Genetic Algorithms: Emergent Behavior in Search and Optimization

Lyle Longley
Under the advisement of Tim Melvin, PhD

Abstract
Evolution helps organisms adapt to their environment, providing complex solutions through seemingly random combinations and permutations of nucleotides. With so much genetic material, the realm of possibilities for unique genetic sequences is practically infinite. How does genetic evolution find sufficient adaptations in such a vast search space? Genetic algorithms (GAs) were created in an effort to model the processes driving evolution.

GAs have since found practical use both in scientific modelling as well as in optimization and problem solving for various engineering applications. By extension with other logical processes GAs can be applied across wide variety of disciplines and problem types. Evolving unique computer programs, or neural network architectures, and predicting protein structures, the stock market, or the weather are all possible. The field of GA use and study is still developing, with researchers coining improved performance out of the GAs as well as finding new ways to use them.

Complex Adaptive Systems
GAs are part of the field of science known as complex adaptive systems. Properties of complex adaptive systems:

- Search space - Finding solutions to problems for which there is a near infinite number of possible solutions, of differing suitability. The realm of possible solutions is often visualized as a horizontal plane.
- Building blocks - Agents in the system organize into independent, self-reinforcing units.
- Emergence - "Building blocks" coalesce to form new, more complex building blocks.
- Hierarchical complexity - The complex organization of "building blocks" at one level form the basis for new, higher order levels of complexity, dependent upon subordinate "building blocks". Examples: atoms combine in "building blocks" at one level form the basis for new, more complex building blocks.
- Agents in the system organize into independent, self-reinforcing units.
- Emergence - "Building blocks" coalesce to form new, more complex building blocks.
- Hierarchical complexity - The complex organization of "building blocks" at one level form the basis for new, higher order levels of complexity, dependent upon subordinate "building blocks". Examples: atoms combine in "building blocks" at one level form the basis for new, more complex building blocks.

"Good Enough"
It is important to note that GAs often do not find the global optimum solution, instead converging on a local optimum. The dynamic between the stochastic nature of GA search and the fitness function driving that search, moves the GA toward a "good enough" solution as effectively as possible. If searching for a global optimum, a combination of GAs and gradient search methods can be very effective.

Anatomy of a GA
- Fitness landscape - The Fitness landscape is the same as the search space, but the fitness of each solution is represented on the vertical axis. Fitness landscapes can be smooth (1) or rugged (2).
- Fitness function - An integral piece of the genetic algorithm, the fitness function tests the chromosomal strings and assigns a score based on their suitability.
- Population - The collection of chromosomal strings being processed by the algorithm.
- Generation - An iteration of the GA consisting of one population which is evaluated, then selected for reproduction, the product of which would be the subsequent generation.
- Chromosomal strings - Possible solutions are encoded in a string. This string can be binary (10010110), or it can consist of variables which represent higher-order ideas such as instructions in a procedure.
- Genetic Algorithms - Emergent Behavior in Search and Optimization
- Historical, antenna design for space missions was a time and labor intensive process.

Application
The most recent advance in GAs has been the introduction of the adaptive genetic algorithm. Instead of having fixed probabilities of crossover and mutation, adaptive GAs vary these parameters in real time based on patterns observed while the GA is running. This allows the GA to introduce more genetic diversity to the population, avoiding convergence to a less fit local optimum.

As you can see, at less than 50 generations the fitness of the adaptive GA population rapidly increases, converging to a much higher fitness than the standard GA.

In addition, parameters such as population size, chromosome length and others will also change the search dynamics for the better or worse. The operable interpretation of behavior and corresponding manipulation of functional parameters is an area ripe for future research.

Interpretation of GA Behavior
- While conceptually simple in design, the emergent usefulness of GAs is the result of a dynamic between abstract internal forces shaping the GA's behavior.
- Schema - patterns in a chromosomal string which can be identified by two outer defined elements. An example using a binary string: 11111, 1000001, and 10101 are all subsets of the schema, where * represents undefined bits.
- Defining length - The distance between the outermost defined elements of a schema, in the above example, is 6.
- Order - The number of defined bits in a schema, the above example is order of 2.
- A Schema is like a cooperating group of genes, with combinations of alleles expressing phenotype of the system. As the schema converges toward higher fitness, the number of defined bits increases. This is because bit configurations of higher fitness become "locked in" to the fitness of solutions.
- The Schema Theorem: For simple GAs, The Schema Theorem explains how short, low order schema with average fitness above the mean for the overall population, will proliferate exponentially. It is important for understanding the inner tension of the GA, pitting the security of relying upon tried-and-true schema against the risk of failing to explore new areas of the search space, potentially finding schema of even higher fitness.
- The Schema Theorem:
  - $S_a = 1 - \text{probability of mutation}$
  - $S_a = 1 - \text{definition length of the schema}$
  - $S_a = 1 - \text{expected fitness of any schema in next generation}$
  - $S_a = 1 - \text{expected fitness of any schema in next generation}$

Acknowledgements
Thanks to Tim Melvin for being a thoughtful, enthusiastic advisor. Also, thanks to Monica Acosta for her guidance, as well as to Draci Rosales and MESA for the opportunity to participate in this project.

References

Figure (1): A smooth fitness landscape showing a global maxima and local maxima.
Figure (2): A rugged fitness landscape
Figure (3): A flowchart of the logical structure of a GA. Although relatively simple in form, complex behavior emerges from the algorithm.
Figure (4): The likelyhood a schema will survive mutation (Sm) or crossover (Sc)
Figure (5): A schema is of order 2. The number of defined bits in this schema is 2.
Figure (6): ST5 antenna - The first GA designed hardware in space
Figure (7): Compares the convergence and fitness of solutions provided by a standard GA (SGA) and an adaptive GA (AGA)

Continued Development
Historically, antenna design for space missions was a time and labor intensive process. Engineers with a vast amount of expertise would spend months building and testing prototypes. For NASA's ST5 mission, the antenna design engineers used GAs to accelerate the design timeline. The antenna produced became the first computer-evolved antenna, surpassing the performance of any human-designed antenna.