

Department of Computer Science and Engineering

Course Code: CSE366

Course Title: Artificial Intelligence

Section: 04

Semester: Spring 24

**Assignment- 02**

**Submitted to:**

**Dr. Mohammad Rifat Ahmmad Rashid**

Assistant Professor

Department of Computer Science and Engineering

East West University

**Submitted by:**

**Name:** B M Shahria Alam

**ID:** 2021-3-60-016

**Date of submission:** 4th April, 2024.

**Objective**

The goal of this assignment is to develop and implement a Genetic Algorithm (GA) to optimize the assignment of multiple robots to a set of tasks in a dynamic production environment. Your primary objectives are to minimize the total production time, ensure a balanced workload across robots, and prioritize critical tasks effectively. Additionally, you will create a detailed visualization to illustrate the final task assignments, robot efficiencies, and task priorities.

**Detailed Requirements**

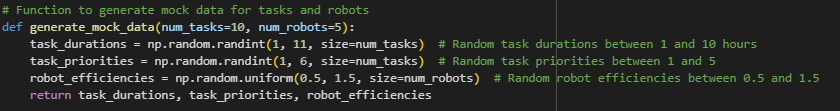
**1. Background:**

• You have a set of tasks, each with a specified duration and priority.

• A pool of robots is available, each with a unique efficiency factor.

• The production environment is dynamic, with tasks and priorities potentially changing over time

**2. Tasks:**

• **Data Preparation:** Generate mock data for tasks (including durations and priorities) and robots (including efficiency factors).

Here we create random duration, priorities of num\_task where num\_tasks=10, it can be any value which need to be solved by the robots. we also define each robot efficiencies of num\_robots which is currently defined 5 but can be pass anything through the parameter.

• **GA Implementation**: Implement a Genetic Algorithm to optimize task assignments considering task duration, robot efficiency, and task priority.

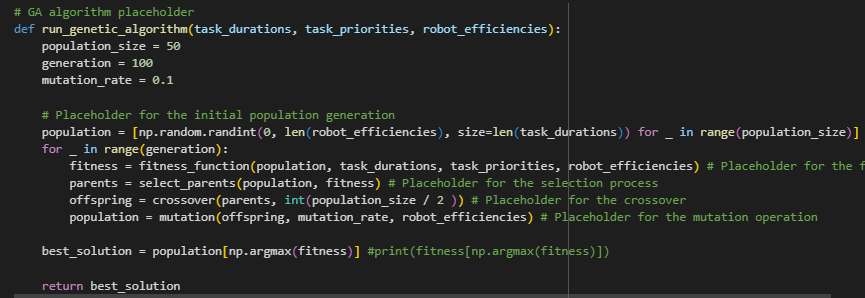
• **Visualization:** Create a grid visualization of the task assignments highlighting key information.

**3. Genetic Algorithm Components:**

• Individual Representation: Represent each potential solution as a vector where each element indicates the robot assigned to each task.

Individual I is represented as a vector of N integers, where N is the number of tasks, and each integer In (where 1≤n≤N) corresponds to the ID of the robot assigned to task n. I=[r1,r2,...,rN]

First we defined initial population randomly by,

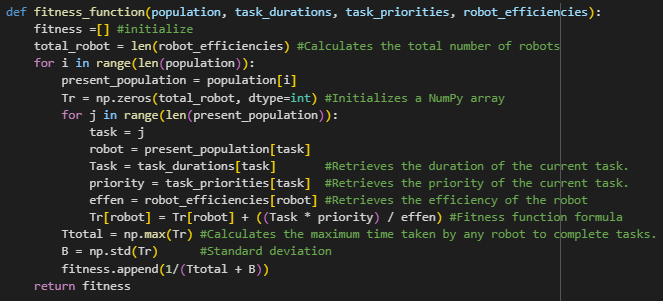


**• Fitness Function:** The fitness function aims to minimize the total production time while ensuring a balanced workload across robots and prioritizing critical tasks. It can be decomposed into several components: Total Time, Workload balance;

• Calculate the total production time, Ttotal, as the maximum time taken by any robot based on its assigned tasks and efficiency.

• Compute workload balance, B, as the standard deviation of the total times across all robots.

• Define the fitness function, F, to minimize both Ttotal and B, incorporating task priorities.



this is our fitness function, where we implemented

Tr = ∑ n∈tasks(r) Dn\*Pn / Er

Ttotal = max (T1, T2, ... , TR)

where:

• tasks(r) is the set of tasks assigned to robot r,

• Dn is the duration of task n,

• Pn is the priority weight of task n,

• Er is the efficiency of robot R

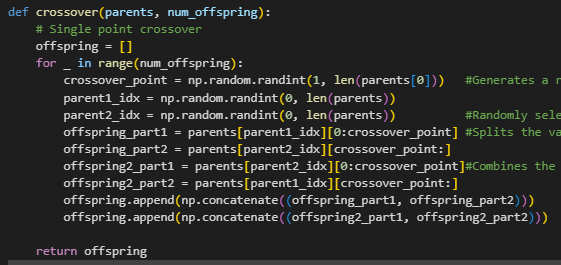
• R is the total number of robots.

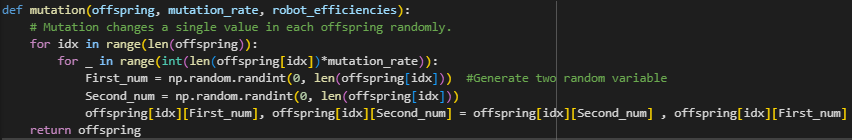
B=σ(T1,T2,...,TR) here this is the standard davition for balacing the workload,

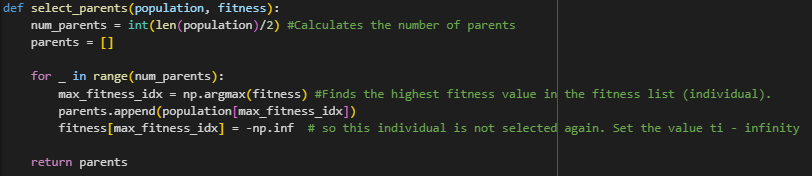
then F(I) = Ttota + B

**4. • Selection, Crossover, and Mutation:**

Implemented Uniform Cost Search (UCS) and A\* (A Star) pathfinding algorithms. The selection process is crucial for guiding the GA towards optimal solutions by choosing individuals from the current population to breed the next generation. We have used Tournament Selection. Where we have choose half of the population to be in mating pool by their fitness value, we choose the values whom’s fitness value Is the lowest because we are trying to minimize the cost and minimize the standard deviation of workload







**5. Visualization**

