East West University

Department of CSE

Assignment 02

Robot Task Optimization Using Genetic Algorithm

Course Code: CSE366

Course Title: Artificial Intelligence

Submitted By:

Md. Arfan Ahmed

ID: 2020-1-60-139

Submitted to:

Dr. Mohammad Rifat Ahmmad Rashid

Assistant Professor, Department of Computer Science and Engineering

Date of Submission: 02 April, 2024

Introduction:
In this report, we represent the development and implementation of a Genetic Algorithm (GA) for optimizing of multiple robots to a set of tasks in a dynamic production environment. The primary objective was to minimize the total production time, ensure a balanced workload across robots, and prioritize critical tasks effectively. The assignment problem involved generating mock data for tasks and robots, implementing the GA algorithm, and visualizing the task assignments along with relevant parameters.

Background:
The assignment problem involved 10 tasks with specified durations and priorities, along with 5 robots characterized by unique efficiency factors. The production environment was dynamic, potentially leading to changes in tasks and priorities over time. The durations and priorities are randomly generated.

Implementation Details:
Mock data for tasks and robots was generated using Python's library. The GA algorithm was then implemented to optimize task assignments, considering task duration, robot efficiency, and task priority. Key parameters such as population size and number of generations were specified to guide the optimization process.

Individual solutions were represented as vectors, with each element indicating the robot assigned to each task. The fitness function aimed to minimize both total production time and workload imbalance while incorporating task priorities. Selection, crossover, and mutation operations were implemented to evolve the population towards optimal solutions.

- **Fitness Function Design:** Designing an effective fitness function that considers both production time and workload balance was challenging. It required careful consideration of task priorities and robot efficiencies. Fitness function is evaluated by sum of max total production time and workload of individuals.

- **Selection**: Tournament selection was employed in our implementation, where 50 tournaments were conducted. In each tournament, a subset of individuals (size 5) was randomly selected from the population. The individual with the lowest fitness score within the tournament was chosen as the winner, ensuring that fitter individuals had a higher chance of being selected as parents for the next generation.

- **Crossover**: Single-point crossover was utilized to generate offspring from selected parent individuals. In this process, a random crossover point was chosen along the genome of the parents. Offspring were then created by swapping the genetic information beyond the crossover point between the parents. This operation facilitated the exploration of new genetic combinations and introduced diversity into the population.

- **Mutation**: Mutation was applied based on a mutation rate of 20%. For each individual in the population, a certain number of mutations were randomly introduced. The mutation rate determined the probability of each gene (task assignment) being mutated. This random alteration of genes helped in introducing new genetic variations, preventing premature convergence, and enabling the algorithm to explore different regions of the solution space.

Visualization:
A grid visualization was created to illustrate task assignments, robot efficiencies, and task priorities. Each cell in the grid represented a task assigned to a robot, with color intensity indicating task duration and annotations providing information on task priority and duration. Robot efficiencies were also annotated for clarity.

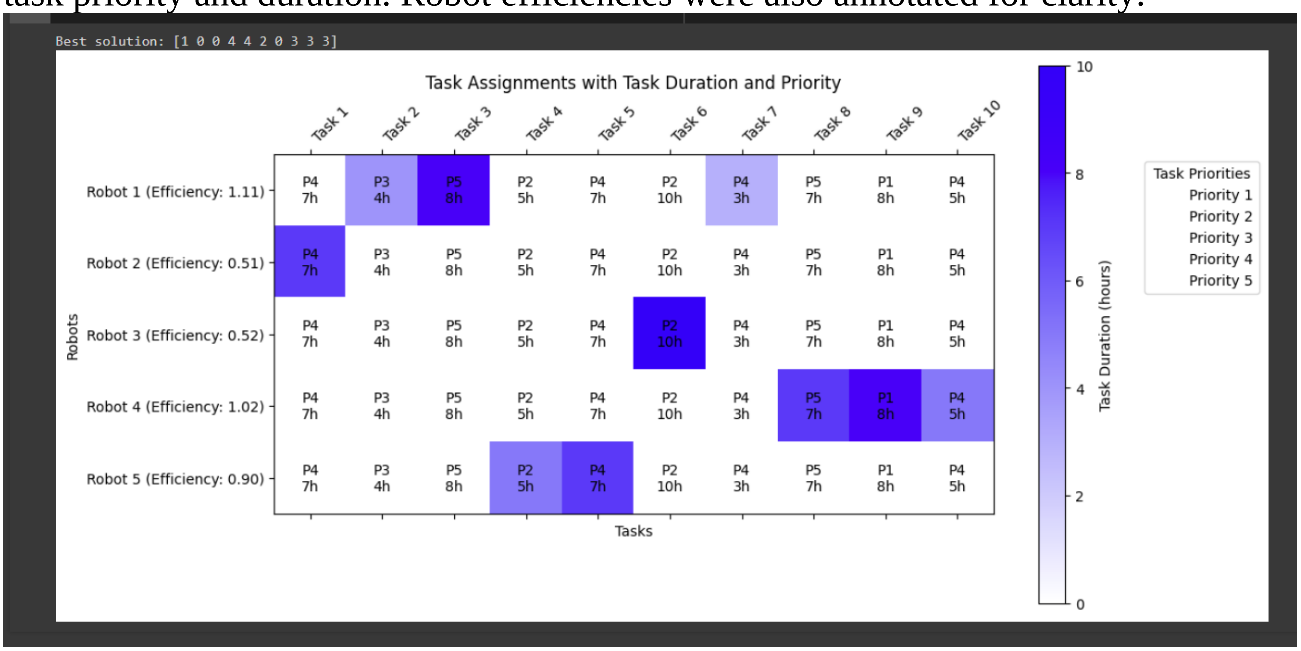

Figure: Visualized output for the Robot task optimization (Best Solution)

Analysis:

- **Influence of Robot Efficiency and Task Priority**: Robot efficiency significantly influenced the optimization process, as more efficient robots could complete tasks faster, leading to shorter production times. Task priority played a crucial role in task assignment, with higher-priority tasks being allocated to more efficient robots to ensure timely completion.
- **Optimization Performance**: The GA successfully optimized task assignments, minimizing production time and balancing workload across robots. Task priorities and robot efficiencies significantly influenced the optimization process.
- **Workload Distribution**: The GA effectively distributed the workload among robots, ensuring that no single robot was overburdened while others remained idle. However, there is potential for further optimization, particularly in dynamically adjusting task assignments based on real-time changes in task priorities and robot efficiency.

Conclusion:

In conclusion, the developed GA successfully optimized task assignments in a dynamic production environment, minimizing production time, balancing workload across robots, and prioritizing critical tasks effectively. Further enhancements could focus on real-time adaptation to changing conditions, leading to even more efficient task allocations.