Event Management Scheduling Optimization

1 Introduction

This report describes the optimization of a 5-day event scheduling problem using a genetic algorithm. The event will host 600 attendees and involve 16 workshops distributed across four session types: Theoretical, Practical, Historical, and Test. The goal is to find a timetable that satisfies all constraints, such as room capacities, session type distribution, teacher availability, and time preferences. We use a metaheuristic approach to generate an optimal solution.

2 Problem Overview

The event spans five days, from Sunday to Thursday. The rooms available are:

- 5 classrooms, each with a capacity of 60 attendees.
- 1 amphitheater with a capacity of 180 attendees.

The attendees must attend 16 sessions, distributed across four session types:

- 4 Theoretical Sessions
- 4 Practical Sessions
- 4 Historical Sessions
- 4 Test Sessions

The topics covered in all session types are:

- A: AI in Healthcare
- B: Ethics in AI
- C: Generative AI
- D: Prompt Engineering

3 Constraints and Penalization

The genetic algorithm needs to satisfy several constraints to generate a valid timetable. The constraints and their associated penalties are as follows:

3.1 Room Conflicts

Each room can only be assigned to one session at any given time. If a room is double-booked, the solution receives a penalty of 100.

3.2 Teacher Conflicts

A teacher should not be scheduled to conduct multiple sessions at the same time. If a teacher is double-booked, the solution receives a penalty of 150.

3.3 Time Ordering

Sessions need to follow a specific order (Theoretical $\rightarrow$ Practical $\rightarrow$ Test). If this order is violated, a penalty of 50 is applied.

3.4 Session Type Distribution

Each session type (Theoretical, Practical, Historical, Test) should have exactly four sessions for each topic. If the number of sessions for any type is incorrect, a penalty of 20 times the absolute difference from 4 is applied.

3.5 Time Preferences

Some sessions should ideally be in specific time slots. For example, Theoretical sessions should be scheduled in the first or second slot, while Historical sessions should be in the third or fourth slot. A penalty of 10 is applied if these preferences are violated.

3.6 Attendee Session Coverage

Each attendee must attend exactly one session for each session type and topic. A penalty of 10 is applied for every missing session.

3.7 Empty Slots

Empty slots are penalized as they represent inefficiency in the scheduling. A penalty of 10 is applied for every empty slot.

4 Genetic Algorithm

4.1 Overview

The genetic algorithm aims to evolve a population of possible schedules, applying crossover, mutation, and selection to produce the best possible solution. The algorithm operates as follows:

1. **Initialization**: A random population of solutions is generated.
2. **Fitness Evaluation**: Each solution is evaluated using the fitness function, which applies penalties for violating the constraints.
3. **Selection**: The best solutions are selected to reproduce.
4. **Crossover**: Two parent solutions are combined to produce offspring.
5. **Mutation**: Small random changes are introduced to the offspring.
6. **Termination**: The algorithm terminates either after a fixed number of generations or when an optimal solution is found.

4.2 Genetic Operations

4.2.1 Crossover

The crossover operation combines two parent solutions to create a child solution. A uniform crossover method is applied, where each session from the parent solutions is randomly selected to form the child. Occasionally, new random sessions are introduced to increase diversity.

Algorithm 1 Crossover

```
Input: Two parent chromosomes parent1, parent2
Output: A child chromosome
for each session in the solution do
  if random choice then
    Add session from parent1
  else
    Add session from parent2
  end if
end for
End
```

4.2.2 Mutation

The mutation operation introduces random changes to a chromosome. A mutation rate determines the likelihood of mutation, and it could involve partial mutation (changing one session) or full reassignment (changing the entire session).

4.2.3 Selection

The selection process uses tournament selection to choose parents based on their fitness scores. Individuals with better fitness have a higher probability of being selected.

4.3 Fitness Function

The fitness function evaluates each solution by calculating the total penalty based on violations of constraints, including room conflicts, teacher conflicts, session ordering, and others. A lower fitness score indicates a better solution.

Algorithm 2 Mutation

```
Input: A chromosome, mutation rate
Output: A mutated chromosome
if random mutation chance then
    Choose between partial or full mutation
    if partial mutation then
        Change a random session
    else
        Reassign a random session
    end if
end if
End
```

Algorithm 3 Selection

```
Input: Population, fitness scores
Output: Selected population
for each individual in the population do
    Perform tournament selection
    if selected individual is better then
        Add to selected population
    end if
end for
End
```

4.3.1 Fitness Evaluation Example

The penalty is calculated as follows:

- Room conflicts: 100 penalty points for double-booked rooms.
- Teacher conflicts: 150 penalty points for double-booked teachers.
- Time ordering violations: 50 penalty points for incorrect session ordering.
- Session type distribution errors: 20 times the absolute difference from the expected count of 4.
- Time preferences violations: 10 penalty points for incorrect time slots.
- Attendee session coverage: 50 penalty points for missing sessions.
- Empty slots: 40 penalty points for unused slots.

5 Results and Evaluation

5.1 Effectiveness of the Algorithm

The genetic algorithm effectively explores the solution space, satisfying most constraints while minimizing penalties. It performs well in optimizing the scheduling problem, producing solutions with reduced room conflicts, teacher conflicts, and violations of session ordering.

5.2 Visualization of the Optimization

The scores of each generation are tracked to show the convergence of the genetic algorithm. The following plot illustrates the best fitness score over generations:

5.3 Schedule Output

Day	Slot Time	Session	Room	Teacher

Day	Slot Time	Session	Room	Teacher
Sunday	8:00 - 10:00	A_Theory	Classroom4	A2
		A_History	Classroom1	A1
		C_Practical	Classroom4	C1
		C_History	Classroom2	C2
		D_History	Amphitheater	D2
Sunday	10:30 - 12:30	A_History	Classroom4	A2
		A_Test	Classroom5	A1
		B_Practical	Amphitheater	B2
		B_Practical	Classroom4	B1
		C_Test	Classroom1	C1
Sunday	13:30 - 15:30	A_Theory	Classroom3	A2
		B_Practical	Amphitheater	B1
		B_History	Amphitheater	B2
		C_Practical	Amphitheater	C2
Sunday	16:00 - 18:00	B_History	Classroom5	B1
		B_History	Classroom1	B2
		C_History	Classroom3	C2
		D_Practical	Classroom1	D1
		D_Practical	Classroom5	D2
Monday	8:00 - 10:00	A_Test	Classroom5	A2
		B_Theory	Classroom2	B2
		C_Theory	Classroom1	C2
		C_History	Classroom4	C1
		D_Practical	Classroom4	D2
Monday	10:30 - 12:30	B_Test	Classroom4	B1
		C_Theory	Classroom1	C2
		C_Test	Classroom3	C1
Monday	13:30 - 15:30	A_History	Classroom4	A1
		B_History	Classroom4	B1
		C_Test	Classroom1	C1
Monday	16:00 - 18:00	A_History	Classroom3	A1
		B_Practical	Amphitheater	B2
		C_Theory	Classroom5	C2
		D_History	Classroom3	D2
		D_Test	Classroom1	D1
Tuesday	8:00 - 10:00	A_Practical	Classroom1	A1

Day	Slot Time	Session	Room	Teacher
		A_History	Classroom1	A2
		B_History	Classroom2	B1
		C_Test	Classroom3	C1
		D_Theory	Classroom5	D2
		D_History	Classroom4	D1
Tuesday	10:30 - 12:30	A_Theory	Classroom2	A1
		C_Theory	Amphitheater	C1
		C_Practical	Amphitheater	C2
		D_Theory	Classroom4	D1
		D_Theory	Classroom3	D2
Tuesday	13:30 - 15:30	A_Theory	Classroom1	A2
		B_Theory	Classroom4	B1
		B_Practical	Classroom5	B2
		C_Test	Classroom5	C1
		C_Test	Classroom2	C2
		D_Practical	Classroom5	D2
		D_Test	Amphitheater	D1
Tuesday	16:00 - 18:00	A_Test	Classroom1	A1
		B_Test	Classroom5	B2
		D_Test	Classroom2	D1
Wednesday	8:00 - 10:00	A_Practical	Classroom5	A1
		A_Test	Classroom1	A2
		B_Practical	Classroom4	B2
		C_Theory	Classroom2	C2
		C_Practical	Classroom5	C1
		D_Theory	Classroom3	D1
Wednesday	10:30 - 12:30	A_Practical	Classroom3	A1
		B_Theory	Classroom2	B2
		B_History	Classroom4	B1
		D_Practical	Classroom5	D2
		D_Test	Classroom1	D1
Wednesday	13:30 - 15:30	A_Practical	Classroom5	A1
		A_Test	Classroom3	A2
		B_Test	Classroom2	B2
		C_Practical	Classroom2	C1
Wednesday	16:00 - 18:00	A_Theory	Classroom4	A2

Day	Slot Time	Session	Room	Teacher
		A_History	Classroom1	A1
		A_Test	Classroom2	A1
		B_Theory	Classroom4	B2
		B_Test	Classroom5	B1
		C_History	Classroom3	C1
		D_Practical	Classroom2	D2
Thursday	8:00 - 10:00	A_Theory	Amphitheater	A2
		C_Practical	Classroom4	C1
		D_Theory	Classroom3	D2
		D_History	Classroom1	D1
Thursday	10:30 - 12:30	B_Theory	Amphitheater	B1
		B_Theory	Classroom5	B2
		C_Theory	Classroom1	C2
		C_History	Classroom1	C1
		D_Test	Amphitheater	D1
		D_Test	Classroom4	D2
Thursday	13:30 - 15:30	B_Test	Classroom4	B1
		D_Theory	Classroom1	D1
		D_History	Classroom3	D2
Thursday	16:00 - 18:00	A_Practical	Classroom2	A2
		A_Practical	Amphitheater	A1
		B_Test	Classroom2	B1
		C_History	Classroom1	C2
		D_History	Classroom2	D1

6 Conclusion

The genetic algorithm provides an efficient solution to the event scheduling problem by minimizing violations of constraints and optimizing the timetable. The method ensures that all participants attend their required sessions, respects room capacities, teacher availability, and adheres to session type distribution and time preferences.

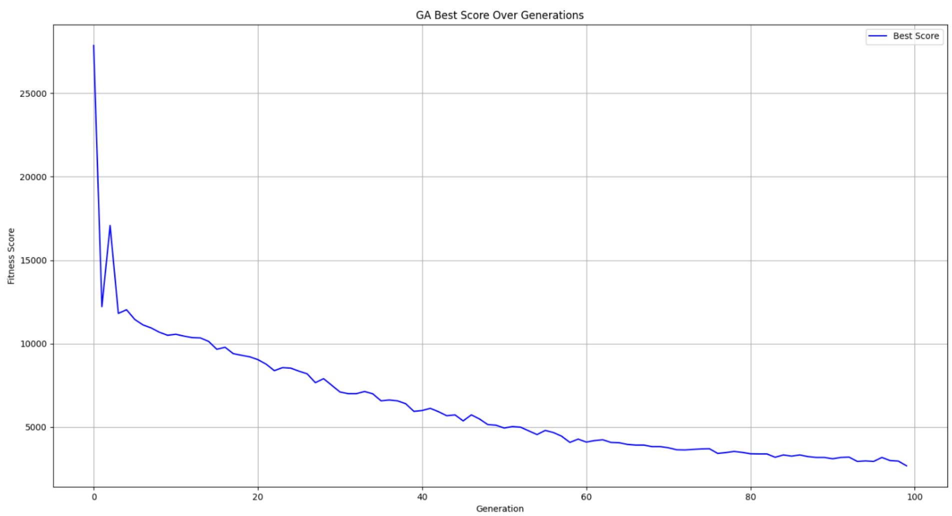

Figure 1: Best Fitness Score Over Generations