Result-Sharing: A Framework for Cooperation in Genetic Programming

Edgar E. Vallejo
Departamento de Ciencias Computacionales
Instituto Tecnológico y de Estudios
Superiores de Monterrey
Campus Estado de México, México

Fernando Ramos
Programa de Graduados en Informática
Instituto Tecnológico y de Estudios
Superiores de Monterrey
Campus Morelos, México

Abstract

We present a framework for cooperation in genetic programming which involves modifications to the program structure of genetic programming with automatic definition of functions and a cooperation stage in which individuals share function-defining branches.

1 SUMMARY

Genetic programming (GP) systems may be viewed as cooperative processes: individuals in the population *cooperate* to evolve a global solution. Result-sharing is a form of cooperation in which individuals assist each other by sharing partial results. Cobb has demostrated that the convergence of genetic algorithms can be improved through result-sharing (Cobb, 1993).

In our model, individuals accept and defect to share partial results using a decision procedure. We exploit the structure of GP with automatic definition of functions (Koza, 1994): function-defining branches are viewed as partial results. Individuals in the population decide to cooperate with a potential partner on the basis of expected payoffs using a decision procedure (Stanley, Ashlock and Tesfatsion, 1994).

We made two modifications to the basic GP with automatic definition of functions to enable result-sharing. The first is a modification to the program structure and the second is the addition of a cooperation stage.

In our implementation, individual's overall program is defined as a *lambda expression* with one argument. This argument is denoted by a variable which is bounded to a procedure during execution. This variable, in turn, is included in the function set of the result-producing branch. Before evaluation, the overall program is applied to the procedure defined by the function-defining branch of its selected cooperation partner. Using this mechanism, the function-defining branch of one individual can be embedded within the result-producing branch of another individual.

In addition, we include a process for the progression of cooperation. After each evolution cycle, we execute a cooperation phase in which individuals exchange procedures. Each cooperation phase, in turn, consists of 4 stages: (1) a *selection stage* in which individuals choose potential cooperation partners; (2) an *evaluation stage* in which individuals estimate the payoff of cooperation; (3) a *decision stage* in which potential partners accept or defect to cooperate; (4) an *update stage* in which programs are updated to include the modifications derived from the decision.

The experiment consist of comparisions between GP with cooperation and GP with no cooperation. Our results indicate that GP with cooperation outperforms GP with no cooperation solving several instances of the lawn mower problem (Koza, 1994) and the six-legged insect locomotion problem (Spencer, 1994).

References

H. Cobb (1993). Is the Genetic Algorithm a Cooperative Learner?. In D. Whitley, *Foundations of Genetic Algorithms 2*. San Mateo, CA: Morgan Kaufmann.

J. Koza (1994). *Genetic Programming II*. Cambridge, MA: The MIT Press.

G. Spencer (1994). Automatic Generation of Programs for Crawling and Walking. In K. Kinnear (ed.), *Advances in Genetic Programming*. Cambridge, MA: The MIT Press.

E. Stanley, D. Ashlock, L. Tesfatsion (1994). Iterated Prisoner's Dilemma with Choice and Refusal of Partners. In C. Langton (ed.), *Artificial Life III*. Reading, MA: Addison Wesley.