Control System Optimization Using Genetic Algorithms within the SoftLab Toolkit

Lisa M. Desjarlais **Mohammad-R. Akbarzadeh-T.** **Craig W. Wright**

NASA Center for Autonomous Control Engineering, University of New Mexico, Albuquerque, NM 87131

Abstract

This paper presents techniques for using genetic algorithms within the SoftLab software environment to automate the optimization of control systems. SoftLab is a framework for the design and simulation of control systems, using soft computing components. It allows users to construct systems by connecting computational modules into dataflow networks. We discuss the set of modules provided for implementing genetic algorithms and demonstrate how even the problem-specific aspects of the genetic algorithm can be easily specified through interpreted scripts, avoiding the need to compile or link in order to set up a new test problem.

SoftLab (Desjarlais & Wright, 1999a) is a framework for the design and simulation of control systems, using soft computing components. It allows users to construct systems by connecting computational modules into dataflow networks. InputVariables (IVs) and OutputVariables (OVs) are the data port objects with which SoftLab Module objects connect to each other, as well as handy places to store a Module's state values for its own use. To apply a genetic algorithm within SoftLab (Desjarlais & Craig, 1999b), the user defines a potential solution to the test problem to be initial values for some subset of the InputVariables embedded in Modules throughout a SoftLab network. Note that this limits the parameters that can be optimized to those whose values are stored in InputVariables.

A SoftLab network that incorporates optimization is hierarchical. It contains at least one supervisory Module, which does not participate in the control loop. (We call a set of Modules forming a self-sufficient simulation/control system a *simulation subnet*, or SSN.) A supervisory Module can, however, connect its IVs and OVs to appropriately typed Variables within the SSN, allowing it to monitor and alter the behavior of Modules within the SSN. A supervisory Module may also have *spies*, which are Modules that are updated each SSN update cycle, but whose removal would not affect the SSN's behavior.

The GeneticAlgorithmSupervisorModule contains a ProblemSpecification object, and old and new Population objects. A Population object contains a vector of Individuals, each of which holds a chromosome and has functions for selection and breeding of new Individuals. A ProblemSpecification object contains a vector of Alleles, each of which pertains to one of the InputVariables that are collectively being optimized. An Allele contains information for converting between the IV value (the *phenotype*) and the sequence of bits that represent it in the chromosome (the *genotype*). It also contains an OutputVariable, which holds a phenotype value for that Allele as its current value. The OV of each Allele in the ProblemSpecification is connected to the corresponding IV, in the simulation subnet, that is being optimized.

The FitnessEvaluationSpyModule's job is to perform a cumulative fitness assessment throughout each trial. It supports fitness functions defined by three equations, specified as strings, of the IV values and time. The fitness is the sum of three terms: the integral over the course of the trial of the first equation, the maximum value taken on by the second, and the minimum value taken on by the third. Upon each update, the spy writes the fitness thus far into an OV. At the end of each trial, the GASupervisorModule reads the per-trial fitness value calculated by the spy and merges it into its calculation of the Individual's total fitness. SoftLab also provides Scheme commands to define the bitstring length and genotype/phenotype conversion. Thus, the problem-specific aspects of the genetic algorithm can be specified entirely through interpreted scripts, without needing to compile or link.

Acknowledgments

This work was supported in part by NASA contract NCCW-0087, NASA grant NCC5-350 under PURSUE, and the New Mexico Space Consortium at the University of New Mexico.

References

L. Desjarlais and C. Wright, "SoftLab: A Soft Computing Framework for Control System Design, Simulation, and Optimization." *1999 ACE/PURSUE Joint Student Conference*, April 19-20, Albuquerque, NM.

L. Desjarlais and C. Wright, "Patterns for Control System Optimization Using Genetic Algorithms in SoftLab." *Ibid*.