# A Bio-inspired Genetic Algorithm with a Self-Organizing Genome: The RBF-Gene Model

Virginie Lefort, Carole Knibbe, Guillaume Beslon, and Joël Favrel

INSA-IF/PRISMa, 69621 Villeurbanne CEDEX, France
{vlefort, cknibbe, gbeslon}@prisma.insa-lyon.fr, joel.favrel@insa-lyon.fr

## 1 Introduction

Although Genetic Algorithms (GAs) are directly inspired by Darwinian principles, they use an over-simplistic model of chromosome and genotype to phenotype mapping. This simplification leads to a lack of performance, mainly because the chromosome structure directly constrains the evolution process.

In biology, the structure of the chromosome is free to evolve. The main feature permitting it is the presence of an intermediate layer (the proteins) between genotype and phenotype: Whatever the size and the locus of a gene, it is translated into a protein and all the proteins are combined to "produce" the phenotype.

Some authors, like Goldberg [1], have tried to introduce some independence between the genotype and the phenotype in GAs but none have really introduced the "protein level". Thus, they do not really part the two levels. We propose a new model of GA introducing such an intermediate level in order to permit evolvability *during* and *by* the evolutionary process to improve convergence.

## 2 The RBF-Gene Algorithm

Inspired by neural networks, we propose to discover the shape of the problem function by combining an unspecified number of basic functions, each of them having a fixed number of parameters. Each "gene" (i.e. each coding sequence) will encode one of these basic functions (called "kernels", e.g. Gaussian functions) exactly as, in biology, the genes encode proteins. The phenotype is then obtained by the interactions of the different kernels in a linear combination. So, the algorithm can be used to approach any bounded $\mathbb{R}^n$ to $\mathbb{R}$ function.

Coding and non-coding sequences are simply distinguished thanks to two genetic sequences ("start" and "stop" sequences). In between, the sequence will be analyzed thanks to a "genetic code" in order to compute the kernel parameters: Each base is associated to one parameter and to one value (0 or 1). Our kernels have three parameters: The mean vector (in $\mathbb{R}^n$), the standard deviation and the kernel weight in the linear combination. To compute the value of a kernel parameter, we extract the subsequence of all the values associated with it, which is then translated thanks to a variable-size Gray code.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 406–407, 2004.

Since there is no global rule to analyze the sequence, it can be modified by any biologically-inspired operator: Switches, local insertions and deletions, large-scale operators, crossover... The chromosome size, the number of kernels, the locus of the genes and the size of the coding sequences are then free to evolve.

## 3 Experiments: The Abalone Regression Task

We tested the algorithm on a regression benchmark: The abalone data set [2]. The goal is to find the age of an abalone given eight biological inputs. The data set contains 4177 experimental points and the fitness function is the root mean squared error. We have done 10 runs of 5000 generations each (each with 10 fold cross-validation). The results show that the RBF-Gene algorithm gives good results, similar or better than those given in [3].

In the very first generations, the algorithm adds a lot of new kernels. Then, it improves the existing kernels, modifying/enlarging the coding sequences to be more precise while the number of kernels remains steady.

The important aspect of the algorithm is the possibility to reorganize its genome while evolving functionally. We could think that the genome size will grow more and more. In practice, merely all chromosomes stabilize their size at around 3000 bases (the initial individuals have 200 bases) representing 37 kernels. We can see here "second order evolution" since long genomes are less *evolvable*.

## 4 Conclusions and Future Work

The RBF-Gene algorithm introduces a real independence between the genotype and the phenotype: All the genes encode proteins and the phenotype is always computable, whatever the number of genes. Moreover, the chromosome structure is dynamically modified during the evolution process to improve future evolution. Although it is still under study, our model gives promising results and proves its ability to add new genes during the evolutionary process.

Future work will focus on biologically inspired operators: By now, we used random operators while biological large-scale ones are based on sequence similarity to enhance their possibilities on reorganizing the structure of the genome.

## References

1. Goldberg, D.E., Deb, K., Kargupta, H., Harik, G.: Rapid accurate optimization of difficult problems using fast messy genetic algorithms. In Forrest, S., ed.: Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, CA, Morgan Kaufmann (1993) 56–64
2. UCI Machine Learning Website (http://www.ics.uci.edu/∼mlearn/MLRepository.html): Abalone data set (consulted in 2003)
3. Automatic Knowledge Miner (AKM) Server: Data mining analysis (request abalone). Technical report, AKM (WEKA), University of Waikato, Hamilton, New Zealand (2003)