A Simple Genetic Algorithm for Reducible Complexity

Lee Graham
School of Computer Science
Carleton University
1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6, Canada
+1-613-520-4333
lee@stellaralchemy.com

Steffen Christensen
School of Computer Science
Carleton University
1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6, Canada
+1-613-520-4333
schriste@scs.carleton.ca

Franz Oppacher
School of Computer Science
Carleton University
1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6, Canada
+1-613-520-4333
oppacher@scs.carleton.ca

ABSTRACT

Challenges are often lofted to explain how gradualistic evolution can evolve systems where function ceases with the removal of any of their multiple parts. We present a genetic algorithm (GA) example using a dynamic fitness function. Given clear definitions of relevant terms, the GA produces such complex systems.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods.*

General Terms

Algorithms, Theory.

Keywords

Genetic Algorithms, Evolution, Complexity

1. INTRODUCTION

In this work we show the Darwinian evolution of systems composed of multiple parts contributing to a function such that all parts are critical. Such systems have been dubbed “irreducibly” complex (IC), implying that they cannot be produced by evolution. The associated pseudoscientific baggage has been adequately addressed elsewhere, and is not further addressed here. From an evolutionary perspective, IC systems may be of interest since they often deviate from the canonical graded progression from random beginnings to an optimum in a static fitness landscape. Several mechanisms for the evolution of IC are found in [1]. Although [1] deals with biology, the carry-over to evolutionary computing (EC) is often meaningful. The GA described here evolves IC via one of these mechanisms.

2. THE GA

Our GA uses a small population of fifty individuals. The genotype is a variable-length sequence of codons which are read in sequence when mapping to the phenotype. The phenotype is a 2D pattern of pieces (or “boxes”) on a game board of thirty rows and fourteen columns. Each codon encodes a row index, a column index, and the type of box to place there (there are four types). Fitness evaluation passes a virtual game “ball” through the board. It carries associated points and is copied, modified, and redirected by boxes along the way. Fitness is calculated by tallying points collected in the game and subtracting a parsimony-related penalty value. The penalty value adjusts game difficulty over time. The GA is described in more detail in [2]. Individuals with non-positive fitness are considered inviable and not eligible to become parents.

The GA checks each viable individual to determine if it is IC. An individual is considered IC if, firstly, it contains multiple parts (boxes), in this case a minimum of five was chosen arbitrarily. Secondly, removal of any box from the phenotype must result in inviability.

3. RESULTS

The GA was run for 600 independent runs of 5,000 generations each. A number of population statistics were recorded at regular twenty-generation logging intervals. In particular, the percentage of viable individuals that were also IC was logged. This percentage, averaged over all runs, begins at zero and climbs rapidly to ~90% by generation 1,000. From there it slowly approaches 100%. Thus, the GA quickly reaches a state of IC near-saturation in the population. Logs of the self-adjusting game penalty value show that the population remains mostly IC while continuing to adapt to ever-increasing game difficulty.

4. CONCLUSION

The evolution of IC presents somewhat atypical scenarios for EC. Many IC-generating mechanisms are known [1]. The GA presented here uses a simulated medium as opposed to a biological one, and a change-of-function mechanism. The change is not an abrupt one, as might be expected in cases involving exaptation, but is instead a gradual change of difficulty.

A problem that arises regarding IC is the defining of terms. Definitions of “system”, “function”, and “part” are generally flexible enough to allow much semantic goal post shifting. By operationalizing these terms for the present GA, we avoid this issue. We define a “system” to be a complete phenotype, a “function” to be attainment of non-zero fitness, and a “part” to be a single box placed on the board. Under these definitions, the GA produces IC systems with upwards of one hundred parts.

5. REFERENCES

[1] The TalkOrigins Archive. *http://www.talkorigins.org*.

[2] Graham, L., and Oppacher, F. Speciation through Exaptation. In *Proceedings of IEEE ALife’07*. 2007.

GECCO’07, July 7–11, 2007, London, England, United Kingdom
ACM 978-1-59593-697-4/07/0007.