is obvious that this kind of visualization can be subsumed by Software Visualization 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 [6] a nice brief overview of multidimensional visualization techniques for visualizing EAs including the use of color, glyphs, or parallel coordinates.

3. Visualization Approach

Our aim was to build a tool for general use in visualization of EAs, i.e., not only for special applications. As a simple application example we have chosen the 0/1-Knapsack Problem [4] that has a wide range of applications. In the remainder of this paper, we use this application as running

Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing,
VL/HCC 2005, pages 299-301, Dallas, TX, USA, 2005.

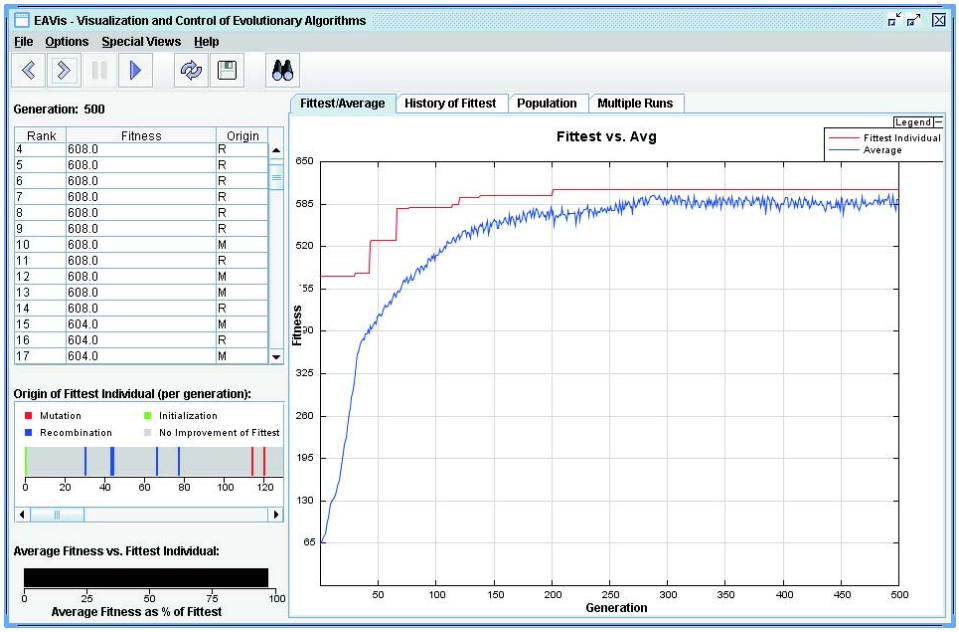

Figure 1. Screenshot of the EAVis tool (tab I opened)

example. EAVis provides a special Java interface by which problem instances, e.g., the implementation of an EA that solves our example problem, can be specified. 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]: by using a VCR-metaphor, the user can control the generation process online; he can play it with a user-defined speed, step-by-step, or backwards; he can stop it, store all data for run comparisons, or perform a reset with a new base population. 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 (including standard deviation), the aim was to create novel views that will convey a more intuitive idea of the quality of the underlying strategy parameters. Additionally, the tool supports an easy possibility to embed an application-dependent *Phenotype View*, i.e., a visualization of the problem itself. A final aim of our tool development is to control the EC process itself, especially the adaption of the strategy parameters.

Several views have been created to facilitate the understanding of the underlying evolutionary processes. The user can easily switch between all views that are arranged on tabs. Exceptions of this tab layout are three smaller views that are permanent visible (shown on the left-hand side of Figure 1). They are discussed at first.

Ranking: The Ranking View is basically a list which continuously displays all individuals in descending order of their fitness values (top left-hand side). This view presents for each individual its rank, the last value of the fitness function, and the evolutionary method which was used by the EA to create this individual (origin). Thus, it can show whether there is a trend towards better fitness derived 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, e.g., 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 (our Allele View is a first step in this direction). However, we plan to add this feature as soon as possible.

Origin of Fittest Individual: To receive 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 (centered on the left-hand side). Each stripe is colored to represent the origin of the present fittest individual, i.e, mutation, recombination, inversion, or initialization.

Average Fitness vs. Fittest Individual: This view is strongly connected with the Average/Fittest Plot. It symbolizes the relationship between the average fitness values and the best fitness value of the actual generation as a horizontal-bar diagram (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.

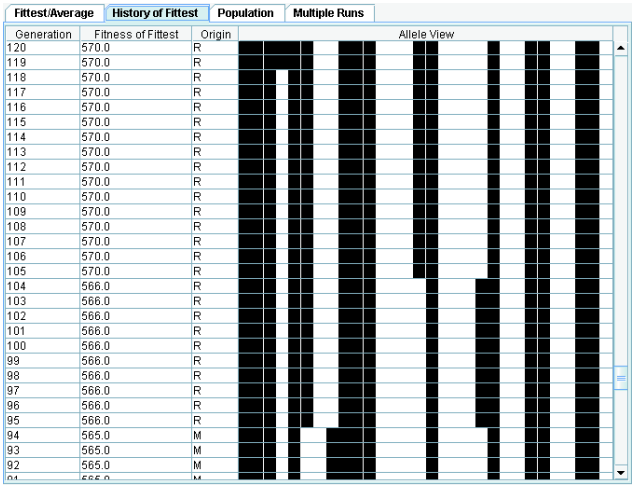

Figure 2. Tab II: History of Fittest

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 (red colored upper line) together with the average fitness value of all individuals (blue colored lower line) of each generation as line graphs. The resulting graphs are highly dependent on the actually used fitness function. Thus, it occurs that the average fitness value can decrease from time to time.

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 Allele View (cp. Figure 2 and [8]). The Allele View shows the alleles of the fittest individual of each generation. An allele is one of a number of alternative forms of the same gene occupying a specific position. In our running example, the item selections for the knapsack are displayed as black and white "barcode", 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 code binary information. Other application examples which use another genetic alphabet lead to more complex color coding schemes.

Population: Individuals represent potential solutions to the problem that we want to solve. So, similarities between different individuals are very interesting. The Population View represents all individuals of the current population as colored dots. Colors act for the individual's origin; positions depend on fitness values and the binary distance of the fittest individual. This allows the visualization of 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 one coordinate system. This is possible for up to seven different runs at the moment. By this means, comparisons between different runs and strategy parameters are supported

4. 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. These views allow the user 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. A future software evaluation will prove its efficiency.

References

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

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

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

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

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

[6] W. M. Spears. An Overview of Multidimensional Visualization Techniques. In T. D. Collins, editor, *Evolutionary Computation Visualization Workshop*, Orlando, Florida, USA, 1999.

[7] W. M. Spears, K. A. De Jong, T. Bäck, D. B. Fogel, and H. de Garis. An Overview of Evolutionary Computation. In *Proceedings of the 6th European Conference on Machine Learning (ECML '93)*, pages 442–459. Springer, 1993.

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