A Greedy-based Algorithm in Optimizing Student's Recommended Timetable Generator with Semester Planner

Khyrina Airin Fariza Abu Samah[1], Siti Qamalia Thusree[2]
Ahmad Firdaus Ahmad Fadzil[3], Lala Septem Riza[4], Shafaf Ibrahim[5], Noraini Hasan[6]
Faculty of Computer and Mathematical Sciences[1, 2, 5, 6]
Universiti Teknologi MARA Cawangan Melaka Kampus Jasin, Melaka, Malaysia[1, 2, 5, 6]
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Shah Alam, Selangor, Malaysia[3]
Department of Computer Science Education, Universitas Pendidikan Indonesia, Indonesia[4]

***Abstract*—Semester planner plays an essential role in students' society that might help students have self-discipline and determination to complete their studies. However, during the COVID-19 pandemic, they faced difficulty organizing time management and doing a manual schedule. It resulted in substantial disruptions in learning, internal assessment disturbances, and the cancellation of public evaluations. Hence, this research aims to optimize the recommended semester planner, Timetable Generator using a greedy algorithm to increase student productivity. We identified three-set control functions for each entered information: 1) validation for the inserted information to ensure valid data and no redundancy, 2) focus scale, and 3) the number of hours to finish the activity. We calculate the priority task sequence to achieve the best optimal solution. The greedy algorithm can solve the optimization problem with the best optimal solution for each situation. Then, we executed it to make a recommended semester planner. From the test conducted, the functionality shows all the features successfully passed. We validate using test accuracy for the system's reliability by evaluating it compared to the Brute Force algorithm, and the trends increase from 60% to 100%.**

Keywords*—Greedy algorithm; optimization; recommendation system; semester planner

I. Introduction

The COVID-19 pandemic has made a huge impact on the educational system around the world, forcing lockdown in schools, universities, and colleges. Hence, Singh and Kant [1] claimed it caused a significant disrupted student learning, internal examinations, and the termination of public exams for qualification due to inferior alternatives. Based on the research, most students are not well prepared for classes. The condensed class schedule negatively impacts their learning experiences [2]. It corresponded with low experiences and academic achievement by the student.

A survey conducted towards 126 students of Universiti Teknologi MARA Melaka Kampus Jasin. They responded that they used paper and pen to organize semester planners manually, and 84.9% of students found it challenging to organize semester planners with upcoming quizzes and tests. Additional time to new learning content is taught to enhance their performance. However, the effect of this treatment is a slight difference throughout the distribution of student performance. While the most benefit is for high-performing students, low-performing students do not benefit. Foulkes et al. [3] highlight that the research argued that it is crucial to determine the content of additional time instruction to explain this pattern.

Besides, balancing studying with the fun stuff in life can be challenging to maintain. Unstructured academic lifestyle is one of the impacts of poor sleep quality in university life [4]. It increases mental the risk of mental illness and poor academic achievement. Panda et al. [5] mentioned that based on research on academic stress students, the stress factors include many assignments, competition with other students, failures, financial problems, poor relationships, lectures, or family matters.

Thus, planning a scheduler is an effective solution to organize and navigate students' time management [6]. A semester planner is an organized schedule where students can create outlines and awareness of the study methods and the effectiveness of their study practices for the current semester and learning goals. Students become self-discipline, deterministic, and more confident [7]. In addition, it helps them build self-regulated learning skills to succeed in various learning environments and workplaces. As a primary time management tool, a scheduler comes in a list of times when events and actions occur. Brioso et al. [8] supported that planning and controlling did not starve due to insufficient input in an intrinsically challenging task.

The primary substance of time management is reducing stress [9]. Of course, building a schedule for the current semester is a tedious task. Students need to fit all the courses into a single scheduler, which would take time and patience. The time management principle is time management on how to plan the specific tasks and work and organized. Otherwise, it can affect entire planning and work. Based on the same survey, 83.8% of the students claimed they never use revision planners' automated class timetable generator.

The research was sponsored by Universiti Teknologi MARA Cawangan Melaka under the TEJA Grant 2021 (GDT 2021/1-28)

Therefore, to overcome the problem, a recommended semester planner using optimization of the greedy-based algorithm was used quickly to find an approximate solution in the optimization prickly. This parallel with the greedy-based algorithm concept, to solve problems in getting the globally optimal solution with the choice seems best by making a locally optimal choice [10]. Moreover, the greedy allocation is compatible with a dynamic heterogeneous resource environment linked to the scheduler via a homogenous communication environment [11]. Thus, Choudhary and Peddoju [12] claimed that it is easier to assess the run with more efficient use of available resources. It allows UiTM students to identify peak study load times to ensure they have plenty of time to complete all assignments. The achievement of each target would bring in the motivation to achieve another target.

II. Related Work and Techniques

This section describes the related works in scheduling and related techniques related to the study.

A. *Related Work in Scheduling*

There are some related works in scheduling such as course timetables, high school timetables [13], exam timetables, and some of the other schedules used in the workplace. Although many intelligent practices have been used in recent years to solve course timetabling, there are still many areas for improvement. Wang et al. [13] claimed that to effectively solve the Greedy Algorithm and provide a high-performance initial Genetic Algorithm population, they suggested using a Greedy and Genetic Fusion Algorithm. The hybrid Cat Swarm Optimization algorithm-based application solves the school schedule problem. It is efficient, easy to use, and fast. The demonstration conducts by experiments with real-life input data to test the efficiency and performance. The test case uses the same timetable and the same description statistics. The result shows that the hybrid CSO-based algorithm performs better in less computational time than many other existing approaches.

For timetable examination, a new proposed combination of three approaches: multiple criteria, maximum independent set, and heuristic graph to solve the problem of timetable examination [14]. There is also have some related work on schedulers in the workplace, such as flight schedules. Optimization control and algorithm design have also been hot issues in China's research into unusual flights. Based on a better knowledge of the NP problem under the flight recovery model, the researcher validates the usefulness of the greedy random adaptive algorithm in the problem-solving process [15].

B. *Related Techniques*

An algorithm is a step-by-step approach to problem-solving, widely used for data processing, calculation, and other related mathematical and computer operations. Different algorithms can easily and quickly perform operations or solve problems in terms of efficiency. There are three related techniques: Evolutionary Algorithm (EA), Greedy Algorithm, and Brute Force.

The Evolutionary Algorithm (EA) development has been significant among the research and optimization techniques set in the last decade [16]. EAs are a set of modern heuristics used with great complexity in many applications. It was the engine of a field known as Evolutionary Computation (EC) that successfully solved complex problems [17]. The advantages of EC techniques derived from gaining fitness and flexibility to the goal combined with robust behavior. Nowadays, EC is considered an adaptable problem-solving concept, particularly in complex optimization problems.

A greedy-based algorithm produces a good solution to some math but not others. As mentioned by Prabhaharan et al. [18], most of the issues they are working on will have two properties; firstly, a greedy choice property is defined as the best way to make any choice to solve sub-problems that might arise later. A greedy algorithm's choice might depend on the decision but not the future or all the subproblem's solutions. It makes iteratively one greedy choice to minimize a problem into an optimal solution. Secondly, an optimal substructure is an optimal solution to solve the problem includes optimal solutions to the subproblem. Then a will problem displays optimal substructure. The problem of optimization is finding the best solution that seems all feasible [19].

Brute Force defines a straightforward problem strategy, commonly dependent on problem statement and concept definition. Definition of 'force' is a computer strategy but not intellect. Another approach to describe Brute Force is 'Just do it', and sometimes, the technique is easiest to use [20]. Exhaustive or Brute Force searches can generate and test the basic solving problem method and algorithm paradigm. Mahoor et al. [21] claimed it consists of the listed systematic potential candidate for solution and testing if each candidate meets the problem statement. Therefore, this approach is used commonly when the problem is small enough that heuristics can reduce the number of possible solutions to a manageable number. Also, identifying a solution recovery model takes more time than speed.

As a result, an evolutionary algorithm is beneficial with conceptual simplicity, while Brute Force is widely applicable and known for its simplicity. However, the greedy-based algorithm is more accessible to implement and faster than the other two. Although the Greedy-based algorithm has not always reached the optimal global solution, most of the time reaches the sub-optimal solution. Thus, greedy-based techniques are the best solution for this project in schedule recommendation of timetable generator with semester planner.

III. Methods

This section divides research methodology into four phases: gather information, design and development and system validation.

A. *Gather Information Phase*

The survey was conducted on 126 respondents of students from UiTM Cawangan Melaka Kampus Jasin on their experience to understand better the real problem faced by them. Table I shows the summary of survey information. About 85.70% of respondents agreed that they have difficulties organizing their study planner, while the rest, 14.30%, voted with no issues. Some of them never use study planners with 58.13%, followed by book or handphone planners with 24.78%, sticky notes with 14.60%, and timetable 2.49%.

TABLE I. SURVEY DETAILS AND RESPONSES TO GATHER INFORMATION

Questions	Item Response	Percentage Response
Difficulties To Organize Study Planner	Yes	85.70%
	No	14.30%
Current Method Study Planner	Never	58.13%
	Book/Handphone Planner	24.78%
	Use Sticky Note	14.60%
	Timetable	2.49%

B. Timetable Generator Flowchart

We use the flowchart as in Fig. 1 to design how the timetable generator operates by defining the system's module, data, and architecture to specify the system requirements. It starts with key-in details information for the user's hobby, extracurricular, and subject details for the current semester. We set the control functions for each information: 1)validation for the inserted information to ensure valid data and no redundancy, 2)focus scale, and 3) the number of hours to finish the activity. The information entered by the user can be updated or deleted before we generate the planner using the Greedy algorithm.

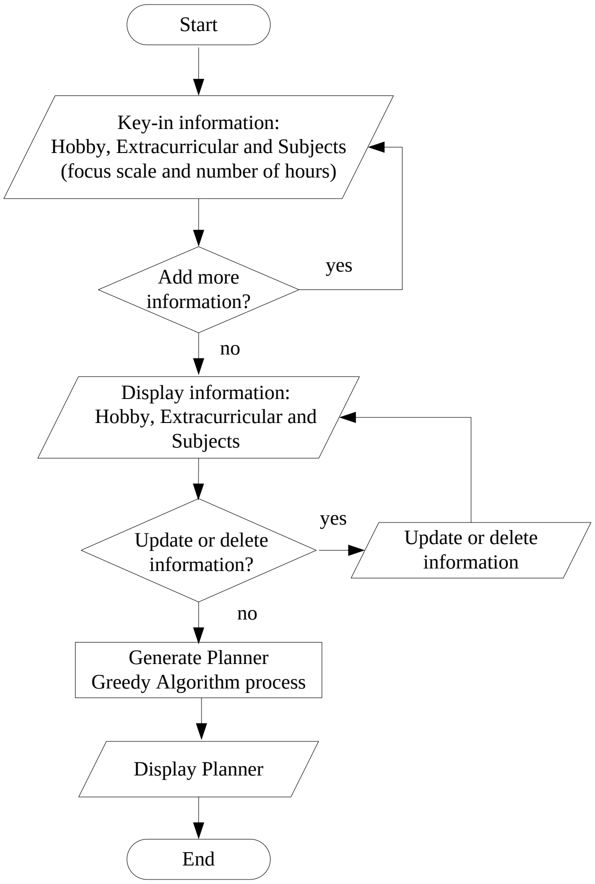

Start
Key-in information:
Hobby, Extracurricular and Subjects
(focus scale and number of hours)
Add more information?
yes
no
Display information:
Hobby, Extracurricular and Subjects
Update or delete information?
yes
Update or delete information
no
Generate Planner
Greedy Algorithm process
Display Planner
End

Fig. 1. Flowchart for Timetable Generator Planner.

C. System Interface

System design or user interface design is designing an interface in software or computerized devices focusing on locks or style. This research aims to design an interface that is easy to use and understandable by the user. User interface design usually refers to graphical user interfaces.

D. Greedy Algorithm Development

Fig. 2 shows the Greedy algorithm flowchart of the phase involved in this algorithm to achieve the best optimization result. The algorithm needs to solve scheduling problems in tasks that need to be completed, required time to complete each task, and priority of each task. Three main processes involved:

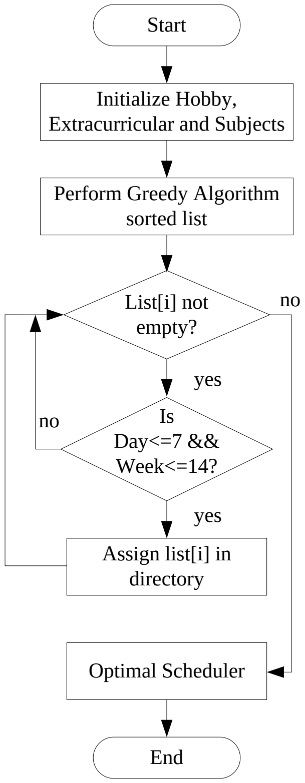

Start
Initialize Hobby, Extracurricular and Subjects
Perform Greedy Algorithm sorted list
List[i] not empty?
no
yes
no
Is Day<=7 && Week<=14?
yes
Assign list[i] in directory
Optimal Scheduler
End

Fig. 2. Phases in Optimizing Greedy Algorithm for Timetable Generator.

1) *Step 1: Initialization:* The algorithm initializes all user variables and sorts the list with the greedy approach. The data will be classified into three lists: name, focus scale, and complete time.

2) *Step 2: Sort list:* Sort the tasks based on decreasing order of priority. However, in some cases, the priorities of different tasks are the same. This algorithm recommends the list of tasks that require a higher focus scale and the shortest time completed the task to minimize the problem. The algorithm would give preference to a higher focus scale and tasks that take less time to complete so that the higher focus scale will lead to a higher score, and more time needed will reduce the score. After being classified into a list, this phase has created another list of pairs [key, value] as shown below. Each of the list pairs is a combination of hobby, extracurricular, and subject.

- Complete time [complete time, name].
- Focus scale [focus scale, name].
- Priority [(focus scale/complete time), name].

Then, the system would proceed with the sorting process for each list in decreasing the order of "key". After all the variables have been sorted, the algorithm will assign lists in a directory with a specific condition.

3) Step 3: Optimal scheduler: Assume that task in priority sequence is n, set priority = {(8, ICT662), (7, Badminton), …n} and tasks use the same resources from of hobby, extracurricular and subject task, with priority score for each task. The algorithm use planner set to store the selected task. The task chosen is ICT662, and Monday is initialized "ICT662". Then the algorithm will go through 14 weeks to check whether the task has an assignment, quiz, or test week. For example, ICT662 has a quiz on week five, then Monday on week five is initialized "Quiz 1 ICT662". An algorithm will proceed initialized in planner set with next day and weeks until all the task in the priority list is empty.

E. System Validation

Lastly, we proceed with system validation to test the functionality and reliability of the system. A functionality test runs to test whether the system's function is running smoothly or not. If there are bugs when running the functionality test, we need to correct the function to ensure it meets the requirements. Functionality testing tests and identifies which function is performed with or without error. If an error occurs in this phase, we need to fix the error before the implementation. For the reliability test, we compare the result of the Timetable Generator with the Brute Force algorithm to check the result's accuracy. We tested this testing phase with developers, lectures, and UiTM Jasin students to know whether this project is performing well or not.

IV. Result and Discussion

A. Functionality Testing

We have to test and check the core's function application, input, button, table, and more during this phase. To test the system's functionality, Table II describes consideration of features criterion where we test it accordingly and fill up the result. There are six features involved: registration and login, subject, hobby and extracurricular, updating user's details, a sequence of the focus scale, complete time and priority, planner and print planner.

The output of Timetable Generator, as in Fig. 3, display the sorted list based on the focus scale, complete time and priority in the list box. These list boxes show the pair between focus scale and complete time with all user tasks. The sorted priority list box is the greedy algorithm result sequence with the score and name of each task. Finally, Fig. 4 shows the output result of the Timetable Generator or the semester planner with an optimal scheduler. We successfully passed all the features, and finally, users can print a semester planner in pdf form as the last feature development.

B. Reliability Testing

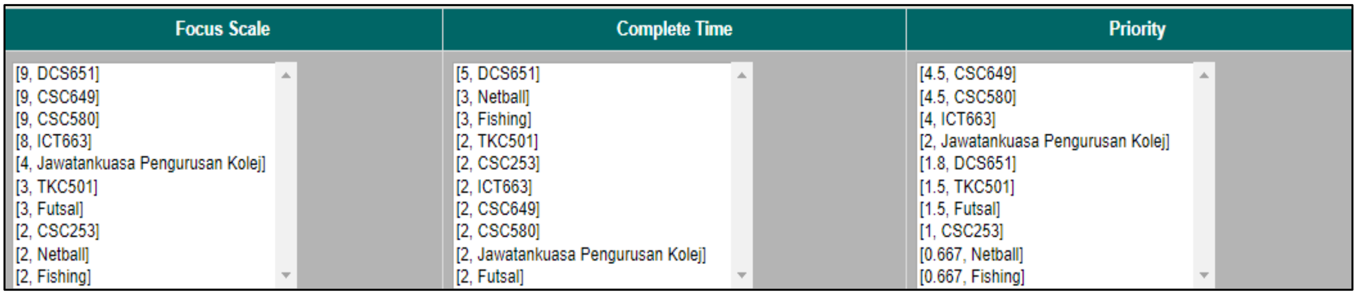
We validate the reliability testing based on the system's accuracy in this phase and use ten tests to run the validation. The system's response might have a different condition with different sequences of priority and Brute Force result on the hobby, extracurricular, and subject enrol. Table III shows the output from the system prototype and Brute Force algorithm result by manual calculation. We test the accuracy testing based on these two rules: First, give preference to higher priorities, leading to a higher score. Second, give preference to tasks that take less time to complete, so tasks that take more time will reduce the score. Next, we calculate the accuracy based on the similarity result sequence of Brute Force and Timetable Generator.

Table IV shows the summary comparison on the research accuracy result, *p* between Brute Force and our proposed study. Only attempt 1 has a slight difference out of five; only three items have similar according to the result sequence. The rest outcome gained is similar for both reliability tests.

We constructed the visualization of an accuracy graph from the result of 10 output. Fig. 5 shows the 2-axes graph, increasing from 0.60 (60%) to 1.0 (100%). It proves that the Timetable Generator system is accurate and can be used for the student to overcome the problem.

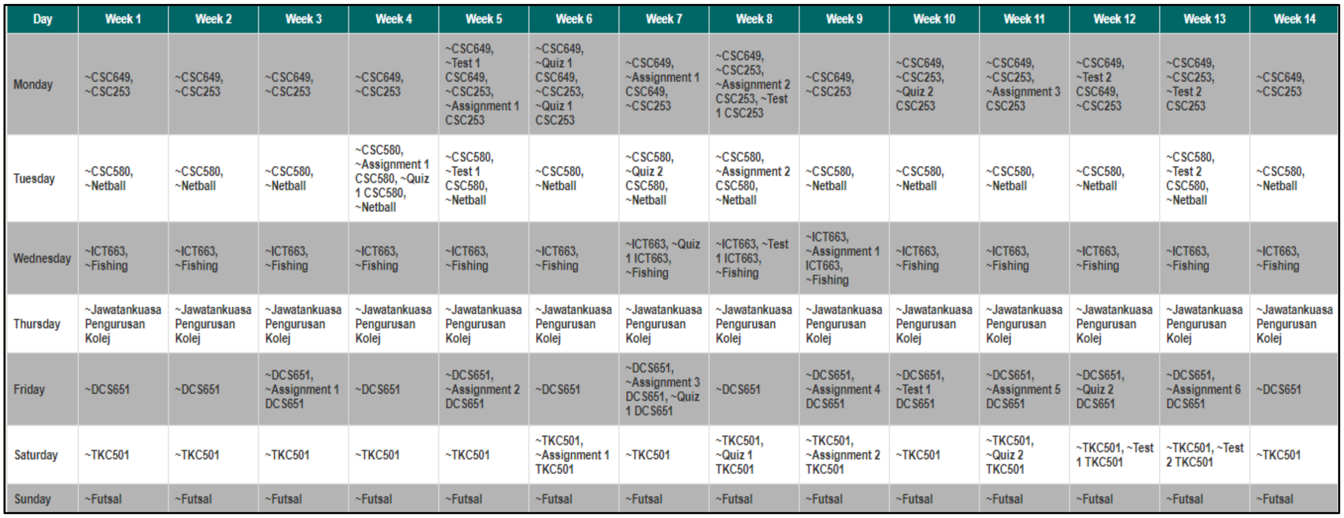
TABLE II. Features Comparison between Recommendation Techniques

Features	**Details**	**Status**
Registration and login	The system can register a new account with an encrypted password and allow users to login into the system with the correct combination of attributes needed.	Passed
Subject, hobby and extracurricular	The system can store the user's subject, hobby, and extracurricular details in the database. Before storing the details, the system will check whether all of the data are inserted or not. The incomplete information would not allow being stored in the database.	Passed
Updating user's details	The system retrieved all user details from the database and displayed them in a grid view. Users can update some data and store it back in the database.	Passed
A sequence of complete time, focus scale and priority	The system can list decreasing order of complete time, focus scale and priority.	Passed
Planner	The system prescribes a schedule of semester planners that is the best based on the priority sequence. The schedule will be displayed in the form of a table that contains seven days and 14 weeks of a semester. Thus, the system will display the subject details based on assignment week, quiz week and test week.	Passed
Print planner	Print the planner in the form of a PDF.	Passed



Focus Scale	Complete Time	Priority
[9, DCS651] [9, CSC649] [9, CSC580] [8, ICT663] [4, Jawatankuasa Pengurusan Kolej] [3, TKC501] [3, Futsal] [2, CSC253] [2, Netball] [2, Fishing]	[5, DCS651] [3, Netball] [3, Fishing] [2, TKC501] [2, CSC253] [2, ICT663] [2, CSC649] [2, CSC580] [2, Jawatankuasa Pengurusan Kolej] [2, Futsal]	[4.5, CSC649] [4.5, CSC580] [4, ICT663] [2, Jawatankuasa Pengurusan Kolej] [1.8, DCS651] [1.5, TKC501] [1.5, Futsal] [1, CSC253] [0.667, Netball] [0.667, Fishing]

Fig. 3. Snapshot Output of a Sorted List of Information in Timetable Generator.



Day	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
Monday	~CSC649, ~CSC253	~CSC649, ~CSC253	~CSC649, ~CSC253	~CSC649, ~CSC253	~CSC649, ~Test 1 CSC649, ~CSC253, ~Assignment 1 CSC253	~CSC649, ~Quiz 1 CSC649, ~CSC253, ~Quiz 1 CSC253	~CSC649, ~Assignment 1 CSC649, ~CSC253	~CSC649, ~CSC253, ~Assignment 2 CSC253, ~Test 1 CSC253	~CSC649, ~CSC253	~CSC649, ~CSC253, ~Quiz 2 CSC253	~CSC649, ~CSC253, ~Assignment 3 CSC253	~CSC649, ~Test 2 CSC649, ~CSC253	~CSC649, ~CSC253, ~Test 2 CSC253	~CSC649, ~CSC253
Tuesday	~CSC580, ~Netball	~CSC580, ~Netball	~CSC580, ~Netball	~CSC580, ~Assignment 1 CSC580, ~Quiz 1 CSC580, ~Netball	~CSC580, ~Test 1 CSC580, ~Netball	~CSC580, ~Netball	~CSC580, ~Quiz 2 CSC580, ~Netball	~CSC580, ~Assignment 2 CSC580, ~Netball	~CSC580, ~Netball	~CSC580, ~Netball	~CSC580, ~Netball	~CSC580, ~Netball	~CSC580, ~Test 2 CSC580, ~Netball	~CSC580, ~Netball
Wednesday	~ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Quiz 1 ICT663, ~Fishing	~ICT663, ~Test 1 ICT663, ~Fishing	~ICT663, ~Assignment 1 ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Fishing	~ICT663, ~Fishing
Thursday	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej	~Jawatankuasa Pengurusan Kolej
Friday	~DCS651	~DCS651	~DCS651, ~Assignment 1 DCS651	~DCS651	~DCS651, ~Assignment 2 DCS651	~DCS651	~DCS651, ~Assignment 3 DCS651, ~Quiz 1 DCS651	~DCS651	~DCS651, ~Assignment 4 DCS651	~DCS651, ~Test 1 DCS651	~DCS651, ~Assignment 5 DCS651	~DCS651, ~Quiz 2 DCS651	~DCS651, ~Assignment 6 DCS651	~DCS651
Saturday	~TKC501	~TKC501	~TKC501	~TKC501	~TKC501	~TKC501, ~Assignment 1 TKC501	~TKC501	~TKC501, ~Quiz 1 TKC501	~TKC501, ~Assignment 2 TKC501	~TKC501	~TKC501, ~Quiz 2 TKC501	~TKC501, ~Test 1 TKC501	~TKC501, ~Test 2 TKC501	~TKC501
Sunday	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal	~Futsal

Fig. 4. Semester Planner using Greedy Algorithm Optimization.

TABLE III. RELIABILITY TESTING BASED ON ACCURACY TEST BETWEEN BRUTE FORCE AND TIMETABLE GENERATOR

No	**Brute Force Result, *p***			**Timetable Generator Result, *p***
	Focus scale, *f*	**Complete time, *t***	**Priority, *p* = *f/t***	
1	[2, ENT600] [1, ITS610] [1, IPK501] [1, CSP650] [1, ISEC]	[2, ENT600] [2, ITS610] [2, IPK501] [2, CSP650] [2, ISEC]	[1, ENT600] [0.5, ITS610] [0.5, IPK501] [0.5, CSP650] [0.5, ISEC]	[1, ENT600] [0.5, ITS610] [0.5, IPK501] [0.5, ISEC] [0.5, SCP650]
2	[6, ENT600] [3, CSC662] [3, ICT662] [2, ITS610]	[5, CSC662] [3, ENT600] [2, ITS610] [2, ICT662]	[1.5, ICT662] [1.333, ENT600] [1, ITS610] [0.6, CSC662]	[1.5, ICT662] [1.333, ENT600] [1, ITS610] [0.6, CSC662]
3	[10, Cooking] [8, CSP650] [7, ENT600] [6, Brass Band] [4, IPK501] [3, ITS610]	[2, CSP650] [2, ENT600] [2, ITS610] [2, IPK501] [2, Brass Band] [1, Cooking]	[10, Cooking] [4, CSP650] [3.5, ENT600] [3, Brass Band] [2, IPK501] [1.5, ITS610]	[10, Cooking] [4, CSP650] [3.5, ENT600] [3, Brass Band] [2, IPK501] [1.5, ITS610]
4	[10, CSP650] [9, CSC445] [8, ICT662] [8, CSC548] [6, Multimedia Club] [6, Debates] [2, Badminton]	[8, Badminton] [6, CSP650] [3, ICT662] [2, Multimedia Club] [2, Debates] [1, CSC445] [1, CSC548]	[9, CSC445] [8, CSC548] [3, Multimedia Club] [3, Debates] [2.667, ICT662] [1.667, CSP650] [0.25, Badminton]	[9, CSC445] [8, CSC548] [3, Multimedia Club] [3, Debates] [2.667, ICT662] [1.667, CSP650] [0.25, Badminton]
5	[10, ISP610] [7, ACIS Club] [6, CSC662] [6, FSKM Club] [4, DCS651] [3, Reading]	[3, DCS651] [3, ISP610] [3, FSKM Club] [2, Reading] [1, ICT662] [1, CSC662]	[7, ACIS Club] [6, CSC662] [3.333, ISP610] [2, ICT662] [2, FSKM Club] [1.5 Reading]	[7, ACIS Club] [6, CSC662] [3.333, ISP610] [2, ICT662] [2, FSKM Club] [1.5 Reading]

	[2, ICT662]	[1, ACIS Club]	[1.333, DCS651]	[1.333, DCS651]
6	[10, CSP650] [4, Watching movie] [4, Reading] [3, IPK501] [3, ITS610] [3, ENT600] [2, ISEC]	[4, Watching movie] [2, IPK501] [2, ITS610] [2, CSP650] [2, ENT600] [2, Reading] [1, ISEC]	[5, CSP650] [2, ISEC] [2, Reading] [1.5, IPK501] [1.5, ITS610] [1.5, ENT600] [1, Watching movie]	[5, CSP650] [2, ISEC] [2, Reading] [1.5, IPK501] [1.5, ITS610] [1.5, ENT600] [1, Watching movie]
7	[10, CSP600] [7, IPK501] [7, Edit photo] [6, JPNR] [4, ENT600] [4, ITS610] [1, EET669]	[8, ITS610] [6, IPK501] [6, CSP600] [5, ENT600] [3, JPNR] [3, Edit photo] [2, EET669]	[2.333, Edit photo] [2, JPNR] [1.667, CSP600] [1.167, IPK501] [0.8, ENT600] [0.5, ITS610] [0.5, EET669]	[2.333, Edit photo] [2, JPNR] [1.667, CSP600] [1.167, IPK501] [0.8, ENT600] [0.5, ITS610] [0.5, EET669]
8	[6, CSP650] [2, ENT600] [2, ITS610] [2, Studying] [1, EET669] [1, IPK501] [1, ISEC]	[8, Studying] [3, CSP650] [2, ITS610] [2, IPK501] [2, ISEC] [1, EET669] [1, ENT600]	[2, ENT600] [2, CSP650] [1, EET699] [1, ITS610] [0.5, IPK501] [0.5, ISEC] [0.25, Studying]	[2, ENT600] [2, CSP650] [1, EET699] [1, ITS610] [0.5, IPK501] [0.5, ISEC] [0.25, Studying]
9	[10, Cooking] [8, IPK501] [8, CSP650] [7, ENT600] [6, Brass band] [6, Watching movie] [3, ITS610] [3, Ping pong]	[4, Watching movie] [3, Ping pong] [2, IPK501] [2, CSP650] [2, ENT600] [2, ITS610] [2, Brass band] [1, Cooking]	[10, Cooking] [4, IPK501] [4, CSP650] [3.5, ENT600] [3, Brass band] [1.5, ITS610] [1.5, Watching movie] [1, Ping pong]	[10, Cooking] [4, IPK501] [4, CSP650] [3.5, ENT600] [3, Brass band] [1.5, ITS610] [1.5, Watching movie] [1, Ping pong]
10	[9, DCS651] [9, CSC649] [9, CSC580] [8, ICT663] [4, JPK] [3, TKC501] [3, Futsal] [2, CSC253] [2, Netball] [2, Fishing]	[5, DCS651] [3, Netball] [3, Fishing] [2, TCK501] [2, CSC253] [2, ICT663] [2, CSC649] [2, CSC580] [2, JPK] [2, Futsal]	[4.5, CSC580] [4.5, CSC649] [4, ICT663] [2, Futsal] [2, JPK] [1.8, DCS651] [1.5, TKC501] [1, CSC253] [0.667, Netball] [0.667, Fishing]	[4.5, CSC649] [4.5, CSC580] [4, ICT663] [2, JPK] [2, Futsal] [1.8, DCS651 [1.5, TKC501] [1, CSC253] [0.667, Netball] [0.667, Fishing]

TABLE IV. SUMMARY OF RESEARCH ACCURACY RESULT

Test / Result	1	2	3	4	5	6	7	8	9	10
Brute force	5	4	6	7	7	7	7	7	8	10
Timetable Generator	3	4	6	7	7	7	7	7	8	10
Accuracy	0.6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

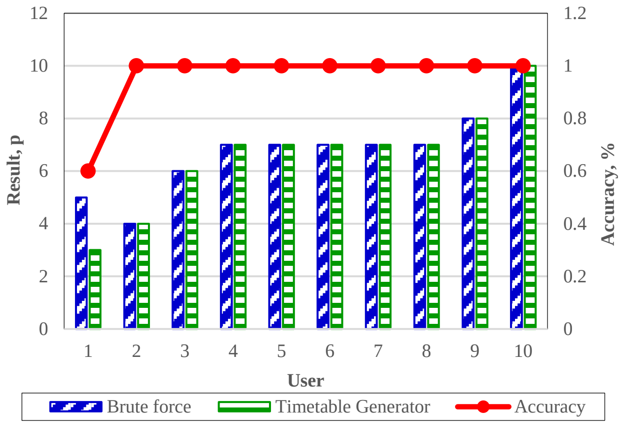


Fig. 5. Phases in Optimizing Greedy Algorithm for Timetable Generator.

V. Conclusion

We presented a recommended semester planner using the optimization technique greedy optimization. Optimization is done to suggest a semester planner according to the user's preferences and at the same time recommend the planner with scheduling all user's activities, including study, hobby and extracurricular for a semester. We managed to validate the Timetable Generator using a reliability test by testing the system's accuracy with the Brute Force algorithm with the achievement of 100% accuracy. Future work is recommended to enhance the system's security. We need to hold the private and confidential data and propose using progressive web applications for a better view and suitable for handphones.

Acknowledgment

The research was sponsored by Universiti Teknologi MARA Cawangan Melaka under the TEJA Grant 2021 (GDT 2021/1-28).

References

[1] S. Singh and K. Kant, "Schools, skills, and learning: The impact of COVID-19 on education," 2020. https://voxeu.org/article/impact-covid-19-education (accessed Oct. 04, 2021).

[2] N. B. Reinke, "The impact of timetable changes on student achievement and learning experiences," Nurse Education Today, vol. 62, pp. 137–142, 2018.

[3] M. Huebener, S. Kuger, and J. Marcus, "Increased instruction hours and the widening gap in student performance," Labour Economics, vol. 47, pp. 15–34, 2017.

[4] L. Foulkes, D. McMillan, and A. Gregory, "A bad night's sleep on campus: an interview to achieve exam success," Sleep Health, vol. 5, no. 3, pp. 280–287, 2019.

[5] S. Panda, M. Mandal, and R. Barman, "Predictors of perceived stress among university students," International Journal of Educational and Management Studies, vol. 5, no. 4, pp. 324–328, 2015.

[6] C. Kridiotis and S. Swart, "A learning development module to support academically unsuccessful 1st-year medical students," African Journal of Health Professions Education, vol. 9, no. 2, pp. 62–66, 2017.

[7] A. M. Hays and S. Sharp, "Supporting postgraduate coursework students through their time of transition," Research Online, vol. 5, no. 2, pp. 40–54, 2018.

[8] X. Brioso, A. Humero, and S. Calampa, "Comparing point-to-point precedence relations and location-based management system in last planner system: A housing project of highly repetitive processes case study," Procedia Engineering, vol. 164, pp. 12–19, 2016.

[9] J. A. Grissom, S. Loeb, and H. Mitani, "Principle time management skills: Explaining patterns in principals' time use, job stress and perceived effectiveness," Journal of Educational Administration, vol. 53, no. 6, pp. 773–793, 2015.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to algorithms. The Knuth-Morris-Pratt Algorithm, 2001.

[11] S. Singh and K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," in International Conference on Emerging Trends in Electrical and Computer Technology, 2011, pp. 933–936.

[12] M. Choudhary and S. K. Peddoju, "A dynamic optimization algorithm for task scheduling in cloud environment," International Journal of Engineering Research and Applications, vol. 2, no. 3, pp. 2564–2568, 2012.

[13] K. Wang, W. Shang, M. Liu, W. Lin, and H. Fu, "A greedy and genetic fusion algorithm for solving course timetable problem," in 17th International Conference on Computer and Information Science (ICIS), 2018, pp. 344–349.

[14] S. Kadry and B. Ghazal, "New proposed design to solve examination timetable problem," in 3rd International Proceedings on Modern Trends in Science, Engineering and Technology, 2015, pp. 36–40.

[15] W. Shaochang, X. Fei, Y. Weixia, and M. Zhe, "Application of greedy random adaptive search algorithm (GRASP) in flight recovery problem," in Second International Conference of Sensor Network and Computer Engineering (ICSNCE 2018), 2018, pp. 78–83.

[16] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multiobjective optimization problems," IEEE Transactions on Evolutionary Computation, vol. 24, no. 2, pp. 380–393, 2019.

[17] V. Kachitvichyanuku, "Comparison of three evolutionary algorithms: GA, PSO, and DE," Industrial Engineering and Management Systems, vol. 11, no. 3, pp. 215–223, 2012.

[18] G. Prabhaharan, B. S. H. Khan, and L. Rakesh, "Implementation of grasp in flow shop scheduling," The International Journal of Advanced Manufacturing Technology, vol. 30, no. 11, pp. 1126–1131, 2006.

[19] C. Cerrone, R. Cerulli, and B. Golden, "Carousel greedy: A generalized greedy algorithm with applications in optimization," Computers & Operations Research, vol. 85, pp. 97–112, 2017.

[20] A. Levitin, Introduction to the design & analysis of algorithm 3rd edition. Pearson, 2012.

[21] M. Mahoor, F. R. Salmasi, and T. A. Najafabadi, "A hierarchical smart street lighting system with brute-force energy optimization," IEEE Sensors Journal, vol. 17, no. 9, pp. 2871–2879, 2017.