Genetic Programming Made Easy

Reproduction Techniques

Olim F. Tuyt & Philip W.B. Michgelsen

January 18, 2016

Abstract

A genetic algorithm consists of three operations: reproduction, crossover and mutation. All three operations can be carried out in different ways, resulting in different genetic algorithm styles. This short treatise will focus on the operation of reproduction and will illustrate three different techniques to perform this operation. The different techniques require different algorithms and generate mathematical differences to the genetic algorithms incorporating them. To better understand what the consequences are of each of the different reproduction techniques, the mathematical differences will be carefully explained.

1 Introduction

In the words of Goldberg, "Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics." Genetic algorithms are search methods that use ideas from genetics to perform a directed randomized efficient search to an optimum set of data points, in a given set of points, determined by some given *fitness function*. They use the idea of survival of the fittest, among data, combined with a structured yet randomized information exchange, given by the ideas of mutation and crossover, to provide a robust and very efficient optimum search algorithm. A genetic algorithm reproduces a sample of organisms, from a population, in such a way that, without heaving to go through the entire population, the fittest data points can be found.[1]

Genetic algorithms are functions that take as input a (i) sample of a particular population, (ii) a fitness function, (iii) a crossover parameter, (iv) a mutation parameter, and an optional bound on the sizes of each generation; genetic algorithms, then return data points, or organisms, in the population, that are the optimum of the population, according to the given fitness function. Genetic algorithms essentially make use of three operations, reproduction, mutation and crossover, to create new generations from the sample of the population, to create a fittest set of organisms. The operation of reproduction is usually carried out first, creating new organisms from old, by incorporating the idea of survival of the fittest; then the operation of crossover is carried out on some percentage of the population, given by the crossover parameter, usually by swapping the head and tail of two bit strings; finally, the operation of mutation is carried out by flipping a single bit in a string (some organism), according to a (biased) coin flip, based on the mutation parameter. In short, "genetic algorithms start with a random population of n strings, copy strings with some bias toward the best, mate and partially swap substrings, and mutate an occasional bit value for good measure."[2] In the following we will call our sample of the population just our population and also see each generation as a population, such that a genetic algorithm becomes a recursive algorithm defined over a population.

To determine what are the fittest organisms in the process of reproduction, a fitness function is used. This is function from the population to the real numbers. This function can be either a *fittest-low fitness function*, a *fittest-high fitness function* or a *fittest-0 function*. A *fittest-low fitness function* is defined s.t the lower the fitness value (possibly negative) an organism is assigned, the fitter the organism; a *fittest-high fitness function* is defined s.t. the higher, the fitter; a *fittest-zero fitness function* is defined s.t. the closer to zero the fittest and 0 is the highest possible fitness value (no negative values are allowed in this case).

In this short treatise on genetic algorithms, three different techniques to carry out the operation of reproduction will be illustrated: both the method of performing the reproduction technique and the mathematical consequences of each reproduction technique will be explained. This gives insight into what reproduction

[1] Goldberg 1989: p. 1.
[2] Goldberg 1989: p. 28.

technique to use in what setting. Sometimes, depending on the population and the fitness functions given, different reproduction techniques can be desirable.

2 Reproduction Techniques

2.1 Roulette Wheel Reproduction

In the *Roulette Wheel Reproduction* technique we reproduce each organism x_i with a probability corresponding to their share of the total fitness.[3] To carry out this reproduction technique a specific bound on the generation size must be given in advance. How *Roulette Wheel Reproduction* procedure is carried out, depends on the fitness function chosen. That is, the procedure is slightly more cumbersome if the fitness function associates lower fitness values to fitter organisms. Therefore we will first introduce the technique with a *fittest-high fitness function*.

Let $X = \{x_1, .., x_n\}$ be a finite population and let $f : X \to \mathbb{R}$ be some fitness function, s.t. the further $f^{>}(x)$ is away from zero the fitter. Let $c \in \mathbb{N}$ be some given bound of the generation size. Consider the following reproduction process:

1. Calculate $f(x_i)$ for all $x_i \in X$;
2. Associated $p_i = \dfrac{f(x_i)}{\sum_{j=1}^{n} f(x_j)}$ with x_i;
3. Create intervals $[0, p_1), ..., [\sum_{j<i}^{n} p_j, \sum_{j\leq i}^{n} p_j), ..., [\sum_{j<n}^{n} p_j, \sum_{j\leq n}^{n} p_j]$, s.t. to each $\langle x_i, p_i \rangle$ there is one interval associated;
4. Uniformly pick c random real numbers $r_1, ..., r_c$ from $[0, 1]$;
5. For each r_j, if r_j falls in interval $\langle x_i, p_i \rangle$, then reproduce x_i.

Note that in the above procedure we have, that the probability that x_i is reproduced, $\mathbb{P}[x_i]$, is exactly its share in the total fitness of the population, that is:

$$\mathbb{P}[x_i] = \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)} \tag{1}$$

Furthermore, the expected number of x_i in the next generation is then just its share in the total fitness times the given bound c on the generation size, that is:

$$\mathbb{E}[\#\text{ of organisms } x_i \text{ found}] = c \cdot \mathbb{P}[_i \text{ will be reproduced}] \tag{2}$$

Let $X_i = \{x \mid x \in X, f(x) = f(x_i)\}$, that is the set of organisms equally fit as organism x_i. Assume X_i consists of l_i organisms, then the expected number of elements of X_i in the next generation is:

$$\mathbb{E}[\#\text{ elements of } X_i \text{ found in next generation}] = l_i \cdot c \cdot \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)} \tag{3}$$

The above described process to carry out the *Roulette Wheel Reproduction* will not work for a *fittest-low* nor for a *fittest-0 fitness function*, because, then the fittest organisms will receive the lowest, or even negative, probability for survival, opposite to what we desire. The procedure can be altered, such that, it can be carried out with a *fittest-0 fitness function*, however, it cannot be tweaked to cope with a *fittest-low fitness function*.[4]

Again assume we have some finite population $X = \{x_1, .., x_n\}$ and let $f : X \to \mathbb{R}$ be a *fittest-0 fitness function*, s.t. the closer $f(x)$ is to zero, the fitter the organism x. If we want to be able to carry out reproduction using the the *Roulette Wheel Reproduction* technique, then we need to alter the probability p_i we assign to organism in step 2. We need to do this in such a way, that we satisfy the specific properties of *Roulette Wheel Reproduction* technique, which are:

[3] cf. Goldberg 1989: p. 30.

[4] If one has a *fittest-low fitness function* and wants to use the *Roulette Wheel Reproduction* reproduction technique nevertheless, then it is often possible to alter the fitness function to either a *fittest-0* or *fittest-high fitness function*. If this is impossible, then the *Total Order Reproduction* is a good alternative.

(i) reproduction probabilities respect the order of fitnesses:

$$f(x_i) < f(x_j) \Rightarrow \mathbb{P}[x_i] > \mathbb{P}[x_j] \text{ (N.B. we have a } \textit{fittest-low fitness function} \text{ now)}$$

(ii) reproduction probabilities respect the weight of the fitness:

$$\frac{f(x_i)}{f(x_j)} = r \Rightarrow \frac{\mathbb{P}[x_i]}{\mathbb{P}[x_j]} = r,$$

To achieve this, we need to ascribe the following probability to x_i, which will be again x_i's probability of survival to x_i:

$$p_i = \frac{\frac{1}{f(x_i)}}{\sum_{k=1}^{n} \frac{1}{f(x_k)}} = \mathbb{P}[x_i] \tag{4}$$

Proof. Clearly all the p_i add up to one, since $\sum_{j=1}^{n} \frac{\frac{1}{f(x_j)}}{\sum_{k=1}^{n} \frac{1}{f(x_k)}} = 1$. Furthermore, the probability assigned to each x_i satisfies the specific properties (i) and (ii) by respectively (i) and (ii) below:

(i) Assume $f(x_i) < f(x_j)$. Then $\frac{1}{f(x_i)} > \frac{1}{f(x_j)}$, thus we have, $\frac{\frac{1}{f(x_i)}}{\sum_{k=1}^{n} \frac{1}{f(x_k)}} > \frac{\frac{1}{f(x_j)}}{\sum_{k=1}^{n} \frac{1}{f(x_k)}}$, as required.

(ii) Assume $\frac{f(x_i)}{f(x_j)} = r$. This is equivalent to $\frac{1}{f(x_i)} = r \cdot \frac{1}{f(x_j)}$. The latter can be expanded to $\frac{\frac{1}{f(x_i}}{\sum_{k=1}^{n} \frac{1}{f(x_k)}} = \frac{r \cdot \frac{1}{f(x_j}}{\sum_{k=1}^{n} \frac{1}{f(x_k)}}$, which is by definition $\mathbb{P}[x_i] = r \cdot \mathbb{P}[x_j]$, or equivalently, $\frac{\mathbb{P}[x_i]}{\mathbb{P}[x_j]} = r$.

□

The foregoing implies that to adjust the *Roulette Wheel Reproduction* technique to cope with *fittest-low fitness functions* we just need to make *two* alterations in the process described above. First we need to check in the first step, after the calculations of all the fitnesses, if the sum of all the fitness values is greater than 0. If so, then we can continue to step 2. Second, we need to alter the p_i associated to each x_i according to the probability in (4).

The *Roulette Wheel Reproduction* technique stays true to the concept of *survival of the fittest*, but can result in a problem called *premature convergence*. If in the fitness variance in the population is high, early in search, and a small number of organisms are much fitter than the others, then these fittest organisms and their crossover products will dominate the next generations very quickly in the population. This can prevent the genetic algorithm from exploring a larger range of the population, leading it to miss the real optima. The dominating organisms will all have very similar fitness scores, that is, they have very low fitness variance, which prevents the roulette wheel from exploiting the fitness differences, resulting in a very early stabilization of the population, on a local optimum. The *Roulette Wheel Reproduction* technique can thus put "too much emphasis on 'exploitation' of highly fit strings at the expense of exploration of other regions of the search space." In *Roulette Wheel Reproduction* the difference among generations depends on the variance of fitness in the parent generation. This problem can be met by choosing a high mutation parameter, that is, to force many organisms to be mutated.[5]

2.2 Total-Order Reproduction

In the *Total-Order Reproduction* technique we reproduce each organism with a probability weighted x_i according to its position in an order on fitness classes. How this procedure is carried out does again depend on the fitness function chosen, but the required change to the procedure is very simple. Furthermore, just as in *Roulette Wheel Reproduction*, this reproduction technique requires a specific bound on the generation, given in advance, to be able to function.

[5] Mitchell 1999: p. 123.

Let $X = \{x_1, ..., x_n\}$ be a finite population. Let $f : X \to \mathbb{R}$ be a *fittest-low fitness function* for our population X. Let $c \in \mathbb{N}$ be some given bound of the generation size. Consider the following reproduction process:

1. Uniformly pick two organisms $x_i, x_j \in X$ (not necessarily distinct);
2. Calculate $f(x_i)$ and $f(x_j)$;
3. If $f(x_i) < f(x_j)$ then reproduce x_i, else reproduce x_j (for *fittest-high functions* choose >)
4. Repeat this procedure c times.

This reproduction process is called the *Total-Order Reproduction Process*, because it neglects the particular fitness values of the organisms, and only focusses on an order among the fitness values. To see this, partition X into subsets X_i s.t. each subset represents a fitness class, that is, all elements in X_i have the same fitness value f_i under f and for every distinct fitness value f_i, there is a distinct subset X_i representing it. We can order the subsets X_i according to their corresponding fitness value (higher to lower) s.t $X_1 > .. > X_k > ... > X_n$. Furthermore we let $X_i = \{x_{i,1}, ..., x_{i,l_i}\}$. This gives us the following survival probability for an organism of f_i:

$$\mathbb{P}[\exists k \text{ s.t. } x_{i,k} \text{ survives}] = \frac{l_i}{c} \cdot \frac{\sum_{j=1}^{i} l_j}{c} = \frac{l_i \cdot \sum_{j=1}^{i} l_j}{c^2}, \text{ where } j < i. \tag{5}$$

With this probability, we can calculate the expected number of elements of some particular subset X_i, found after reproduction, with a bound c on the generation size:

$$\mathbb{E}[\#\text{ elements of } X_i \text{ found in next generation}] = c \cdot \frac{l_i \cdot \sum_{j=1}^{i} l_j}{c^2} = l_i \cdot \frac{\sum_{j=1}^{i} l_j}{c} \tag{6}$$

In the above equation (6) we see that the expected number of instances of some particular subset X_i found, only depends on (i) the number of organisms with fitness f_i and (ii) the ratio of organism with fitness lower than f_i and the bound on the generation. Furthermore, we see that the function f is absent in equation (6) (cf. equation (3)), hence, only the order between the particular fitness values matters, not the size of their particular values. In *Total Order Reproduction* only the *number* of organisms with higher and lower-or-equal fitness determine which organisms are most likely to be reproduced. This is very different for Roulette Wheel Reproduction, where not only the *number*, but also the share of the particular fitness value of the organism in the total fitness of the population is taken into account.

A consequence of carrying out a *Total-Order Reproduction* is that the number of fittest organisms in a generation is expected never to increase. That is, the cardinality l_n of X_n is expected to remain the same in all generations, assuming a crossover and mutation parameter of 0; in *Roulette Wheel Reproduction* this cardinality is expected to keep increasing until the whole generation consists of fittest organisms.

2.3 Expected Number Control Reproduction

In the *Expected Number Control Reproduction* technique we reproduce ...
it can be seen as a waekened form of *ellitism* (in 'ellitism' the k best organisms to be retained in the next *generation*)[6]
Let $X = \{x_1, ..., x_n\}$ be a finite population. Let $f : X \to \mathbb{R}$ be a *fittest-high fitness function* for our population X. Define $\mu = \frac{\sum_{k=1}^{n} f(x_k)}{n}$, i.e. average fitness of population. Furthermore, define a floor function $\lfloor \cdot \rfloor : \mathbb{R} -> \mathbb{Z}$, which returns the integral part of a real number, e.g. $\lfloor 1.3 \rfloor = 3$ and $\lfloor 0.9 \rfloor = 0$. Do the following procedure for all $x_i \in X$:

1. Calculate $m_i = \frac{f(x_i)}{\mu}$;
2. Set $p_i = m_i - \lfloor m_i \rfloor$;
3. Reproduce x_i, $\lfloor m_i \rfloor$ times;
4. Pick uniformly some random real number $r \in [0, 1]$: if $1 - r \leq p_i$ then reproduce x_i.

[6] Mitchell 1999: p. 126

Note that in the above procedure we have, that the probability that x_i is reproduced, $\mathbb{P}[x_i]$, is:

$$\mathbb{P}[x_i] = \min\{\frac{f(x_i)}{\mu}, 1\} \tag{7}$$

It is clear from the above equation (7) that if some organism has a fitness above average, then it reproduce with probability 1, otherwise, it reproduces with a probability smaller than 1. The expected number copies of a single organism x_i is:

$$\mathbb{E}[\#x_i] = \lfloor m_i \rfloor + m_i - \lfloor m_i \rfloor = m_i = \frac{f(x_i)}{\mu} \tag{8}$$

The foregoing equations, (8) and (7), make it clear why this reproduction technique is called *Expected Number Control*, since this reproduction procedure guarantees that the fittest organisms are *reproduced*. This is however still no *elitisms*, because these fittest organisms can still disappear in the population after 1 round of the recursive genetic algorithm, due to crossover and mutation. Furthermore, from the foregoing, we see that *Expected Number Control* has the nice property that the ratio of the expected number of x_i and of x_j, in the next generation, is proportionate to the ratio of their fitnesses, that is:

$$f(x_i) = rf(x_j) \Rightarrow \mathbb{E}[\#x_i] = r\mathbb{E}[\#x_j], \tag{9}$$

in the case of a *fittest-high fitness function*, where x_i is r times fitter than x_j. As in the case of *Roulette Wheel Reproduction* technique the above described procedure will, however, not work with a *fittest-0* and *fittest-low fitness function*. The procedure can be adjusted such that it works for the first, but it cannot work in general for the latter. To adjust *Expected Number Control* to work for *fittest-0 fitness function*, the following needs to be done:

$$\text{set } m_i = \frac{\frac{1}{f(x_i)}}{\mu}.$$

To keep the nice property in (9), now in the case of *fittest-0 fitness function*, that is:

$$f(x_i) = \frac{1}{r}f(x_j) \Rightarrow \mathbb{E}[\#x_i] = r\mathbb{E}[\#x_j] \tag{10}$$

where we have $\frac{1}{r}$ to capture that x_i is r times fitter than x_j. It suffices to set m_i as above, since:

$$\mathbb{E}[\#x_i] = \frac{\frac{1}{f(x_i)}}{\mu} = \frac{1}{f(x_i)\cdot\mu} = \frac{1}{\frac{1}{r}f(x_j)\cdot\mu} = \frac{1}{r}\mathbb{E}[\#x_j]. \tag{11}$$

A major difference between *Expected Number Control Reproduction* and the other reproduction techniques described in this essay, is that *Expected Number Control* does not demand a bound on the size of the generation, specified before reproduction can be carried out. Moreover, the size of the new generation obtained by *Expected Number Control* can not even be known in advance, because it depends on the (biased) coin flips (i.e. the uniform pick of $r \in [0, 1]$) for the left-over non-integer parts of the m_i. Furthermore, just as *Roulette Wheel Reproduction*, reproduction by *Expected Number Control* suffers from *premature convergence* and even more severely. If the population that will be reproduced by *Expected Number Control*, has a high fitness variance, then the far outlying fittest organisms, will be reproduced in very large numbers, because their expectation depends (partly) on the mean fitness, meaning the genetic algorithm will, depending on the crossover and mutation parameters, stabilize very early on the optima found in the original random sample population.

3 Bibliography

1. Goldberg, D.E., (1989), *"Genetic Algorithms in Search, Optimization, and Machine Learning"*, Addison-Wesley Publishing Company.

2. Mitchell, M. (1999), *"An Introduction to Genetic Algorithms"*, MIT press.