East West University

Department of Computer Science and Engineering

LAB REPORT

Course Code: CSE366

Course Title:

Section:

Lab Number: 02

Experiment Name:

Robot Task Optimization Using Genetic Algorithm

Submitted To: Dr. Md. Habibur Rahman.

Name	ID
Fatema Jalal	2021-1-60-108

This code is an implementation of a Genetic Algorithm (GA) to solve a task allocation problem in a multi-robot system. Let's break down the code:

Importing Libraries: The code begins by importing necessary libraries such as NumPy for numerical computations and Matplotlib for visualization.

Mock Data Generation: Mock data for task durations, task priorities, and fixed robot efficiencies are generated randomly. This data represents the characteristics of tasks to be assigned to robots.

GA Implementation Parameters: Parameters for the GA implementation are defined, including population size, number of generations, and mutation rate.

Initial Population Generation: An initial population of solutions is generated randomly. Each solution represents a possible assignment of tasks to robots.

Fitness Function: The calculate_fitness function evaluates the fitness of a solution. It calculates the total production time, workload balance, and incorporates task priorities to compute a fitness value. The fitness value is a measure of how well a solution performs.

Mutation Operation: The mutate function performs mutation on a solution with a given mutation rate. In this case, mutation involves swapping the robots assigned to two randomly selected tasks.

Visualization Function: The visualize_assignments function is used to visualize task assignments. It creates a heatmap where rows represent robots and columns represent tasks. The cell values indicate task durations, and task priorities are displayed as text within the cells.

Example Usage: The code provides an example usage where it visualizes the assignment before and after mutation. It first visualizes the assignment from the initial population, then applies mutation to one solution and visualizes the mutated assignment.

Overall, this code demonstrates a basic implementation of a genetic algorithm for optimizing task allocation in a multi-robot system, with visualization to help understand the task assignments and their characteristics.

Output:

The output of the code consists of two visualizations:

Before Mutation:

This visualization shows the initial assignment of tasks to robots before the mutation operation is applied.

Each cell in the grid represents a task assigned to a robot, where the intensity of the color indicates the duration of the task, and the number inside the cell indicates the task priority.

The row labels represent the robots, and the column labels represent the tasks.

The legend on the right side of the plot indicates the task priorities.

After Mutation:

This visualization shows the assignment of tasks to robots after the mutation operation is applied.

The mutation operation randomly selects two tasks within the assignment list and swaps their assigned robots.

The resulting assignment may have changed compared to the initial assignment due to the mutation operation.