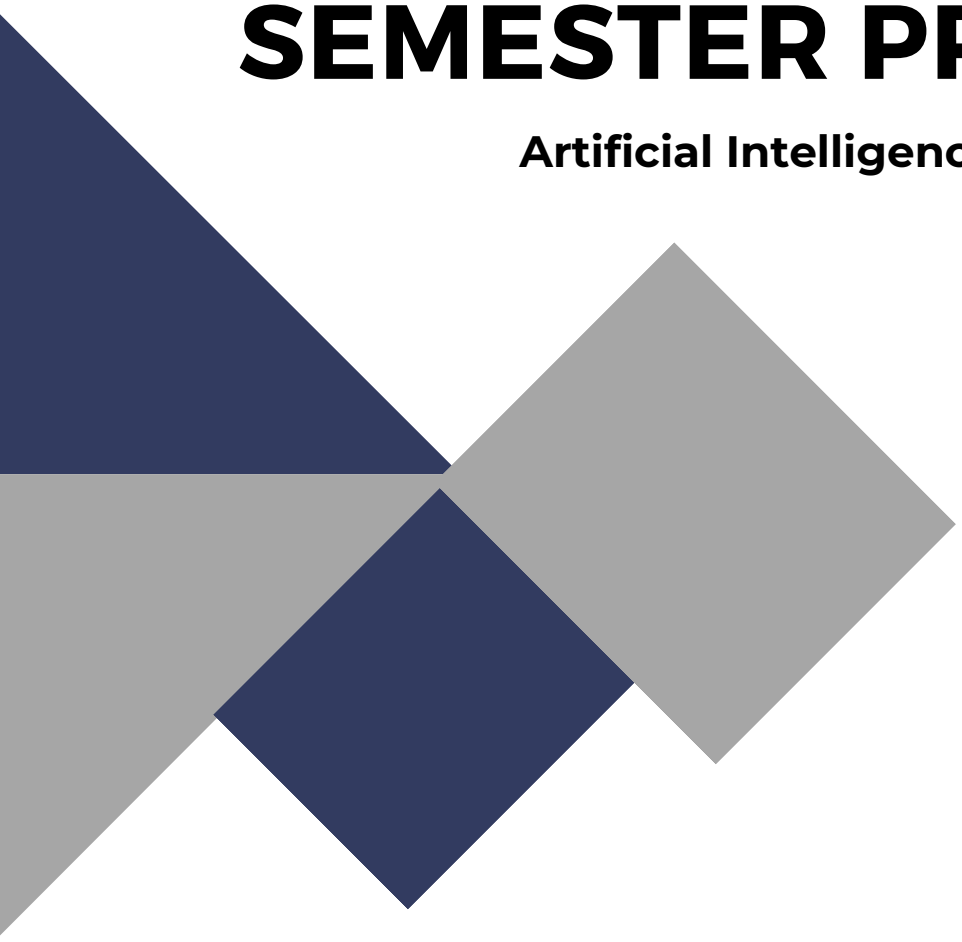
SEMESTER PROJECT

Artificial Intelligence



SPRING 2024

Section: Y

TIMETABLE SCHEDULING USING GENETIC ALGORITHM

Muhammad Kashif

21I-0851

Table of Contents

Timetable Scheduling

Artificial Intelligence Project Report

Introduction

In the modern academic landscape, efficient timetable scheduling is pivotal for universities to ensure smooth operations and optimal utilization of resources. This project delves into the classical problem of timetable scheduling, aiming to create schedules for each semester with minimal clashes between sections, professors, and rooms. The primary goal is to design a robust algorithmic solution that automates the timetable creation process while adhering to a set of hard and soft constraints.

The key challenges addressed in this project include:

- Ensuring classes are scheduled in available classrooms.
- Accommodating section sizes with appropriate room capacities.
- Avoiding scheduling conflicts for professors and sections.
- Optimizing the allocation of courses to professors and sections.
- Minimizing the number of clashes and interruptions in the timetable.

To tackle these challenges, a genetic algorithm approach is employed, leveraging binary encoding for chromosomes and a fitness function that penalizes clashes and conflicts. The algorithm iteratively refines schedules to achieve optimal solutions that meet the specified constraints.

This report provides a concise overview of the project's methodology, including data representation, constraint handling, algorithm implementation, and results analysis. It showcases the steps involved in the project and presents the outcomes achieved through the application of the genetic algorithm to timetable scheduling.

Data Preparation and Representation

The project starts by importing necessary libraries such as **Counter** from collections, **pandas** for data manipulation, **numpy** for numerical operations, **openpyxl** for Excel file handling, and **random** for generating random numbers. These libraries are essential for data preparation, representation, and algorithm implementation

Resource Initialization

For the scheduling of both theory and lab sessions, essential resources such as rooms, labs, lab courses, and lab instructors were initialized. This ensures that the algorithm has access to crucial information for optimizing schedules effectively.

I initialized a few lists for the lab courses, labs, and lab instructors. For theory courses, I utilized an Excel file named 'Courses.xlsx', which contained course allocations across different departments.

Genetic Algorithm Parameters

To guide the optimization process, the following parameters were defined:

- Population Size: I set the population size to 32 chromosomes, representing potential timetable schedules.
- Number of Generations: The algorithm evolved over 10,000 generations to refine the schedules.
- Mutation Probability: A mutation probability of 0.085 was applied, determining the likelihood of introducing changes in the chromosomes during evolution.

Creating and Cleaning DataFrame

The **create_and_clean_dataframe** function is designed to handle the conversion of Excel data into a structured Pandas DataFrame, adhering to specific constraints based on the sheet name provided. This function also performs data cleaning operations to ensure the DataFrame is ready for further processing.

I implemented this function to read the Excel file and extract relevant information, such as course, section, course instructor, and other details based on the sheet provided. If the sheet does not match any available sheet names, an exception is raised to handle errors gracefully. Additionally, the function merges the 'Section' column with 'Batches' and drops any rows with missing values in key columns.

Mapping Timetable to DataFrame

The **map_timetable_to_dataframe** function plays a crucial role in mapping the generated timetable (represented as a list) to an empty timetable DataFrame. This mapping ensures that the timetable data is structured and organized in a format that can be easily analyzed and manipulated.

In this function, each element in the generated timetable list is mapped to the corresponding index in the DataFrame, representing rooms and timeslots. This process populates the DataFrame with scheduled courses, instructors, and other relevant details for each timeslot and room combination.

Binary Encoding of Chromosome Information

The **binary_encode_chromosome** function is a crucial step in the genetic algorithm's chromosome representation. This function converts the given chromosome information into binary format, facilitating efficient manipulation and optimization during the timetable scheduling process.

I implemented this function to binary encode the chromosome information provided in a list format. The information includes course details, type (theory/lab), section, section strength, professor, lecture days, timeslots, rooms, and room sizes. Each element in the list is converted to its binary representation, removing the '0b' prefix typically seen in Python's binary representation.

The binary encoded chromosome is structured in the same order as the input chromosome information, ensuring that essential details are preserved and can be manipulated effectively during the genetic algorithm's evolution process.

This binary encoding process is fundamental for representing timetable schedules as chromosomes in a format that the algorithm can operate on, optimizing schedules based on defined constraints and preferences.

Fitness Evaluation Function

The **fitness** function plays a crucial role in evaluating the fitness score of a given timetable chromosome based on a set of hard and soft constraints.

The hard constraints include:

- Ensuring teachers and students do not have overlapping classes.
- Limiting the number of classes per day for teachers and students.
- Penalizing heavily for violations of these constraints.

The soft constraints include:

- Room capacity constraints.
- Section-room and room-section assignments.
- Professor workload constraints.
- Section course limits.

The function calculates a score based on how well the timetable meets these constraints, with higher scores indicating better adherence to the constraints and a more optimized timetable schedule.

Selection, Crossover, and Mutation Functions

The **selection**, **crossover**, and **mutation** functions are essential components of the genetic algorithm's evolution process.

- The **selection** function identifies the top-performing chromosomes (parents) based on their fitness scores. In this implementation, a simple top fitness selection method is used to select the top 4 parents from the population.
- The **crossover** function combines genetic material from two parents to create two offspring (children). The chromosomes are split into equal parts, and the children inherit different parts from each parent.
- The **mutation** function introduces random changes (mutations) in the offspring chromosomes with a predefined probability. This helps maintain diversity in the population and explore new solutions during evolution.

Genetic Algorithm Function

The **genetic_algorithm** function orchestrates the entire genetic algorithm process for timetable scheduling optimization. This function takes the initial population of timetables as its parameter and evolves the population over a specified number of generations.

The key steps in the genetic algorithm process include:

1. Selection: Parents with higher fitness scores are selected from the population for reproduction. A simple top fitness selection method is used in this implementation.

2. Crossover: Parents produce offspring (children) through crossover operations, where genetic material is exchanged between parents to create diverse offspring.
3. Mutation: Offspring undergo random mutations with a predefined probability, introducing variability in the population and exploring new solutions.
4. New Population Creation: A new population is created in each generation by combining parents, performing crossover, and introducing mutations.
5. Fitness Evaluation and Improvement: The fitness scores of chromosomes in the population are evaluated, and improvements are made iteratively to enhance overall fitness.
6. Selection of Best Chromosome: At the end of the specified generations, the chromosome with the highest fitness score is selected as the optimal timetable schedule.

The function iteratively refines the population, striving to improve the fitness score of chromosomes with each generation. It stops either when the maximum fitness score (100 in this case) is achieved or after completing the specified number of generations.

The genetic algorithm process encompasses selection, crossover, mutation, and fitness evaluation, culminating in the identification of the best timetable schedule that meets defined constraints and optimization criteria.

Empty Timetable DataFrame Creation

An empty timetable DataFrame is initialized to structure the timetable scheduling process. This DataFrame, named **timetable**, is organized with time slots as columns and rooms as indices, providing a framework for scheduling courses and allocating resources efficiently.

Course Allocation Data Preparation

The course allocation data from different sheets in the 'Courses.xlsx' file is read and cleaned using the **create_and_clean_dataframe** function. Sheets such as 'Computing-Theory', 'S&H', and 'MG' are processed and merged into a unified DataFrame, **all_courses_df**. This DataFrame contains essential course information like course names, sections, instructors, and other details.

Alleles Generation

All possible alleles representing course options are generated based on the course allocation data. These alleles encompass a variety of course combinations, ensuring diversity in timetable scheduling possibilities. The **alleles** list contains these course options, facilitating chromosome creation during the genetic algorithm process.

Initial Population Generation

The initial population of chromosomes, representing individual timetable schedules, is created using the generated alleles and the **population_size** parameter. Each chromosome comprises randomly selected alleles for course scheduling across rooms and time slots, forming the basis for evolutionary optimization through the genetic algorithm.

This initialization phase sets the stage for timetable scheduling optimization, leveraging data structures and genetic algorithm principles to generate optimal timetable schedules that adhere to defined constraints and preferences.

DataFrame Creation and Styling

The timetable DataFrame is created to visualize lab assignments or timetable schedules. After populating the DataFrame with lab assignments data, styling is applied to enhance readability and aesthetics using Pandas styling capabilities.

Genetic Algorithm Execution

The genetic algorithm is executed on the specified initial population and generation size using the **genetic_algorithm** function. This process involves iterative evolution of the population, improving fitness scores, and generating optimal timetable schedules.

The algorithm runs until either the maximum fitness score (100 in this case) is achieved, or the predefined number of generations is completed. In the screenshot provided, the algorithm successfully found a chromosome with the maximum possible fitness score, indicating an optimized timetable schedule.

Fitness Evaluation of Final Timetable

Upon completion of the genetic algorithm, the fitness score of the final generated timetable is evaluated using the **fitness** function. The fitness score provides a quantitative measure of how well the timetable adheres to defined constraints and optimization criteria.

The output of the fitness evaluation indicates the fitness score of the final timetable, which in my case is 100 out of a maximum possible score of 100.

Output and Results

The output of my timetable scheduling algorithm showcases the successful optimization of timetable schedules while adhering to various constraints and preferences. The screenshots below depict key outputs and results of my program.

```
Chromosome of maximum possible fitness found!
Generation:  3232
Fitness score of the final timetable: 100 (max possible = 100)
```

FAST Timetable (Monday)

	08:30-09:50	10:05-11:25	11:40-01:00	01:15-02:35	02:50-04:10	04:25-05:45
Room						
C301	BS Batch 2022 , SE-2B Introduction to Software Engineering (SE) Ms. Saba Kanwal	BS Batch 2020 , AI-6J Fundamentals of Natural Language Processing (AI) Dr. Omer Beg	BS Batch 2020 and Earlier Batches Repeat Course , AI-A Machine Learning (BS-AI) Dr. Uzair Iqbal	BS Batch 2022 , DS-2C Differential Equations (DS) Sumaira Azhar	BS Batch 2022 , DS-2C Islamic Studies/Ethics (DS) Abdul Salam	MS Elective Courses (All Programs) , MCS-A Data Visualization Dr. Faisal Cheema
C302	BS Batch 2021 , CS-4B/Z Database Systems (CS) Dr. Shujaat Hussain	BS Batch 2022 , CS-2A Digital Logic Design (CS) Ms. Nirmal Tariq	BS Batch 2022 , SE-2E Introduction to Software Engineering (SE) Dr. Khubaib Amjad	BS Batch 2020 , CS-6C Artificial Intelligence (CS) Mr. Saad Salman	BS Batch 2019 , CY-T Security Operations and Administration Mr. M. Ahsan Shakeel	BS Batch 2020 , SE-6Q Web Engineering (SE) Mr. Zaheer Sani
C303	BS Batch 2022 , CS-2A Object Oriented Programming (CS) Mr. Shams Farooq	BS Batch 2021 and Earlier Batches Repeat Course , C Data Structures (All Programs) Ms. Amna Irum	BS Batch 2021 , CS-4B Fundamentals of Management (CS) Ayesha Yaqoob	BS Batch 2021 and Earlier Batches Repeat Course , D Data Structures (All Programs) Mr. Bilal Khalid Dar	BS Batch 2022 , DS-2B Differential Equations (DS) Arif Hussain	MS (CNS) , MCNS-A Applied Programming M. Adil ur Rehman
C304	PhD , PCS-2A Advanced Machine Learning Dr. Usman Habib	BS Batch 2020 , CS-6F Statistical Modelling Dr. Mukhtar Ullah (EE)	BS Batch 2021 , CS-4D Fundamentals of Management (CS) Muhammad Hassaan	BS Batch 2019 , CS-8B Digital Marketing (CS) Husbi Ahmed	BS Batch 2021 , CS-4A/Y Design and Analysis of Algorithms (CS) Mr. Owais Idrees	BS Batch 2020 , SE-6P Applied Artificial Intelligence (SE Elective - I) Ms. Shahela Saif

FAST Timetable (Monday LABS)

	Time Slot	Lab Course	Lab Section	Instructor	Lab Location
0	08:30-11:30	Digital Logic Design Lab (CS)	CS-2H	Ms. Ifrah Masood (EE)	C-Rawal 2
1	08:30-11:30	Digital Logic Design Lab (CS)	CS-2E	Mr. Abdul Hannan	B-Digital
2	08:30-11:30	Digital Logic Design Lab (CS)	CS-2B	Mr. Abdul Hannan1	C-Margalla 2
3	08:30-11:30	Digital Logic Design Lab (CS)	CS-2J	Mr. Fakhar Abbas (EE)	C-GPU Lab
4	08:30-11:30	Digital Logic Design Lab (CS)	CS-2D	Mr. Abdul Hannan2	A-CALL 2
5	08:30-11:30	Digital Logic Design Lab (CS)	CS-2F	Ms. Ifrah Masood1 (EE)	A-Karakoram 1
6	08:30-11:30	Digital Logic Design Lab (CS)	CS-2K	Mr. Fakhar Abbas1 (EE)	C-Rawal 3
7	08:30-11:30	Digital Logic Design Lab (CS)	CS-2G	Mr. Abdul Hannan3	C-Rawal 1
8	08:30-11:30	Digital Logic Design Lab (CS)	CS-2A	Mr. Fakhar Abbas1 (EE)	A-Karakoram 2
9	08:30-11:30	Digital Logic Design Lab (CS)	CS-2C	Ms. Ifrah Masood2 (EE)	C-Margalla 1
10	11:45-02:45	Object Oriented Programming Lab (CS)	CS-2J	Ms. Sher Bano	C-Rawal 2
11	11:45-02:45	Object Oriented Programming Lab (CS)	CS-2A	Ms. Sher Bano1	B-Digital
12	11:45-02:45	Object Oriented Programming Lab (CS)	CS-2E	Ms. Amna Hassan	C-Margalla 2

FAST Timetable (Tuesday)

	08:30-09:50	10:05-11:25	11:40-01:00	01:15-02:35	02:50-04:10	04:25-05:45
C301	('Final Year Project-I (DS)', 'BS Batch 2019 , DS-N', 'FYP Committee')	('Statistical Modelling', 'BS Batch 2020 , CS-6D', 'Dr. Shahzad Saleem (EE)')	('Artificial Intelligence (AI)', 'BS Batch 2021 , AI-4K', 'Mr. Umair Arshad')	('Object Oriented Programming (CS)', 'BS Batch 2022 , CS-2J', 'Ms. Marium Hida')	('Software Design and Architecture (SE)', 'BS Batch 2021 , SE-4P', 'Dr. Uzair Khan')	('Programming Fundamentals (All Programs)', 'BS Batch 2022 and Earlier Batches Repeat Courses , E', 'Mr. Majid Hussain')
C302	('Digital Logic Design (CS)', 'BS Batch 2022 , CS-2F', 'Dr. Niaz Ahmed (EE)')	('Statistical Modelling', 'BS Batch 2020 , CS-6C', 'Mr. Muhammad Almas Khan')	('Object Oriented Programming (AI)', 'BS Batch 2022 , AI-2C', 'Mr. Adil Majeed')	('Artificial Intelligence (CY)', 'BS Batch 2020 , CY-6T', 'Mr. Shoaib Saleem')	('Process Mining and Simulation (SE Supporting-III)', 'BS Batch 2020 , SE-6P', 'Mr. Irfan Ullah')	('Information Security (SE)', 'BS Batch 2020 , SE-6Q', 'Dr. Mudassar Aslam')
C303	('Advanced Statistics (DS)', 'BS Batch 2021 , DS-4M', 'Dr. Ahmed Din')	('Introduction to Software Engineering (SE)', 'BS Batch 2022 , SE-2G', 'Dr. Isma ul Hassan')	('Design and Analysis of Algorithms (CS)', 'BS Batch 2021 , CS-4C/D', 'Dr. Kashif Munir')	('Networks and Cyber Security-II (CY)', 'BS Batch 2020 , CY-6T', 'Dr. Qaiser Shafi')	('Programming Fundamentals (All Programs)', 'BS Batch 2022 and Earlier Batches Repeat Courses , H', 'Dr. Muneeb Ullah')	('Database Systems (AI)', 'BS Batch 2021 , AI-4J', 'Ms. Ayesha Kamran')
C304	('Discrete Structures (DS)', 'BS Batch 2022 , DS-2C', 'Dr. Abid Rauf')	('Discrete Structures (SE)', 'BS Batch 2022 , SE-2C', 'Ms. Shafaq Riaz')	('Discrete Structures (SE)', 'BS Batch 2022 , SE-2D', 'Mr. Majid Hussain')	('Programming Fundamentals (All Programs)', 'BS Batch 2022 and Earlier Batches Repeat Courses , C', 'Dr. Subhan Ullah')	('Artificial Neural Networks (AI)', 'BS Batch 2020 , AI-6K', 'Mr. Hassan Raza')	('Software Engineering (CS)', 'BS Batch 2020 , CS-6C', 'Dr. Ejaz Ahmed')