Lab 06

Genetic Algorithm

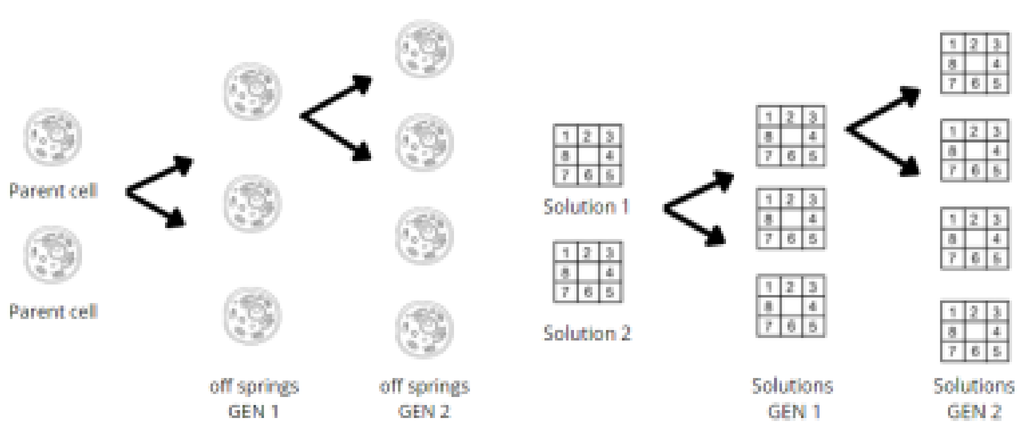
1. What is Genetic Algorithm?

A **genetic algorithm** is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found.

2. Comparison between Natural Evolution and Genetic Algorithm

This notion can be applied for a search problem. We consider a set of solutions for a problem and select the set of best ones out of them.



3. Genetic Algorithm

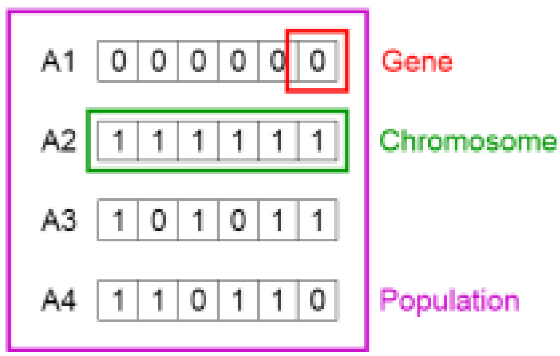
Five phases are considered in a genetic algorithm

1. Initial population

The process begins with a set of individuals which is called a Population. Each individual is a solution to the problem you want to solve.

An individual is characterized by a set of parameters (variables) known as Genes. Genes are joined into a string to form a Chromosome (solution).

In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We say that we encode the genes in a chromosome.



2. Fitness function

The **fitness function** determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a **fitness score** to each individual. The probability that an individual will be selected for reproduction is based on its fitness score

3. Selection

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation.

Two pairs of individuals (parents) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

There are many types of selection:-

1- Roulette wheel

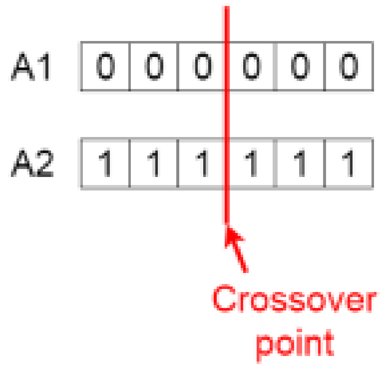
2- Random

3- Tournament selection

4. Crossover

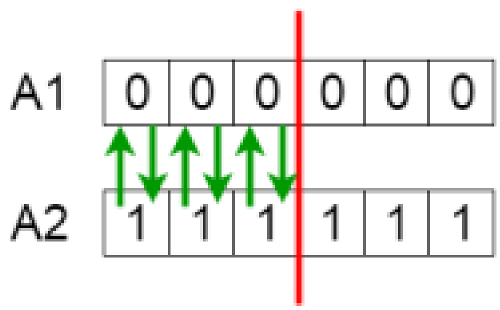
Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes.

For example, consider the crossover point to be 3 as shown below.



Crossover point

Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached.



Exchanging genes among parents

The new offspring are added to the population.

Types are:-

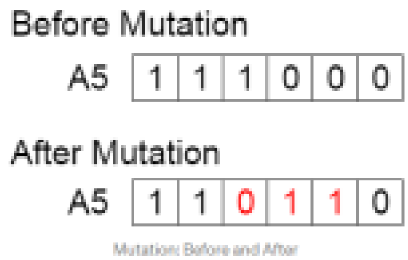
1- Single point

2- Two point

3- Uniform crossover

5. Mutation

I am here In certain new offspring formed, some of their genes can be subjected to a **mutation** with a low random probability. This implies that some of the bits in the bit string can be flipped.

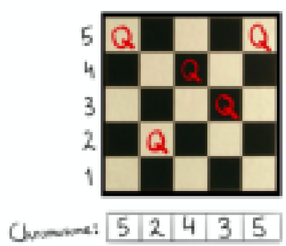


Mutation: Before and After

4. Mapping of G.A on N-Queens Problem

1. Initial population

Firstly, we need to create a chromosome representation. For showing a chromosome, the best way is to represent it as a list of length N where in our case N=5. The value of each index shows the row of the queen in a column. The value of each index is from 1 to 5..



In the initialization process, we need to arrange a random population of chromosomes (potential solutions) are created. Here is the initial population, I took 4 chromosomes, each of which has a length 5. They are

[5 2 4 3 5]

[4 3 5 1 4]

[2 1 3 2 4]

[5 2 3 4 1]

2. Fitness function

First of all, the fitness function is pairs of non-attacking queens. So, higher scores are better is better for us. In order to solve the fitness function for the chromosome [5 2 4 3 5], I assigned each queen uniquely as Q1, Q2, Q3, Q4 and Q5. And to find the fitness function value I made the following equation:

Fitness function = F1+F2+F3+F4+F5

where:

> F1 = number of pairs of nonattacking queens with queen Q1.
>
> F2 = number of pairs of nonattacking queens with queen Q2.
>
> F3 = number of pairs of nonattacking queens with queen Q3.
>
> F4 = number of pairs of nonattacking queens with queen Q4.
>
> F5 = number of pairs of nonattacking queens with queen Q5.

Here for example if we already counted pair Q1 and Q2 to F1, we should not count the same pair Q2 and Q1 to F2.

So we found that for chromosome [5 2 4 3 5] the fitness function will be equal to 7.

Then we need to compute the probability of being chosen from the fitness function. This will be needed for the next selection step. First, we need to add all fitness functions which will be equal as the following:

7+6+6+5=24

Then we need to compute the probability of being chosen from the fitness function. We need to divide the fitness function by the sum of the fitness function and multiply it to 100%.

[5 2 4 3 5] probability of being chosen = 7/24 *100% = 29%

[4 3 5 1 4] probability of being chosen =6/24 * 100% = 25%

[2 1 3 2 4] probability of being chosen =6/24 * 100% = 25%

[5 2 3 4 1] probability of being chosen =5/24 * 100% = 21%

3. Selection

In the next step, we randomly choose the two pairs to reproduce based on probabilities which we counted on the previous step. In other words, a certain number of chromosomes will survive into the next generator using a selection operator. Here selected chromosomes to act as parents that are combined using crossover operator to make children. In addition to this, we pick a crossover point per pair.

Here we took randomly following chromosomes based on their probabilities:

[4 3 5 1 4]

[5 2 4 3 5]

[4 3 5 1 4]

[2 1 3 2 4]

We can notice that we did not take the chromosome [5 2 3 4 1] because its probability of being chosen is the least among chromosomes.

For the first pair

[4 3 5 1 4]

[5 2 4 3 5]

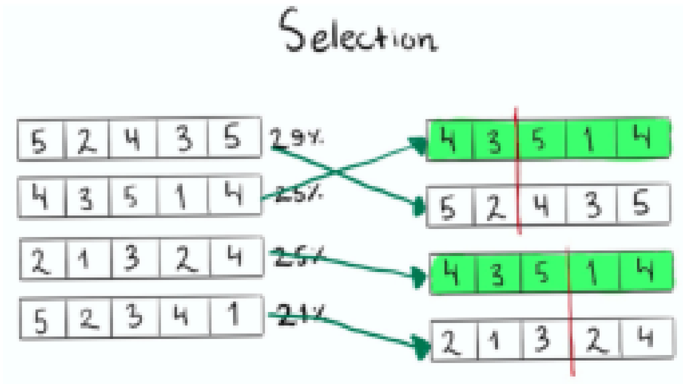
The crossover point will be picked after two genes.

In the case of the second pair

[4 3 5 1 4]

[2 1 3 2 4]

The crossover point will be picked after three genes.



4. Crossover

In the crossover, selected chromosomes act as parents that are combined using crossover operator to make children. In other words, it combines the genetic information of two parents to generate new offspring.

Here we can see that children generated from the first pair ([4 3 5 1 4] and [5 2 4 3 5]) are the following:

[4 3 4 3 5]

[5 2 5 1 4]

From the second pair ([4 3 5 1 4] and [2 1 3 2 4]) the children are the following:

[4 3 5 2 4]

[2 1 3 1 4]

In other words, in order to create the first child from pair in crossover process, we took the parent #1 chromosome first part and parent #2 chromosome second part which makes the new individual which consists of

[(first part of parent #1 chromosome) [(second part of parent #2 chromosome)]

In order to create the second child from the same pair we took the parent #1 chromosome second part and parent #2 chromosome first part which makes the new individual which consists of

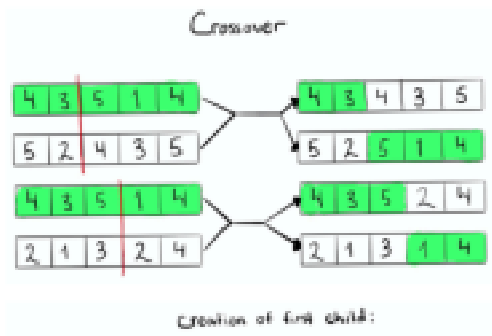
[(second part of parent #1 chromosome) [(first part of parent #2 chromosome)]

So in our case when we create the children of pair [4 3 5 1 4] and [5 2 4 3 5], for producing the first child, we took [(first part of parent #1 chromosome) [(second part of parent #2 chromosome)] = [4 3 4 3 5]

For producing second child, we took [(second part of parent #1 chromosome) [(first part of parent

#2 chromosome)] = [5 2 5 1 4]

The same process we will do to the second pair ([4 3 5 1 4] and [2 1 3 2 4]).



5. Mutation

The next step is mutation. In the mutation process, we alter one or more gene values in chromosomes which we found after crossover. So it randomly changes a few gens and the mutation probability is low. So in our example, our mutation will look like the following:

[4 3 4 3 5] →[4 3 1 3 5]

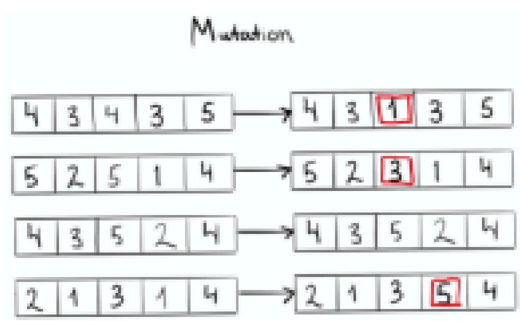
[5 2 5 1 4] →[5 2 3 1 4]

[4 3 5 2 4] →[4 3 5 2 4]

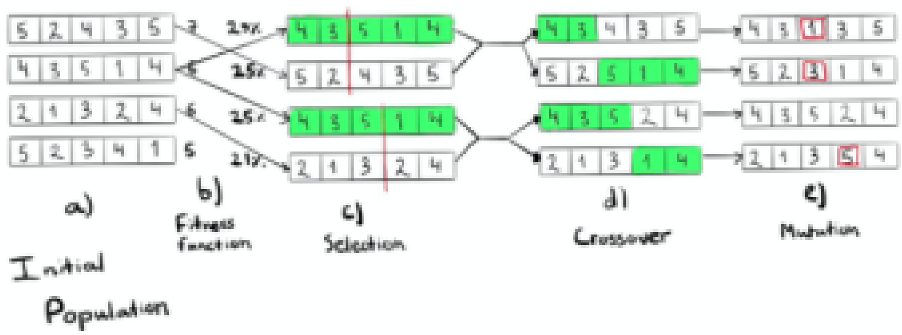
[2 1 3 1 4] →[2 1 3 5 4]

where we can notice that the third gene in the chromosome [4 3 4 3 5] changed from 4 to 1.

Also the third gene in the chromosome [5 2 5 1 4] changed from 5 to 3. In addition to this, the fourth gene in the chromosome [2 1 3 1 4] changed from 1 to 5.



So until this, the genetic algorithm to solve the 5-Queen algorithm will look like the following:



In the next step, we need to update the generation. New chromosomes will update the population but the population number will not change. So the chromosomes

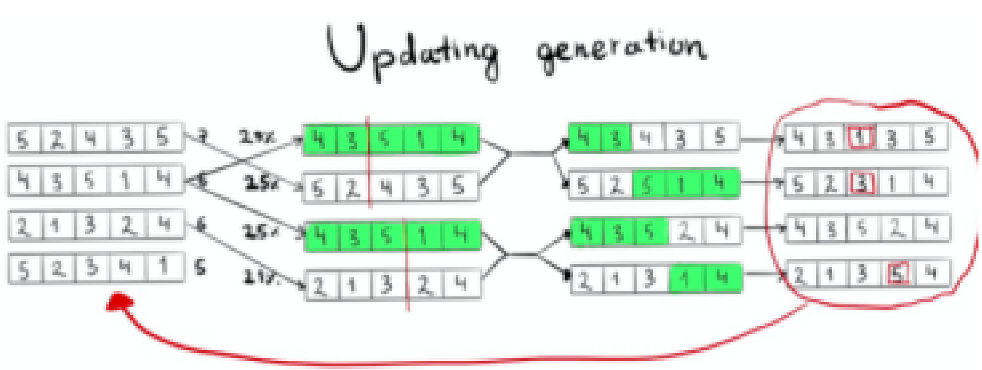
[4 3 1 3 5]

[5 2 3 1 4]

[4 3 5 2 4]

[2 1 3 5 4]

Will be our new population.



Exercise:

A thief enters a shop carrying bag which can carry 35 kgs of weight. The shop has 10 items, each with a specific weight and price. Now, the thief's dilemma is to make such a selection of items that it maximizes the value (i.e., total price) without exceeding the knapsack weight. We have to help the thief to make the selection.

Available Items are:

Item No.	Weight	Value
1	3	266
2	13	442
3	10	671
4	9	526
5	7	388
6	1	245
7	8	210
8	8	145
9	2	126
10	9	322

Initial population:

[[0 1 0 1 1 0 0 1 1 1]

[1 1 1 1 0 1 1 1 0 0]

[0 1 0 0 0 0 1 1 0 1]

[0 0 1 0 1 1 0 0 0 0]

[0 0 1 1 0 0 0 0 0 1]

[0 1 0 1 1 0 1 0 0 0]

[1 1 1 0 0 0 1 0 1 0]

[0 0 0 0 1 1 1 0 0 0]

Fitness Function:

$$fitness = \sum_{i=1}^{n} c_i v_i; if \sum_{i=1}^{n} c_i w_i \leq kw$$

$$fitness = 0; otherwise$$

where,

n = chromosome length

Ci = ith gene

Vi = ith value

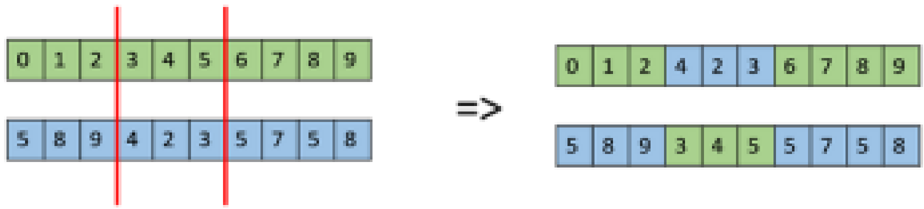
Wi = ith weight

kw = allowed weight

To generate new population:

Two-point crossover

Example of two-point cross-over. Choose the positions as in the



example.

Mutate offspring got from cross over at 2 random positions. Select 2 random positions: if you find 1 change it to 0 and vice versa.

Always keep 8 best chromosomes, 50% from initial population and 50% from new generation. If your program taking time you can iterate for fixed number of iterations.

Change the give code to meet the following requirements:

1) Change the fitness function to one, given here and run the same code to see if its working correctly.

Fitness Function:

$$fitness = \sum_{i=1}^{n} c_i v_i ; if \sum_{i=1}^{n} c_i w_i \leq kw$$

$$fitness = 0; otherwise$$

2) Do not use the initial population generation method given in the code. Instead use the population given in manual.

Initial population:

[[0 1 0 1 1 0 0 1 1 1]

[1 1 1 1 0 1 1 1 0 0]

[0 1 0 0 0 0 1 1 0 1]

[0 0 1 0 1 1 0 0 0 0]

[0 0 1 1 0 0 0 0 0 1]

[0 1 0 1 1 0 1 0 0 0]

[1 1 1 0 0 0 1 0 1 0]

[0 0 0 0 1 1 1 0 0 0]

3) Change the above code to implement the two point cross over instead of random selection.
4) Change the given code to select 60% samples from new generation and 50% from initial population. Which is 50-50 right now.

Submission Instructions

Always read the submission instructions carefully.

Rename your Jupyter notebook to your roll number and download the notebook as .ipynb extension.

To download the required file, go to File->Download .ipynb Only

submit the .ipynb file. DO NOT zip or rar your submission file.

Submit this file on Google Classroom under the relevant assignment.

Late submissions will not be accepted.

END