



**Faculty of Engineering & Technology  
Electrical & Computer Engineering Department**

**ARTIFICIAL INTELLIGENCE  
ENCS3340  
Project 1: Genetic Algorithm**

---

**Prepared by:**

**Name:** Ghassan Qandeel      **ID:** 1212397

**Name:** Mohammad khmour      **ID:** 1212517

**Supervised by:**

Dr. Ismail Khater

Dr. Yazn Abu Faraha

**Date:** 20-5-2024

**Abstract:**

Consider a manufacturing plant with several machines such as cutting machines, drilling machines, and assembly stations. Each product requires a sequence of operations on these machines. The scheduling problem here is to determine the optimal sequence and timing for each product to minimize the overall production time or maximize throughput while considering machine capacities and job dependencies. This project aims to develop a genetic algorithm to optimize job shop scheduling in a manufacturing plant setting.

**Input:**

- List of Jobs with their operation in machines.

**Constraints:**

- Can not do operation on machine if it busy in another machine (should wait)
- Can not interrupt the operation of machine while it starts.
- Just one machine can do on the job in same time.

**Output:**

- The best solution after doing genetic algorithm like Gant chart, this solution will be implemented in Gant chart will describe each machine any job work and any time is start time, and end time also waiting time if happened.

**objective function:**

the objective function typically aims to minimize the total completion time of all jobs. it implemented in the calculate\_fitness function, which evaluates the fitness of a given job schedule. The fitness value is the total time of the schedule, and the goal is to minimize this value.

### **problem formulation:**

for this problem we see it hard to formulation it firstly, but after we understand all the constraints is describe above and in the project file, firstly we put the input in file and read it with number of machines, we take the input data implement it in same way of population then shuffle it with respect the constrains (randomly with respect for constraint), Here the population is ready and is the form of chromosome and population

### **Chromosome representation:**

We assume the easier way to represent chromosome is following form:

**[job1, job2, job3, .....]**

### **Fitness function:**

The function takes the chromosome and return the max time needed to end all job, also return dictionary that have jobs as keys and machines with start and end as value for each job in specific machine. Because we choice the objective function is to calculate the production time, we will take the time as fitness value and the lowest time is best parents.

### **Cross-over:**

After we take the suitable parents, we make the cross-over operation randomly by 80% percent to happen, and we select the cutting point randomly to make efficient solution and maybe this solution be the optimal one.

### **Mutation:**

We take the offspring children from cross-over operation to make mutation, it can change on solution by shift one job randomly, this operation can happen by 10% percent.

## Test cases:

Job1: M1[9] -> M4[5] -> M5[12] -> M1[6]

Job2: M2[10] -> M3[7] -> M4[14]

Job3: M3[10] -> M2[7] -> M4[14]

Job4: M5[6] -> M3[7] -> M2[7]

Job5: M3[5] -> M2[7] -> M3[8]

Job6: M1[10]

## Output:

