EAST WEST UNIVERSITY

Assignment 2

Course Name: Artificial Intelligence
Course Code: CSE 366
Section - 3

Assignment Name: Genetic Algorithm - Robot Resource Optimization

Submitted By
Name: Tithi Paul
ID: 2021-2-60-057
Dept. of Computer Science & Engineering

Submitted To
Dr. Mohammad Rifat Ahmmad Rashid
Assistant Professor
Department of Computer Science and Engineering
East West University

Genetic Algorithm - Robot Resource Optimization

Introduction

Task assignment optimization is important in many fields where resource efficiency is critical, such as manufacturing, logistics, and robots. In this report, we present a genetic algorithm (GA) approach to optimizing task assignments to robots, considering both robot efficiency and task priority. The simulation's objectives are to assess how these variables affect optimization and examine how robot workload is distributed.

Objectives

The objectives of this simulation and analysis are as follows:

Task Assignment Optimization: Create a genetic algorithm that assigns tasks to robots in an efficient manner while taking task priority and robot efficiency into account.

Assessment of Optimization Elements: Examine how different robot efficiency levels and task priorities affect the optimization procedure and the designation of the final tasks.

Workload Distribution Analysis: Examine how jobs are assigned to robots in order to guarantee a balanced workload and effective resource use.

Identification of Optimization Opportunities: Identify potential areas for further optimization in the genetic algorithm implementation to enhance convergence speed and solution quality.

Approach

In my approach, The Genetic Algorithm evolves solutions by selecting top-performing parent solutions, merging them to create offspring, and introducing random variations through mutation. It uses simulated data on tasks and robot efficiencies, evaluating solutions based on the efficiency and balance of robot workloads.

Implementation Details

Libraries Used:

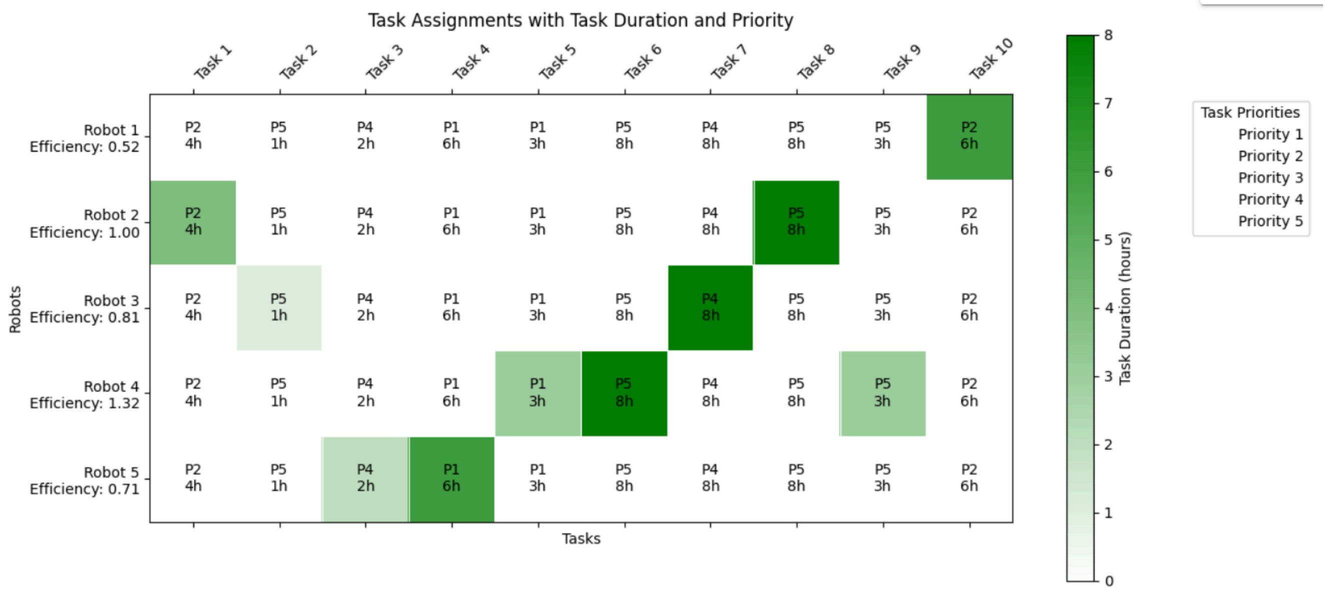
- NumPy for numerical operations.

- Matplotlib for visualization.

Functions:

- **Data generation function (`generate_mock_data`):** Random data for task durations, task priorities, and robot efficiencies are generated to simulate a realistic scenario.
- **Fitness evaluation function (`evaluate_fitness`):** Each solution's fitness is evaluated based on optimal time and effectiveness.
- **Genetic Algorithm Implementation:**
  The GA iteratively generates solutions via selection, crossover, and mutation.
  - Selection: The best solutions are chosen by parents based on fitness scores.
  - Crossover: Offspring are created by combining parts of parent solutions.
  - Mutation: Random variations are introduced to offspring solutions.
- The main function is to run the genetic algorithm (`run_genetic_algorithm`).
- **Visualization function (`visualize_assignments_improved`):** Final task assignments are visualized to understand the distribution of tasks among robots.

Simulation



	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
Robot 1 Efficiency: 0.52	P2 4h	P5 1h	P4 2h	P1 6h	P1 3h	P5 8h	P4 8h	P5 8h	P5 3h	P2 6h
Robot 2 Efficiency: 1.00	P2 4h	P5 1h	P4 2h	P1 6h	P1 3h	P5 8h	P4 8h	P5 8h	P5 3h	P2 6h
Robot 3 Efficiency: 0.81	P2 4h	P5 1h	P4 2h	P1 6h	P1 3h	P5 8h	P4 8h	P5 8h	P5 3h	P2 6h
Robot 4 Efficiency: 1.32	P2 4h	P5 1h	P4 2h	P1 6h	P1 3h	P5 8h	P4 8h	P5 8h	P5 3h	P2 6h
Robot 5 Efficiency: 0.71	P2 4h	P5 1h	P4 2h	P1 6h	P1 3h	P5 8h	P4 8h	P5 8h	P5 3h	P2 6h

Fig: 1

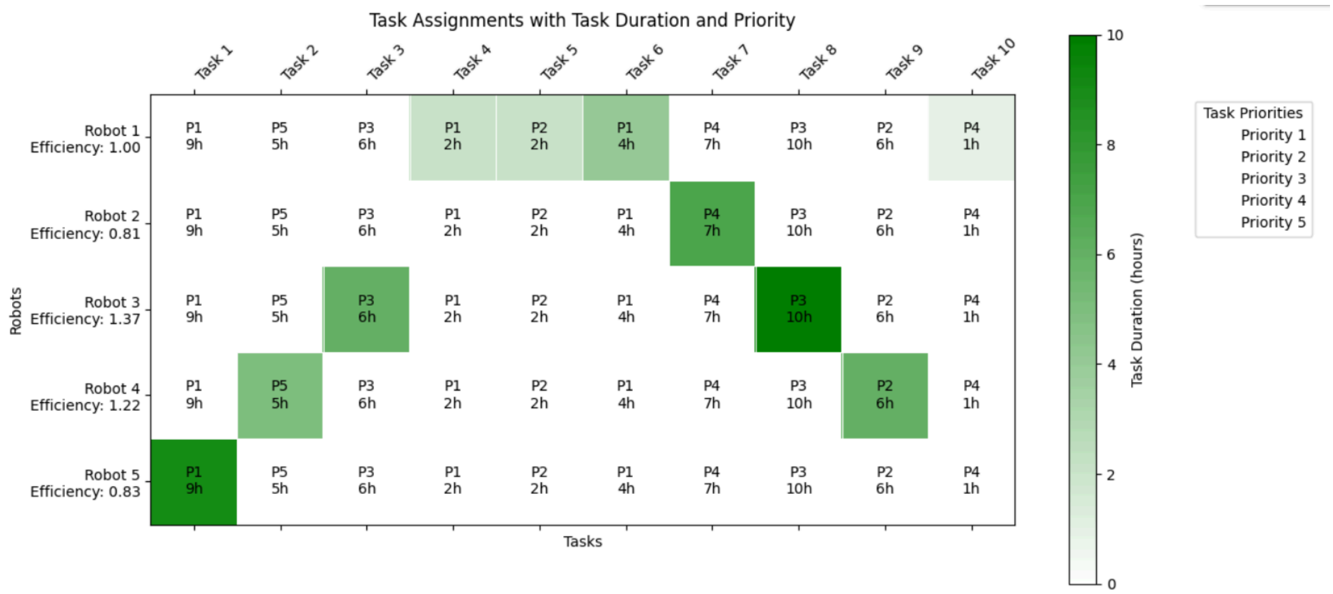


Fig: 2

Analysis of Result

The simulation results provide valuable insights into the optimization process:

1. Impact of Robot Efficiency and Task Priority:

- Higher robot efficiency reduces total time but may lead to imbalance if not distributed properly.
- Task priorities significantly influence assignment decisions, affecting both total time and balance.

2. Workload Distribution:

- The genetic algorithm effectively optimizes workload distribution among robots, leading to a balanced allocation of tasks.
- Robots with higher efficiencies handle more tasks, contributing to overall efficiency optimization.

3. Potential for Further Optimization:

- Fine-tuning parameters such as population size, mutation rate, and crossover probability present opportunities to improve convergence speed and solution quality.
- Implementing adaptive mechanisms to adjust robot efficiencies dynamically based on workload could enhance optimization further.

Conclusion

By taking robot efficiency and task priority into account, the evolutionary algorithm efficiently optimizes task assignments, distributing the workload evenly among the robots and cutting down on overall time. The method shows that it can handle complex job assignments in real-world circumstances through repeated refinement. However, even better optimization outcomes may come from additional improvements in dynamic adaptation and parameter adjustment. This thorough analysis emphasizes the need to take these variables into account to arrive at the best solutions, offering insightful advice to organizations on how to optimize job allocation procedures and boost productivity and resource usage across a range of industries.