An Examination of Hypermutation and Random Immigrant Variants of mrCGA for Dynamic Environments

Gregory R. Kramer and John C. Gallagher

Department of Computer Science and Engineering
Wright State University, Dayton, OH, 45435-0001
{gkramer, johng}@cs.wright.edu

1 Introduction

The mrCGA is a GA that represents its population as a vector of probabilities, where each vector component contains the probability that the cooresponding bit in an individual's bitstring is a one [2]. This approach offers significant advantages during hardware implementation for problems where power and space are severely constrained. However, the mrCGA does not currently address the problem of continuous optimization in a dynamic environment. While, many dynamic optimization techniques for population-based GAs exist in the literature, we are unaware of any attempt to examine the effects of these techniques on probability-based GAs. In this paper we examine the effects of two such techniques, hypermutation and random immigrants, which can be easily added to the existing mrCGA without significantly increasing the complexity of its hardware implementation. The hypermutation and random immigrant variants will be compared to the performance of the original mrCGA on a dynamic version of the single-leg locomotion benchmark.

2 Dynamic Optimization Variants of mrCGA

The hypermutation strategy, proposed in [1], increases the mutation rate following an environmental change and then slowly decreases it back to its original level. For this problem the hypermutation variant was set to increase the mutation rate from 0.05 to 0.1. Random immigrants is another strategy that diversifies the population by inserting random individuals [4]. Simulating the insertion of random individuals is accomplished in the probability vector by shifting each bit probability toward its original value of 50%. For this problem the random immigrants variant was set to shift each bit probability by 0.12. To ensure fair comparisons between the two variants, the hypermutation rate, and the bit probability shift were empirically determined to produce roughly the same divergence in the GA's population.

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 454–455, 2003.

3 Testing and Results

The mrCGA and its variants were tested on the single-leg robot locomotion problem. The goal for this problem is to evolve a five neuron CTRNN (Continuous Time Recurrent Neural Network) controller that allows the robot to walk forward at optimal speed. Each benchmark run consisted of 50,000 evaluation cycles with the leg's length and angular inertia changed every 5,000 evaluation cycles. The algorithms were each run 100 times on this problem. Performance was evaluated by examining the quality of the final solution achieved prior to each leg model change. A more formal examination of the single-leg locomotion problem can be found in [3].

Comparisons between the mrCGA, hypermutation, and random immigrant results show that the best solutions are achieved by the hypermutation variant. The average pre-shift error for the mrCGA is 18.12%, whereas the average pre-shift error for the hypermutation variant shows a 2.27% decrease to 15.85%. In contrast, the random immigrant variant performed worse than mrCGA, with a 4.18% increase in error to 22.30%.

4 Conclusions

Our results show that for the single-leg locomotion problem, hypermutation increases the quality of the mrCGA's solution in a dynamic environment, whereas the random immigrant variant produces slightly lower scores. Both of these variants can be easily added to the existing mrCGA hardware implementation without significantly increasing its complexity. In the future we plan to categorize the effects of the hypermutation and random immigrant strategies on the mrCGA for a variety of generalized benchmarks. This categorization will be useful to help determine which dynamic optimization strategy should be employed for a given problem.

References

1. Cobb, H.G. (1990) *An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments.* Technical Report AIC-90-001, Naval Research Laboratory, Washington, USA.
2. Gallagher, J.C. & Vigraham, S. (2002) *A Modified Compact Genetic Algorithm for the Intrinsic Evolution of Continuous Time Recurrent Neural Networks.* The Proceedings of the 2002 Genetic and Evolutionary Computation Conference. Morgan-Kaufmann.
3. Gallagher, J.C., Vigraham, S., & Kramer, G.R. (2002) *A Family of Compact Genetic Algorithms for Intrinsic Evolvable Hardware.*
4. Grefenstette, J.J. (1992) *Genetic algorithms for changing environments.* In R. Maenner and B. Manderick, editors, *Parallel Problem Solving from Nature 2*, pages 137–144. North Holland.