Solving Course Timetabling: Case Study of University of Tabuk, Saudi Arabia

[1]Eman Ebniah Alamrani, [2]Mashael Maashi, [3]Manal Souliman Alhadidi, [4]Mona Souliman Althwai, [5]Amirah Abdulhameed Althwai, [6]Hanan Rashed Alharbi, [7]Hifa Ali Alhwity, [8]Gadah Mohammad Albalawi

[1]Technical and Vocational Training Corporation
[2]King Saud University
[3]Ministry of education
[4,5,6]Taibah University
[7,8]Ministry of education

Abstract- **The need to schedule specific resources over prefixed timeslots is highly required in many institutions all over the world. This is known as a timetabling problem which is a combinatorial optimization problem. The timetabling problem is an NP-complete problem. University/Course timetabling (U/CTTP) is considered one of these problems. It requires scheduling of a certain number of courses over a time within a specific resource such as teachers, rooms, students, and other special equipment. It also requires several constraints which must be satisfied. Each institution has a different perspective on its timetabling and differs from other institutions because of the diversity of constraints. This project aims to solve the course timetabling problem at the Faculty of Computer Science and Information Technology - Girls Section Tabuk University. A genetic algorithm (GA) will be employed to address this problem. A real data set from the faculty of computer science and information technology- Girls Section is applied as our benchmarking data set. This project will require a computer terminal with suitable programming software.**

Keywords: **Genetic algorithm; Course timetabling issues.**

Introduction

This project aims to address the issues related to the University of Tabuk's course scheduling, with a particular focus on the Faculty of Computer Science and Information Technology - Girls Section. The university's existing manual timetabling method makes it difficult to deploy, alter, and manage time efficiently. It also makes it difficult to settle problems in classes and classrooms. To get beyond these restrictions, the project suggests creating an automated course schedule with a Genetic Algorithm (GA), the main objective of which is to reduce soft constraint violations [1].

The first step is to become familiar with the University of Tabuk's current course scheduling system, especially as it relates to the Faculty of Computer Science and Information Technology - Girls Section. This knowledge entails determining the advantages and disadvantages of the present manual method, laying the groundwork for its eventual improvement. Members of the academic committee create timetables by hand, and they must meet certain criteria, such availability of time slots and classrooms. Nevertheless, there are a lot of obstacles this approach must overcome, such as implementation, modification, and time consumption issues. Moreover, the manual method is not successful in settling disputes in classes and courses. As such, the incorporation of an automated solution is imperative to improve and streamline the timetabling process[2,3].

The project seeks to gather actual data from the Faculty of Computer Science and Information Technology - Girls Section to properly solve the issue of course scheduling. To create a benchmarking dataset that effectively reflects the unique requirements and limits of the university, this data gathering is essential. The suggested automatic timetabling system will be developed and evaluated using the dataset as a basis [4,5]. Faculty members who oversee the scheduling process are interviewed as part of the data-gathering procedure. A 15% sample of students will also get a student survey to get important information about their preferences and level of satisfaction with their course schedules. The amalgamation of academic staff and student comments will facilitate a thorough comprehension of the current obstacles and necessities [6].

The focus of the project is developing a prototype model that uses a Genetic Algorithm (GA) to solve the Faculty of Computer Science and Information Technology - Girls Section's course scheduling issue. Heuristic search algorithms that are successful in solving optimization issues are called genetic algorithms. They are modelled after the process of natural selection. The GA will be used to automatically create conflict-free schedules that follow both hard and soft restrictions in the context of course timetabling [7,8]. The prototype intends to minimize the cost of soft constraint violations and increase schedule satisfaction for both professors and students by enabling the quick and effective development of an automated course schedule.

The project's scope comprises several aspects, such as identifying the characteristics of the problem, such as the study's objectives, hard and soft limitations, data volume, and sample size. This entails not just comprehending the operational features of the current scheduling system but also assessing user preferences and satisfaction through surveys and faculty interviews. The project's hardware specifications call for printers, scanners, and PCs with particular CPUs and RAM capacities. Software tools that will be used include Microsoft Visual C# Program Language, Microsoft Office for presentations and documentation, and graphic programs like Adobe Photoshop CS6 and Smart Draw [8].

When it comes to project creation, the goal is to solve course scheduling issues through study. Creating a prototype with a sample dataset of computer science and information technology courses is the first stage in the process. After that, this prototype will be subjected to the GA to produce an automated course schedule [9,10]. The project won't handle instructor and student tasks during certain periods due to time restrictions; it will be assumed that these assignments have already been finished. For the first semester of the academic year 2014, the prototype's effectiveness will be assessed by contrasting its results with the present course scheduling system of the Faculty of Computer Science and Information Technology - Girls Section [11].

The project's product scope is designed to give faculty members a complete answer to the issue of course scheduling. Its goal is to minimize the cost of soft constraint breaches by providing automatically created schedules that are devoid of conflicts. The expected advantages include better workload management, enhanced timetabling efficiency, and higher levels of satisfaction from both teachers and students. The suggested approach, which is based on the application of a Genetic Algorithm, aims to maximize both hard and soft restrictions by assigning courses and rooms throughout certain time intervals [12].

To sum up, this project is an important attempt to improve the University of Tabuk's course scheduling procedure, with a particular emphasis on the Faculty of Computer Science and Information Technology - Girls Section. A potential method for automating the timetabling process and producing conflict-free schedules that adhere to both hard and soft restrictions is the integration of a genetic algorithm. The cooperative process that includes student surveys and faculty interviews guarantees that the produced solution satisfies the needs and preferences of the university community. The performance of the prototype will be closely monitored throughout the project to gather important information on the viability and efficacy of putting in place an automated course scheduling system. Through these initiatives, the project hopes to improve instructional procedures and the overall satisfaction of stakeholders within the university setting [11,13,14].

Literature Review

A Genetic algorithm to solve the problem of scheduling resources is proposed in it is the method based on two phases the first phase where phase I deals with timeslot movements, and phase II deals with room movements. The algorithm used here is a blend of simulated annealing and steepest descent. The experimental results indicate that Simulated Annealing and steepest descent combination performs better, faster, and with much less effort compared to manually generated results, the improvement in achieving constraints. An extension of constraint logic programming that allows for weighted partial satisfaction of soft constraints is described in and applied to the development of an automated timetabling system for Purdue University [15,16]. A new repair search algorithm is proposed in to improve upon initially generated (partial) assignments of the problem variables. The model and search methods applied to the solution of the large lecture room component are presented and discussed along with the computational results. The proposed method can reflect diverse requirements of students during course enrolment. The automated search was able to find suitable times and classrooms for almost all classes. A prototype study of a computer application uses Genetic Algorithms (GAs) in timetable management problems. The prototyping result offers a good view of how the timetable is optimized. The result shows that there is an evolution between the original solution and the ending solution. The problem with the prototype is that the mutation does not using soft constraints to change some chromosomes and gives only brand-new solutions for the solutions mutated. A formulation of a genetic algorithm is proposed in [12,17]. The proposed method provides a good quality course timetable. The experimental results show that the best chromosomes based on fitness function from the pool of chromosomes and similarly through crossover. Future work needed to use this genetic approach technique for solving our real-world course timetabling problems. A genetic algorithm with a guided search strategy and a local search technique is proposed for the university course timetabling problem in [18,19]. The guided search strategy is used to create offspring in the population based on a data structure that stores information extracted from previous good individuals. The local search technique is used to improve the quality of individuals. The proposed genetic algorithm is tested on a set of benchmark problems in comparison with a set of state-of-the-art methods from the literature. The experimental results show that the proposed genetic algorithm can produce promising results for the university course timetabling problem. A new hybrid algorithm to solve course timetabling problems based on the Genetic Algorithm and Great Deluge algorithm is proposed in [20,21]. The hybridized method is applied to standard benchmark course timetable problems and is able to produce promising results. In [22,23], PSO (promoting a scheme for solving complex problems) was applied to solving course timetabling problems to reduce the computational complexity. A timeslot was designated in a particle's encoding as the scheduling unit. Two types of PSO, the inertia weight version and the constriction version, were evaluated. Experimental results demonstrate that the proposed scheme of constriction PSO with interchange heuristic can generate satisfactory course timetables that meet the requirements of teachers and classes according to the various applied constraints.

Methodology

The project's initiation involves the identification and assignment of a Project Manager, setting priorities for professors and students regarding project requirements, and selecting the programming language for design and implementation. The information gathering phase relies on two primary sources within the Faculty of Computer Science and Information Technology - Girls Section: interviews with faculty members overseeing the scheduling process and a student survey distributed to a 15% sample. The resources identified from this phase include better utilization of labs and classrooms and adherence to a maximum of 50 students per section.

A sub-section of the information-gathering phase details Timetable Constraints, categorizing them into Hard and Soft Constraints. Hard constraints are those that cannot be violated, ensuring the feasibility of the schedule, while soft constraints, or preferences, aim to enhance the quality of the timetable. The list of Hard Constraints includes ensuring no conflicts for students and staff, no double booking of rooms, adherence to faculty availability, specific requirements for course registration, and constraints related to lecture timings, room capacities, and specialized room needs. Soft Constraints involve preferences such as avoiding overtime for staff and locations, conflicts deemed acceptable by the department, load balancing of resources, preferred starts and usage, primary suitability of resources, even distribution of sections for courses, and considerations related to scheduling labs and tutorials.

The Soft Constraints continue with preferences like assigning activities to rooms close in size to the number of attending students, load balancing to use all available resources evenly, preferred starts for activities, preferred room usage, primary suitability of resources, and even distribution of course sections, avoiding scheduling labs or tutorials on the same day as lectures, refraining from scheduling tutorials in the morning before lectures, scheduling courses requiring effort and focus in the first hours of the day, and considering the department's division of study days into two groups: Sunday, Tuesday, and Thursday, and Monday and Wednesday [24].

The methodology outlines a comprehensive approach to address the course timetabling problem, starting with scope initiation, followed by meticulous information gathering and a literature survey. The identified constraints, both hard and soft, provide a robust foundation for designing the genetic algorithm that will form the core of the automatic timetabling system [25, 26]. This methodology ensures that the developed solution aligns with the unique requirements and preferences of the Faculty of Computer Science and Information Technology - Girls Section at the University of Tabuk. The systematic approach from scope initiation to constraint definition sets the stage for the subsequent phases of development and evaluation, offering a well-structured framework for addressing the challenges associated with manual timetabling and enhancing the overall efficiency of the educational scheduling process.

Genetic Algorithms (GAs) serve as powerful search heuristics, rooted in natural selection processes and routinely applied to address optimization and search problems. Belonging to the broader category of evolutionary algorithms (EA), GAs leverage principles inspired by natural evolution, encompassing inheritance, mutation, selection, and crossover. GAs generates individuals, represented as encoded forms known as "chromosomes" or "genomes," wherein chromosomes undergo combination or mutation to produce new individuals. The process of "crossover," akin to recombination in sexual reproduction, involves creating an offspring's chromosome by alternating segments from two parent chromosomes of fixed length.

To prepare the data for the genetic algorithm, we must now encode the timetable in a binary format. In this method, a series of 0s and 1s are used to represent each element, including rooms, timeslots, and courses. The data is made simpler by this binary encoding, which also makes it compatible with genetic algorithm processes and computationally efficient. A binary code is allocated to every element, making genetic operators like crossover and mutation easier to manipulate. This simplified form helps the optimization process by supporting the iterative creation of better schedules that respect the restrictions that have been defined.

The initial phase of the GA involves data encoding and decoding, where solutions are transformed into encoded chromosomes, typically in binary format, streamlining algorithmic treatment. Each chromosome comprises genes, and the number of bits in each gene depends on the maximum data size intended for storage. For instance, (00011) can represent 3 with 5 bits, storing data from 0 to 31, while (011) can represent 3 with 3 bits, accommodating data from 0 to 7.

- **Chromosome Encoding**:

Encoding

This is a **chromosome**

000110110101011011010101111100101100010111100 1

The **chromosome** can be spited into **genes**

000110110101011011010 10 111110010110001011110 01

A **gene** can be spited to get all the **data**

00011 011 0101 011 0110 1010

3 3 5 3 6 10

The first step in the Genetic Algorithm (GA) procedure is to initialize a population of N chromosomes, each of which has randomly produced DNA. The following actions carried out repeatedly, are as follows: selection: the process of building a mating pool by evaluating each chromosome's fitness; reproduction: this involves selecting two parents iteratively based on relative fitness, crossing over to create a new child by combining their DNA; the new child is then added to a new population; and finally, the old population is replaced with the new one, which forces a return to the selection step.

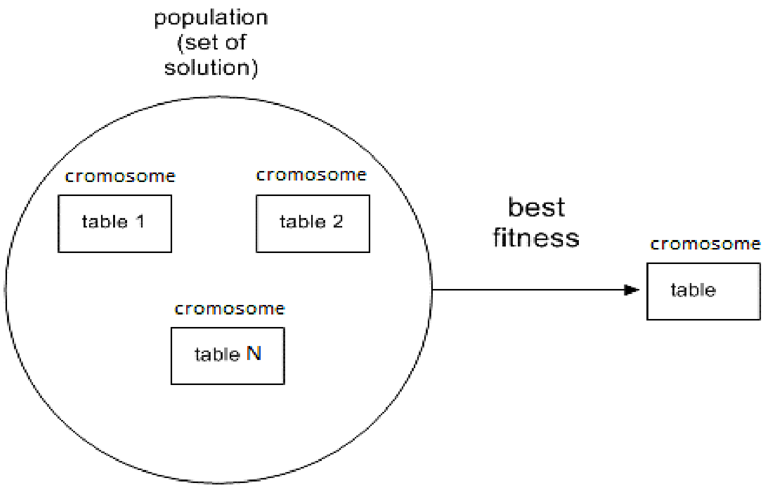


Figure 1: Initial Population

Furthermore, many chromosomes are formed inside the original population, each corresponding to a table of events with genes designating the course, section, and timeslot format (7 bits for lecture start time and 7 bits for lecture finish time).

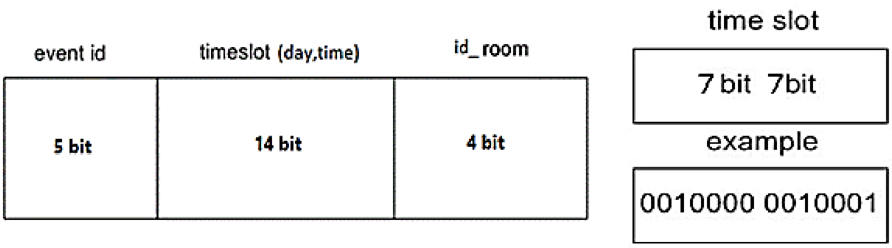


Figure 2: Timeslot of the lectures

There is also an interpretation of the first six segments, which represent the lecture that begins on Sunday, April (00), at 8 o'clock (1000) and ends at 9 o'clock (1001) on the same day, April (00). The Genetic Algorithm (GA) uses the Darwinian concept of selection in its Selection step, evaluating population fitness to choose parents who will produce offspring. There are two essential elements in this choosing procedure. The first step in evaluating fitness is creating a fitness function, which generates a score that represents each population member's level of fitness. Although real-world circumstances entail survival without scoring, in the classic genetic algorithm, generating optimal solutions necessitates a numerical assessment. Second, once the fitness for every population chromosome has been determined, a mating pool is established. Different strategies can be used, such as the elitist strategy, which puts variety at risk by having the highest-scoring chromosomes become exclusive parents. As an alternative, a probabilistic approach which uses a population of five chromosomes as an example—is advised for the mating pool. By normalizing fitness scores and allocating a range of 0 to 1 as a proportion of overall fitness, this approach enables a more complex and diverse parent selection process throughout evolution.

Table 1: The fitness score for given chromosomes

Chromosome	Fitness
C1	3
C2	4
C3	0.5
C4	1.5
C5	1

Table 2: The normalized fitness

Chromosome	Fitness	Normalized Fitness	Percentage
C_1	3	0.5	30%
C_2	4	0.4	40%
C_3	0.5	0.05	5%
C_4	1.5	0.15	15%
C_5	1	0.3	10%

Table 3: The Probability of parent

Parent	Probability
C_1	30%
C_2	40%
C_3	5%
C_4	15%
C_5	10%

To create a new population from the current one, crossover evolution and mutation are both used in the Genetic Algorithm (GA) Reproduction stage. Two or more chromosomes are used in the crossover evolution process, which can be carried out as a single or multiple cross-over to form new ones.

- **The simple crossover evolution**:

- **The multiple crossover evolution:**

  - **Three chromosomes crossover evolution**

Chromosome C_1: ABC

Chromosome C_2: DEF

Chromosome C_3: GHI

Crossover generation: 27 chromosomes: ADG, ADH, ADI, AEG, AEH, AEI, AFG, AFH, AFI, BDG, BDH, BDI, BEG, BEH, BEI, BFG, BFH, BFI, CDG, CDH, CDI, CEG, CEH, CEI, CFG, CGH, CFI.

Limitations, however, may result in incorrect or inferior chromosomes if this procedure is unable to produce better results. To overcome this, mutation is used to provide unpredictability and speed up the development of solutions that were not anticipated, providing a new angle on the fitness function.

Original Chromosome:

100111101011 101100101001 101001101010 101111001010

2539 2857 2669 3018

After mutation:

100111101011 101010101001 101001101101 101111001010

2539 2857 2669 3018

Within a chromosome, the mutation modifies gene values selectively and does not impact other solutions. A new population of solutions is created after crossing and mutation to improve the overall quality. This loop can end under several circumstances, including a fitness threshold, a finite number of chromosomes, or a predetermined number of generations. By acting as stop criteria, these circumstances enable the evaluation of evolutionary functions' efficiency. Until a predetermined number of generations (g) is achieved, the selection, reproduction, and evaluation loop are applied recursively, with the new population replacing the existing one. The method eventually converges to a final solution that displays n chromosomes (timetables) with the soft constraint violations represented by minimized objective values. The project implements and codes the complete procedure.

- **Two chromosomes C_1 and C_2**

Chromosome C_1 :

1011100110111010100010100011 1/10100100101101010011010001111

Chromosome C_2 :

01101110100110110101000101000/11100110110100100100101010011

Crossover generation:

10111001101111010100010100011 1/01101110100110110101000101000

10111001101111010100010100011 1/11100110110100100100101010011

01101110100110110101000101000/10100100101101010011010001111

11100110110100100100101010011/10100100101101010011010001111

Crossover generation: 4 chromosomes

Results and Discussion

We used an actual data set for this study from the Girls Section of the Computer Science Program at the University of Tabuk Faculty of Computers and Information Technology. We'll utilize a sample dataset as well as a prototype model for our automated course scheduling.

Table 4: Timeslot & days

Timeslot & days	**8-8.30**	**8.30-9**	**9-9.30**	**9.30-10**	**10-10.30**	**10.30-11**	**11-11.30**	**11.30-12**	**12-12.30**	**12.30-1**	**1-1.30**	**1.30-2**	**2-2.03**	**2.30-3**	**3-3.30**	**3.30-4**
Su	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Mo	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Tu	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
We	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
Th	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

Table 5: Timetable in a Binary format

j\i	**000 0**	**000 1**	**001 0**	**001 1**	**010 0**	**010 1**	**011 0**	**011 1**	**100 0**	**100 1**	**101 0**	**101 1**	**110 0**	**110 1**	**111 0**	**111 1**
000	000 000 0	000 000 1	000 001 0	000 001 1	000 010 0	000 010 1	000 011 0	000 011 1	000 100 0	000 100 1	000 101 0	000 101 1	000 110 0	000 110 1	000 111 0	000 111 1
001	001 000 0	001 000 1	001 001 0	001 001 1	001 010 0	001 010 1	001 011 0	001 011 1	001 100 0	001 100 1	001 101 0	001 101 1	001 110 0	001 110 1	001 111 0	001 111 1
010	010 000 0	010 000 1	010 001 0	010 001 1	010 010 0	010 010 1	010 011 0	010 011 1	010 100 0	010 100 1	010 101 0	010 101 1	010 110 0	010 110 1	010 111 0	010 111 1
011	011 000 0	011 000 1	011 001 0	011 001 1	011 010 0	011 010 1	011 011 0	011 011 1	011 100 0	011 100 1	011 101 0	011 101 1	011 110 0	011 110 1	011 111 0	011 111 1
100	100 000 0	100 000 1	100 001 0	100 001 1	100 010 0	100 010 1	100 011 0	100 011 1	100 100 0	100 100 1	100 101 0	100 101 1	100 110 0	100 110 1	100 111 0	100 111 1

Within this portion of the research, we explore the difficult procedure of using a Genetic Algorithm (GA) to solve the University/Course Timetabling Problem (U/CTTP) at the University of Tabuk's Faculty of Computer Science and Information Technology - Girls portion. The first step is a thorough analysis of the benefits and drawbacks of the existing manual timetabling system, setting the stage for the automated alternative that will be suggested later. A key component of the project's approach is the identification and classification of hard and soft constraints, which are necessary for the GA's efficient design. The foundation of the GA process is a methodical approach to data encoding, population initialization, selection, crossover, mutation, and the iterative creation of new solutions. The work also offers a thorough grasp of the transition from encoded chromosomes to optimal schedules, shedding light on the subtleties and difficulties involved in using GAs. A practical component is added to the study by utilizing an actual dataset from the Faculty of Computer Science and Information Technology - Girls Section. This emphasizes how the produced solution aligns with the specific requirements of the institution. The effectiveness of the prototype is assessed at the end of the analysis section. The results are compared with those of the current manual timetabling system, and the potential advantages—such as better workload management, increased efficiency, and higher satisfaction among teachers and students are highlighted. The outcomes show that putting in place an automated course scheduling system is both feasible and effective, which is a major step forward in resolving the issues that manual timetabling techniques provide.

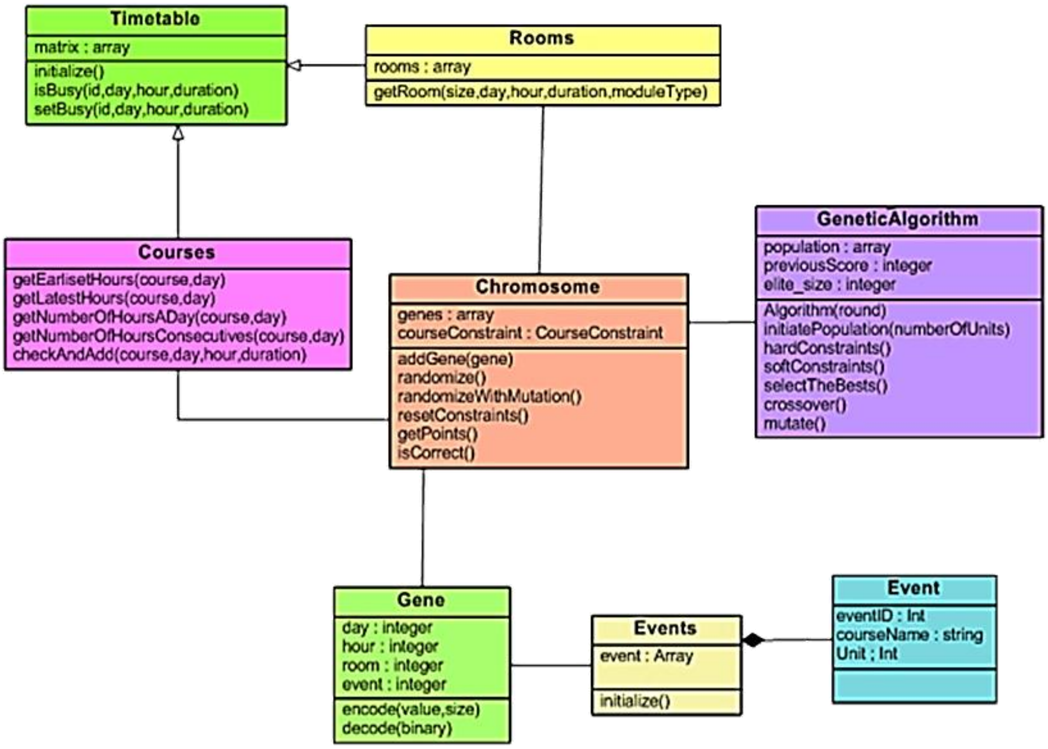


Figure 3: Class Diagram

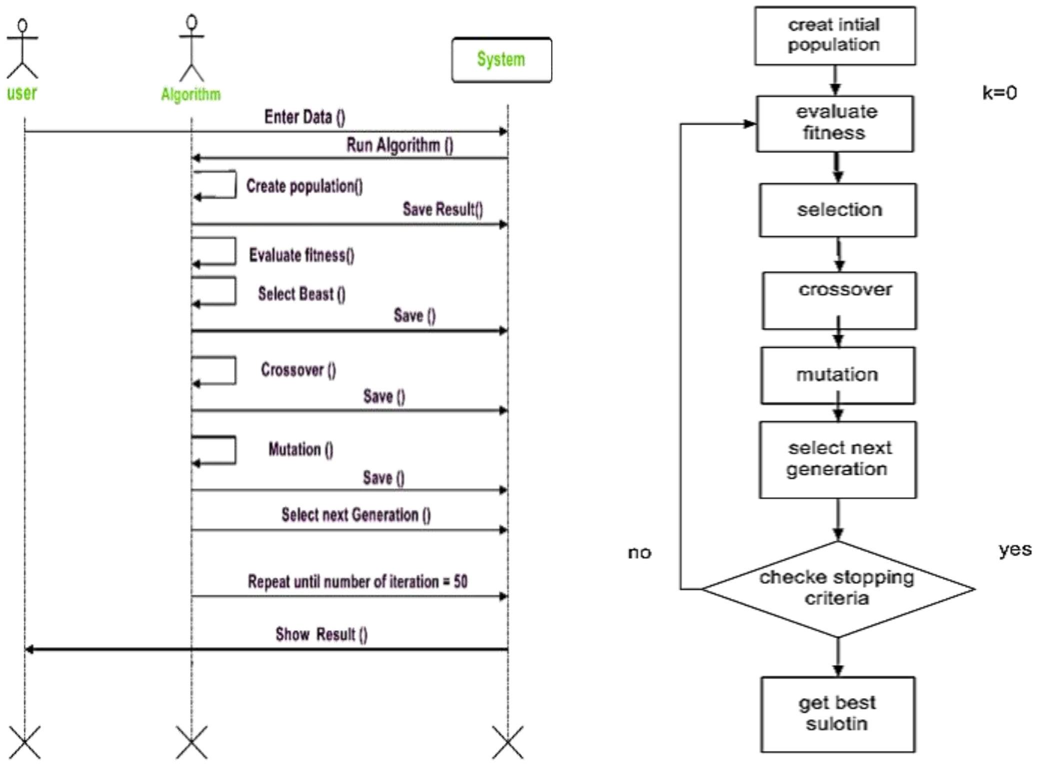


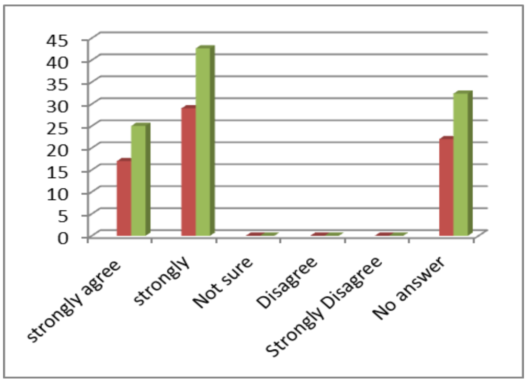
Figure 4: Sequence Diagram and Activity Diagram GA

15% of the 450 students in the University of Tabuk's Faculty of Computers and Information Technology responded to a survey, and it was found that a sizable percentage of them were unhappy with their course schedules. According to the replies, 22.05% of respondents agreed and 25% strongly agreed that they were unhappy with their course schedules. Additionally, 54.5% of the students changed their course schedules in concert with their lecturers as proactive actions. Remarkably, a quarter of the pupils said they would want their course schedules to be completely uninterrupted. Furthermore, 47.22% of the students had worked with their professors to adjust their course schedules. Of those, a portion preferred to have just little breaks ideally no more than an hour because greater breaks presented practical difficulties. Additionally, 20.41% of students said they would rather not have lectures beyond 2:00 p.m., and 30.56% of students said they would like to avoid delayed leave hours. The system's features are outlined with the primary goal of scheduling each course at a time that minimizes conflicts with other courses and room assignments in mind. The more course combinations, faculty availability, room limitations, and other factors that are taken into account, the harder the work becomes. By using data analysis to take into account variables including faculty preferences, historical trends, and student course requests, student course conflicts can be reduced. The suggested remedy combines several methods from scientific literature, specifically mentioning genetic algorithms as a possible instrument.

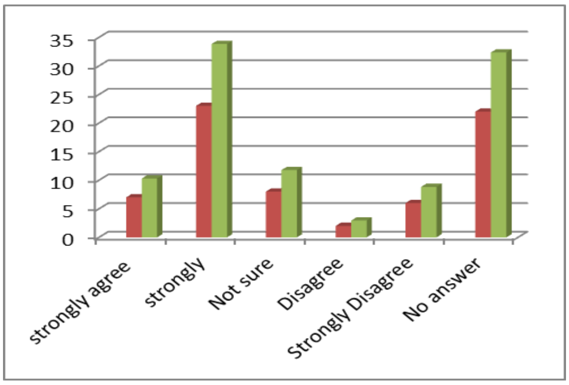
Table 6: Analysis of the Survey

Statements	Strongly agree	Agree	Not sure	Disagree	Strongly Disagree	No answer
I have time gaps in my course timetable, (if not agree go to no. 3).	25%	42.64%	0	0	0	32.35%
I am happy with the time gaps in my course timetable.	10.29%	33.82%	11.76%	2.94%	8.82%	32.35%
I have free days in my course timetable.	4.41%	10.29%	5.88%	20.58 %	58.82%	
I have one course only on one day in my timetabling.	16.17%	0	2.94%	27.94 %	52.94%	
I have courses at 3 p.m. and later.	16.17%	7.35%	2.49%	22.05 %	51.47%	
I am not happy with my course timetable.	25%	22.05%	17.64%	20.588 %	14.70%	
I modified my course timetable after agreement with the course's teacher academic officer.	23.52%	30.88%	10.29%	13.23 %	22.05%	

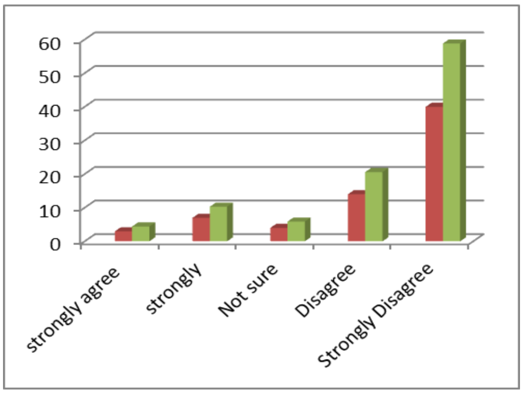
Q1. I have time gaps in my course timetable, (if not agree go to no. 3).



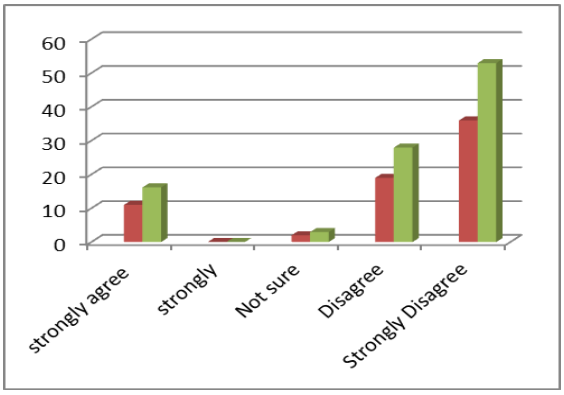
Q2. I am happy with the time gaps in my course timetable.



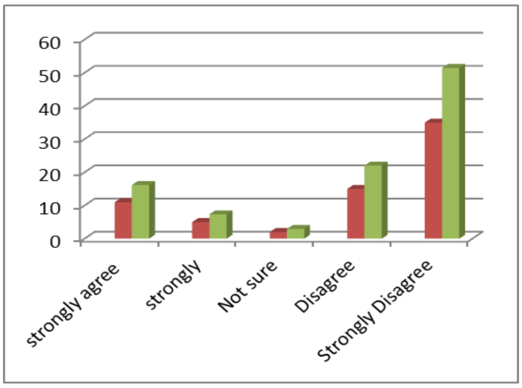
Q3. I have free days in my course timetable.



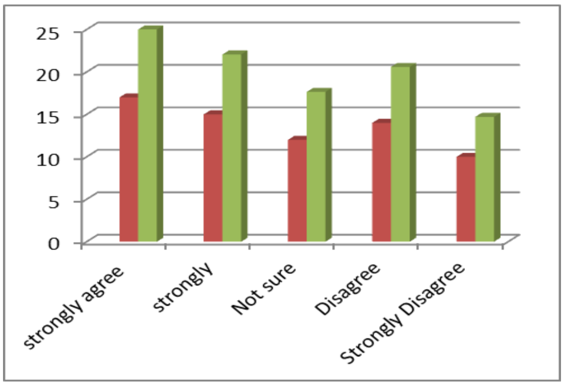
Q4. I have one course only at one day in my timetabling.



Q5. I have courses at 3 p.m. and later.



Q6. I am not happy with my course timetable.



Q7. I modified my course timetable after agreement with the course's teacher\academic officer.

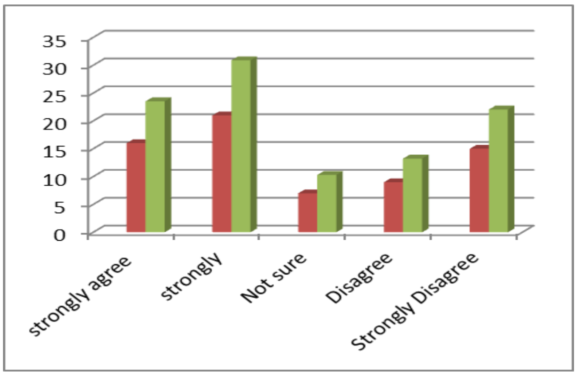


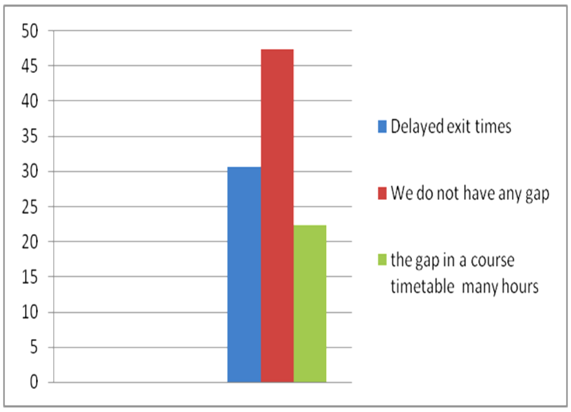
Figure 5: Question 01-07 for Analysis

Table 7: Analysis of the open questions of the survey

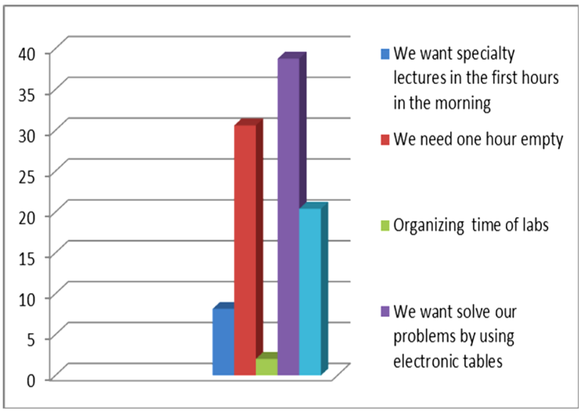
Analysis Open questions	
8	**Why you are not happy with your current course timetable?**

Answer	Delayed exit times	25.93%	**39.71% Who answered**
	We do not have any gap	37.04%	
	the gap in a course time table many hours	37.04%	
	Other reasons	60.29%	
9	**For what reason you have modified your current course timetable?**		
Answer	Delayed exit times	30.56%	**52.94% Who answered**
	We do not have any gap	47.22%	
	the gap in a course time table many hours	22.22%	
	Other reasons	47.06%	
10	**Any suggestions to improve the course timetable?**		
Answer	We want specialty lectures in the first hours of the morning	8.16%	**72.06% Who answered**
	We need one hour of empty	30.61%	
	Organizing time of labs	2.04%	
	We want to solve our problems by using electronic tables	38.78%	
	We do not want lectures delayed after 2 p.m.	20.41%	
	Other reasons	27.94%	

Q8: For what reason you have modified your current course timetable?



Q9: Any suggestions to improve the course timetable?



Q10: Why you are not happy with your current course timetable?

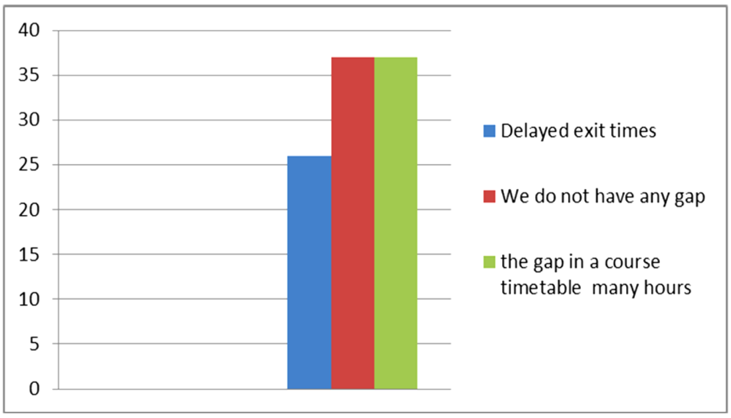


Figure 6: Question 8-10 for Analysis

The system's efficiency, error-free functioning, and versatility across several modes from completely automated timetabling to human data entry are highlighted in the specs. The system is made to automatically create schedules based on a variety of factors and preferences, guaranteeing that all necessary courses are offered to accommodate the preferences of the students. Thorough testing has been done to find and fix problems and defects, guaranteeing the dependability of the system. The system's adaptability is highlighted, enabling administrators to quickly change preferences and resources like the number of classrooms and courses to suit changing demands. The overall goal of the suggested Automatic Courses Timetabling system is to provide a flexible and efficient way to handle the issues raised by the survey.

Regarding complexity, considering the Genetic Algorithm's (GA) reliance on selection, crossover, mutation processes, and factors such as population size (n) and number of generations (G), there is no generic temporal complexity for GA in course scheduling. The GA overhead is reflected in a simpler metric, g * n, which aims for solutions with a high rate of convergence. The technique is designed to be implementable across several operating systems and computer languages that support object-oriented ideas, hence ensuring reliability. Performance-wise, the system's effectiveness is assessed across about 20 run times, demonstrating its capacity to find the best solutions throughout the course scheduling procedure. Together, these factors form the methodology's guiding principles, offering a solid framework for the creation and assessment of the Automatic Courses Timetabling system.

Conclusion

This thorough study tackles the significant issues related to course scheduling in the Faculty of Computer Science and Information Technology - Girls Section of the University of Tabuk. Due to the shortcomings of the current manual timetabling method in terms of efficiency, flexibility, and satisfying various restrictions, an automated course scheduling system utilizing a Genetic Algorithm (GA) was proposed and put into place. A thorough examination of the existing system, the determination of hard and soft restrictions, and a methodical approach to GA procedures were all part of the research. The prototype was shown to be effective through the use of actual data, highlighting possible advantages including better task management and increased instructor and student satisfaction. The study aligned the proposed system with the unique demands of the university community by including student preferences, as disclosed through a survey. The suggested system prioritizes efficiency, error-free operation, and adaptability to offer a flexible solution to the problems associated with manual timetabling and contribute significantly to the improvement of educational scheduling procedures. The article presents a requirements management plan that takes into account the issues raised by the survey and emphasizes the necessity of effective solutions given the current state of technology and the speed at which advancement is occurring. Large academic institutions, especially universities, struggle to keep track of several sections and courses, therefore prompt and effective solutions are required. The creation of an Automatic course timetabling system is the suggested solution, which aims to efficiently expedite the process of producing course schedules. The method would prevent scheduling conflicts by accounting for a variety of resources, such as the number of instructors, classes, and classrooms. This automatic course scheduling system's prototype aims to give students a quick and effective tool for organizing their schedules.

REFERENCES:

[1] A. ELEN and İ. ÇAYIROĞLU, "Solving of Scheduling Problem with Heuristic Optimization Approach," *Technol. (Karabuk Univ.*, vol. 13, no. 3, pp. 159–172, 2010, [Online]. Available: http://jestech.karabuk.edu.tr/arsiv/2010-03/PDF/5 - A.Elen.pdf

[2] Complexity, "Retracted: Improved Genetic Algorithm to Solve the Scheduling Problem of College English Courses," *Complexity*, vol. 2023, pp. 1–1, 2023, doi: 10.1155/2023/9839523.

[3] A. Rezaeipanah, S. S. Matoori, and G. Ahmadi, "A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search," *Appl. Intell.*, vol. 51, no. 1, pp. 467–492, 2021, doi: 10.1007/s10489-020-01833-x.

[4] G. Alnowaini and A. A. Aljomai, "Genetic Algorithm for Solving University Course Timetabling Problem Using Dynamic Chromosomes," *2021 Int. Conf. Technol. Sci. Adm. ICTSA 2021*, 2021, doi: 10.1109/ICTSA52017.2021.9406539.

[5] S. N. Tung, J. B. Jaafar, I. A. Aziz, H. G. Nguyen, and A. N. Bui, "Genetic Algorithm for Solving Multi-Objective Optimization in Examination Timetabling Problem," *Int. J. Emerg. Technol. Learn.*, vol. 16, no. 11, pp. 4–24, 2021, doi: 10.3991/ijet.v16i11.21017.

[6] S. Abdennadher and M. Marte, "University course timetabling using constraint handling rules," *Appl. Artif. Intell.*, vol. 14, no. 4, pp. 311–325, 2000, doi: 10.1080/088395100117016.

[7] S. Tung Ngo, J. Jafreezal, G. Hoang Nguyen, and A. Ngoc Bui, "A Genetic Algorithm for Multi-Objective Optimization in Complex Course Timetabling," *ACM Int. Conf. Proceeding Ser.*, pp. 229–237, 2021, doi: 10.1145/3457784.3457821.

[8] M. Elliot, F. S. Gbenga, and M. E. J, "Enhanced Heuristic Teaching Timetabling Algorithm Using Genetic algorithm," *Int. J. Sci. Technol. Res.*, vol. 9, no. 4, pp. 3804–3814, 2020.

[9] A. A. Gozali and S. Fujimura, "Solving University Course Timetabling Problem Using Multi-Depth Genetic Algorithm," *SHS Web Conf.*, vol. 77, no. October 2019, p. 01001, 2020, doi: 10.1051/shsconf/20207701001.

[10] S. Imran Hossain, M. A. H. Akhand, M. I. R. Shuvo, N. Siddique, and H. Adeli, "Optimization of University Course Scheduling Problem using Particle Swarm Optimization with Selective Search," *Expert Syst. Appl.*, vol. 127, pp. 9–24, 2019, doi: 10.1016/j.eswa.2019.02.026.

[11] W. Wen-Jing, "Improved adaptive genetic algorithm for course scheduling in colleges and universities," *Int. J. Emerg. Technol. Learn.*, vol. 13, no. 6, pp. 29–42, 2018, doi: 10.3991/ijet.v13i06.8442.

[12] A. O. Modupe, O. E. Olusayo, and O. S. Olatunde, “Development of a University Lecture Timetable using Modified Genetic Algorithms Approach,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 9, p. 2277, 2014, [Online]. Available: www.ijarcsse.com
[13] N. G. A. P. H. Saptarini, I. W. Suasnawa, and P. I. Ciptayani, “Senior high school course scheduling using genetic algorithm,” *J. Phys. Conf. Ser.*, vol. 953, no. 1, 2018, doi: 10.1088/1742-6596/953/1/012067.
[14] J. B. Matias, A. C. Fajardo, and R. M. Medina, “A fair course timetabling using genetic algorithm with guided search technique,” *Proc. 2018 5th Int. Conf. Bus. Ind. Res. Smart Technol. Next Gener. Information, Eng. Bus. Soc. Sci. ICBIR 2018*, pp. 77–82, 2018, doi: 10.1109/ICBIR.2018.8391170.
[15] M. AlJarrah, A. AlSawalqah, and S. AlHamdan, “Developing Courses Timetable System for Academic Departments Using Genetic Algorithm,” *Jordanian J. Comput. Inf. Technol.*, no. February 2021, p. 1, 2017, doi: 10.5455/jjcit.71-1465735040.
[16] E. A. Abdelhalim and G. A. El Khayat, “A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA),” *Alexandria Eng. J.*, vol. 55, no. 2, pp. 1395–1409, 2016, doi: 10.1016/j.aej.2016.02.017.
[17] . S. K. J., “Exam Timetabling Problem Using Genetic Algorithm,” *Int. J. Res. Eng. Technol.*, vol. 03, no. 05, pp. 649–654, 2014, doi: 10.15623/ijret.2014.0305120.
[18] Oyeleye, C. Akinwale, “Performance evaluation of simulated annealing and genetic algorithm in solving examination timetabling problem,” *Sci. Res. Essays*, vol. 7, no. 17, pp. 1727–1733, 2012, doi: 10.5897/sre11.2103.
[19] M. S. Kohshori, D. Zeynolabedini, M. S. Liri, and L. Jadidi, “Multi Population Hybrid Genetic Algorithms for University Course Timetabling Problem,” *Int. J. Inf. Technol. Comput. Sci.*, vol. 4, no. 6, pp. 1–11, 2012, doi: 10.5815/ijitcs.2012.06.01.
[20] M. S. Kohshori and M. S. Abadeh, “Hybrid Genetic Algorithms for University Course Timetabling,” *Int. J. Comput. Sci. Issues*, vol. 9, no. 2, pp. 446–455, 2012, [Online]. Available: http://www.ijcsi.org/papers/IJCSI-9-2-2-446-455.pdf
[21] H. Larget, “Genetic Algorithms used in Timetable Management By,” no. April, 2012.
[22] S. Yang and S. N. Jat, “Genetic algorithms with guided and local search strategies for university course timetabling,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 1, pp. 93–106, 2011, doi: 10.1109/TSMCC.2010.2049200.
[23] X. Deng, Y. Zhang, B. Kang, J. Wu, X. Sun, and Y. Deng, “An application of genetic algorithm for university course timetabling problem,” *Proc. 2011 Chinese Control Decis. Conf. CCDC 2011*, pp. 2119–2122, 2011, doi: 10.1109/CCDC.2011.5968555.
[24] J. H. Holland, *Adaptation in neural and artificial system*. The MIT Press, 1975.
[25] S. R.Sutar and R. S. Bichkar, “University Timetabling based on Hard Constraints using Genetic Algorithm,” *Int. J. Comput. Appl.*, vol. 42, no. 15, pp. 1–7, 2012, doi: 10.5120/5765-7964.
[26] J. Wright, “Changing Expectations of Academic Libraries,” pp. 4846–4852, 2014, doi: 10.4018/978-1-4666-5888-2.ch476.