A primer on Genetic Algorithms

Genetic Algorithms (Gas) are simplified models of our current understanding of natural selection.

For any species, individuals of that species are likely to differ. As a slightly ridiculous analogy: among giraffes, some individuals may have genes which result in longer necks, compared to equivalent genes that result in shorter necks in their fellow shorter necked giraffes.

As long as there is genetic variation (i.e. difference) in a population, then some members of that population are likely to be more successful (i.e. 'fit') than others. To continue the slightly ridiculous analogy, if a new type of tree colonised a giraffe habitat, and that tree species grew taller than existing tree species, then giraffes with longer necks may be able to graze the taller trees, and therefore have a new food supply denied to shorter necked giraffes. The longer necked giraffes would be 'fitter' to their environment, and since they could obtain more food, be more likely to successfully reproduce and thus 'pass on' their genes for growing long necks. Over time, the population would change from relatively shorter necked giraffes to longer necked ones.

From this very simplified and toy example, it can be seen that genetic variation is required for a population to evolve (i.e. change its overall genetic make-up), but also some type of selection pressure (such a competition for food among individuals in order to survive to reproduce) is required for that evolution to create a 'fitter' population. Much genetic change will result in organisms that are less well adapted (in the same way that randomly altering a rather well optimised machine is more likely to result in a worse machine than a better one). Selection pressure is what ensures that only the rare beneficial change is likely to be kept. Poorly adapted individuals are more likely to be unable to reproduce (e.g. they collect less food, or get eaten, or can't resist diseases).

GAs are usually designed to start with a '*random*' population of genes. They then use an *evaluation function* to determine the 'raw score' of how well each gene 'performs'.

Usually the evaluation function determines some sort of 'cost' that we wish to minimise. In this case we could determine a fitness from a raw score (rawScore) by calculating 1/(rawScore + 1). We added the one to the denominator so we don't get a divide-by-zero error if the value of rawScore is 0. The lower the raw score becomes, the higher the fitness will be.

Normalisation is usually done on the fitness in order to have the total population fitness equal 1, which simplifies the working of the selection function described below. Normalisation simply entails determining the total summed fitness of the entire population, then dividing each gene's fitness by that total.

An individual that performs well is more likely to reproduce. In GAs, as in the wild, there is no guarantee that a fit individual will be chosen (or survive) to reproduce. Accidents can happen to even fit organisms, and genetically 'inferior' organisms can sometimes have luck on their side, but on average fitter individuals are more likely to reproduce, and have more offspring, than less fit individuals. (Allowing such probabilistic selection of individuals is an advantage. If only the fittest individuals were used (every time), then even valuable 'subsets' of genetic material in less fit individuals would be discarded along with the unfortunate individuals themselves. Such brutal destruction of genetic variation tends to make a population 'converge' too quickly to a set of identical clones (and without variation, further evolution is impossible), without exploring the potential of 'mixing' up the variation available in the initial population).

A GA selection method which favours 'fit' individuals is 'fitness proportionate selection' (also known as roulette-wheel selection). The more fit an individual is, the more likely they will be chosen to produce a new individual for the next generation. A simple implementation of this method requires a population to be sorted on normalised fitness, with the fittest individual at the start of the list. A random number is chosen between 0 and 1, and we step through the list summing the individual fitness scores until our sum is no longer less than our random number. The individual which we stop at is the one chosen for reproduction.

To ensure the very fittest individual is not accidently lost (as proportional selection doesn't guarantee that even the fittest individual will ever be selected), usually the very fittest individual is cloned and the clone added to the new population. The remaining N – 1 children in a new population of size N are produced by either mutating a clone of an existing selected parent (usually only a small percentage of children are mutants, 5% of the population is a common GA figure) or producing a child by using crossover on two selected parents. The small amount of mutation allows genetic novelty to be added to the population for evaluation (although most mutations are likely to be 'bad', so we only use it sparingly).

Crossover is the main reproductive operator because of the following heuristic: very roughly described, if two parents had 'successful' genetic material, perhaps combining subsets of the genetic material from each parent would capture the 'best' of both parents. Of course, it is possible the child will get the 'worst' of both parents, in which case selection pressure is very likely to prevent that child from reproducing, and thus eventually 'weed out' the 'bad' genetic material. For those who are interested (and not busy with assignments), the 'building block hypothesis' elaborates on the heuristic behind crossover.

The new child population can then be evaluated, and the 'cycle of life' repeated.