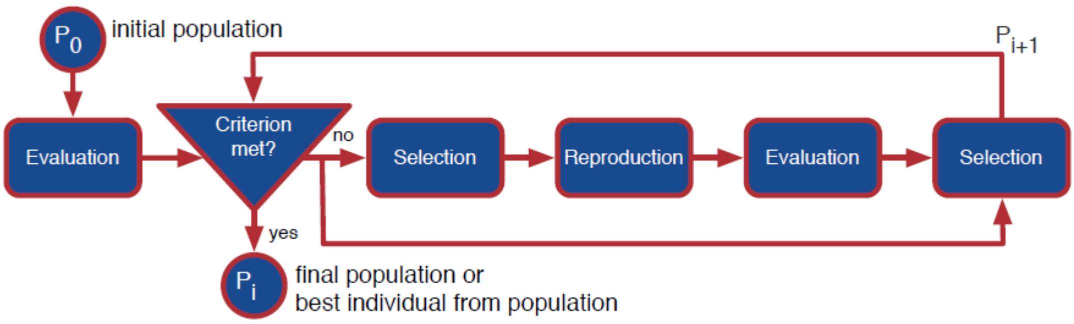
Genetic Algorithm (Continued)

Genetic algorithms are usually used to identify optimal solutions to complex problems. This can clearly be easily mapped to search methods, which are aiming toward a similar goal. Genetic algorithms can thus be used to search for solutions to multi-value problems where the closeness of any attempted solution to the actual solution (**fitness**) can be readily evaluated. In short, a **population** of possible solutions (**chromosomes**) is generated, and a fitness value for each chromosome is determined. This fitness is used to determine the likelihood that a given chromosome will survive to the next generation or reproduce. Reproduction is done by applying **crossover** to two (or more) chromosomes, whereby features (**genes**) of each chromosome are combined together. Mutation is also applied, which involves making random changes to particular genes.



Lab Tasks

Exercise 12.1.

Consider the problem from the last lab focusing on maximizing the following function and Perform the mentioned modifications.

$$f(x) = \frac{-x^2}{10} + 3x$$

where x is allowed to vary between 0 and 31. You must perform following tasks in the code.

a. Representation of states (solutions): To solve this using a genetic algorithm, we must encode the possible values of x as chromosomes. For this problem, we will encode x as a binary integer of length 5. Thus, the chromosomes for our genetic algorithm will be sequences of 0's and 1's with a length of 5 bits, and have a range from 0 (00000) to 31 (11111).

b. Fitness function:

The fitness function for it will be:

$$f(x) = \frac{-x^2}{10} + 3x$$

To begin the algorithm, we select an initial population of 10 chromosomes at random. The resulting initial population of chromosomes is shown in Table 1. Next we take the x-value that each chromosome represents and test its fitness with the fitness function. The resulting fitness values are recorded in the third column of Table 1.

Chromosome Number	Initial Population	x-Value	Fitness Value $f(x)$	Selection Probability
1	0 1 0 1 1	11	20.9	0.1416

2	1 1 0 1 0	26	10.4	0.0705
3	0 0 0 1 0	2	5.6	0.0379
4	0 1 1 1 0	14	22.4	0.1518
5	0 1 1 0 0	12	21.6	0.1463
6	1 1 1 1 0	30	0	0
7	1 0 1 1 0	22	17.6	0.1192
8	0 1 0 0 1	9	18.9	0.1280
9	0 0 0 1 1	3	8.1	0.0549
10	1 0 0 0 1	17	22.1	0.1497

c. **Operators:**
   i. Apply cross over in every generation.
   ii. Apply mutation after every 3 generations.

d. **Termination criteria:** Your loop should stop when the value of one of your candidate's fitness functions is greater or equal to 90%.

Your Task is to modify the implementation according to the given instructions:

1- Now take 6 bit binary i.e. range from 0 to 63 (000000 – 111111), take 7 random values from 0 to 63 as initial population.
2- Change the parent selection method from Roulette wheel to
   i. Rank Selection
   ii. Random Selection
3- Modify the cross over to 2-point cross over.

Exercise 12.2.

Suppose a genetic algorithm uses chromosomes of the form x = abcdefgh with a fixed length of eight genes. Each gene can be any digit between 0 and 9. Let the fitness of individual x be calculated as:

$$f(x) = (a + b) - (c + d) + (e + f) - (g + h)$$

and let the initial population consist of four individuals with the following chromosomes:

x1 = 6 5 4 1 3 5 3 2

x2 = 8 7 1 2 6 6 0 1

x3 = 2 3 9 2 1 2 8 5

x4 = 4 1 8 5 2 0 9 4

a) Evaluate the fitness of each individual and arrange them in order with the fittest first and the least fit last.
b) Cross the first and third fittest individuals (ranked 1st and 4^{th}) using a uniform crossover.
c) Produce 6 new chromosomes and the solution should have a chromosome that gives the maximum of the fitness function