CS348 Optimization Techniques Project Report

Project Title: Café Workflow Optimization using MIP & GA

Team Members:

- Faris Ali Alduraibi (421107654)
- Saleh Saed Alghool (422117042)
- Sulaiman Saleh Alduraibi (431108067)
- Hamad Alsuhaibani (431109462)

---

1. Introduction

In the fast-paced environment of a café, multiple customer orders must be processed simultaneously using limited equipment (e.g., espresso machines, ovens, blenders). Delays due to resource contention can lead to customer dissatisfaction and reduced throughput.
This project addresses the workflow scheduling problem using two optimization techniques:

- **Mixed Integer Programming (MIP)** for exact solutions.
- **Genetic Algorithm (GA)** for heuristic, scalable scheduling.
  The objective is to minimize the total completion time (makespan).

---

2. Problem Statement

Given:

- A set of orders with a fixed sequence of tasks (e.g., Espresso → Blending → Heating)
- A set of shared resources (e.g., Espresso machines, Blender, Oven)

Goal:

- Assign each task to a valid resource

- Ensure no two tasks using the same resource overlap
- **Minimize the overall makespan**

---

3. Mathematical Model

`Variables:`

- `assign[i, r] = 1 if task i is assigned to resource r, 0 otherwise`
- `start[i] = start time of task i`
- `makespan = maximum finish time across all tasks`

`Objective:`

- `minimize makespan`

`Constraints:`

- `Each task assigned to exactly one resource`
- `Tasks on the same resource must not overlap (Big-M constraint)`
- `Every task finishes before the makespan`

---

4. Tools and Libraries

- **Python 3.10**
- **pandas**, **numpy** — data management
- **matplotlib** — visualizations (Gantt and convergence)
- **PuLP** — MIP modeling and solver
- **Jupyter Notebook** — experimentation and presentation

---

5. Code Overview

Data Setup

- 10 orders × 3 tasks: Espresso, Blending, Heating
- Tasks mapped to compatible resources:
    - Espresso → Espresso_1 / Espresso_2
    - Blending → Blender
    - Heating → Oven
- Task durations randomly generated between 2–5 minutes

MIP (Using PuLP)

- Binary assignment variables + start times
- Big-M constraints ensure non-overlapping use of the same resource
- Solved using `PULP_CBC_CMD`

GA (Genetic Algorithm)

- Population of random task-to-resource schedules
- Fitness = makespan
- Crossover: half/half from parents
- Mutation: randomly reassigns resource
- Tracked best fitness across 50 generations
- Visualized convergence curve for insight

---

6. Results

MIP Sample Output:

Order	Task	Resource	StartTime	Duration
O1	Espresso	Espresso_1	0	3
O1	Blending	Blender	3	2
O1	Heating	Oven	5	4

MIP Makespan: 13

➢ 6.1 Gantt Chart (MIP)

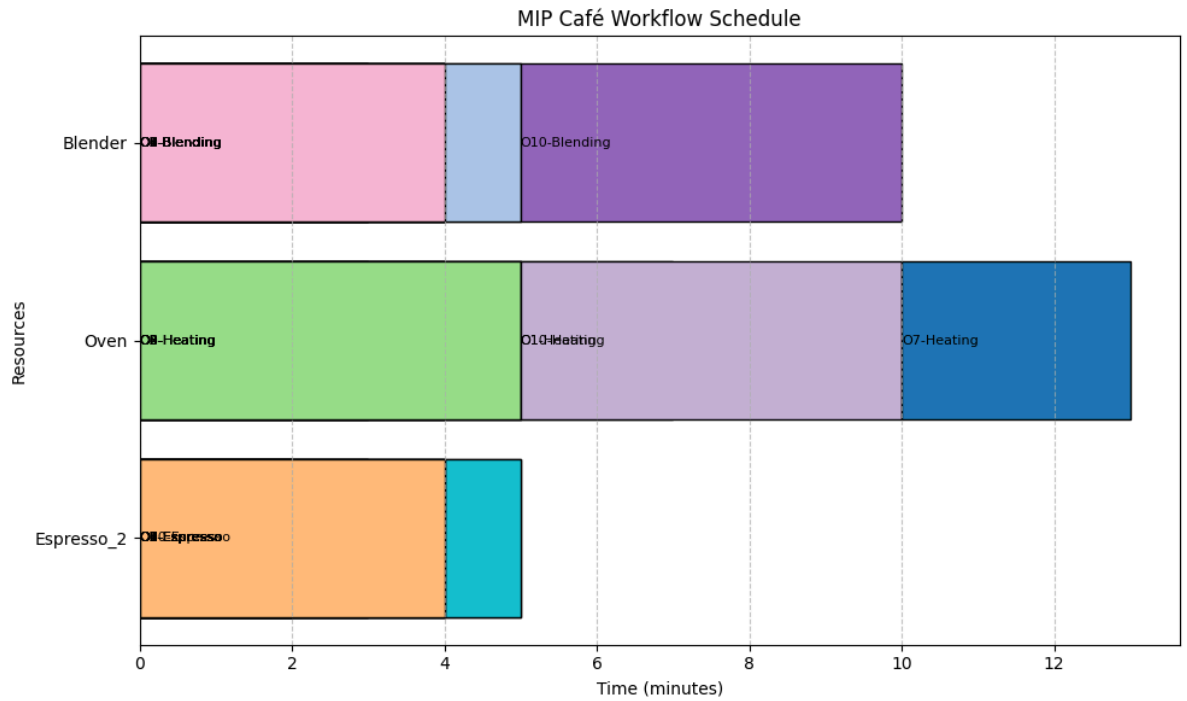

Figure 1: The figure below visualizes the task allocation across café resources based on the MIP-optimized schedule.

GA Output:

- GA schedule found heuristic schedule with makespan 41
- **Convergence plot** shows early stabilization in the best fitness across generations

➢ 6.2 GA Convergence Chart

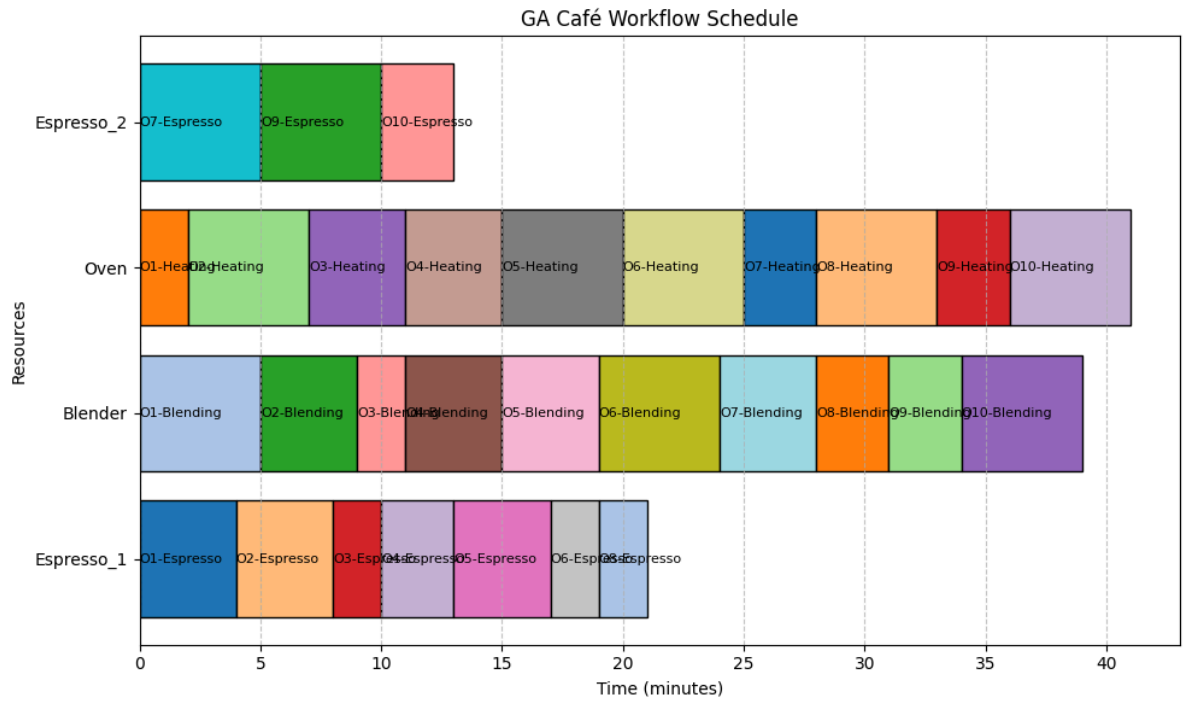

Figure 2: This chart shows the convergence behavior of the Genetic Algorithm over generations, tracking the best makespan at each iteration.

---

7. Analysis

Performance:

Method	Makespan	Approach	Scalability
MIP	13	Exact	Low
Genetic Algorithm	41	Heuristic	High

Trade-offs:

- **MIP** guarantees optimality, but is slow for large-scale problems
- **GA** is fast and scalable but may converge early or miss optimal solutions

---

8. Conclusion

This project demonstrates how two optimization techniques — MIP and GA — can be used to solve task scheduling problems in a resource-constrained environment.

- **MIP** achieved perfect scheduling with minimal makespan.
- **GA** provided a flexible, scalable alternative suitable for larger scenarios.
  Jupyter Notebook enabled step-by-step experimentation, visualization, and analysis of performance.

The entire project, including source code, notebook, and documentation, is version-controlled and publicly available on GitHub for transparency and collaboration.

---

9. Deliverables

- Python Script: `CS348_cafe_workflow.py`
- Jupyter Notebook: `project_notebook.ipynb`
- Final Report (this)
- GitHub Repository: https://github.com/FarisAlduraibi/CS348-Cafe-Workflow.git