IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA ITB10424
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

Chapter XIII

Controlling Robots with Fractal Gene Regulatory Networks

Peter J. Bentley, University College London, UK

ABSTRACT

Fractal proteins are a new evolvable method of mapping genotype to phenotype through a developmental process, where genes are expressed into proteins comprised of subsets of the Mandelbrot set. The resulting network of gene and protein interactions can be designed by evolution to produce specific patterns, which in turn can be used to solve problems. This chapter introduces the fractal development algorithm in detail and describes the use of fractal gene regulatory networks for learning a robot path through a series of obstacles. The results indicate the ability of this system to learn regularities in solutions and automatically create and use modules.

INTRODUCTION

Life is complex. This is true in the chemical interactions of proteins and genes within a single cell, or in the cellular interactions in a multicellular organism. While evolution is mostly to blame for this, there can be little doubt that complexity could not arise if the vast intricacies of molecular interactions and physical forces were not present. Open-ended evolution (evolution in which solutions get progressively more complex) relies on the right kind of genetic representation in the right kind of environment. In nature, this is DNA — a molecule that relies on chemical interactions in order to function.

Translating these ideas into computer science is not a simple prospect. As we know in evolutionary computation, using a fixed binary string as a genotype prohibits complexity growth. But variable-length representations such as genetic programming do not guarantee an increase in complexity either (unless you count introns as complexity). Even if a seemingly ideal representation is found, often it is not evolvable, or it only achieves its complexity increases through the careful high-level structures (e.g., modularity) imposed on it by the developer.

This work takes a different approach. A developmental process maps variable-length genotypes to phenotypes, through the use of *fractal proteins.* Genes are expressed into complex fractal shapes (subsets of the Mandelbrot set) that interact according to their forms. The resulting network of gene interactions can be designed by evolution to produce specific gene activation patterns, which in turn can be used to solve problems. In this chapter the use of fractal gene regulatory networks for learning a robot path through a series of obstacles is described.

BACKGROUND

Development

Development is the set of processes that lead from egg to embryo to adult. Instead of using a gene for a parameter value as we do in standard EC (i.e., a gene for long legs), natural development uses genes to define proteins. If expressed, every gene generates a specific protein. This protein might activate or suppress other genes, might be used for signalling amongst other cells, or might modify the function of the cell it lies within. The result is an emergent "computer program" made from dynamically forming gene regulatory networks (GRNs) that control all cell growth, position and behaviour in a developing creature (Wolpert et al., 2001). An introduction to biological development can be found in Bentley (2002), with greater detail in Wolpert et al. (2001) and Kumar and Bentley (2003a).

There is currently much research being performed on computational development. Problems of scalability, adaptability and evolvability have led many researchers to attempt to include processes such as growth, morphogenesis or differentiation in their evolutionary systems (Jackson & Tyrrell, 2002; Miller & Banzhaf, 2003; Sipper, 2002). However, research focussing on the creation of GRNs is less common in computer science. Jakobi (1995) was one of the first to explicitly design a system that enabled (and even extracted) GRNs, which were then used to develop neural networks for controlling robots. Jakobi's genetic representation was highly flexible, but had no real concept of chemistry to affect his proteins. While Jakobi was enthused by the results at the time, the subsequent lack of further development implies that his proposed representations were not as evolvable as first thought. Another example is the work of Eggenberger (1997), which still remains impressive, demonstrating the use of gene behaviours such as regulation, and cellular behaviours such as differentiation and simple morphogenesis. Eggenberger modelled biological development in some detail, and showed how different forms made from cell clusters could be evolved using these techniques. Nevertheless, Eggenberger's work has little concept of gene expression or protein folding, and the evolvability and scalability of his system remains open to question. More recently,