Improving Genetic Algorithm Performance in the Majority Problem

Nicola Mc Donnell, Enda Howley, Jim Duggan
National University of Ireland, Galway
nixmcd@gmail.com, {enda.howley, jim.duggan}@nuigalway.ie

Keywords: Cellular Automata, Majority Problem, Evolutionary Computation, Genetic Algorithm

Abstract

Evolutionary Computation can evolve rules to solve the cellular automata majority problem, however, published results show that genetic algorithms (GA) performed sub-optimally compared with the human-designed rules. This paper demonstrates that GA performance can improve with a larger number of agents, and where the test cases used in the fitness evaluation is more challenging with later generations.

1. Introduction

A binary elementary Cellular Automata (CA) is an array of cells, where each cell is either '*on*' or '*off*'. The automata have a restricted view of the world and can only observe the states of the other automata within their neighborhood. At each time-step the automata, observing its neighborhood, activate a rule to decide what their next state should be. At any time, the λ value of a CA is the fraction of automata that are '*on*'.

In the majority problem (MP) the challenge for the CA is to decide if the majority of cells are on or off. This is a non-trivial as only a cell's neighborhood is visible.

In the classical MP problem there are 149 automata in the grid and the neighborhood size is 3 [1,2,3,4]. To evaluate how good a rule is at solving the MP, it's run against a set of tests with random initial configurations (ICs). When generating ICs for the tests, if each bit is selected randomly, the resulting ICs will have a λ distribution with a mean of 0.5 and standard deviation of 0.04; this is referred to as an un-biased distribution.

2. Previous Work

There are several effective human designed rules for the MP. The Gacs-Kurdyumo-Levin (GKL) rule [3], proposed in 1978, has an accuracy of 81.6%. In 1995, Davis and Das [2], improved upon the GKL rule giving accuracy of 81.8% and 82.178%, respectively.

Bio-inspired search techniques from Evolutionary Computing have been applied. Mitchell [1] and Crutchfield [4] used Genetic Algorithm (GA). Their best reported rule had an accuracy of 77.5%. Andre [2] used Genetic Programming (GP) to evolve a rule with reported accuracy of 82.326%.

The GA used in Mitchell and Crutchfield's work evolved a population of 100 different rules for 100 generations. However, the ICs had a biased distribution of λ; it was uniformly distributed between 0.0 and 1.0. The initial genomes for the GA also had this biased distribution. Mitchell [1] reported that preliminary experiments indicated a need for this biased distribution so that the GA makes progress in early generations. In explaining why the GA did not perform as well, Mitchell [1] commented that the biased distribution turns out to impede the GA in later generations, because as increasingly fitter rules are evolved, the IC sample becomes less and less challenging for the GA. Andre [2] observed that the lower GA performance may have been the consequence of factors such as small populations size, and this finding is examined in our approach.

3. Experimental Results

Our results, summarized in Figure 1, show a comparison of the original GA having 100 agents, with it having 1,000 agents and having 1,000 agents where the λ distribution of the ICs is increased from the original biased distribution to an unbiased distribution.

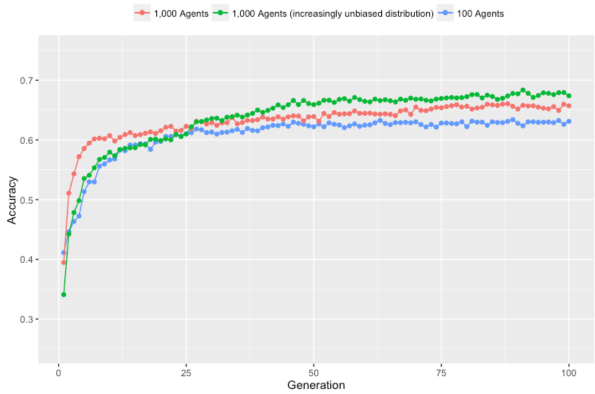

Figure 1 Comparison of GA with 100 and 1,000 agents.

These results demonstrate that the GA's success is very sensitive to how it is set up. In our future work will explore using online feedback to configure and guide the GA.

4. References

[1] Mitchell, Melanie, James P. Crutchfield, and Peter T. Hraber. "Evolving cellular automata to perform computations: Mechanisms and impediments." *Physica D: Nonlinear Phenomena* 75.1 (1994): 361-391.

[2] Andre, David, Forrest H. Bennett III, and John R. Koza. "Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem." *Proceedings of the 1st annual conference on genetic programming.* MIT Press, 1996.

[3] Gács, Péter, Georgy L. Kurdyumov, and Leonid A. Levin. "One-dimensional uniform arrays that wash out finite islands." *Problemy Peredachi Informatsii* 14, no. 3 (1978): 92-96.

[4] Crutchfield, J. P., Mitchell, M., & Das, R. (2003). Evolutionary design of collective computation in cellular automata. *Evolutionary dynamics—exploring the interplay of selection, neutrality, accident, and function*, 361-411.