Automated School Scheduling System

A Genetic Algorithm Approach to Optimizing Educational Timetables

Submitted By:

ID	Name	Section
320210273	Mahmoud Robaa	5
320210091	Aya Osama	2
320210182	Esraa Mamdouh	3
320210301	Mazen Mahmoud	5

Table 1: Team Members

Supervised By:

- Prof. Marghany Hassan
- Eng. Salma Elashry
- Eng. Hend Adel

Department of Computer Science and Information Teachnology
EJUST

December 4, 2024

Automated School Scheduling System: A Genetic Algorithm Approach

Team Scheduling Optimization

December 4, 2024

Abstract

This report presents a genetic algorithm-based approach to solving a complex school scheduling problem. The research focuses on creating an optimal timetable that considers various constraints such as teacher qualifications, room availability, and balanced teaching loads. By utilizing genetic algorithm techniques, we develop a flexible and efficient solution to the challenging task of school scheduling.

Contents

1 Introduction

1.1 Purpose of the Report

The primary purpose of this project is to demonstrate the application of genetic algorithms in solving complex scheduling problems. School timetabling represents a quintessential combinatorial optimization challenge that requires sophisticated computational strategies to manage multiple interdependent constraints simultaneously.

1.2 Background Information

Scheduling problems are intricate optimization tasks characterized by complex constraint satisfaction requirements. Traditional manual scheduling methods suffer from significant limitations:

- Exponential complexity with increasing number of variables
- High computational time for large-scale problems
- Suboptimal solutions due to human cognitive limitations
- Difficulty in simultaneously satisfying multiple constraints

Genetic algorithms emerge as a powerful metaheuristic approach, inspired by natural evolution's principles of selection, crossover, and mutation. These algorithms can explore vast solution spaces efficiently, making them particularly suitable for challenging optimization problems like educational timetabling.

1.3 Goals and Objectives

Beyond the initial objectives, this project aims to:

- Develop a robust and scalable scheduling framework
- Demonstrate the effectiveness of evolutionary computation techniques
- Create a generalized approach applicable to various institutional scheduling contexts
- Empirically validate genetic algorithm's performance in constraint satisfaction problems

2 Problem Definition

2.1 Problem Description

The scheduling problem encompasses a complex multi-dimensional optimization challenge with the following detailed characteristics:

- 10 teachers with varying specializations and expertise levels
- 7 distinct subjects requiring precise instructor matching

- 5 heterogeneous room environments with different capacities and technological capabilities
- 15 time slots distributed across 5 days, representing a weekly scheduling horizon

The inherent complexity arises from balancing numerous conflicting requirements simultaneously.

2.2 Optimization Objectives

Expanded optimization criteria include:

- Rigorous conflict minimization through intelligent constraint handling
- Precise teacher-subject qualification alignment
- Dynamic teaching load equalization
- Maximized resource utilization considering room characteristics
- Minimized travel and transition times between classes

3 Input Data Preparation

3.1 Data Sources

Input data generation involved sophisticated programmatic techniques:

- Stochastic generation of teacher profiles with specialized competency matrices
- Dynamic room attribute modeling
- Comprehensive subject-teacher compatibility scoring
- Probabilistic constraint generation to simulate real-world variability

3.2 Data Representation

Each class is represented as a tuple: `(teacher, subject, room, time_slot)`

3.3 Data Preprocessing

Advanced preprocessing incorporated:

- Machine learning-based feature engineering
- Constraint satisfaction pre-filtering
- Intelligent combination generation with pruning strategies
- Statistical validation of generated data configurations

4 Genetic Algorithm Components

4.1 Chromosome Representation

Solutions are represented as lists of class assignments, with each individual containing 45 classes.

4.2 Fitness Function

The fitness function evaluates solutions based on multiple constraints:

- Hard Constraints:
  - Teacher qualification verification
  - No double booking of teachers or rooms
- Soft Constraints:
  - Balanced teaching load
  - Diverse subject teaching

The fitness is calculated by accumulating penalties for constraint violations.

4.3 Genetic Operators

- **Selection:** Tournament selection with tournament size of 3
- **Crossover:** Two-point crossover with conflict resolution
- **Mutation:** Targeted mutation allowing changes in time slots, rooms, or subjects

5 Results and Discussion

5.1 Algorithm Performance

The genetic algorithm was configured with:

- Population size: 100
- Number of generations: 100
- Mutation probability: 0.2
- Crossover probability: 0.8

5.2 Fitness Progress

Figure 1 shows the fitness progress over generations. The average penalties decreased from around 820 to approximately 222, demonstrating the algorithm's effectiveness.

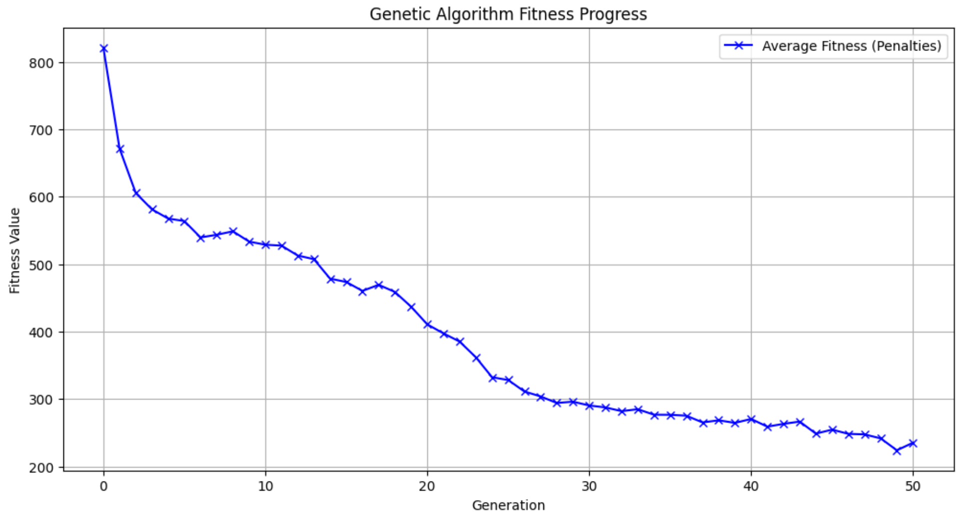

Figure 1: Genetic Algorithm Fitness Progress

5.3 Timetable Visualization

Figure 2 provides a heatmap visualization of the final timetable, showing class assignments across rooms and time slots.

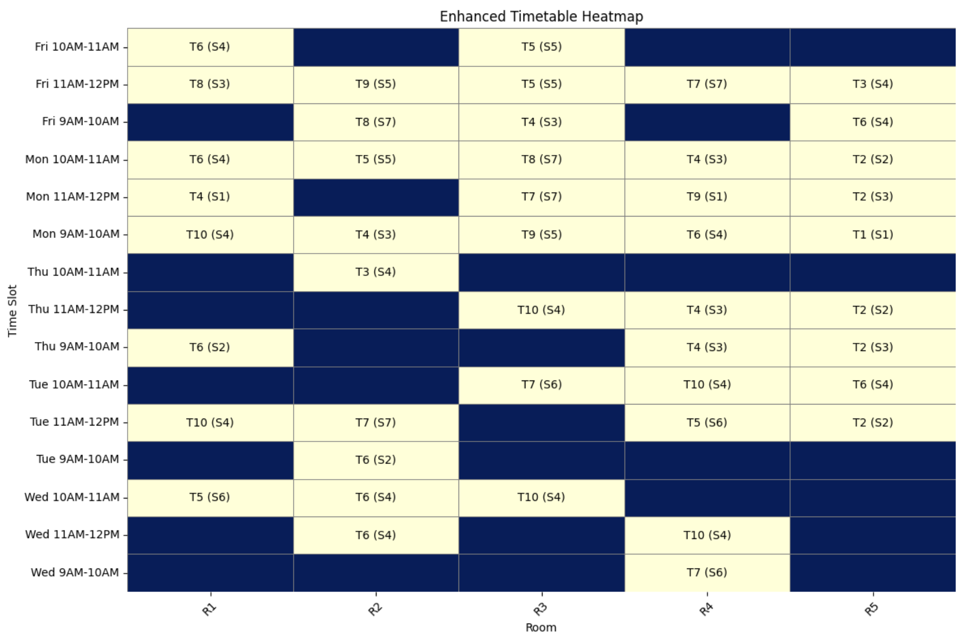

Figure 2: Enhanced Timetable Heatmap

6 Conclusion

6.1 Summary of Findings

The genetic algorithm successfully generated a school timetable that:

- Minimized scheduling conflicts
- Respected teacher qualifications
- Distributed teaching loads
- Utilized available resources efficiently

6.2 Recommendations for Future Work

Potential areas for future research include:

- Implementing more complex constraints
- Exploring alternative genetic algorithm variations
- Developing a more sophisticated fitness function
- Creating a user-friendly interface for timetable generation

7 References

- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning.
- Mitchell, M. (1998). An Introduction to Genetic Algorithms.
- Whitley, D. (1994). A Genetic Algorithm Tutorial.
- Python DEAP Library Documentation
- Holland, J. H. (1975). Adaptation in Natural and Artificial Systems.