INTI JOURNAL | eISSN:2600-7320
Vol.2019:69

Automated Student Timetable Scheduling System based on Genetic Algorithm

Amirul Azuani Romle[1], Noor Ain Rosly[2] Meng Chuan Haw[3]

Faculty of Information Technology, INTI International University, Nilai, Negeri Sembilan, Malaysia.

***Email:** aazuani.romle@newinti.edu.my, noorain.rosly@newinti.edu.my, i16009811@student.newinti.edu.my

Abstract

Timetable scheduling is always a major challenge in an education setting due to the complexity of the timetabling environment including large number of students, changing in study plan, diverse courses and limited number of classrooms. These factors lead to several major problems identified during the enrollment period such as timetable clashing between students, continuous hours of lecture and difficulty to find a matching slot for the clashed courses. In order to solve these problems, automated timetable scheduling system based on genetic algorithm is proposed which is estimated to reduce the chances of class clashing and prevent continuous lecture time. Genetic algorithm works based on natural evolutions that comprises of several iterations. The iterations will continue to generates new generation cycles until the optimum schedule is met. By using minimal data entry such as the course information, the system will be able to find the best timeslot. As a result, student and lecturer will have some gap between their lecturer hour while the classroom and laboratory can be assigned more effectively as well as classroom clashing which continuously happened in the manual scheduling system can be avoided. Hence, genetic algorithm will be able to find a better solution for timetable scheduling as well as optimizing the efficiency of the timetabling unit in INTI International University.

Keywords

Scheduling system, timetabling system, genetic algorithm, artificial intelligence

Introduction

In every education institution, timetable scheduling process is highly important as it represent organization daily routines work schedules. This process involves main stakeholders in education institution such as lecturers, students and faculty administrator. Optimum scheduling leads to maximum efficiency and accuracy of organization's routines activities [1]. Scheduling can be a very tough process especially in large organization as the process is subject to various factors [2]. The leading factors influence the scheduling process are identified as number of courses, available

International Conference on Innovation and Technopreneurship 2019
Submission: 24 May 2019; Acceptance: 29 July 2019

lecturers, available classroom, and student's individual study plan. All of the factors mentioned above except for student's individual study plan are easier to handle as they can be design in advance, prior to the semester begins. Unlike those factors, student's individual study plan is hard to propose earlier as it will solely depend on the student performance for each semester. In many cases, failure in any particular course in the current or previous semester will hinder the progression in the initial study plan as the student will not be able to progress with the planned course in the current semester. Therefore, faculty will usually need to offer the previous course on top of the current course. This will impact on the available resources which could lead to tougher scheduling process.

Some education institutions are also suffering from scarce resources of lecturers and infrastructure which would add up the complexity of the scheduling process [3]. In education institution, generating timetable is the most common and essential scheduling process that assign the course with venue in a specific timeslot. Currently, timetable scheduling system that is used in Faculty of IT, in INTI is still using manual scheduling. This manual scheduling requires administrator to assign every single course into the timeslot and do necessary adjustment based on the occurring conflicts. The manual scheduling is often tedious, time consuming and may involve more man power [4]. Therefore, a solution which is based on automated scheduling system using genetic algorithm is proposed to automatically generate the best solution for timetabling scheduling system which estimated to increase the efficiency and improve the performance of the faculty services towards the stakeholders.

Genetic algorithm is established on a heuristic search from the theory of natural evolution [5]. In natural evolution, reproduction took place when two individuals produce an offspring. The best genes from two individuals combines together to produce a better offspring [6]. The reproduction of offspring also is influence by mutation. A mutation happens when individual's gene is altered permanently. Mutation could lead to better offspring, harmful or no changes to the new offspring [7]. This concept of natural evolution is applied in the genetic algorithm which will be use to generate a better solution for timetabling scheduling system in INTI.

Methods

Genetic algorithm used for the proposed automated timetabling scheduling in Faculty of IT, INTI International University was initially design by conducting interview with the administrator of Timetable Unit in INTI. The purpose of this interview is to understand the flow of the system including the system requirements as well as the system constraints. This phase is very crucial as it will determine the functionality and feasibility of the system towards the faculty [8]. Eleven questions related to the current timetabling process is asked face to face.

Apart from interview with the administrator, we also conducted a questionnaire survey to both lecturers and students in the Faculty of IT. The purpose of the survey is to learn about the issue face by the end user with the current timetabling system as well as their requirements towards the proposed system. For simplicity and practically, we used Google Form to distribute the survey to the respondents. It was recorded that a total of 55 respondents with 35 of them are students and remaining 20 respondents are lecturers.

A. Chromosome representation.

Based on the understanding gained from the interview session, we design the genetic algorithm solution which is the chromosome. Chromosome comprises of binary digit patterns which can be assigned to each modules. Then, the binary digits of every module is summed up to build a complete chromosome.

B. Creating population

The initial population is randomly generated based on the number of modules as well as restrictions or constraints in the real timetabling scheduling system. The number of the population can also be decided by the researcher.

C. Fitness Evaluation Function

As population is randomly generated, a score need to be assigned to every individual chromosome in the population. This is to evaluate each chromosome on how they are close to the required constraints and solutions. The score is calculated by using the evaluation function. In many situation, genetic algorithm for scheduling system will use the inverse of the individuals with conflicts.

D. Selection

Selection phase is to select the fittest individuals based on the given score held by each individual. The lesser the score indicates the fittest individuals that will be chosen for reproduction.

E. Crossover and Mutation

In this process either crossover or mutation can take place. At some condition both can also be used together. For crossover, a crossover point is randomly chosen from the genes. A new offspring is created by changing the genes of parents (which refers to a pair of chromosome). The new offspring that results from the crossover is added to the population.

The purpose of mutation is to ensure the diversity as well to avoid pre-mature convergence in the population. Mutation is done by alternating the binary digits of the genes.

The above process is recursively continuing until the algorithm find a converged population where there is no offspring is produce are differ from the previous generation.

Results and Discussions

By using the prototyping system to test the efficiency of automated scheduling using genetic algorithm the data that used includes:

- 3 Classroom
- 4 Timeslot [Mon 9-10, Mon 10-11, Tue 9-10, Tue 10-11]
- 4 Lecturer
- 7 Subject [Subject including the correspond lecturer]
- 3 Department

First Generation of Timetable (Contain 2 conflicts)

Class #	Dept	Course (number,max of students)	Room (Capacity)	Instructor(Id)	Meeting Time (Id)
01	Level 1	PRG2201 (C1,25)	R1 (25)	Dr James Wom (I1)	Tue 10:00 - 11:00 (MT4)
02	Level 1	PRG2202 (C2,35)	R2 (45)	Dr Steven (I3)	Tue 09:00 - 10:00 (MT3)
03	Level 2	SDT2201 (C3,25)	R1 (25)	Mr.TChalla (I2)	Mon 09:00 - 10:00 (MT1)
04	Level 2	SDT4201 (C4,30)	R3 (35)	Mrs Marvell (I4)	Mon 10:00 - 11:00 (MT2)
05	Level 2	INT2201 (C5,35)	R3 (35)	Mrs Marvell (I4)	Tue 10:00 - 11:00 (MT4)
06	Level 3	IBM4201 (C6,45)	R3 (35)	Dr Steven (I3)	Mon 09:00 - 10:00 (MT1)
07	Level 3	IBM2402 (C7,45)	R2 (45)	Mr.TChalla (I2)	Mon 09:00 - 10:00 (MT1)

Figure 1.1

The conflict occurs over the class 03 and 07. Lecturer (Mr. TChalla) having two classes at the same timeslot (Mon 9 to 10). Second conflict occurs over the class 06, the course maximum capacity (C6 with 45) is more than the classroom capacity (R3 with 35).

Fifth Generation of Timetable (Contain 1 conflict)

Class #	Dept	Course (number,max of students)	Room (Capacity)	Instructor(Id)	Meeting Time (Id)
01	Level 1	PRG2201 (C1,25)	R1 (25)	Mr.TChalla (I2)	Tue 09:00 - 10:00 (MT3)
02	Level 1	PRG2202 (C2,35)	R2 (45)	Dr Steven (I3)	Tue 09:00 - 10:00 (MT3)
03	Level 2	SDT2201 (C3,25)	R1 (25)	Dr James Wom (I1)	Mon 10:00 - 11:00 (MT2)
04	Level 2	SDT4201 (C4,30)	R3 (35)	Mrs Marvell (I4)	Mon 10:00 - 11:00 (MT2)
05	Level 2	INT2201 (C5,35)	R3 (35)	Mrs Marvell (I4)	Tue 10:00 - 11:00 (MT4)
06	Level 3	IBM4201 (C6,45)	R2 (45)	Dr James Wom (I1)	Mon 10:00 - 11:00 (MT2)
07	Level 3	IBM2402 (C7,45)	R2 (45)	Mr.TChalla (I2)	Mon 09:00 - 10:00 (MT1)

Figure 1.2

The conflict reduces but there are also left one conflict in the class 03 and 06. Lecturer (Dr James Wom) having two classes at the same timeslot.

Last Generation (No conflict)

Class #	Dept	Course (number,max of students)	Room (Capacity)	Instructor(Id)	Meeting Time (Id)
01	Level 1	PRG2201 (C1,25)	R1 (25)	Mr.TChalla (I2)	Tue 09:00 - 10:00 (MT3)
02	Level 1	PRG2202 (C2,35)	R2 (45)	Dr Steven (I3)	Tue 09:00 - 10:00 (MT3)
03	Level 2	SDT2201 (C3,25)	R1 (25)	Mr.TChalla (I2)	Mon 09:00 - 10:00 (MT1)
04	Level 2	SDT4201 (C4,30)	R3 (35)	Mrs Marvell (I4)	Mon 10:00 - 11:00 (MT2)
05	Level 2	INT2201 (C5,35)	R3 (35)	Mrs Marvell (I4)	Tue 10:00 - 11:00 (MT4)
06	Level 3	IBM4201 (C6,45)	R2 (45)	Dr James Wom (I1)	Mon 10:00 - 11:00 (MT2)
07	Level 3	IBM2402 (C7,45)	R2 (45)	Mrs Marvell (I4)	Mon 09:00 - 10:00 (MT1)

BUILD SUCCESSFUL (total time: 0 seconds)

Figure 1.3

After several iteration of new generation, the conflict totally solved, and the timetable is completed without any conflicts.

The estimate result for a human to do this scheduling takes around 15 minutes to complete while automated scheduling process is done in less than 1 second after the system implemented (Figure 1.3). From the above result, we can see that all the lecturer's scheduled will not have more than one class in the same timeslot.

As seen from the results given above, Figure 1.1 –Figure 1.3 it is observe that genetic algorithm enhances the scheduling on every iteration of new generation produced. The algorithm will continue the iteration until the best generation achieved. Thus genetic algorithm will try to find the fittest solution for the timetable.

Conclusions

The prototype system proofed that the automated timetable scheduling is much more efficient than the manual scheduling and able to avoid the clashing issue that is very common in manual timetabling system. It is known that, when the number of resources increased to a larger number, human will face difficulty to do the scheduling manually while in the other hand to satisfy all the constraints. The automated timetable scheduling system using genetic algorithm saves a lot of time without compromising the system constraints. Hence, it is concluded that genetic algorithm would be able to offer a better solution for timetabling scheduling activities in Faculty of Information Technology, INTI International University.

Acknowledgment

The work published in this paper is a part of Final Year Project, Faculty of Information Technology, INTI International University.

References

Pillay, N., & Banzhaf, W. (2010). An informed genetic algorithm for the examination timetabling problem. Applied Soft Computing, 10(2), 457-467. doi:10.1016/j.asoc.2009.08.011

Almeida, M. W., Medeiros, J. P., & Oliveira, P. R. (2015). Solving the Academic Timetable Problem Thinking on Student Needs. 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). doi:10.1109/icmla.2015.184

Matias, J. B., Fajardo, A. C., & Medina, R. P. (2018). A Hybrid Genetic Algorithm for Course Scheduling and Teaching Workload Management. 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM). doi:10.1109/hnicem.2018.8666332

Abouelhamayed, A. F., Mahmoud, A. S., Shaaban, T. T., Salama, C., & Yousef, A. H. (2016). An enhanced genetic algorithm-based timetabling system with incremental changes. 2016 11th International Conference on Computer Engineering & Systems (ICCES). doi:10.1109/icces.2016.7821985

Shatnawi, Safwan & Albalooshi, Fawzi & Rababa'h, Khaleel. (2012). Generating Timetable and Students schedule based on data mining techniques. International Journal of Engineering Research and Applications. 2. 1638-1644.

Diveev, A., & Bobr, O. (2017). Variational Genetic Algorithm for NP-hard Scheduling Problem Solution. Procedia Computer Science, 103, 52-58. doi:10.1016/j.procs.2017.01.010

Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. Computers & Industrial Engineering, 86, 43-59. doi:10.1016/j.cie.2014.11.010

Abouelhamayed, A. F., Mahmoud, A. S., Shaaban, T. T., Salama, C., & Yousef, A. H. (2016). An enhanced genetic algorithm-based timetabling system with incremental changes. 2016 11th International Conference on Computer Engineering & Systems (ICCES). doi:10.1109/icces.2016.7821985