

Visualization Tool for Evolutionary Algorithms

Thomas Egger

Andreas Kerren*

Algorithms and Data Structures Group
Institute of Computer Graphics and Algorithms
Vienna University of Technology
Favoritenstraße 9-11, A-1040 Vienna, Austria

ABSTRACT

Analyzing the large amount of data which is produced by evolutionary algorithms is a difficult task. In this paper we present a Java-based visualization tool that displays such data on different levels. In this way it facilitates a better understanding of the underlying evolutionary computational models.

1. INTRODUCTION

Evolutionary computation 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 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 the evolutionary methods a new generation of the population is produced and a so-called fitness function evaluates each new generation [3].

The complexity of evolutionary algorithms make them difficult to understand. Detailed analysis is made more difficult by 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. Temporary developed for a very special application the aim is to get a tool for general use in visualization of EAs. First, we give a short overview about the application area of our tool in section 2. Section 3 describes the visualization of EAs and presents some screenshots. Finally, section 4 illustrates related works and gives concluding remarks.

*Corresponding author: kerren@ads.tuwien.ac.at

2. APPLICATION

As a first application we consider a population of neural networks (NNs) having the goal to learn various complex tasks, such as a non-linear motion of an object. Here, these neural networks can be regarded as the brains of different two-winged animals, e.g. birds. The brains have no knowledge when the algorithm starts and their task is to learn a complex movement of some body parts of the animals. The motion that they produce (by flapping animal's wings) enables the animals to move from one point in space to another, cp. Figure 1.



Figure 1: Wing stroke of an animal

The original population of our EA is initially created by random setting of the synaptical weights. During the EA's run, all occurrences of NNs are tested by a fitness function F which evaluates their ability to move the animal. This results in a ranking of the different NNs. The best individuals of the population (here 500 individuals) are entitled to inherit their "genes", i.e. their synaptical weights, to the next generation of networks. Mutation and inversion (a special kind of mutation) of genes of randomly selected individuals also result in the forming of individuals in a new population. Individual NNs that are weak with regard to the fitness function are discarded: they are not entitled to spread their genes in further generations. After 100 generations of NNs (break condition) the EA terminates and the result, the fittest individual of the population, is displayed.

3. VISUALIZATION

In order to display the ongoing change of the population, several views have been created to facilitate the understanding of the underlying evolutionary processes.

Ranking: The ranking view is basically a list which continuously displays the ten fittest NNs, see Figure 2. In our case the fitness function F is simply the distance that the animal can cover in a certain amount of time. Remember that the brain of each animal is a neural network. This view presents for each of the NNs its unique ID, the last value of F , and the evolutionary method which was used by the EA to create this neural network.

-Rangliste		5%	
1	469	F: 4.627	R
2	462	F: 4.615	R
3	67	F: 4.593	M
4	56	F: 4.557	R
5	323	F: 4.496	R
6	299	F: 4.496	R
7	239	F: 4.473	R
8	386	F: 4.467	M
9	32	F: 4.449	R
10	295	F: 4.437	R

Figure 2: Ranking

Log list: Figure 3 presents the log list view which shows a history of the fittest individual of each generation. The synaptical weights of the fittest NN are displayed as a colored "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 genes are passed on from generation to generation.

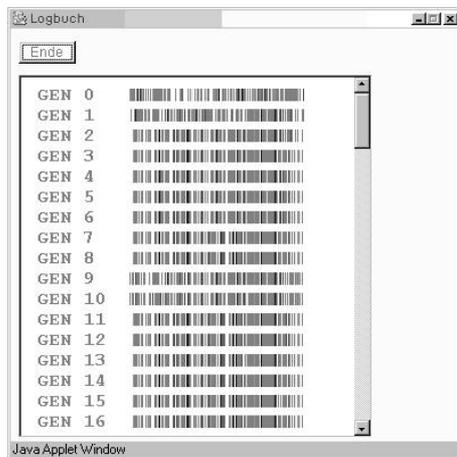


Figure 3: Log list

Evolution map: A more complex view is the creation of a map that holds various information about the evolution of the whole system over the time. Three different geometrical shapes are drawn onto a white drawing area depending on time and the fittest individual, cp. Figure 4. After each generation loop of the EA the origin of the fittest animal is calculated. It might be the result of a mutation, inversion or recombination of individuals of the previous generation. Depending on whether it has arisen from some individuals by these evolutionary methodes, the visualization tool draws the following geometrical shapes:

- *Mutation:* A line in various degrees of thickness and length depending on time and impact of the individual is drawn, i.e. the more important the genetical leap of this individual was, the thicker the line is, and the later in the progress we are the longer, the line will be.
- *Inversion:* A box filled with red color is drawn in various degrees of size and thickness, again depending on time and impact of the individual.
- *Recombination:* A circle is analogously drawn.

Currently, the positioning of these geometrical shapes is partially random, but we work on a better representation of this evolution trace. As a result, a graphic is created similar to certain pictures you might encounter in a museum of modern art. The difference is that these drawings can be interpreted by the user in such a way, that the history of the population evolution of NNs can be easily analyzed.

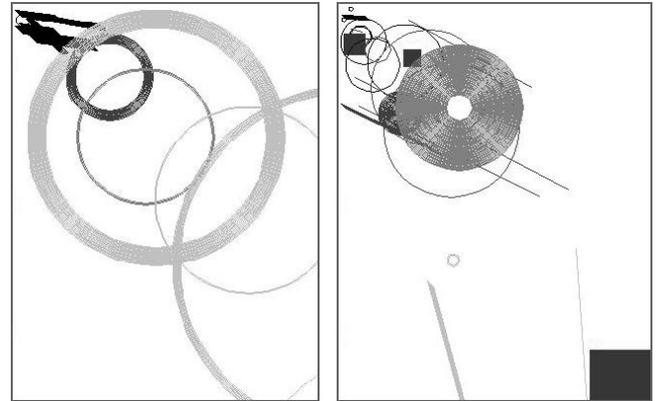


Figure 4: Two evolution maps

4. CONCLUSION

There is only few research in visualization of EAs. The VIS tool is a newer example that was discussed in a paper of A. S. Wu et al. [4]. Other visualization approaches were presented at different conferences on genetic and evolutionary computation e.g. by H. Pohlheim [2] or T. D. Collins [1]. In this overview we have shortly summarized a number of visualization aspects for evolutionary computations. A tool was illuminated that can facilitate the understanding of evolutionary algorithms through visualizing changes within a population of NNs. This application should only be considered as a first test case of our development. Currently, we try to extend the capabilities of our prototype implementation and to discover novel representations of the huge amount of data which is produced by EAs.

5. REFERENCES

- [1] T. D. Collins. Using Software Visualization Technology to Help Genetic Algorithm Designers. In *Proc. of the 7th International Conference on Genetic Algorithms (ICGA '97)*, pages 307–314. Morgan Kaufmann, 1997.
- [2] H. Pohlheim. Visualization of Evolutionary Algorithms – Set of Standard Techniques and Multidimensional Visualization. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO '99)*, pages 533–540. Morgan Kaufmann, 1999.
- [3] W. M. Spears, K. A. De Jong, T. Bäck, D. B. Fogel, and H. de Garis. An Overview of Evolutionary Computation. In *Proc. of the 6th European Conference on Machine Learning (ECML '93)*, pages 442–459. Springer, 1993.
- [4] A. S. Wu, K. A. De Jong, D. S. Burke, J. J. Grefenstette, and C. Loggia Ramsey. Visual Analysis of Evolutionary Algorithms. In *Proc. of the Congress on Evolutionary Computation (CEC '99)*, pages 1419–1425. IEEE Press, 1999.