Genetic Algorithm
By Ben Shedlofsky

Executive Summary

The Genetic algorithm is a way using computers to figure out how a population evolves genetically. These understandings of how a population evolves towards a more healthy population; as well as how it evolves through breeding purposes. Each generation develops a new offspring and some offspring converge towards a healthier population and some don't at all. This algorithm helps figure out in mathematical terms how a population breeds through a computer simulation.

The major problem was understanding step by step how a Genetic algorithm works and how it doesn't work. The other problem was figuring out to present it to the audience without the audience not understanding what I presented or some magic show of how it works.

In a computational format the Genetic Algorithm uses binary or base 2 instead of using base 10. This makes it a little harder to read and understand from a business end user. Once you understand that there are just 1's and 0's instead of 0 thru 9 digits, you start to recognize that this uses a computational system to derive populations.

Breeding for each generation happens either by two parents or by one parent. The two parents are called crossover and usually happen the most. While the one parent or mutation usually happen very low percentage. This is a very similar makeup to a biological population like monkey breeding amongst themselves or single cell organisms breeding amongst themselves too. Each one of us has genes or trait that makes us who we are and these differences in our genes or traits make us unique to the individual. For example, the color of your skin or another example is if you wear glasses or not makes us unique in many ways. These traits we have define us from our parents and grandparents and make us who we are because this biological makeup defines us as human beings.

I realized that this curiosity of looking at data to define how biological populations breed has been going on since 1954 and continued to this day. Once the desktop computer came out; it made the use of doing this computational algorithm a lot easier to handle. This in turn gave way to pre-built software that helps make this will lead to more discoveries about the Genetic Algorithm.

Problem Description

The problem was figuring out how to educate people on the Genetic Algorithm. There were different subjects I touched upon, but I had to figure out in a general order how to present the Genetic Algorithm. The reason being so that everyone understands how the Genetic Algorithm works because the problem is a lot of people don't get how a population evolves in a computational format.

Analysis Technique/Results

Genetic algorithms (GAs) attempt to mimic computationally the processes by which natural selection operates, and apply them to solve business and research problems. Developed by John Holland in the 1960s and 1970s genetic algorithms provide a framework for studying the effects of such biologically inspired factors as mate selection, reproduction, mutation, and crossover of genetic information. In the natural world, the constraints and stresses of a particular environment force the different species (and different individuals within species) to compete to produce the fittest offspring. In the world of genetic algorithms, the fitness of various potential solutions are compared, and the fittest potential solutions evolve to produce ever more optimal solutions. (Larose, 240)

Not surprisingly, the field of genetic algorithms has borrowed heavily from genomic terminology. Each cell in our body contains the same set of chromosomes, strings of DNA that function as a blueprint for making one of us. Then each chromosome can be partitioned into genes, which are blocks of DNA designed to encode a particular trait, such as eye color. A particular instance of a gene (e.g., brown eyes) is an allele. Each gene is to be found at a particular locus on the chromosome. Recombination, or crossover, occurs during reproduction, where a new chromosome is formed by combining the characteristics of both parents' chromosomes. Mutation, the altering of a single gene in a chromosome of the offspring, may occur randomly and relatively rarely. The offspring's fitness is then evaluated, either in terms of viability (living long enough to reproduce) or in the offspring fertility. (Larose, 240)

In the field of genetic algorithms, a chromosome refers to one of the candidate solutions to the problem, a gene is a single bit or digit of the candidate solution, an allele is a particular instance of the bit or digit (e.g., 0 for binary-encoded solutions or the number 7 for real-valued solutions). Recall that binary numbers have base 2, so that the first "decimal" place represents "ones," the second represents "twos," the third represents "fours," the for represents "eights," and so on.

So the binary string 10101010 represents

$(1 * 128) + (0*64) + (1 * 32) + (0 * 16) + (1 * 8) + (0 * 4) + (1 * 2) + (0 * 1) =$ 170. (Larose, 241)

A genetic algorithm requires two things to be defined:

1. a genetic representation of the solution domain.
2. a fitness function to evaluate the solution domain (Wikipedia, 2)

A fitness function is a genetic representation and measures the quality of the represented solution. It is always problem dependant. GA then proceeds to initialize a population of solutions randomly, then improve it through repetitive application of mutation, crossover, inversion and selection operators. (Wikipedia, 2)

Selection refers to the method used for selecting which chromosomes will be reproducing. The fitness function evaluates each of the chromosomes (candidate solutions), and the fitter the chromosome, the more likely it will be selected to reproduce. (Larose, 241)

Crossover performs recombination, creating two new offspring by randomly selecting a locus and exchanging subsequences to the left and right of the locus between two chromosomes chosen during selection. For example, in binary representation, two strings 11111111 and 00000000 could be crossed over at the sixth locus in each to generate the two new offspring 1111000 and 00000111. (Larose, 241)

Here is an example of a crossover:

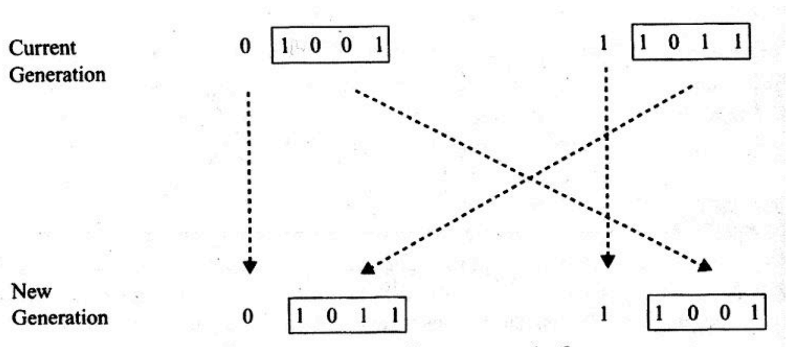

Figure 6.2 Performing crossover at locus two on the first two parents.

Mutation randomly changes bits at a particular locus in a chromosome. Mutation has a very small probability. For example, after crossover the 11111000 child string is mutated at locus two to become 10111000. This introduces new information to the genetic pool and protects against converging too quickly to a local optimum. (Larose, 241)

Most genetic algorithms function by interactively updating a collection of potential solutions called a population. Each member of the population is evaluated for fitness on each cycle. A new population then replaces the old population using the operators above, with the fittest members being chosen for reproduction or cloning. (Larose, 241)

The process of a Genetic Algorithm starts first with the initialization. Individual solutions randomly generated from initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. (Wikipedia, 2)

Next step is the selection process. During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selections rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming. (Wikipedia, 2)

Then you got reproduction as the next step. I discussed already about mutation versus crossover.

The final step is termination. A solution is found that satisfies minimum criteria. Fixed number of generation reached. Allocated budget (computation time/money) is reached. The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results. The last choice is manual inspection or just combinations of all the above choices.

The pro observations for the Genetic Algorithm are as follows. GA often locates good solutions. This is an effective heuristic when dealing with a very large solution space. Mutation introduces new information to the genetic pool and protects against converging too quickly to a local optimum.

The Cons observations for the Genetic Algorithm are as follows. There can be time delay when executing the program. GA tends to converge towards local points rather then global points. Operate dynamic data sets is difficult and may prevent early converge towards solution. Specific optimization problems because simpler optimization algorithms. There are better solutions than genetic algorithms. GA cannot effectively solve problems which only the fitness measure is right/ wrong. No way to converge on a solution.

Here is some more information about the Genetic Algorithm. Fitness function is important factor for speed and efficiency of the algorithm. Selection is important genetic operator; as well as the importance between crossover versus mutation.

Here is an example of a genetic algorithm:

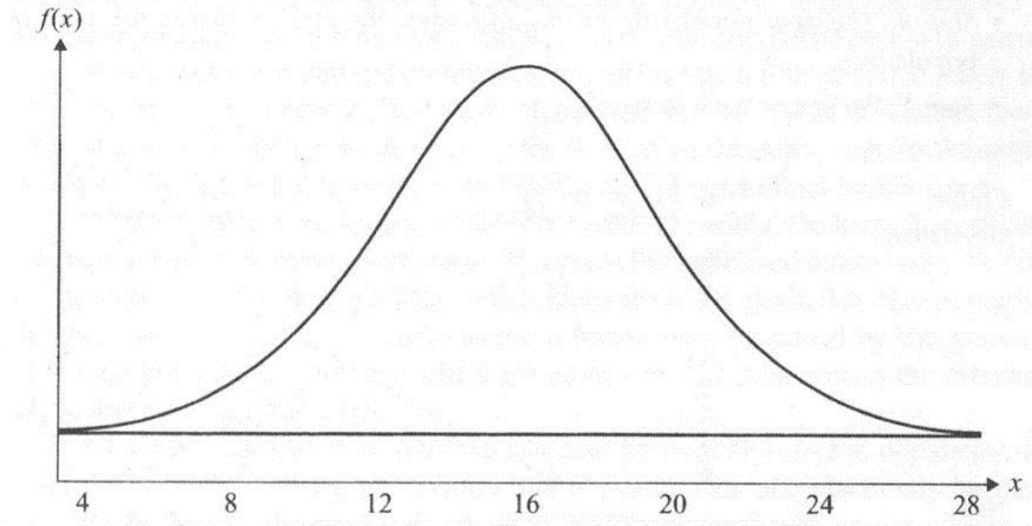

Figure 6.1 Finding the maximum value of the normal **(16, 4)** distribution.

This is a graph of a genetic algorithm.

SIMPLE EXAMPLE OF A GENETIC ALGORITHM AT WORK

Let's examine a simple example of a genetic algorithm at work. Suppose that our task is to find the maximum value of the normal distribution with mean $\mu = 16$ and standard deviation $\sigma = 4$ (Figure 6.1). That is, we would like to find the maximum value of

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[\frac{-1}{2\sigma^2}(X-\mu)^2\right] = \frac{1}{\sqrt{2\pi}\,(4)} \exp\left[\frac{-1}{2(4)^2}(X-16)^2\right]$$

The standard deviation is 4 and the normal distribution is 16. This is an example of a genetic algorithm equation.

We allow X to take on only the values described by the first five binary digits, that is, 00000 through 11111, or 0 to 31 in decimal notation.

First Iteration

- *Step 0: Initialization.* We define the *crossover rate* to be$p_c = 0.75$ and the *mutation rate* to be $p_m = 0.002$.
- *Step 1:* Our population will be a set of four chromosomes chosen randomly from the set 00000 – 11111. So $n = 4$ and $l = 5$. These are *00100* (4), *01001* (9), *11011* (27), and *11111* (31).
- *Step 2:* The fitness $f(x)$ is calculated for each chromosome in the population (Table 6.1).
- *Step 3:* Iterate through the following steps until n offspring have been generated.
  - *Step 3a: Selection.* We have the sum of the fitness values equal to

$$\sum_i f(x_i) = 0.001108 + 0.021569 + 0.002273 + 0.000088 = 0.025038$$

Then the probability that each of our chromosomes will be selected for parenthood is found by dividing their value for $f(x)$ by the sum 0.025038. These

TABLE 6.1 Fitness and Probability of Selection for Each Chromosome

Chromosome	Decimal Value	Fitness	Selection Probability
00100	4	0.001108	0.04425
01001	9	0.021569	0.86145
11011	27	0.002273	0.09078
11111	31	0.000088	0.00351

are also shown in Table 6.1. Clearly, chromosome *01001* gets a very large slice of the roulette wheel! The random selection process gets under way. Suppose that chromosome *01001* and *11011* are selected to be the first pair of parents, since these are the two chromosomes with the highest fitness.

- *Step 3b: Crossover.* The locus is randomly chosen to be the second position. Suppose that the large crossover rate of p_c, 0.75, leads to crossover between *01001* and *11011* occurring at the second position. This is shown in Figure 6.2. Note that the strings are partitioned between the first and second bits. Each child chromosome receives one segment from each of the parents. The two chromosomes thus formed for the new generation are *01011* (11) and *11001* (25).
- *Step 3c: Mutation.* Because of the low mutation rate, suppose that none of the genes for *01011* or *11001* are mutated. We now have two chromosomes in our new population. We need two more, so we cycle back to step 3a.
- *Step 3a: Selection.* Suppose that this time, chromosomes *01001* (9) and *00100* (4) are selected by the roulette wheel method.
- *Step 3b: Crossover.* However, this time suppose that crossover does not take place. Thus, clones of these chromosomes become members of the new generation, *01001* and *00100*. We now have $n = 4$ members in our new population.

• *Step 4.* The new population of chromosomes therefore replaces the current population.

• *Step 5.* We iterate back to step 2.

TABLE 6.2 Fitness and Probability of Selection for the Second Generation

Chromosome	Decimal Value	Fitness	Selection Probability
00100	4	0.001108	0.014527
01001	9	0.021569	0.282783
01011	11	0.045662	0.598657
11001	25	0.007935	0.104033

Second Iteration

• *Step 2:* The fitness $f(x)$ is calculated for each chromosome in the population (Table 6.2).

- *Step 3a: Selection.* The sum of the fitness values for the second generation is $\sum_i f(x_i) = 0.076274$, which means that the average fitness among the chromosomes in the second generation is three times that of the first generation. The selection probabilities are calculated and shown in Table 6.2.

We ask you to continue this example in the exercises.

This is a step by step process of an example of a genetic algorithm.

An example of a pseudo code for a genetic algorithm is:

```
1. Choose initial population
2. Evaluate the fitness of each individual in the population
3. Repeat
        1.  Select best-ranking individuals to reproduce
        2.  Breed new generation through crossover and mutation (genetic operations) and give birth to offspring
        3.  Evaluate the individual fitnesses of the offspring
        4.  Replace worst ranked part of population with offspring
4. Until Termination
```

The history of the genetic algorithm started with computer simulations of evolution started in 1954 with the work of Nils Aall Barriclelli. In 1957 Austrailian geneticist Alex Fraser published a series of papers on simulation of artificial selection on organisms with multiple loci control a measurable trait. Then in the early 1960s there were computer simulations of evolution by biologists. The methods of the genetic algorithm were described in books by Fraser, Burnell and Crosby. Fraser's simulations included all the essential elements of modern genetic algorithms. Hans Bremermann published papers in the 1960s adopting population of solution to optimization problems, undergoing recombination, mutation, and selection. This also included elements of modern genetic algorithms.

Artificial evolution became widely recognized optimization method as a result work from Ingo Rechenberg and Hans-Paul Schwefel in the 1960s and early 1970s. They solved complex engineering problems through evolution strategies. Genetic algorithms became popular through work of John Holland in the early 1970s. John Holland's book Adaptation in Natural and Artificial Systems (1975) work originated with studies in cellular automata. He and his students conducted these experiments and introduced formalized framework. This framework predicated quality of next generation and became known as Holland's Schema Theorem.

Genetic Algorithm's research remained largely theoretical until the mid 1980s because of The First International Conference on Genetic Algorithms was held. Academic interest grew and increase desktop computational power allowed practical application with new techniques. In the late 1980s General Electric started selling the world's first genetic algorithm product. In 1989 Axelis Inc. made the world's second GA product and first for desktop computers.

Assumptions

The main assumption was how the genetic algorithm works and how it doesn't work. Another assumption was that everyone would understand this easily.

Issues

The main issue was realizing how I would describe this to a general audience.

References

Genetic Algorithm, (n.d.), Retrieved March 22, 2008 from http://en.wikipedia.org/wiki/Genetic_Algorithm

Larose, Daniel T., Data Mining Methods and Models, U.S.A.: John Wiley & Sons, Inc., 2006