

2AIT608 - Machine Learning

Genetic Algorithms

Dr Dimitris C. Dracopoulos

email: d.dracopoulos@westminster.ac.uk

1 Introduction to Evolutionary Computing

Inspired by (Darwinian) natural selection and genetics.

Three main categories:

- Genetic Algorithms
- Genetic Programming
- Evolutionary Programming developed by Lawrence Fogel is based on mutation. No crossover is involved.

2 Genetic Algorithms

Genetic algorithms (GAs) is a type of optimisation techniques applicable to a number of different applications (not only optimisation).

Traditional optimisation algorithms are based on two main categories:

- *Indirect methods:* Set the gradient to zero and solve a set of nonlinear equations.
- *Direct methods:* Start from a specific point and follow the direction of the local gradient (gradient descent).

Both of the above classical techniques operate locally (therefore they are restricted to local minima). The second disadvantage of the classical techniques is that the space for many real world problems is not smooth or continuous and the derivatives do not exist.

3 Genetic Algorithms vs Traditional Optimisation Techniques

1. GAs work with a coded representation of the parameters, not directly with the parameters.
2. GAs search from many different points simultaneously, not from a single point.
3. GAs use objective (cost or utility) functions, not derivatives or other knowledge which might not be available or does not exist.
4. GAs use probabilistic rules, not deterministic rules.

4 Operation of the Genetic Algorithm

The following steps describe the application of a genetic algorithm:

1. Start with an initial population (e.g. random) of candidate solutions.
2. Repeatedly apply a number of genetic operators to generate a new population.
3. Denote the best individual of the last generation (population) as the solution.

5 Genetic Operators applied by a Genetic algorithm

The operators that a genetic algorithm is using are:

- *Reproduction*: Select individuals with higher fitness than others to reproduce so that their children are found in the next generation. Unfit individuals die with higher probability than fitter ones.
- *Crossover*: Combine two reproduced individuals so that their children are copies in the next generation.
- *Mutation*: Probabilistic change of part of an individual.

Besides mutation and crossover, other specialised genetic operators can be applied as well, depending on the nature of the problem.

5.1 Reproduction

The process of selection an individual so that its offspring is copied to the next generation.

The simplest way to do so, is to choose an individual based on *proportional fitness* (also known as roulette wheel selection):

- Given a weighted roulette wheel representing the percentage of individuals fitness over all population, spin the wheel and select the individual which the wheel selected (Figure 1).

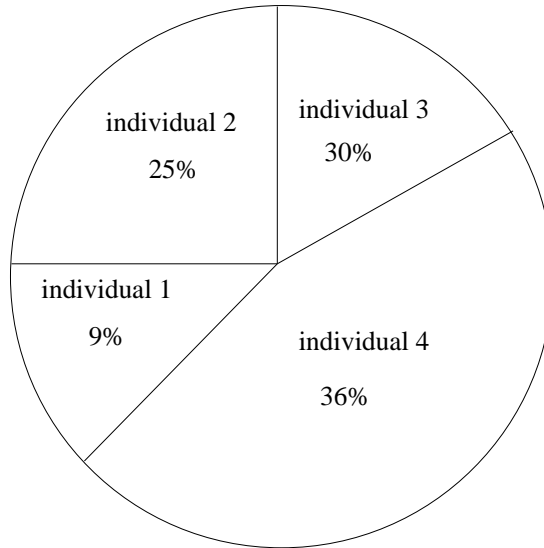


Figure 1: Reproduction based on roulette wheel selection.

The process of roulette wheel can be also described as follows:

Calculate for each individual its proportion of the total population fitness, such that the sum of all the individual fitness equals 1.0. On a line between 0.0 and 1.0, the better individuals will take up a larger proportion of the line than will less fit individuals. An individual is then chosen by generating a random number between 0.0 and 1.0 and then selecting the individual in whose region of the number line the random number falls. By repeating this each time an individual needs to be chosen, the better individuals will be chosen more often than the poorer ones, thus fulfilling the requirements of survival of the fittest.

5.2 Crossover

Randomly select one (or more) point(s) in two selected individuals, exchange the corresponding parts, creating two offspring. Copy the offspring to the new generation.

This is illustrated in Figure 2.

5.3 Mutation

Mutation is the occasional (with small probability) random alteration of a gene of a chromosome (chromosome is the term used for the representation of an individual).

- This has the purpose of reintroducing useful genetic material (genes) that has been lost.

The mutation operator is illustrated in Figure 3.

6 Example:

Find the minimum of function $f(x) = x^2$ in the range $[0, 31]$.

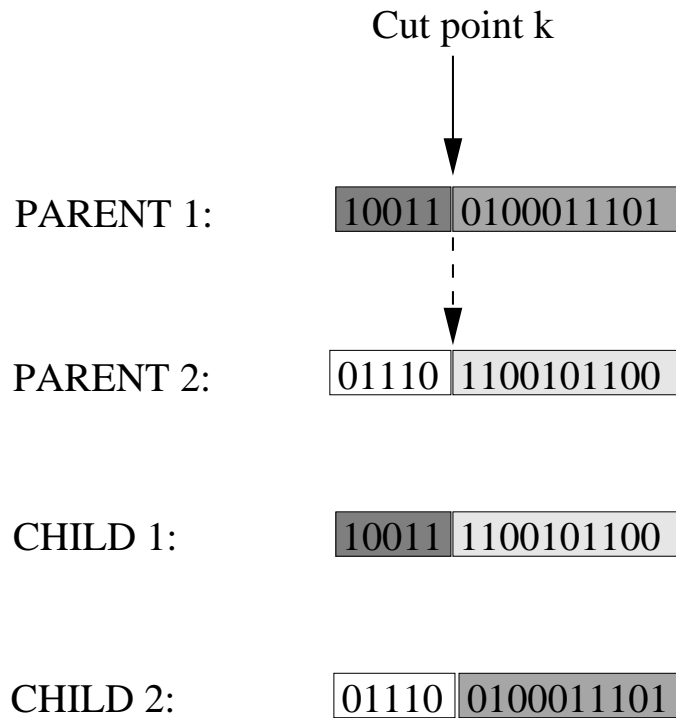


Figure 2: A one-point crossover operation.

original chromosome: 110100011011100

mutated chromosome: 110101011011100

Figure 3: The mutation genetic operation.

- A candidate solution can be represented in a binary form, using 5 bits. Thus, 11111 represents a candidate solution corresponding to $x = 31$.
- The genetic algorithm iterative steps (shown in the pseudocode of the next section) can be applied then.

7 Pseudocode of a Genetic Algorithm

The pseudocode of a genetic algorithm is shown in Figure 4.

8 The Schema Theorem - Why Genetic Algorithms Work

Without loss of generality, assume a population of binary strings with length $l = 7$.

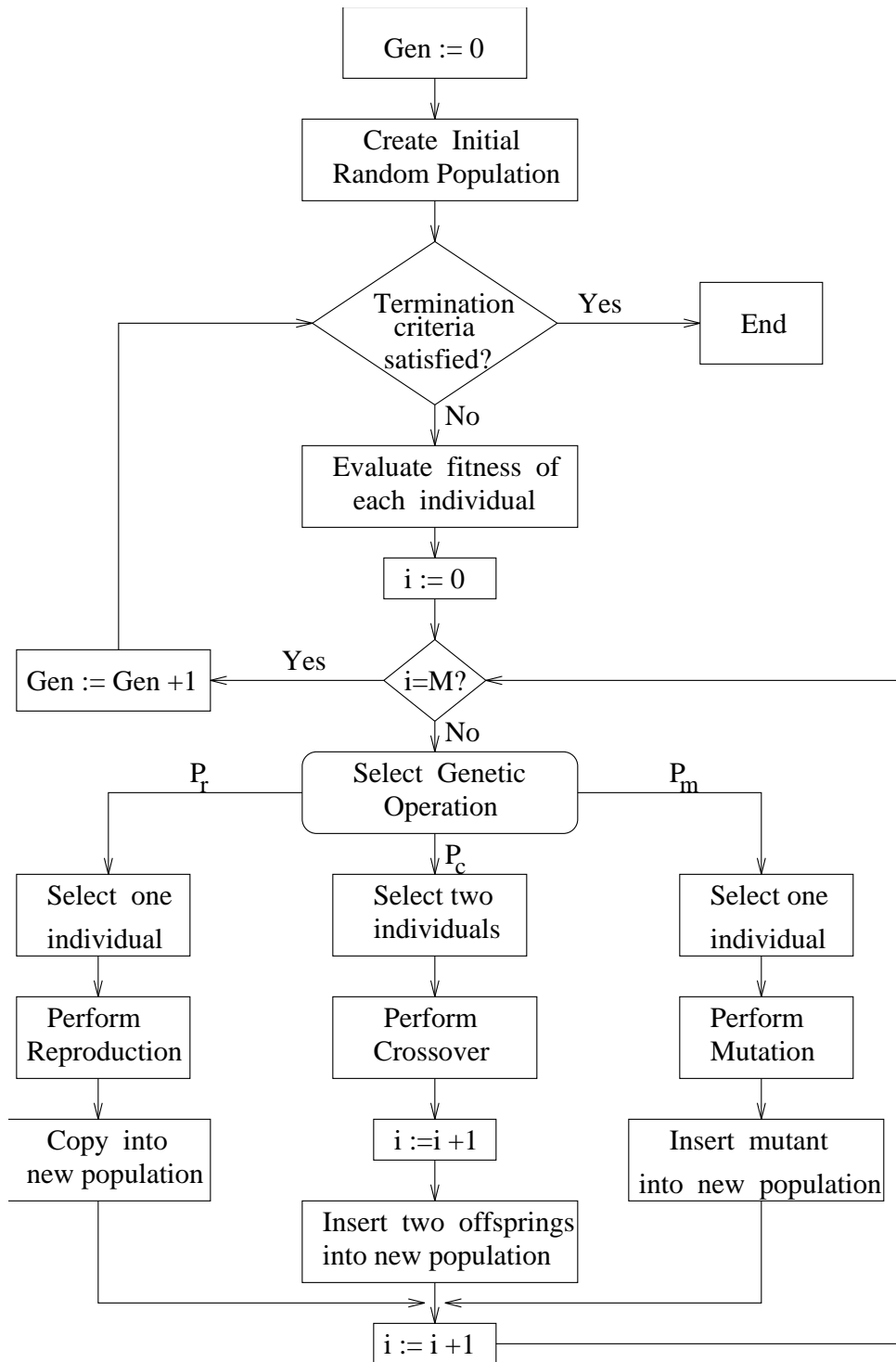


Figure 4: The conventional genetic algorithm.

There are 2^l different strings consisting of 0s and 1s.

A schema H is a string taken from the three-letter alphabet 0, 1, * where * is the don't care character.

For example, consider schema $H = *11*0**$. Then string $A = 0111000$ is an example of schema H , but $B = 0011000$ is not (the second bit does not match).

- In total there are 3^l schemata over a binary string of length l .
- In a string population with n members there are at most $n \cdot 2^l$ schemata. This is because each string member belongs to 2^l schemata (the first position of a schema must contain the first bit of the string or *, the second position of a schema must contain the second bit of the specific string or *, etc.).
- The *order* $o(H)$ of a schema H is defined as the number of fixed positions. For example the order of $011*1**$ is 4.
- The *defining length* $d(H)$ of a schema H is the distance between the first and last specific string positions. For example, the defining length of $011*1**$ is 4.

To determine the expected number of schemata in the population because of reproduction:

Assume at time t , there are m examples of a specific schema H contained in the population $A(t)$, i.e. $m = m(H, t)$.

During reproduction, a string is copied to the next generation according to its fitness, with probability:

$$p_i = \frac{f_i}{\sum f_j} \quad (1)$$

where f_i the fitness of individual string i .

After picking n members for the next population using reproduction, the expected number of schemata H is:

$$m(H, t + 1) = m(H, t) \cdot n \cdot \frac{f(H)}{\sum f_j} \quad (2)$$

where $f(H)$ is the average fitness of the strings representing H at t .

Since the average fitness of the whole population is $\bar{f} = \frac{\sum f_j}{n}$, the previous equation can be written as:

$$m(H, t + 1) = m(H, t) \frac{f(H)}{\bar{f}} \quad (3)$$

A particular schema grows as the ratio of the average fitness of the schema to the average fitness of the population. Schemata with fitness above the population average will receive an increasing number of samples in the next generation.

The effect of crossover in schemata:

Consider the following string A and two representative schemata:

```
A = 0 1 1 1 0 0 0
H1 = * 1 * * * * 0
H2 = * * * 1 0 * *
```

Choosing position 3 as the crossover point, destroys schema H_1 but not H_2 . Therefore the probability of destroy schema H_2 is $1/6$ (the probability of choosing a crossover point within the defining length of the schema).

The probability of survival of H_2 is $1 - 1/6 = 5/6$.

The probability of survival of any schema H is:

$$p_s = 1 - \delta(H)/(l - 1) \quad (4)$$

where $\delta(H)$ is the defining length of H and l the length of the alphabet.

Assuming that crossover occurs with probability p_c then the probability of schema H to be destroyed is $p_c \cdot \frac{\delta(H)}{l-1}$ and the survival probability is:

$$p_s \geq 1 - p_c \cdot \frac{\delta(H)}{l - 1} \quad (5)$$

The combined effect of reproduction and crossover is given by:

$$m(H, t + 1) \geq m(H, t) \frac{f(H)}{\bar{f}} \left[1 - p_c \cdot \frac{\delta(H)}{l - 1} \right] \quad (6)$$

Schema H grows more than schema H_2 if its defining length is shorter than the defining length of H_2 and its fitness is above the fitness of H_2 .

Determining the effect of mutation on schema H :

Assume that mutation of a gene occurs with probability p_m . Then the probability of a gene surviving is $1 - p_m$ and the probability of schema H surviving is:

$$(1 - p_m)^{o(H)} \quad (7)$$

where $o(H)$ is the order of schema H .

If $p_m \ll 1$ then:

$$(1 - p_m)^{o(H)} \approx 1 - o(H) \cdot p_m \quad (8)$$

The combined effect of reproduction, crossover and mutation on the expected number of copies that a particular schema H receives in the next generation is (ignoring small cross product terms):

$$m(H, t + 1) \geq m(H, t) \frac{f(H)}{\bar{f}} \left[1 - p_c \cdot \frac{\delta(H)}{l - 1} - o(H) \cdot p_m \right] \quad (9)$$

The above is referred to as the *Schema Theorem*.

9 Different Selection Techniques for GA Individuals

- *Proportional Fitness Selection*: The roulette wheel selection, described before.

- *Tournament Selection*: This is closer to the type of competition in nature where two animals will fight for the right to mate. Various rules exist for tournament selection, but in general two or more individuals are chosen randomly from the population, and their fitness are compared. The individual with the highest fitness wins. This type of selection does not require the evaluation of the fitness for all individuals in the population. It is possible to evaluate only the fitness of an individual as and when it is chosen for a tournament.
- *Rank Selection*: Sort the individuals' fitness and rank them accordingly. The ranks can be compared rather than the fitness. One of the advantages that this approach has is, that small differences between the fitness of individuals can be exaggerated.

10 Different types of Fitness Measure

- *Raw Fitness r_i* : The fitness as defined by a objective function. This could be the error associated with an individual. This is simpler than others as it is defined directly in terms of the problem definition. A large value can indicate a better or a worse individual (depending on how the raw fitness is defined).
- *Standardised Fitness s_i* : This converts the raw fitness so that a lower value corresponds to a better fitness. If a lesser value of the raw fitness corresponds to a better value then $s_i = r_i$.

Otherwise the standardised fitness can be derived as:

$$r_i = r_{\max} - r_i \quad (10)$$

- *Adjusted Fitness a_i* : This lies between 0 and 1 and it has larger values for better individuals:

$$a_i = \frac{1}{1 + s_i} \quad (11)$$

The advantage of adjusted fitness is that it exaggerates the importance of small differences between individuals. This is important in later generations of a GA run.

- *Normalised Fitness n_i* : It is larger for better individuals of the population and the sum of the normalised fitness values is 1. It can be defined as:

$$n_i = \frac{a_i}{\sum_k a_k} \quad (12)$$

The normalised fitness is particularly useful if fitness proportional selection is applied.