ISSN No:-2456-2165

Optimization of Course Timetable For Individual Student Using Genetic Algorithm

Norhasliza bt Muhamad Nor
Department of Information Technology and Communication
Ungku Omar Polytechnic
Perak, Malaysia

Abstract:- Students in institutions of higher education such as polytechnics use class timetable that have been provided by the polytechnic, but different with students who have a problem of failure of the course in the previous semester. Students, who have failed one or more courses, need to build their own timetable for next semester. This is because they need to slots the failed courses in their timetable for next semester. The process of constructing the timetable manually is time consuming because many trials need to be done, can cause errors and is not necessarily timetable that has been generated is the most optimal in terms of the number of credit hours. They need to produce a course timetable with the most optimal credit hours without any problems in the timetable. The generation of individual course timetable manually requires a lot of effort and takes a very long time due to many constraints and the possibilities of making mistakes is very high. The study presented in this paper focuses on optimization using genetic algorithm in solving the course timetabling problem for individual student. A conceptual model is adapted from the process of Genetic Algorithms has been developed and furthermore one program has been developed using this model. The program is tested to ensure that the results produced by this program are free from any errors and this program used to evaluate the proposed adapted conceptual model. The results from this evaluation found that, optimization of course timetable for individual student using genetic algorithm achieved a good result. It can be concluded that genetic algorithm can be used to solve the problem in generated the course timetable for individual student and help student to get the optimize timetable for their individual course timetable.

Keywords:- *Genetic Algorithm; Optimization; Course Timetable; Constraint.*

I. INTRODUCTION

A timetable is a set of meetings in time. A meeting is combination of resources (rooms, people etc), some of which is specified by problem and some of which may be allocated as part of solution [1]. In the institutions of education, timetables are very important element. The timetable is built to allow process of teaching and learning in an organized and planned. The timetable will normally be completed to the use for one semester. Students and lecturers use a timetable to find out the order to attend class every day. There are timetables for class, timetables for lecturer and timetables for room.

Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives [2]. Each subject offered by the institutions has the number of teaching and learning hours that must be followed for that semester. The generation of timetable can ensure that the total number of learning hours of the course is followed. In the institutions of higher education such as polytechnics, class timetable will be provided, students do not need to build their own class timetable, they just need to use timetables prepared for their class. However, for students who have to repeat a course due to the failed of courses in the previous semester or they have course that have not been taken in the previous semester, had to build their timetable manually. This is because of they have to attend classes for that subject and should insert time for that class in their schedule.

This study was conducted to help most of the students who are having problems at the beginning of each semester to construct individual course timetable manually, they cannot use the provided course timetable because of many constraints. This manual process involves a very long time and there is a human error. Students that failed course often lose motivation to continue their studies due to these problems, and the time taken to finish their study take longer because of the timetable they make is not optimal.

II. THE TIMETABLING PROBLEM

In the institutions of higher education, timetable was provided at the beginning of each semester for lecturers and students. They just need to use the timetable provided. Different for students who have to repeat a course due to the failed of course in the previous semester or they have course that have not been taken in the previous semester, they had to build their timetable manually. This is because of they have to attend classes for that course and should insert time for that class in their timetable. This kind of student have some course to take different from other student in their class, which means that the master timetable already exists, but this kind of student cannot follow the existing class timetable and have to construct their own timetable. Yet production of individual course timetable manually requires effort and a very long time. The possibility of making mistakes is very high. This is because the students have to try all possibilities to get the best choice. Students also may be generating a timetable that does not conform to the maximum criteria, when they can actually generate better timetable, with optimization qualifying credit hours taken. In completing individual course timetable manually, students in polytechnic have to consider a few things such as course that needs to be repeated, course that has not been taken, course that are being offered, the classes timing, and maximum credit hours can be taken on that semester. In addition, students must follow the rules like comply with the prerequisite for course, comply with the minimum and maximum number of credit hours and making sure there is no overlap time for a complete timetable.

Timetabling is that the task of constructing a timetable while satisfying some constraints. There are basically two varieties of constraints, soft constraints and hard constraints. Soft constraints are those if we violate them in scheduling, the output continues to be valid, but hard constraints are those which if we violate them; the timetable is not any further valid [3]. The timetabling problem is an important practical problem that is frequently encountered in educational institutions, such as schools and universities. The timetabling problem has received special attention from the scientific community in the last few decades. This is mainly due to the fact that manual generation of timetables is very time consuming and the resulting timetables are usually inefficient and may be costly in terms of money and resources [4]. Many algorithms and solutions have been researched and implemented to address course timetabling such as Hybrid Evolutionary Approach, Generic Algorithm and Local Search, Graph Coloring and Constraint Manipulation, Soft Computing Techniques, Object-Oriented Scheduler and many others [2]. But many of the research do not cover the problem for individual course timetabling problem. This study focuses on optimization of course timetable for individual student in Polytechnic Malaysia. The following section examines previous work using genetic algorithms to solve the university course timetabling problem.

III. GENETIC ALGORITHM AND RELATED WORKS

Genetic algorithms (GAs) were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. [5]. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached [6].

[7], proposed a Genetic algorithm to solve the school timetable problem for a South African primary and high school schedules. The overall process is a two phase approach. The first phase is a GA that focuses on producing feasible timetables, while the second GA phase improves the quality of timetables found during the first phase. The study proved that the proposed algorithm has produced good quality timetables for the school timetabling problem by meeting all hard constraints.

In paper by [8], genetic algorithm is used to solve university course timetabling problem. At first, a model of problem to be solved is defined. Then, the genetic representation is determined and a fitness function is established according to the constraints. Finally, a case of university course timetabling from real-world is discussed and solved. It is demonstrated that the method proposed in this paper is feasible and efficient. In this study, the model of university course timetabling problem is redefined. Then a process of using genetic algorithm to solve this problem is given. In the meantime, a kind of genetic representation is determined to indicate the relationship of teachers, courses, classes, classrooms and time periods, and a fitness function is established according to the constraints. Finally, a case of university course timetabling problem from real-world is analyzed and solved.

In paper by [9], the testing and comparison of FGA and GA were conducted for the problem of scheduling lectures at STMIK XYZ. Based on the results obtained, FGA can produce a schedule without any hard constraint violations. FGA can be used to solve multi-objective problems. FGA has a smaller number of generations than GA. However, overall GA is superior in producing schedules without any problems.

IV. GENETIC ALGORITHM TO OPTIMIZE COURSE TIMETABLE

Genetic algorithm was selected to solve the problem of constructing the optimal timetable for course timetable for the individual student in Polytechnic. Genetic algorithms work with a population consisting of individuals, each individual represents a possible solution. There are several processes in genetic algorithms, modelling is done by using process in genetic algorithm. The main steps in genetic algorithm include initialization, selection, crossover, mutation. This is mainly used to generate useful solutions in optimization and search problems [10].

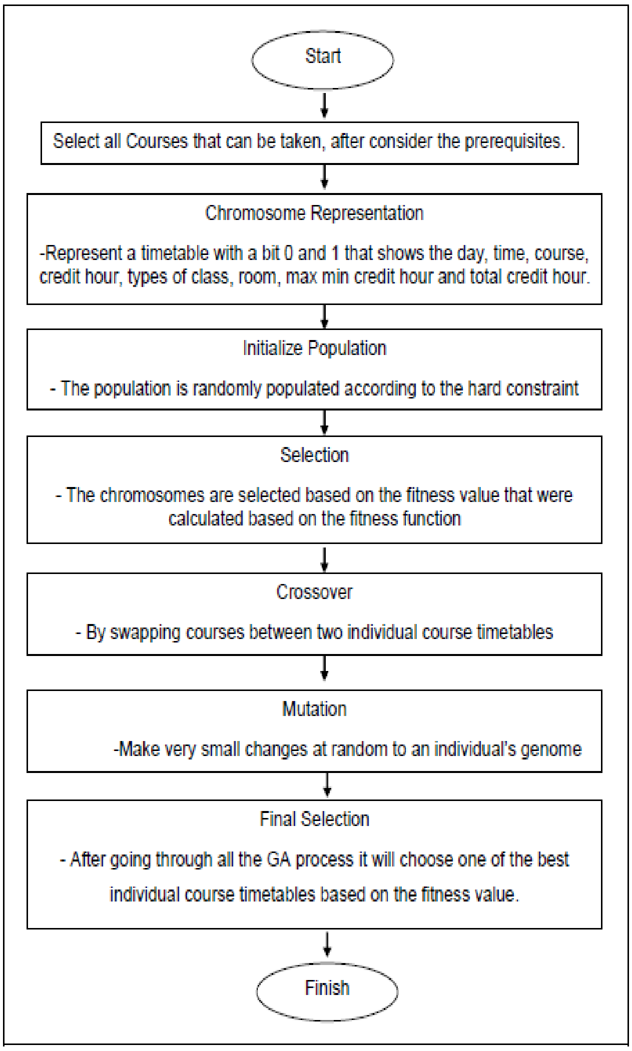

Start
Select all Courses that can be taken, after consider the prerequisites.
Chromosome Representation
-Represent a timetable with a bit 0 and 1 that shows the day, time, course, credit hour, types of class, room, max min credit hour and total credit hour.
Initialize Population
- The population is randomly populated according to the hard constraint
Selection
- The chromosomes are selected based on the fitness value that were calculated based on the fitness function
Crossover
- By swapping courses between two individual course timetables
Mutation
-Make very small changes at random to an individual's genome
Final Selection
- After going through all the GA process it will choose one of the best individual course timetables based on the fitness value.
Finish

Fig. 1. GA Process in Developing Course Timetable for Individual Student

According to the flowchart in Figure 1, the process started with selecting all the courses that can be taken in a current semester, after consider the prerequisites. Then GA starts with generate the Cromosome Representation which represent Courses, Credit Hour, Class, Types, Room, Day, Time, Maximum Minimum Credit Hour and total credit hour into binary alphabet and after that generate Individual Representation that consist of possible solutions to the problem. After that, initial population will be randomly generated according to the hard constraint. In Selection process, chromosomes are selected based on the fitness values that were calculated based on the fitness function. Next, Crossover will take place by swapping courses between two individual course timetables. Then, Mutation makes very small changes at random to an individual’s genome. Final selection will choose one of the best individual course timetables based on the fitness value.

A. *Course Selection for Individual Student*

The first process is the selection of the course that will be taken. Student, who failed a course, should consider a prerequisite in selecting course for the next semester. This is because the Polytechnic system has state that there is a course which is a prerequisite to other course, but there is also a course with no prerequisites. The first step for students who failed the course is providing a list of courses with no prerequisites with the failed course or with the course that has not been taken yet. These courses should be listed along with classes that offer the course in that semester. List of courses listed will be selected to achieve the most optimal timetable.

One problem used as an example, the problems that happen to a student who failed a course FP201 programming fundamentals in 2nd semester of study. In the third semester, this student will construct the individual course timetable. This student must slot the failed course, that is FP201 also other subjects that can slot together. This example is taken to show the process that will happen using the techniques of genetic algorithm in the optimization of course timetable for the individual student. List of subjects that can be taken by this student is FP201 Programming Fundamentals, AF104 Bahasa Mandarin, AE301 Communicative English 2, FP303 Computer Network, FP305 Data Structure, AW101 Occupational Safety and Health, FP304 Database System, AA301 Tamadun Islam, FP501 Open Source Operating System and FP511 Human Computer Interaction.

B. *Chromosome Representative*

The problem consists of the following entities: Course Name, Day and Time, Room, Credit Hour, Class Name, Total Minimum Credit Hour, Total Maximum Credit Hour and Total Credit Hour. According to above entities, course timetable for individual student problem is to optimize the construction of timetable with all desired hard constraints are met and soft constraints are satisfied as far as possible. In this phase, possible solution is encoded as a string of bits. Table I shows an example of chromosome representation for day/ time and Table II shows an example of chromosome representation for course name.

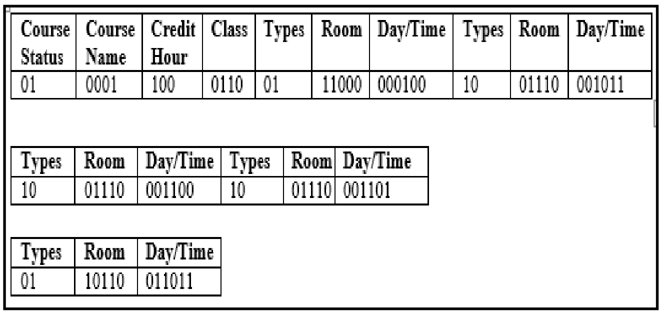
TABLE I. THE CHROMOSOME REPRESENT THE DAY/ TIME

Time / Day	8-9	9-10	10-11	11-12	12-1	1-2	2-3	3-4	4-5	5-6
Monday	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010
Tuesday	001011	001100	001101	001110	001111	010000	010001	010010	010011	010100
Wednesday	010101	010110	010111	011000	011001	011010	011011	011100	011101	011110
Thursday	011111	100000	100001	100010	100011	100100	100101	100110	100111	101000
Friday	101001	101010	101011	101100	101101	101110	101111	110000	110001	110010

TABLE II. THE CHROMOSOME REPRESENT THE COURSE NAME

Course	Bit
FP201- PROGRAMMING FUNDAMENTALS	0001
AF104 – BAHASA MANDARIN	0010
AE301- COMMUNICATIVE ENGLISH 2	0011
FP303 – COMPUTER NETWORK	0100
FP305 – DATA STRUCTURE	0101
AW101 – OCCUPATIONAL SAFETY AND HEALTH	0110
FP304 – DATABASE SYSTEM	0111
AA301 – TAMADUN ISLAM	1000
FP501 – OPEN SOURCE OPERATING SYSTEM	1001
FP511 – HUMAN COMPUTER INTERACTION	1010

Figure 2 shows an example of individual for course FP201 Programming Fundamentals for class DIP2A.



Course Status	Course Name	Credit Hour	Class	Types	Room	Day/Time	Types	Room	Day/Time
01	0001	100	0110	01	11000	000100	10	01110	001011

Types	Room	Day/Time	Types	Room	Day/Time
10	01110	001100	10	01110	001101

Types	Room	Day/Time
01	10110	011011

Fig. 2. Individual for FP201 class DIP2A

C. Initialize Population

Initial Population randomly generated, consist of individuals after taking consideration to hard constraints. For this research the generation of population number has been chosen to be at 100 populations.

D. Selection

Selection (or reproduction) is an operator that makes more copies of better strings in a new population. Selection is usually the first operator applied on population. Selection operator selects good strings in a population and forms a mating pool. This is one of the reasons for the selection operation to be sometimes known as reproduction operator. Thus, in selection operation the process of natural selection cause those individual that encode successful structures to produce copies more frequently. Roulette wheel selection, Rank selection, Steady state selection, Elitism, Boltzmann Selection and Tournament Selection is a method of selection of chromosomes [11].

1) *Elitism:* Elitism is a strategy chosen for this research. Elitism is a kind of selection in which the best individual passed to the next generation as such without any modification. Elitism prevents the best individual to undergo the reproduction process so as to pass them without any modification into next generation [12]. Elitism is used in this study because it is able to copy the best chromosome to the new population, so that the best chromosome will not lost by the crossover and mutation that were implemented. This is important to prevent the loss of the best solution to get the optimum individual course timetable.

E. Fitness

Fitness function is used to evaluate chromosomes whether they are feasible and how good they are [13]. In this research, each population will be considered how many constraints it has violated. Specific value assigned to the hard constraints and soft constraints as the penalty, a violation of a hard constraint has more penalty value than a soft constraint. The sum of all penalty values calculated for each chromosome, this value is called fitness value. Chromosome with less value of fitness is better than chromosome with higher value of fitness. Table III shows the penalty for each constraint.

TABLE III. PENALTY VALUE FOR CONSTRAINTS

	Constraints	Penalty
Hc1:	Must slot failed course	500
Hc2:	A student can't be assigned to more than one course during any time slot	500
Hc3:	Student must achieve minimum credit hours allowed	500
Hc4:	Student cannot take more than maximum number of credit hours allowed	500
Sc1:	There is a gap between the next subjects.	10
Sc2:	Selection of good arrangement of timetable that do not cause students rushing form one place to another place that far away from the previous class for the next class.	10

A value for Hard Constraints, Hc1 = 500, Hc2 = 500, Hc3 = 500 and Hc4 = 500. A value for Soft Constraints, Sc1 = 10 and Sc2 = 10.

$$Fitness = \sum_{i=1}^{n} P\ (i)$$

$$P\ (i) = \sum_{c=1}^{m} P(c)$$

P (i) is the penalty for constraint i, P (c) represent the penalty of the corresponding violation, n is the number of hard and soft constraints, while m is the number of violations.

Fitness = (Hc1 x m1) + (Hc2 x m2) + (Hc3 x m3) + (Hc4 x m4) + (Sc1 x m5) + (Sc1 x m6)

F. Crossover

A crossover operator is used to recombine two strings to get a better string. The next step after selection is crossover which generates a second generation population of solutions from those selected through selection. Two Point Crossover used in this study. Crossover occurs between individuals in different populations. When crossover occurs, a new population is produced. Population with the best solution will be used. Figure 3 shows the two point crossover process while figure 4 shows the new population after the two point crossover process.

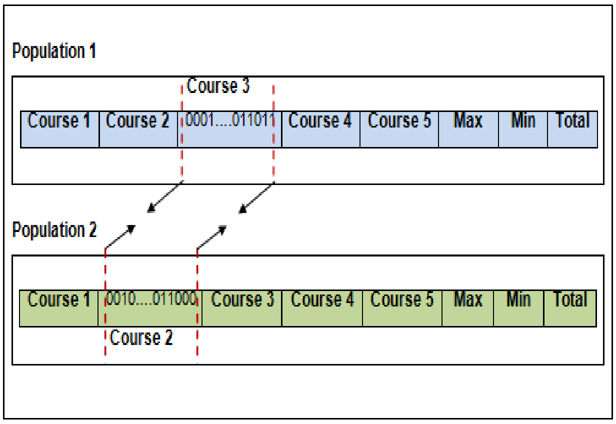

Population 1
Course 3
Course 1 | Course 2 | 0001....011011 | Course 4 | Course 5 | Max | Min | Total
Population 2
Course 1 | 0010....011000 | Course 3 | Course 4 | Course 5 | Max | Min | Total
Course 2

Fig. 3. Two Point Crossover between Courses

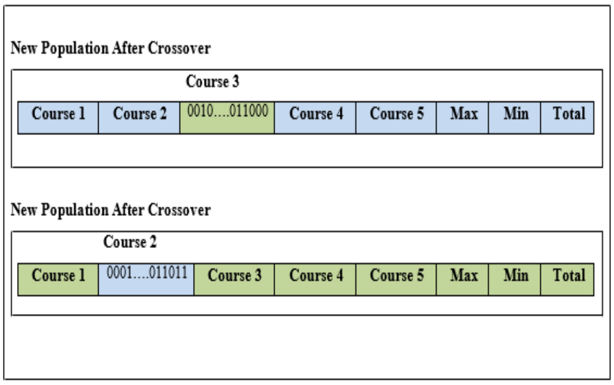
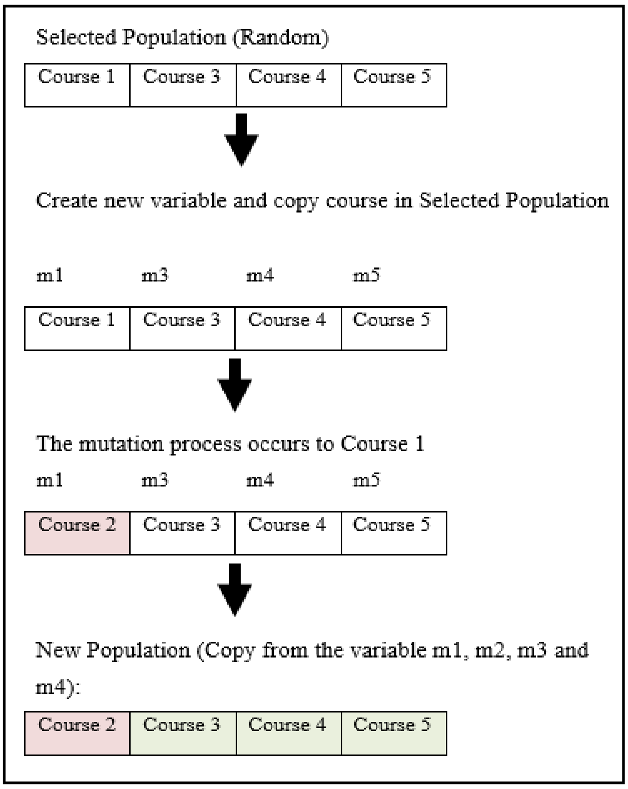

New Population After Crossover
Course 3
Course 1 | Course 2 | 0010....011000 | Course 4 | Course 5 | Max | Min | Total
New Population After Crossover
Course 2
Course 1 | 0001....011011 | Course 3 | Course 4 | Course 5 | Max | Min | Total

Fig. 4. New Population after Crossover

G. Mutation

Mutation operation is very simple. It just takes randomly a number of rows in the chromosome matrix and moves it to another randomly chosen slot (change the day and/or the time) or location. In every operation, the domain of different variables is respected. Note that the tool works with variable values as indexes to the real values. These real values for examples courses' names, lecturer' names, etc can be bound to these variables. The relationship between the different variables should be set at the beginning of the process [14].


Selected Population (Random)

Course 1 | Course 3 | Course 4 | Course 5

Create new variable and copy course in Selected Population

m1 m3 m4 m5

Course 1 | Course 3 | Course 4 | Course 5

The mutation process occurs to Course 1

m1 m3 m4 m5

Course 2 | Course 3 | Course 4 | Course 5

New Population (Copy from the variable m1, m2, m3 and m4):

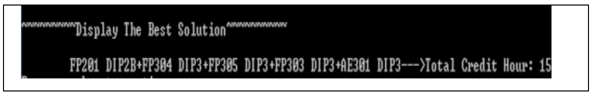
Course 2 | Course 3 | Course 4 | Course 5

Fig. 5. Mutation process

Figure 5 shows the example of mutation process and the result from the mutation.

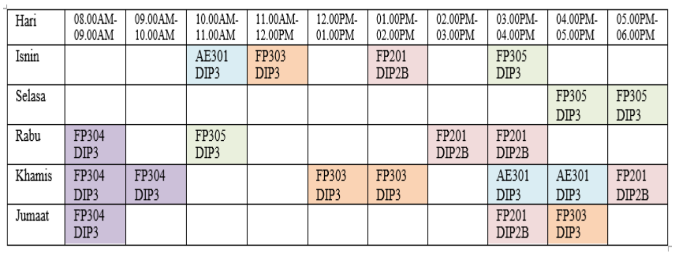
H. Final Selection

The last process is the final selection. Final selection will be done again after the completion of the process of mutation. This should be done because there is a change to the population during the process of crossover and mutation. Each chromosome will be stored in a master file with its fitness value, either after the crossover or mutation. This is necessary because the final selection will be made at the end of the process to select the best solution to the problem. The selection is done using a fitness value as the previous selection process, to select the best population, in order to get the best solution. Figure 6 shows an example of the output generated by the final selection from one of the student scenarios used in the study.


```
^^^^^^^^^^Display The Best Solution^^^^^^^^^^^

     FP201 DIP2B+FP304 DIP3+FP305 DIP3+FP303 DIP3+AE301 DIP3--->Total Credit Hour: 15
```

Fig. 6. Output generated from the final selection



Hari	08.00AM-09.00AM	09.00AM-10.00AM	10.00AM-11.00AM	11.00AM-12.00PM	12.00PM-01.00PM	01.00PM-02.00PM	02.00PM-03.00PM	03.00PM-04.00PM	04.00PM-05.00PM	05.00PM-06.00PM
Isnin			AE301 DIP3	FP303 DIP3		FP201 DIP2B		FP305 DIP3		
Selasa									FP305 DIP3	FP305 DIP3
Rabu	FP304 DIP3		FP305 DIP3				FP201 DIP2B	FP201 DIP2B		
Khamis	FP304 DIP3	FP304 DIP3			FP303 DIP3	FP303 DIP3		AE301 DIP3	AE301 DIP3	FP201 DIP2B
Jumaat	FP304 DIP3							FP201 DIP2B	FP303 DIP3	

Fig. 7. Timetable drawn based on output from the final selection

Figure 7 shows the timetable drawn based on output from the final selection using genetic algorithm.

Total credit hours = 4 (FP201) + 3 (FP304) + 3 (FP305) + 3 (FP303) + 2 (AE301) = 15

Using the timetable in Figure 7, four items have been tested. The result found:

i. The failed course has been listed
ii. No repetition of the same course
iii. No clash between the courses listed
iv. Total number of credit hours is correct

V. CONCLUSION

Optimization of course timetable for individual student using genetic algorithm in this study has been validated in the test and achieved good results. This can conclude that genetic algorithm can be used to solve the problem in generating the course timetable for individual student and help student to get the optimize timetable for their individual course timetable.

It is very difficult to take soft constrains as consideration to optimize the course timetable for individual student. Example, to get the timetable that has a gap between courses. Because of that issue, soft constrain is not used to calculate the penalty in this study. Further research can be made to take soft constrains as consideration in order to optimize the course timetable for individual student.

REFERENCES

[1]. A. Jain, S. Jain, and P. K. Chande, "Formulation of Genetic Algorithm to Generate Good Quality Course Timetable," vol. 1, no. 3, pp. 248–251, 2010.

[2]. S. Nem, R. Kasbon, F. O. I. Mean, and S. Mahamad, "Combination-based Solution for Timetable Organizer 2 Literature Review 3 System Design," pp. 24–28, 2013.

[3]. D. Mittal, H. Doshi, M. Sunasra, and R. Nagpure, "Automatic Timetable Generation using Genetic Algorithm," *Ijarcce*, vol. IX, no. Vi, pp. 245–248, 2015, doi: 10.17148/ijarcce.2015.4254.

[4]. M. Hosny and S. Fatima, "A Survey of Genetic Algorithms for the University Timetabling Problem," vol. 13, pp. 34–39, 2011.

[5]. M. Mitchell, *An introduction to genetic algorithms.* MIT Press, 1998.

[6]. S. Naseem and J. Shengxiang, "A Guided Search Genetic Algorithm for the University Course Timetabling Problem," no. August, pp. 10–12, 2009.

[7]. R. Raghavjee and N. Pillay, "Using Genetic Algorithms to Solve the South African School Timetabling Problem Rushil Raghavjee Nelishia Pillay," pp. 286–292, 2010.

[8]. X. Deng, Y. Zhang, B. Kang, J. Wu, X. Sun, and Y. Deng, "An application of genetic algorithm for university course timetabling problem," *2011 Chinese Control Decis. Conf.*, pp. 2119–2122, May 2011, doi: 10.1109/CCDC.2011.5968555.

[9]. M. E. Al Rivan and B. Bhagaskara, "Perbandingan Fluid Genetic Algorithm dan Genetic Algorithm untuk Penjadwalan Perkuliahan," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, p. 350, 2020, doi: 10.32736/sisfokom.v9i3.879.

[10]. A. Jose and P. Chandrasekar, "A Study on Optimization Algorithms for Clustering Gene Expression Data," vol. 2, no. 1, pp. 1263–1270, 2014.

[11]. S. Gandhi, D. Khan, and V. S. Solanki, "A Comparative Analysis of Selection Scheme," no. 4, pp. 131–134, 2012.

[12]. A. Sharma, "International Journal of Advanced Research in Computer Science and Software Engineering Review Paper of Various Selection Methods in Genetic Algorithm Types of Selection Method," vol. 3, no. 7, pp. 1476–1479, 2013.

[13]. W. Chinnasri, S. Krootjohn, and N. Sureerattanan, "Performance Comparison of Genetic Algorithm ' s Crossover Operators on University Course Timetabling Problem," 2012.

[14]. C. Chiam, "FPTT Timetable Scheduling System Using Genetic Algorithm," 2011. Accessed: Jun. 13, 2014. [Online]. Available: http://digitalcollection.utem.edu.my/id/eprint/6302.