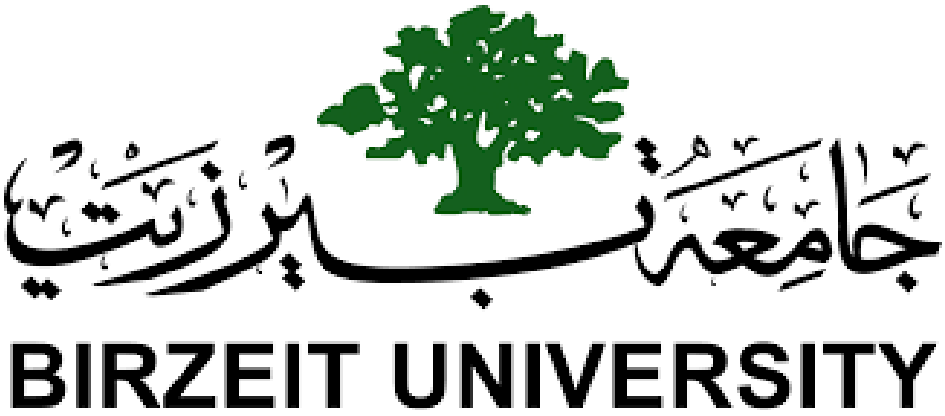


Faculty of Engineering and Technology

Electrical and Computer Engineering Department

ARTIFICIAL INTELLIGENCE - ENCS3340

Project # 1 Report

Prepared by:

Student name :Hala Jebreel Student name:Diana Naseer

Student number : 1210606 Student number: 1210363

Supervised by: Dr. Yaza Abu Farha

Section : 4

- # Problem formulation:
  - **The Job Shop Scheduling Problem** is solved using a genetic algorithm, which is aim the makespan , or total completion time, of a set of jobs processed on a set of machines. Each job consists of a sequence of operations, each requiring a specific machine for a given duration. Chromosome representation in this context involves encoding each job as a sequence of operations, where the order of operations within each job is a permutation of the indices of the operations. As an example, if there are two jobs each with three operations, a chromosome will be like [[2, 0, 1], [0, 2, 1]] , indicating the order in which the operations for each job are to be performed.

  - **Chromosome** : The fitness of a chromosome is evaluated by decoding the sequence into an actual schedule and calculating the makespan. The crossover operation combines two parent chromosomes to produce offspring by selecting a random crossover point and swapping the subsequences of the parents at this point. For instance, given parents: [[2, 0, 1], [0, 2, 1]] and [[1, 2, 0], [2, 0, 1]] with a crossover point after the first element, the offspring might be [[2, 2, 0], [0, 0, 1]] and [[1, 0, 1], [2, 2, 1]].

    The crossover function combines two parent chromosomes to produce two offspring. A single crossover point is randomly selected, and the sequences from the parents are combined at this point to create the children. This method ensures that each child inherits a mix of operations from both parents.

    - **Crossover:** The crossover function combines two parent chromosomes to produce two offspring. A single crossover point is randomly selected, and the sequences from the parents are combined at this point to create the children. This method ensures that each child inherits a mix of operations from both parents.

  - **Mutation** : Mutation in the genetic algorithm introduces diversity by randomly swapping two operations within a job's sequence. For instance, with a mutation rate of 0.1, a chromosome like [[2, 0, 1], [0, 2, 1]] might mutate to [[0, 2, 1], [0, 2, 1]] by exchanging the positions of the first and second operations in the first job. The

algorithm initializes a population of random chromosomes, evaluates their fitness, selects parents using methods such as tournament selection, applies crossover and mutation to create offspring, and forms a new population. This process repeats iteratively until an optimal or satisfactory schedule is achieved.

Mutation introduces variability into the population by randomly swapping two operations within a job sequence. This helps to explore new areas of the solution space and avoid local minima.

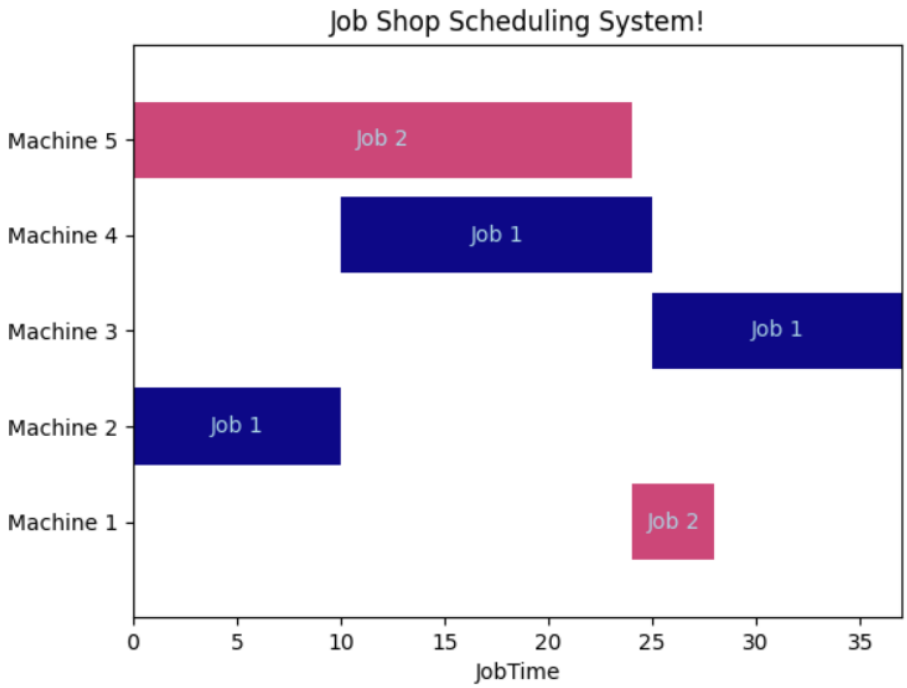
- The best chromosome discovered is decoded to provide the optimal schedule, which is then visualized using a Gantt chart. This method efficiently explores the solution space of the JSSP, leveraging the genetic algorithm's capabilities to handle complex scheduling problems and achieve high-quality schedules.

  - Objective Function

  The objective function evaluates the fitness of each chromosome by calculating the make span, which is the total time required to complete all jobs. The **calculate_time** function decodes the individual and schedules the operations on the machines, tracking the end times to determine the make span.

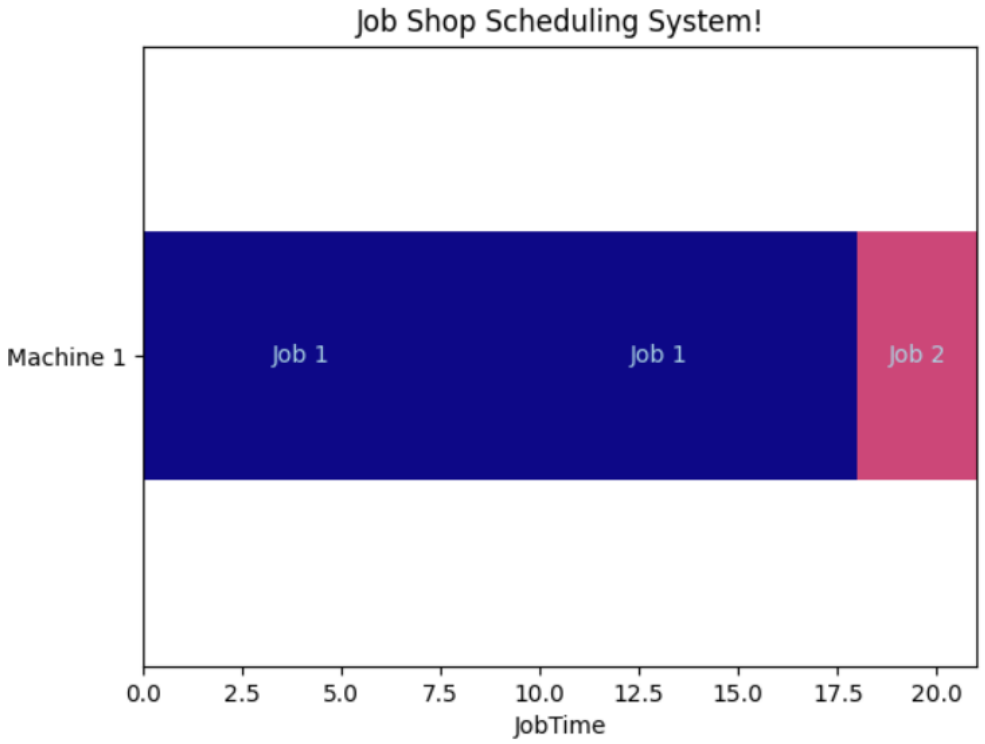
Test’s Cases:

```
Please enter the number of machines: 5
Please enter the number of jobs: 2
Please enter the name of Job 1: Cut
Enter the number of Job_Operations for Job Cut: 3
Please enter machine, and duration(time) for the operation 1 (in this format:M1 10): M2 10
Please enter machine, and duration(time) for the operation 2 (in this format:M1 10): M3 12
Please enter machine, and duration(time) for the operation 3 (in this format:M1 10): M4 15
Please enter the name of Job 2: Drilling
Enter the number of Job_Operations for Job Drilling: 2
Please enter machine, and duration(time) for the operation 1 (in this format:M1 10): M1 4
Please enter machine, and duration(time) for the operation 2 (in this format:M1 10): M5 24
```



```
                    {Welcome to Job Shop Scheduling!}

Please enter the number of machines: 1
Please enter the number of jobs: 2
Please enter the name of Job 1: J1
Enter the number of Job_Operations for Job J1: 2
Please enter machine, and duration(time) for the operation 1 (in this format:M1 10): M1 10
Please enter machine, and duration(time) for the operation 2 (in this format:M1 10): M1 8
Please enter the name of Job 2: J2
Enter the number of Job_Operations for Job J2: 1
Please enter machine, and duration(time) for the operation 1 (in this format:M1 10): M1 3
```



```
                    {Welcome to Job Shop Scheduling!}

Please enter the number of machines: 3
Please enter the number of jobs: 2
Please enter the name of Job 1: C1
Enter the number of Job_Operations for Job C1: 2
Please enter machine, and duration(time) for the operation 1 (in this format:M1 10): M2 10
Please enter machine, and duration(time) for the operation 2 (in this format:M1 10): M3 15
Please enter the name of Job 2: C2
Enter the number of Job_Operations for Job C2: 1
Please enter machine, and duration(time) for the operation 1 (in this format:M1 10): M6 14
Invalid machine index: M6. Please enter a machine index between M1 and M3.
Please enter machine, and duration(time) for the operation 1 (in this format:M1 10): M1 9
```