Common Selection Methods

Melanie Mitchell

Adapted from *An Introduction to Genetic Algorithms*

An important decision to make in using a genetic algorithm is how to perform selection—that is, how to choose the individuals in the population that will create offspring for the next generation, and how many offspring each will create. The purpose of selection is, of course, to emphasize the fitter individuals in the population in hopes that their offspring will in turn have even higher fitness. Selection has to be balanced with variation from crossover and mutation (the "exploitation/exploration balance"): too-strong selection means that suboptimal highly fit individuals will take over the population, reducing the diversity needed for further change and progress; too-weak selection will result in too-slow evolution. As was the case for encodings, numerous selection schemes have been proposed in the GA literature. Below I will describe some of the most common methods. And as was the case for encodings, these descriptions do not provide rigorous guidelines for which method should be used for which problem; this is still an open question for GAs. (For more technical comparisons of different selection methods, see Goldberg and Deb 1991; Bäck and Hoffmeister 1991; de la Maza and Tidor 1993; and Hancock 1994.)

Fitness-Proportionate Selection with "Roulette Wheel" and "Stochastic Universal" Sampling

Holland's original GA used fitness-proportionate selection, in which the "expected value" of an individual (i.e., the expected number of times an individual will be selected to reproduce) is that individual's fitness divided by the average fitness of the population. The most common method for implementing this is "roulette wheel" sampling, described in chapter 1: each individual is assigned a slice of a circular "roulette wheel," with the size of the slice being proportional to the individual's fitness. The wheel is spun N times, where N is the number of individuals in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation. This method can be implemented as follows:

1. Sum the total expected value of individuals in the population. Call this sum T.
2. Repeat N times:
   - Choose a random integer r between 0 and T.
   - Loop through the individuals in the population, summing the expected values, until the sum is greater than or equal to r. The individual whose expected value puts the sum over this limit is the one selected.

This stochastic method statistically results in the expected number of offspring for each individual. However, with the relatively small populations typically used in GAs, the actual number of offspring allocated to an individual is often far from its expected value (an extremely unlikely series of spins of the roulette wheel could even allocate all offspring to the worst individual in the population). James Baker (1987) proposed a different sampling method—"stochastic universal sampling" (SUS)—to minimize this "spread" (the range of possible actual values, given an expected value). Rather than spin the roulette wheel N times to select N parents, SUS spins the wheel once—but with N equally spaced pointers, which are used to selected the N parents. Baker (1987) gives the following code fragment for SUS (in C):

```
ptr = Rand(); /* Returns random number uniformly distributed in [0,1] */
for (sum = i = 0; i < N; i++)
    for (sum += ExpVal(i,t); sum > ptr; ptr++)
        Select(i);
```

where i is an index over population members and where $\mathrm{ExpVal}(i,t)$ gives the expected value of individual i at time t. Under this method, each individual i is guaranteed to reproduce at least $\lfloor \mathrm{ExpVal}(i,t) \rfloor$ times but no more than $\lceil \mathrm{ExpVal}(i,t) \rceil$ times. (The proof of this is left as an exercise.)

SUS does not solve the major problems with fitness-proportionate selection. Typically, early in the search the fitness variance in the population is high and a small number of individuals are much fitter than the others. Under fitness-proportionate selection, they and their descendents will multiply quickly in the population, in effect preventing the GA from doing any further exploration. This is known as "premature convergence." In other words, fitness-proportionate selection early on often puts too much emphasis on "exploitation" of highly fit strings at the expense of exploration of other regions of the search space. Later in the search, when all individuals in the population are very similar (the fitness variance is low), there are no real fitness differences for selection to exploit, and evolution grinds to a near halt. Thus, the "rate of evolution" depends on the variance of fitnesses in the population.

Sigma Scaling

To address such problems, GA researchers have experimented with several "scaling" methods—methods for mapping "raw" fitness values to expected values so as to make the GA less susceptible to premature convergence. One example is "sigma scaling" (Forrest 1985; it was called "sigma truncation" in Goldberg 1989a), which keeps the selection pressure (i.e., the degree to which highly fit individuals are allowed many offspring) relatively constant over the course of the run rather than depending on the fitness variances in the population. Under sigma scaling, an individual's expected value is a function of its fitness, the population mean, and the population standard deviation. A example of sigma scaling would be

$$\text{ExpVal}(i,t) = \begin{cases} 1 + \frac{f(i)-\overline{f}(t)}{2\sigma(t)} & \text{if } \sigma(t) \neq 0 \\ 1.0 & \text{if } \sigma(t) = 0, \end{cases}$$

where $\text{ExpVal}(i,t)$ is the expected value of individual i at time t, $f(i)$ is the fitness of i, $\overline{f}(t)$ is the mean fitness of the population at time t, and $\sigma(t)$ is the standard deviation of the population fitnesses at time t. This function, used in the work of Tanese (1989), gives an individual with fitness one standard deviation above the mean 1.5 expected offspring. If $\text{ExpVal}(i,t)$ was less than 0, Tanese arbitrarily reset it to 0.1, so that individuals with very low fitness had some small chance of reproducing.

At the beginning of a run, when the standard deviation of fitnesses is typically high, the fitter individuals will not be many standard deviations above the mean, and so they will not be allocated the lion's share of offspring. Likewise, later in the run, when the population is typically more converged and the standard deviation is typically lower, the fitter individuals will stand out more, allowing evolution to continue.

Elitism

"Elitism," first introduced by Kenneth De Jong (1975), is an addition to many selection methods that forces the GA to retain some number of the best individuals at each generation. Such individuals can be lost if they are not selected to reproduce or if they are destroyed by crossover or mutation. Many researchers have found that elitism significantly improves the GA's performance.

Boltzmann Selection

Sigma scaling keeps the selection pressure more constant over a run. But often different amounts of selection pressure are needed at different times in a run—for example, early on it might be good to be liberal, allowing less fit individuals to reproduce at close to the rate of fitter individuals, and having selection occur slowly while maintaining a lot of variation in the population. Later it might be good to have selection be stronger in order to strongly emphasize highly fit individuals, assuming that the early diversity with slow selection has allowed the population to find the right part of the search space.

One approach to this is "Boltzmann selection" (an approach similar to simulated annealing) in which a continuously varying "temperature" controls the rate of selection according to a preset schedule. The temperature starts out high, which means that selection pressure is low (i.e., every individual has some reasonable probability of reproducing). The temperature is gradually lowered, which gradually increases the selection pressure, thereby allowing the GA to narrow in ever more closely to the best part of the search space while maintaining the "appropriate" degree of diversity. For examples of this approach, see Goldberg 1990, de la Maza and Tidor 1991 and 1993, and Prügel-Bennett and Shapiro 1994. A typical implementation is to assign to each individual i an expected value,

$$\mathrm{ExpVal}(i,t) = e^{f(i)/T} / < e^{f(i)/T} >_t,$$

where T is temperature and $<>_t$ denotes the average over the population at time t. Experimenting with this formula will show that, as T decreases, the difference in $\mathrm{ExpVal}(i,t)$ between high and low fitnesses increases. The desire is to have this happen gradually over the course of the search, so temperature is gradually decreased according to a predefined schedule. De la Maza and Tidor (1991) found that this method outperformed fitness-proportionate selection on a small set of test problems. They also (1993) compared some theoretical properties of the two methods.

Fitness-proportionate selection is commonly used in GAs mainly because it was part of Holland's original proposal and because it is used in the Schema Theorem, but, evidently, for many applications simple fitness-proportionate selection requires several "fixes" to make it work well. In recent years completely different approaches to selection (e.g., rank and tournament selection) have become increasingly common.

Rank Selection

Rank selection is an alternative method whose purpose is also to prevent too-quick convergence. In the version proposed by Baker (1985), the individuals in the population are ranked according to fitness, and the expected value of each individual depends on its rank rather than on its absolute fitness. There is no need to scale fitnesses in this case, since absolute differences in fitness are obscured. This discarding of absolute fitness information can have advantages (using absolute fitness can lead to convergence problems) and disadvantages (in some cases it might be important to know that one individual is far fitter than its nearest competitor). Ranking avoids giving the far largest share of offspring to a small group of highly fit individuals, and thus reduces the selection pressure when the fitness variance is high. It also keeps up selection pressure when the fitness variance is low: the ratio of expected values of individuals ranked i and $i+1$ will be the same whether their absolute fitness differences are high or low.

The linear ranking method proposed by Baker is as follows: Each individual in the population is ranked in increasing order of fitness, from 1 to N. The user chooses the expected value Max of the individual with rank N, with $Max \geq 0$. The expected value of each individual i in the population at time t is given by

$$\mathrm{ExpVal}(i,t) = Min + (Max - Min)\frac{\mathrm{rank}(i,t) - 1}{N - 1}, \tag{1}$$

where Min is the expected value of the individual with rank 1. Given the constraints $Max \geq 0$ and $\sum_i \mathrm{ExpVal}(i,t) = N$ (since population size stays constant from generation to generation), it is required that $1 \leq Max \leq 2$ and $Min = 2 - Max$. (The derivation of these requirements is left as an exercise.)

At each generation the individuals in the population are ranked and assigned expected values according to equation **??**. Baker recommended $Max = 1.1$ and he showed that this scheme compared favorably to fitness-proportionate selection on some selected test problems. Rank selection has a possible disadvantage: slowing down selection pressure means that the GA will in some cases be slower in finding highly fit individuals. However, in many cases the increased preservation of diversity that results from ranking leads to more successful search than the quick convergence that can result from fitness-proportionate selection. A variety of other ranking schemes (such as exponential rather than linear ranking) have also been tried. For any ranking method, once the expected values have assigned, the SUS method can be used to sample the population (i.e., choose parents).

As was described in chapter 2 above, a variation of rank selection with elitism was used by Meyer and Packard for evolving condition sets, and my colleagues and I used a similar scheme for evolving cellular automata. In those examples the population was ranked by fitness and the top E strings were selected to be parents. The $N - E$ offspring were merged with the E parents to create the next population. As was mentioned above, this is a form of the so-called $(\mu + \lambda)$ strategy used in the evolution strategies community. This method can be useful in cases where the fitness function is noisy (i.e., is a random variable, possibly returning different values on different calls on the same individual); the best individuals are retained so that they can be tested again and thus, over time, receive increasingly reliable fitness estimates.

Tournament Selection

The fitness-proportionate methods described above require two passes through the population at each generation: one pass to compute the mean fitness (and, for sigma scaling, the standard deviation) and one pass to compute the expected value of each individual. Rank scaling requires sorting the entire population by rank—a potentially time-consuming procedure. Tournament selection is similar to rank selection in terms of selection pressure, but it is computationally more efficient and more amenable to parallel implementation. Two individuals are chosen at random from the population. A random number r is then chosen between 0 and 1. If $r < k$ (where k is a parameter, for example 0.75), the fitter of the two individuals is selected to be a parent; otherwise the less fit individual is selected. The two are then returned to the original population and can be selected again. An analysis of this method was given in Goldberg and Deb 1991.

Steady-State Selection

Most GAs described in the literature have been "generational"—at each generation the new population consists entirely of offspring formed by parents in the previous generation (though some of these offspring may be identical to their parents). In some schemes, such as the elitist schemes described above, successive generations overlap to some degree—some portion of the previous generation is retained in the new population. The fraction of new individuals at each generation has been called the "generation gap" (De Jong 1975). In steady-state selection, only a few individuals are replaced in each generation: usually a small number of the least fit individuals are replaced by offspring resulting from crossover and mutation of the fittest individuals. Steady-state GAs are often used in evolving rule-based systems (e.g., classifier systems; see Holland 1986) in which incremental learning (and remembering what has already been learned) is important and in which members of the population collectively (rather than individually) solve the problem at hand. Steady-state selection has been analyzed by Syswerda (1989, 1991); by Whitley (1989); and by De Jong and Sarma (1993).