Assignment: Genetic Algorithm for Scheduling Using Permutation Representation

Objective:

Implement a Genetic Algorithm (GA) to solve a single-machine scheduling problem with sequence-dependent setup times, minimizing total weighted completion time.

Problem Description:

You are given a set of n independent jobs.

Each job (i) has:

- **Processing time (p_i)**
- **Weight/priority (w_i)**

For every ordered pair of jobs ((i, j)), there is a **setup time ($s_{i,j}$)**, representing the time required to switch from job (i) to job (j).

A solution is represented as a permutation of the n jobs:

$$X = [x_1, x_2, \dots, x_n]$$

The completion times are computed as:

$$C_{x_1} = p_{x_1}$$

$$C_{x_k} = C_{x_{k-1}} + s_{x_{k-1},x_k} + p_{x_k}$$

The objective is to minimize the total weighted completion time:

$$T(X) = \sum_{k=1}^{n} w_{x_k} C_{x_k}$$

Requirements:

You must implement a Genetic Algorithm with the following characteristics:

Population size: 50

Max generations: 100

Elitism: 2. The top 2 best individuals copied directly into the next generation.

Selection: tournament (k=3)

Crossover: Implement the following permutation-preserving crossover methods:

Order Crossover, and

PMX (Partially Mapped Crossover), and

Cycle Crossover

Mutation: Implement the following permutation-preserving mutation methods:

Swap mutation, and

Inversion mutation, and

Scramble mutation

You must test **all 9 possible** crossover-mutation combinations.

Fitness: Since GA maximizes fitness, the fitness of each chromosome is the inverse of its total weighted completion time ($fitness(X) = \frac{1}{T(X)}$).

Representation: permutation of jobs

Required Outputs:

Your program must output:

1. The best job sequence found by the GA
2. The minimum total weighted completion time
3. The number of generations executed

Visualization Requirement:

You must generate a plot showing the GA's performance:

Maximum fitness per generation

Average fitness per generation

Dataset:

n = 7

Processing times P_i:

1:12, 2:7, 3:15, 4:5, 5:9, 6:11, 7:8

Weights w_i:

1:4, 2:9, 3:3, 4:7, 5:5, 6:6, 7:8

Setup time matrix $s_{i,j}$:

Row 1: 0 4 6 5 7 3 4

Row 2: 4 0 5 6 4 5 6

Row 3: 6 5 0 4 5 7 6

Row 4: 5 6 4 0 3 4 5

Row 5: 7 4 5 3 0 6 4

Row 6: 3 5 7 4 6 0 5

Row 7: 4 6 6 5 4 5 0

Deliverables:

1. Source code implementing the Genetic Algorithm **from scratch**

2. The max/average fitness plot

3. A report including:

Description of encoding, operators, and parameter settings

Fitness plot

Final solution and performance discussion