Genetically Programming Networks to Evolve Memory Mechanisms

Arlindo Silva
Center for Informatics and Systems
of the University of Coimbra
Polo II – Pinhal de Marrocos
3030 Coimbra - Portugal

Ana Neves
Center for Informatics and Systems
of the University of Coimbra
Polo II – Pinhal de Marrocos
3030 Coimbra - Portugal

Ernesto Costa
Center for Informatics and Systems
of the University of Coimbra
Polo II – Pinhal de Marrocos
3030 Coimbra - Portugal

Abstract

Many agents need memory to survive in their environment or to solve some specific task. We present an evolutionary approach to the development of agents with memory. This approach is based on a form of distributed genetic programming, named genetically programmed networks. Memory mechanisms are implemented, not by the program structure or some form of indexed memory, but by the evolution of recurrent connections between programs associated to the nodes of a genetically programmed network.

1 INTRODUCTION

In this article we propose Genetically Programmed Networks (GPN) [Silva99] as an alternative process to evolve agents with memory. Based on results obtained using GPN to solve two benchmark problems, the Ant Problem and the Tartarus Problem, we defend that evolving GPN can be more efficient than other evolutionary approaches previously used to evolve agents with memory, namely program structure [Koza92], indexed memory [Teller94] and concurrent genetic programming [Trenaman98].

2 GENETICALLY PROGRAMMED NETWORKS

A Genetically Programmed Network is constituted by a sequence of programs. Each program can be pictured as associated with a node in a network. Besides the nodes with associated programs, the network also has a set of inputs, a set of outputs and connections between them. The nodes are the computing elements in the network and each one uses the attached program to compute its output based on data flowing in from its connections. Connections act as a mean of transportation for data between inputs, outputs and nodes. There is no explicit representation for this network. In fact, the network structure is implicitly defined in the way the programs make use of a carefully defined terminal set. The evolutionary process used to evolve the programs, and, consequently, GPN, is an extension of genetic programming (GP), as defined in [Koza92], to multi-tree individuals.

3 EXPERIMENTAL RESULTS

We used GPN to evolve agents for two well-known benchmark problems, the Ant Problem [Koza92] and the Tartarus Problem [Teller94]. Comparison between the best GPN based approach and several others is presented in Table 1 and shows GPN to be the most efficient approach. Comparison is made in terms of the effort [Koza92] needed to solve the Ant Problem and of the best individuals evolved for the Tartarus Problem.

Table 1: Comparison between GPN and other GP based approaches for the Ant and Tartarus Problems

Ant Prob.	Effort	Tartarus	Best	Avg.
GP	450,000	GP	98	63
PDGP	426,000	GP+Mem.	212	177
EP	136,000	CGP	221	205
GPN Neural Net	59,500	GPN Rule System	252	227

References

[Koza92] J. Koza, "Genetic programming: On the programming of computers by means of natural selection", Cambridge, MA, MIT Press, 1992.

[Silva99] A. Silva, A. Neves and E. Costa, "Evolving Controllers for Autonomous Agents Using Genetically Programmed Networks", to appear in *EuroGP99: Proceedings of 2nd European Workshop on Genetic Programming.*

[Teller94] A. Teller, "The Evolution of Mental Models", *Advances in Genetic Programming*, Complex Adaptive Systems, pp. 199-220, MIT Press, 1994.

[Trenaman98] A. Trenaman, "Concurrent Genetic Programming and the Use of Explicit State to Evolve Agents in Partially-Known Environments", *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pp. 391-398, Morgan Kaufmann, 22-25 July 1998.