Why GAs are Hard to Use

Chris Thornton
Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QN
UK
Email: Chris.Thornton@cogs.susx.ac.uk
WWW: http://www.cogs.susx.ac.uk/users/christ/index.html
Tel: (44)1273 678856

December 20, 1995

Abstract

Genetic Algorithms (GAs) are increasingly used for such purposes as deriving programs [1] and producing designs for robots [2]. According to the building-block hypothesis and schema analysis of Holland [3] the GA is an efficient search method. However, empirical work has shown that in some cases the method is outperformed by simpler processes such as random-permutation hill climbing [4] and [5]. The present paper reexamines Holland's framework (as formulated by Goldberg [6]) and finds that such in-practice failures are effectively predicted by the schema analysis. The high efficiency of the GA method is commonly attributed to its 'implicit parallelism'. However, this efficiency is hard to realise because there is a deep contradiction between the building-block hypothesis and the schema theorem.

1 Introduction: natural and simulated evolution

In natural evolution, populations of individuals compete to survive and reproduce. Relatively fit individuals survive longer and thus reproduce more. Over time, the fitter examples of random variations accumulate and average fitness tends to increase. According to the Darwinian theory, this process of **natural selection** is responsible for the development of all life forms on earth. Researchers hope to harness its power for computational purposes by implementing simulations of the process. In these simulations, the individuals are candidate solutions to some problem and fitness is a measure of solution quality. The aim is thus to 'evolve' high-quality solutions through simulated natural selection.

A common way of pursuing this approach involves use of the **crossover-based genetic algorithm** or **C-GA** [6]. In this approach, candidate solutions are represented as strings of characters or **genotypes**. Reproduction involves the production of a new individual through the splicing together of genotypes from two 'parents'. Parent genotypes are split at a certain point, forming a left part and a right part. The right part from one parent is then joined to left part from other, and vice versa. This produces two offspring genotypes which then replace relatively unfit individuals from the population.

2 Schema analysis

At first sight, the C-GA appears to be a way of randomly exploring the space of possible genotypes. However, Holland's **schema analysis** [3] provides an alternative picture. In this analysis we assume that the GA is a way of processing genotype *features* rather then genotypes themselves — a feature being simply a set of values in specific positions. A particular feature is defined in terms of a **schema**. This is a genotype-like string with specific values in some positions and 'don't care' values (asterisks) in others. An example is

```
*10**0****
```

pure reproduction,

5 The shortness and low-epistasis assumptions

According to the schema theorem, schemas will only be processed correctly if they are of ‘short’ defining length. In fact a schema will only be processed correctly if its fitness advantage (the excess of its fitness over the average fitness) is greater than its ‘vulnerability’ — the ratio of its defining length to the genotype length. How easily is this ‘shortness’ assumption satisfied?

independently attributable to individual parts of the genotype, but only to their interactions.

6 The building block hypothesis

The credibility of the C-GA does not rest solely on the schema theorem. It also rests on the so-called **building-block hypothesis**. This states that the crossover GA works well when short, low-order, highly fit schemas recombine to form even more highly fit, higher-order schemas. In fact, as Forrest and Mitchell [4] note, 'the ability to produce fitter and fitter partial solutions by combining blocks is believed to be the primary source of the GA's search power.' Unfortunately, when we come to examine the assumptions introduced by the building-block hypothesis, we find that they directly contradict those introduced by the schema theorem.

The building-block hypothesis assumes that the fitness of any one block is typically affected by the other blocks on the genotype. If this were not the case it would be meaningless to talk about a 'building-block process' operating over and above the usual evolutionary process. Thus the building-block hypothesis implicitly assumes a situation of high epistasis and thus contradicts the low-epistasis assumption introduced by the schema theorem.

When we come to consider the length implications of the building-block hypothesis we uncover a further contradiction. During the building-block process, the schemas that require processing at any given stage are actually the blocks that have been put together by the prior building-block process. Except at the assumptions introduced by the schema theorem. The situation is illustrated schematically in Figure 1. Given the importance of the building-block hypothesis within the GA paradigm this clash of assumptions occurring at the most fundamental level of the analysis is of special interest. As Forrest and Mitchell [4] have commented there is a 'need for a deeper theory of how low-order building blocks are discovered and combined into higher-order schemas.'

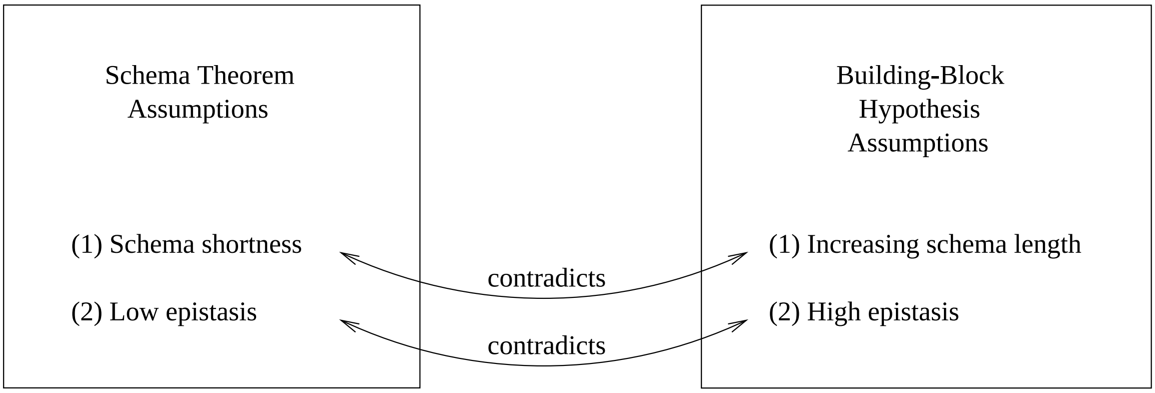

Figure 1: Inherent contradictions in the schema/building-block framework.

7 Summary

As Forrest and Mitchell have noted, confidence in the efficacy of the GA is still largely based on the building-block hypothesis and the schema theorem. The schema theorem shows that schemas with high fitness are given exponentially increasing numbers of trials through rep9.43s 120.4ink.75912Td[d 120.e4t16000

References

[1] Koza, J. (1992). *Genetic Programming: on the Programming of Computers by Means of Natural Selection.* Cambridge, Massachusetts: MIT Press.

[2] Cliff, D., Harvey, I. and Husbands, P. (1993). General visual robot controller networks via artificial evolution. In D. Casement (Ed.), *Proceedings of the Society of Photo-optical Instrumentation Engineers Conference (SPIE-93), Session on Intelligent Robots and Computer Vision XII: Algorithms and Techniques.* Boston MA.

[3] Holland, J. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor: University of Michigan Press.

[4] Forrest, S. and Mitchell, M. (Forthcoming). Relative building-block fitness and the building-block hypothesis. In D. Whitley (Ed.), *Foundations of Genetic Algorithms 2.* San Mateo, CA: Morgan Kaufmann.

[5] Lang, K. (1995). Hill climbing beats genetic search on a boolean circuit synthesis task of koza's. *Proceedings of The Twelfth International Conference on Machine Learning* (pp. 340-343).

[6] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

[7] O'Reilly, U. (Forthcoming). An analysis of genetic programming. PhD Thesis.