Universiti Teknologi MARA | Cawangan Melaka

PCMJ

Progress in Computing and Mathematics Journal

volume 1

https://fskmjebat.uitm.edu.my/pcmj/

Progress in Computing and Mathematics Journal
College of Computing, Informatics, and Mathematics
Universiti Teknologi MARA Cawangan Melaka, Kampus Jasin
77300, Merlimau, Melaka Bandaraya Bersejarah

PCMJ

Progress in Computing and Mathematics Journal

volume 1

Progress in Computing and Mathematics Journal (PCMJ)
College of Computing, Informatics, and Mathematics
Universiti Teknologi MARA Cawangan Melaka, Kampus Jasin
77300, Merlimau, Melaka Bandaraya Bersejarah

EDITORS

PCMJ
Progress in Computing and Mathematics Journal
volume 1

PREFACE

Welcome to the inaugural volume of the **Progress in Computing and Mathematics Journal (PCMJ)**, a publication proudly presented by the College of Computing, Informatics, and Mathematics at UiTM Cawangan Melaka.

This journal represents a significant step in our commitment to fostering a vibrant research culture, initially providing a crucial platform for our undergraduate students to showcase their intellectual curiosity, dedication to scholarly pursuit, and potential to contribute to the broader academic discourse in the fields of computing and mathematics. However, we envision PCMJ evolving into a beacon for researchers both nationally and internationally. We aspire to cultivate a space where groundbreaking research and innovative ideas converge, fostering collaboration and intellectual exchange among established scholars and emerging talents alike.

The manuscripts featured in this first volume, predominantly authored by our undergraduate students, are a testament to the hard work and dedication of these budding researchers, as well as the guidance and support provided by their faculty mentors. They cover a diverse range of topics, reflecting the breadth and depth of research interests within our college, and set the stage for the high-quality scholarship we aim to attract in future volumes.

As editors, we are honored to have played a role in bringing this journal to fruition. We extend our sincere gratitude to all the authors, reviewers, and members of the editorial board for their invaluable contributions. We also acknowledge the unwavering support of the college administration in making this initiative possible.

We hope that PCMJ will inspire future generations of students and researchers to embrace research and innovation, to push the boundaries of knowledge, and to make their mark on the world of computing and mathematics.

Editors
Progress in Computing and Mathematics Journal (PCMJ)
College of Computing, Informatics, and Mathematics
UiTM Cawangan Melaka

TABLE OF CONTENTS

PCMJ

Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

WEB BASED PERSONALIZED UNIVERSITY TIMETABLE FOR UITM STUDENTS USING GENETIC ALGORITHM

Mohd Radhi Fauzan Bin Jamli
College of Computing. Informatics and Mathematics

Ahmad Firdaus bin Ahmad Fadzil
College of Computing. Informatics and Mathematics

Article Info

Received: February 2024
Accepted: August 2024
Available Online: October 2024

Abstract

The research endeavors to develop a web-based system specifically designed to create personalized university timetables for Universiti Teknologi MARA (UiTM) students using genetic algorithms, aiming to address the urgent need for a customizable timetable solution catering to the diverse scheduling requirements of both repeater and non-repeater students while optimizing course group selection to minimize conflicts and enhance scheduling flexibility. The complexity of timetable generation stems from the varied course groupings and scheduling constraints inherent in UiTM's curriculum, leading to challenges for students, particularly repeaters, in enrolling in courses across different semesters and groupings, resulting in conflicts and inefficiencies. Traditional methods of timetable generation lack the adaptability needed to tackle these complexities, necessitating the development of an innovative solution. The proposed approach utilizes genetic algorithms to dynamically produce optimized timetables based on individual student needs, with real-time data scraping from 'iCRESS' ensuring the system stays up-to-date with the latest course information for accurate timetable generation. Within the genetic algorithm framework, each timetable is represented as a chromosome, forming a population of potential timetables refined through successive generations by genetic operators like crossover and mutation. Student input initiates the process, with user interaction allowing for timetable customization based on personal preferences. Extensive experimentation with genetic algorithm parameters has yielded promising results, notably a parameter set (population size = 12, generation size = 30, mutation rate = 0.2) demonstrating robust performance, achieving optimal timetables with swift convergence and minimal conflicts. This configuration excelled in efficiency and scalability, offering a viable solution for timetable generation at scale. Future work entails enhancing system robustness through comprehensive contingency planning, real-time data integration, and algorithmic optimization, with a focus on refining the genetic algorithm and exploring parallel processing techniques to further enhance efficiency and scalability.

Keywords: University Timetabling; Genetic Algorithm; Metaheuristic Algorithm

INTRODUCTION

The University Course Timetabling Problem (UCTP) is a complex challenge that arises in academic institutions when scheduling courses for students (Houhamdi et al., 2019; Chen et al., 2021). This project aims to tackle the problem by developing a solution that addresses course conflicts and incorporates students' preferred course groups. Timetabling conflicts can occur when students are required to enrol in courses with different groups or when they need to retake certain courses, resulting in scheduling complexities. By addressing these conflicts and considering students' time or group preferences, this project seeks to optimize the course timetabling process and enhance the overall scheduling experience for students.

To tackle UCTP, this project will be employing Genetic Algorithm (GA) as the primary method. According to Herath (2017), genetic algorithm is a computational technique inspired by the process of natural selection and evolution. He added that this technique involves the use of genetic operators such as selection, crossover, and mutation to evolve a population of potential solutions and identify the best timetable configurations. By utilizing a genetic algorithm, this project leverages its ability to explore a vast search space and find optimal or near-optimal solutions for the timetabling problem (Herath, 2017). This approach has shown promising results in solving similar 1 optimization problems, making it a suitable choice for addressing the complexities of course timetabling (Wong et al., 2022).

The proposed system of this project will be a web-based platform that incorporates several key features to enhance the timetabling experience for UiTM students. Firstly, the system will allow students to input their required courses for the current semester without the need to specify the course groups manually. Instead, the system will employ an optimized algorithm to search for suitable timetables that accommodate all the required courses without conflicts. This feature aims to simplify the timetable generation process for students and save them time and effort. Additionally, the system will provide a customizable approach to timetabling. Once the generated timetable is displayed, students will have the flexibility to "lock" at least one course. By locking a course, students indicate their preference to maintain the current course group while reshuffling the rest of the timetable. This feature allows students to have some control over their timetables and make adjustments according to their preferences while still ensuring that the locked courses remain unchanged.

Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

LITERATURE REVIEW

This chapter presents a comprehensive review of the literature on university timetabling. It covers the constraints in University Timetabling, Genetic Algorithm adaption to timetabling problem and related works. This literature review serves as a foundation for the development and evaluation of the proposed web-based system for generating personalized university timetables using a genetic algorithm.

Constraints in University Timetabling

Timetabling in educational institutions involves addressing various constraints to create well-structured and efficient schedules. These constraints are crucial for meeting the requirements of both students and institutions. In university timetabling, several common constraints must be considered.

- Course Availability

One important constraint is course availability, which refers to specific periods during which each course can be scheduled. It is essential to avoid conflicting schedules for courses that share the same resources or require specific facilities.

- Room Capacity

Another critical constraint is room capacity. Each room or venue where classes are held has a maximum capacity. It is necessary to ensure that the number of students assigned to a room does not exceed its capacity, thus maintaining a safe and conducive learning environment (Mahlous et al., 2023).

- Student Grouping

Student grouping is also a significant constraint, particularly in large universities with multiple groups or sections for courses. It is important to schedule students from the same program or cohort together, facilitating effective group learning and interaction among peers (Ilyas et al., 2015).

- Student Preferences

In addition to the mandatory constraints, student preferences can also play a role in timetabling. These preferences may include preferred time slots, instructors, or room locations. While accommodating all student preferences may not always be possible, considering them can contribute to student satisfaction and engagement (Mahlous et al., 2023).

Genetic Algorithm adaption to Timetabling Problem

Genetic algorithm is a type of metaheuristic algorithm that is inspired by the process of natural selection. GA is used to solve a wide variety of problems, including university timetabling. GA works by creating a population of potential solutions, and then iteratively evolving and improving these solutions through a process of selection, crossover, and mutation (Markal et al., 2020). Selection is the process of choosing the best solutions from the population (Thakare et al., 2020). Figure 2.1 below shows the pseudocode to generate the population.

```
Generate a random ordering of courses;
        For all courses
            For all available room
                 For all available timeslot
                      Assign room and timeslot
                      If no conflicts
                          Exit;
                 End for
             End for
        End for
 End;
```

Figure 1: The pseudocode to generate population

Crossover is the process of combining two solutions to create a new solution. Mutation is the process of randomly changing a solution. GA is able to find good solutions to the university timetabling problem by exploring a large number of potential solutions and by avoiding local optima (Abdelhalim et al., 2016). Figure 2 and Figure 3 below show the steps involved in the genetic algorithm process for university timetabling and pseudocode of mutation operation respectively.

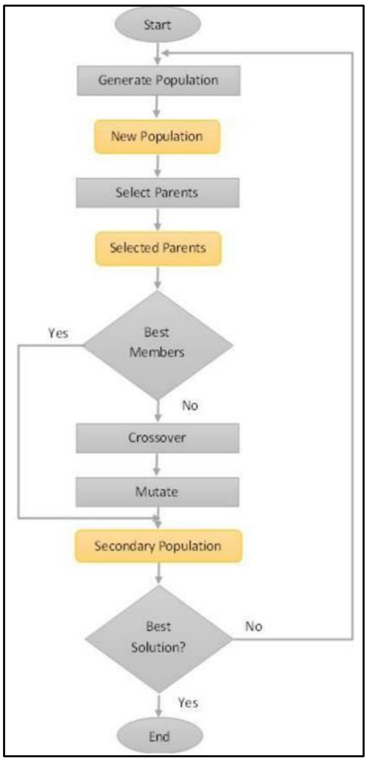

Figure 2: The flowchart of Genetic Algorithm

Mutation operation

```
input: solution, mutation probability (P_m)
if random(0,1) < P_m then
    index := index of the pair of an event and a suitable room
             that has the maximum template value
    fitness1 := the current fitness value of solution
    tempsolution := solution
    timeslot := solution(index)
    for each value in domain of timeslot do
        tempsolution(index) := value
        fitness2 := the fitness value of tempsolution
        if fitness2 < fitness1 then
            fitness1 := fitness2
            timeslot := value
        end if
    end for
    solution(index) := timeslot
else
    index := random(0, length(solution) − 1)
    solution(index) := random(value in the domain of timeslot)
end if
Update template(index) of solution
output: Return solution
```

Figure 3: The pseudocode of mutation operation

Figure 1 shows that the first step is to generate a population of chromosomes. A chromosome is a representation of a timetable. Each chromosome is a string of genes, where each gene represents a course. The genes are arranged in a time order, with each gene

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

representing a course that is to be taught at a particular time (Pérez et al., 2021). Figure 4 below shows chromosome representation in a timetable.

	R1	R2	R3	. . . . .		Rn
T1	e_{14}			e_{17}		e_1
T2		e_6	e_7		e_2	
T3		e_3		e_5	e_9	
T4	e_{19}	e_4	e_{15}		e_8	e_{10}
T5	e_7					
T6				e_{13}		
T7	e_{16}					
T8	e_{11}		e_{20}	e_{21}		
T9	e_{12}	e_{18}			e_{22}	

Figure 4: Chromosome representation in Timetable

The next step is to evaluate the fitness of each chromosome. The fitness of a chromosome is a measure of how well it satisfies the constraints of the timetabling problem. The constraints can include things like course conflicts, room availability, and lecturer availability. Once the fitness of each chromosome has been evaluated, the next step is to select the fittest chromosomes. The fittest chromosomes are those that have the highest fitness. The fittest chromosomes are then used to create a new population of chromosomes. The new population of chromosomes is created using a process called crossover. Crossover is a process where two chromosomes are combined to create a new chromosome (Pérez et al., 2021). The new chromosome inherits genes from both parent chromosomes. The new population of chromosomes is then evaluated and the fittest chromosomes are selected. This process is repeated until a satisfactory timetable is found (Premasiril, 2019). Figure 2.5 shows the pseudocode of the crossover operation.

Crossover operation

```
input: solution1, solution2
for i:= 1 to length of solution do
    sum := template1_i + template2_i
    if sum = 0 then
        solution3_i := solution1_i
    else
        rand := random(0, sum − 1)
        if rand < template1_i then
            solution3_i := solution2_i
        else
            solution3_i := solution1_i
        end if
    end if
end for
output: Return solution3
```

Figure 5: Pseudocode of crossover operation

Related Works

A thorough examination of pertinent literature highlights a range of methodologies and strategies employed in university timetabling, with comparisons drawn across various aspects such as platforms, methods, data collection techniques, handling of course groups and conflicts, integration of student preferences, and customization options. Diverse platforms, including web-based applications and standalone software, illustrate the adaptability of solutions, while methodological approaches encompass metaheuristic algorithms like genetic algorithms, simulated annealing, and tabu search. Data collection methods typically involve accessing university databases or conducting interviews for course details and scheduling parameters. While not universally addressed, effective solutions for course groupings and conflicts are proposed in some studies, and several studies incorporate student preferences to tailor timetables accordingly. Although some systems allow for timetable customization, this feature is not consistently implemented across studies. Analyzing existing systems provides valuable insights for developing a tailored timetabling solution for UiTM students, considering the institution's specific requirements and constraints.

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

Table 1: Related Works

Reference	Platform	Method / Techniques	Data Collection	Handle course groups and conflicts	Incorporate student preferences	Enable timetable customization
Thakare et al., 2020	Web-based	GA	University Database	Y	Y	N
Vianna et al., 2020	Standalone software	VNS + TS	University Database	Y	Y	N
Brenda et al., 2021	Standalone software	GA	Interview	Y	Y	N
Rashmi et al., 2021	Standalone software	GA	University Database	Y	Y	N
Chaudhari et al., 2022	Web-based	GA	School Database	Y	Y	N
Mahlous et al., 2023	Standalone software	GA	University Database	Y	Y	N
Proposed System	Web-based	GA	University Database	Y	Y	Y

METHODOLOGY

In the context of solving optimization and search problems, Genetic Algorithms (GAs) are computational techniques inspired by natural selection and genetics. GAs employ principles of population-based evolution, utilizing genetic operators like selection, crossover, and mutation to iteratively enhance solutions over generations. This adaptive and stochastic nature allows GAs to explore vast search spaces effectively, making them well-suited for addressing complex problems where traditional methods may struggle to find optimal solutions. In this section, we will delve into the specific methods implemented in the GA and explore their relevance to the project. Figure 6 presents the flowchart of the GA, and each process will be examined in detail below.

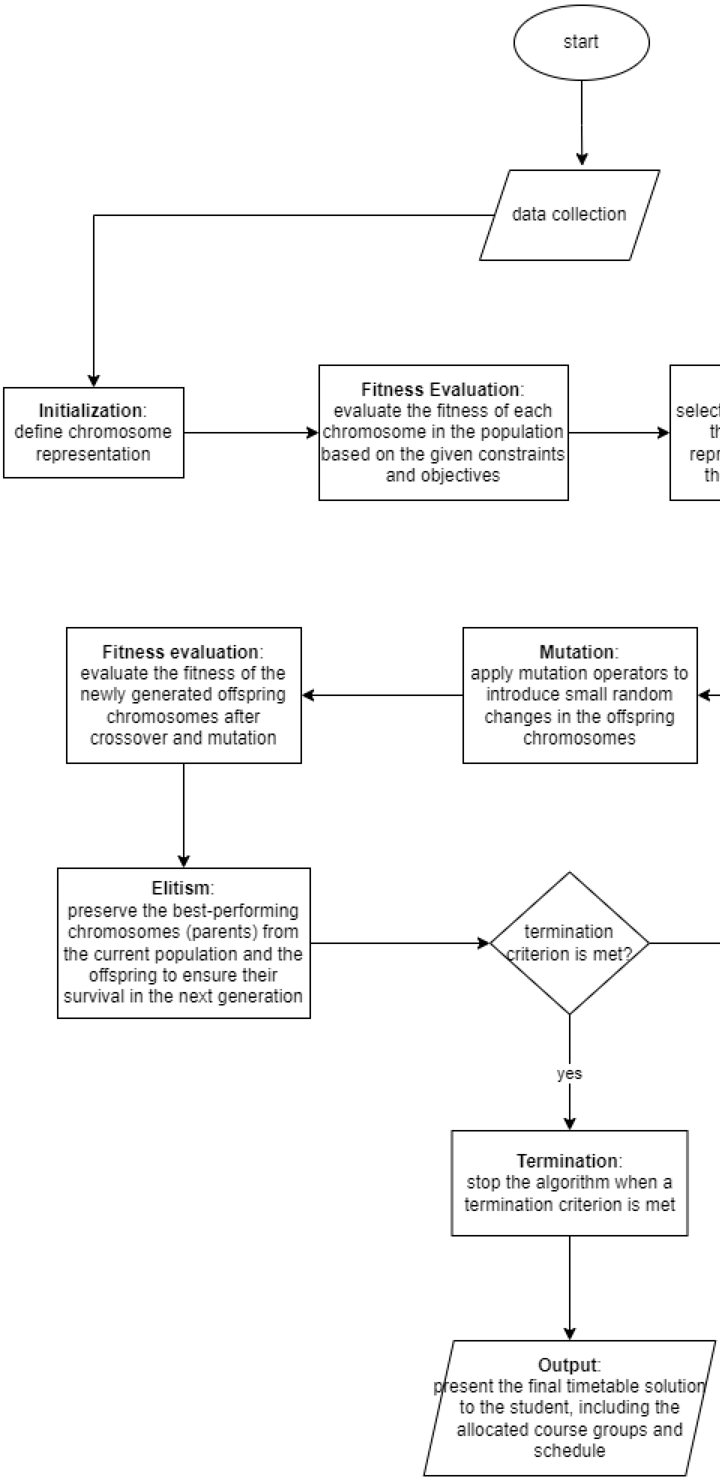

Figure 6: Genetic Algorithm Flowchart

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

Data Collection and Preparation

In this step, course information will be scrapped and processed from ‘icress’ website. This data includes course codes, time periods, locations, course groups and any additional details needed for generating the timetable.

Campus : M3 - Course : CSC508
Valid as 16/06/2023 13:15:41

NO	DAY TIME	GROUP	MODE	STATUS	ROOM
1.	MONDAY (08:00 AM-10:00 AM)	M3CS2534B	BOTH - Fulltime and Part-time	First Timer and Repeater	MAKMAL KOMPUTER 4
2.	TUESDAY (16:00 PM-18:00 PM)	M3CS2534C	BOTH - Fulltime and Part-time	First Timer and Repeater	BILIK KULIAH 12
3.	WEDNESDAY (08:00 AM-10:00 AM)	M3CS2304B	Fulltime	First Timer and Repeater	MAKMAL KOMPUTER 10
4.	WEDNESDAY (10:00 AM-12:00 PM)	M3CS2304C	BOTH - Fulltime and Part-time	First Timer and Repeater	MAKMAL KOMPUTER 11 (MM)
5.	WEDNESDAY (08:00 AM-10:00 AM)	M3CS2534A	Fulltime	First Timer and Repeater	BILIK KULIAH 10
6.	THURSDAY (16:00 PM-18:00 PM)	M3CS2304A	Fulltime	First Timer and Repeater	MAKMAL KOMPUTER 2 (BIG DATA)
7.	THURSDAY (14:00 PM-16:00 PM)	M3CS2304B	Fulltime	First Timer and Repeater	BILIK BACAAN KOLEJ
8.	THURSDAY (10:00 AM-12:00 PM)	M3CS2534B	BOTH - Fulltime and Part-time	First Timer and Repeater	BILIK KULIAH 17
9.	THURSDAY (14:00 PM-16:00 PM)	M3CS2534C	BOTH - Fulltime and Part-time	First Timer and Repeater	MAKMAL KOMPUTER 8
10.	FRIDAY (10:00 AM-12:00 PM)	M3CS2304A	Fulltime	First Timer and Repeater	BILIK TUTORIAL 02
11.	FRIDAY (10:00 AM-12:00 PM)	M3CS2304C	BOTH - Fulltime and Part-time	First Timer and Repeater	DEWAN SEMINAR 2B
12.	FRIDAY (08:00 AM-10:00 AM)	M3CS2534A	Fulltime	First Timer and Repeater	MAKMAL KOMPUTER 10

Figure 7: Course information from ‘iCress’ website

Initialization

During initialization, chromosome will be defined for representation in genetic algorithm. In this case, the chromosome represents a potential timetable solution for a student which is represented in array data structure (refer Figure 8). It could be a sequence of genes, where each gene represents a course with its allocated group and time slot (refer Figure 9). This step sets the foundation for creating the initial population of potential timetables.

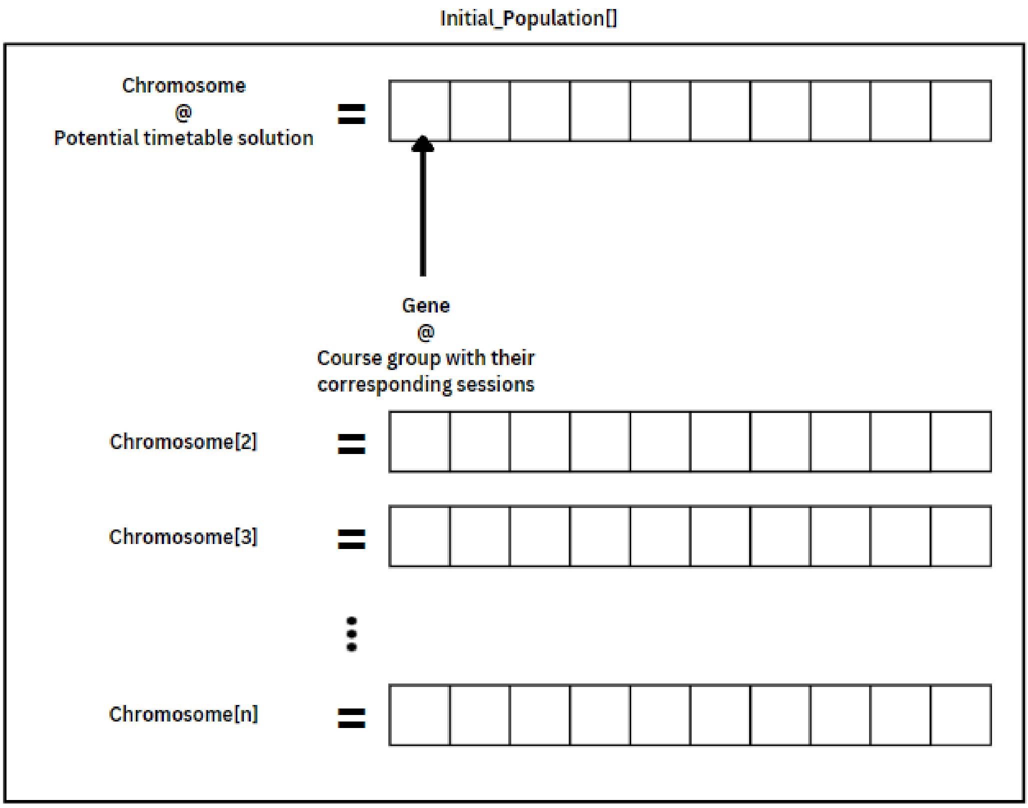

Figure 8: Population representation

```
class Course:
    def __init__(self, name, group, sessions):
        self.name = name
        self.group = group
        self.sessions = sessions

class Session:
    def __init__(self, day, time_start, time_end):
        self.day = day
        self.time_start = time_start
        self.time_end = time_end
```

Figure 9: Gene representation

Fitness Evaluation

The fitness evaluation step involves assessing the quality of each chromosome (timetable) in the population. Fitness value will be calculated for each timetable, considering factors such as course overlaps (Figure 10 illustrates the initial algorithm idea to check number of course overlaps in a chromosome or timetable). Higher fitness values are assigned to timetables that satisfy the constraints and preferences more effectively, indicating their potential as better solutions (refer Figure 11).

```python
def count_clashes(chromosome):
    clashes = 0
    for i in range(len(chromosome)):
        for j in range(i + 1, len(chromosome)):
            course1 = chromosome[i]
            course2 = chromosome[j]
            for session1 in course1.sessions:
                for session2 in course2.sessions:
                    if session1.day == session2.day:
                        # Check for time overlap or conflict
                        if do_time_slots_overlap(session1.time_start,
                                                 session1.time_end,session2.time_start,
                                                 session2.time_end):
                            clashes += 1  # add number of clashes
    return clashes

def do_time_slots_overlap(start1, end1, start2, end2):
    return start1 < end2 and end1 > start2
```

Figure 10: Algorithm to count course overlapping in a timetable

```python
def calculate_fitness(chromosome):
    # Number of course overlaps
    clashes = count_clashes(chromosome)

    # Calculate the overall fitness value
    fitness = 1 / (clashes + 1)  # Higher fitness for fewer clashes

    return fitness
```

Figure 11: The fitness function

Crossover

Crossover involves the creating of new offspring by exchanging genetic material between selected parent chromosomes. Crossover techniques are applied (e.g., single-point crossover or multi-point crossover) to exchange course groups or time slots between the parents

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

(refer Figure 12). This process explores different combinations of courses and groups, potentially generating offspring with improved fitness values.

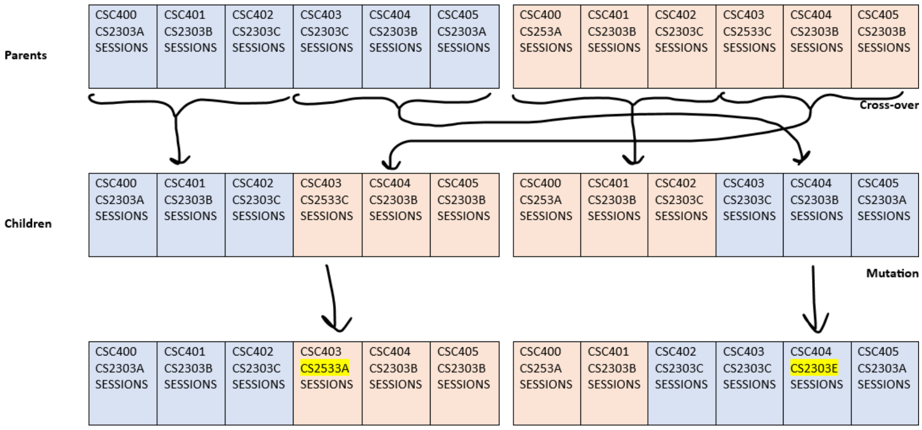

Figure 12: Crossover and Mutation operation

Elitism

Elitism involves preserving the best-performing chromosomes from the current population and the offspring. Certain number of top-performing timetables will be selected based on their fitness values and carry them forward to the next generation without any changes (refer Figure 13). Elitism helps maintain the best solutions found so far throughout the evolutionary process.

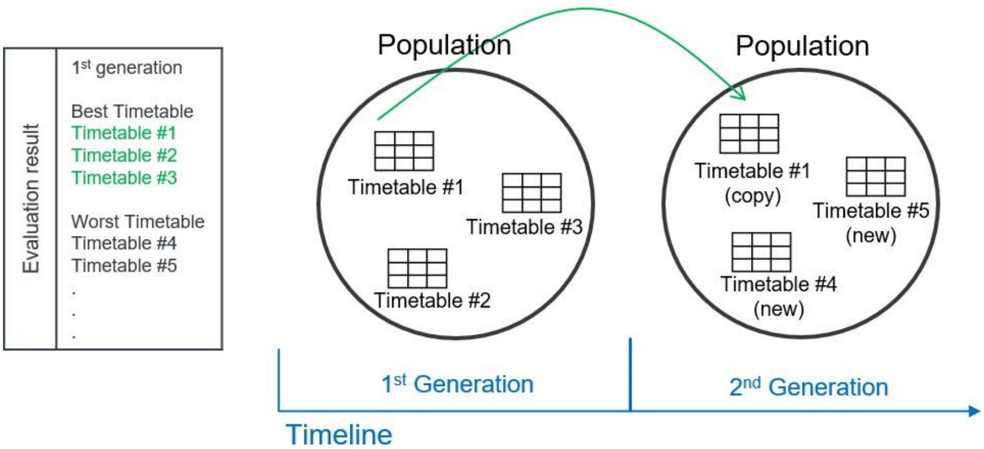

Figure 13: Elitism implementation

PCMJ

Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

RESULT AND DISCUSSION

Test Summary

In the quest to improve the efficiency and reliability of the Genetic Algorithm (GA) for personalized university timetable generation, a comprehensive convergence test has been meticulously executed. This test, crucial within the testing protocol, involves two distinct input scenarios: the 'Basic Input' configuration with minimalistic course group specifications and the 'Advanced Input,' considering extreme configurations of each course. This strategic approach allows for an evaluation of the GA's performance across a spectrum of real-world scheduling complexities. The test systematically varies three critical genetic parameters: population size, generation size, and mutation rate, each assuming three different values. Population sizes of 4, 8, and 12, generation sizes of 10, 30, and 50, and mutation rates of 0.10, 0.20, and 0.30 are scrutinized, offering insights into the algorithm's convergence patterns over diverse timeframes and levels of genetic diversity.

In summarizing the extensive experiments conducted for both basic and advanced constraints, the Genetic Algorithm's (GA) performance across diverse parameter sets has been thoroughly evaluated. For basic constraints, the initial parameter set (population size = 4, generation size = 10, mutation rates = 0.1, 0.2, 0.3) consistently demonstrated swift and reliable convergence, achieving optimal timetables with negligible conflicts and efficient execution times. Meanwhile, in the context of advanced constraints, parameter set (population size = 12, generation size = 30, mutation rate = 0.2) emerged as particularly robust, showcasing rapid convergence at generation 7 with optimal timetables and no conflicts. This configuration demonstrated adaptability to increased computational demands, striking an optimal balance between scalability and performance. Therefore, parameter set (population size = 12, generation size = 30, mutation rate = 0.2) stands out as the most favourable, excelling in both efficiency and convergence speed across the spectrum of hard constraints in timetable generation.

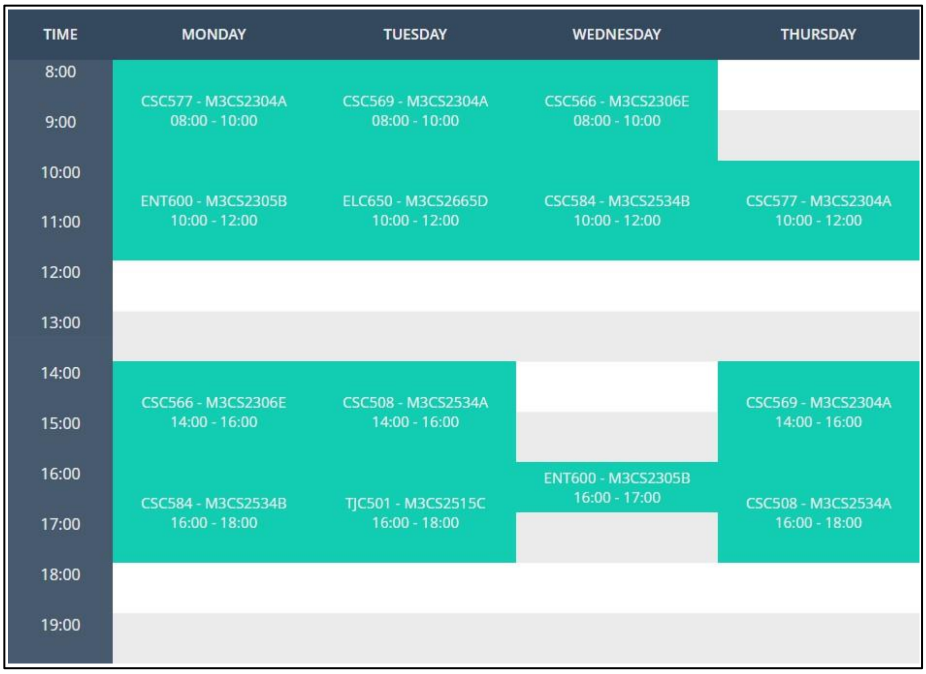

Figure 14: Generated Timetable using the best genetic parameters where p=12, g=30 and m=0.2

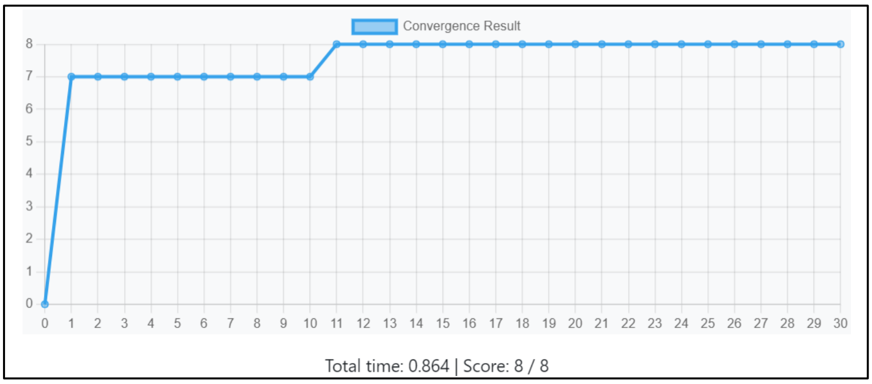

Figure 15: Convergence result for p=12, g=30 and m=0.2

Conclusion

In the pursuit of an improved scheduling solution for UiTM students, the project initially focuses on developing a user-friendly web-based system aimed at simplifying the timetable creation process. Students can input required courses without specifying course groups. Subsequently, the project delves into the algorithmic core, aiming to design an efficient timetable generation algorithm using a genetic approach. This algorithm is tailored to optimize course group and time slot allocations, addressing challenges posed by diverse course groups and potential conflicts. To assess the system's practicality and reliability, the project employs convergence testing to ensure the genetic algorithm consistently generates optimal timetables aligned with student constraints and preferences. By scrutinizing the system's alignment with student needs, this evaluation framework aims to affirm its effectiveness in meeting the diverse scheduling needs of UiTM students consistently.

REFERENCES

Chaudhari, Y. S., Dmello, V. W., Shah, S. S., & Bhangale, P. (2022). Autonomous Timetable System Using Genetic Algorithm. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1687-1694). doi: 10.1109/ICSSIT53264.2022.9716370.

Chen, M., Sze, S., Goh, S. L., Sabar, N., & Kendall, G. (2021, July 27). A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities. IEEE Access, PP, 1-1. https://doi.org/10.1109/ACCESS.2021.3100613

Abdelhalim E. A. and G. A. El Khayat, "A utilization-based genetic algorithm for solving the university timetabling problem (uga)," *Alexandria Engineering Journal, vol. 55, no. 2,* pp. 1395–1409, 2016.

Herath, A. K. (2017). Genetic Algorithm For University Course Timetabling Problem. Electronic Theses and Dissertations, 443. Retrieved from https://egrove.olemiss.edu/etd/443

Houhamdi, Z., Athamena, B., Abuzaineddin, R., & Muhairat, M. (2019). A multi-agent system for course timetable generation. *TEM Journal*, *8*, 211-221. https://doi.org/10.18421/TEM81-30.

Ilyas, R., Iqbal, Z. (2015). Study of Hybrid Approaches used for University Course Timetable Problem (UCTP). [Ebook]. DOI: 10.1109/ICIEA.2015.7334198.

Mahlous, A.R., & Mahlous, H. (2023). Student timetabling genetic algorithm accounting for student preferences. *PeerJ Computer Science*, 9, e1200. https://doi.org/10.7717/peerj-cs.1200.

Markal, S., et al. (2020). Timetable Generator. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 22(2), 29-33. doi:10.9790/0661-2202022933.

Pérez, E. C., Rios, O. M., Bautista, D. P., Sanchez, S. S., & Acevedo, F. A. (2021). A Genetic Algorithm Solution for Scheduling Problem. In *2021 XVII International Engineering Congress (CONIIN)* (pp. 1-10). doi: 10.1109/CONIIN54356.2021.9634725.

Premasiril (2019). University Timetable Scheduling Using Genetic Algorithm Approach Case Study: Rajarata University OF Sri Lanka.

Rashmi, K. R., & Abhishek, M. B. (2021). Automated University Timetable Generation using Prediction Algorithm. *International Research Journal of Engineering and Technology (IRJET)*, 8(6). Retrieved from https://www.irjet.net/archives/V8/i6/IRJET-V8I6419.pdf

Thakare, S. (2020, August 4). Automated Timetable Generation using Genetic Algorithm. *International Journal of Engineering Research*, V9. doi: 10.17577/IJERTV9IS070568.

Wong, C. H., Goh, S. L. & Likoh, J. (2022). A genetic algorithm for the real-world university course timetabling problem. *2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA), Selangor* pp. 46-50, doi: 10.1109/CSPA55076.2022.9781907

PCMJ

Progress in Computing and Mathematics Journal

UNIVERSITI TEKNOLOGI MARA | Cawangan Melaka

eISSN 3030-6728