Journal of Physics: Conference Series

A Heuristics Approach for Classroom Scheduling Using Genetic Algorithm Technique

To cite this article: Izah R. Ahmad *et al* 2018 *J. Phys.: Conf. Ser.* **995** 012050

A Heuristics Approach for Classroom Scheduling Using Genetic Algorithm Technique

Izah R. Ahmad, Suliadi Sufahani*, Maselan Ali and Siti N.A.M. Razali

Department of Mathematics and Statistics, Faculty of Applied Science Technology, UniversitiTun Hussein Onn Malaysia, 86400 Parit Raja, Johor, Malaysia

E-mail: izah.fida@gmail.com, suliadi@uthm.edu.my, maselan@uthm.edu.my, asyikinr@uthm.edu.my

Abstract. Reshuffling and arranging classroom based on the capacity of the audience, complete facilities, lecturing time and many more may lead to a complexity of classroom scheduling. While trying to enhance the productivity in classroom planning, this paper proposes a heuristic approach for timetabling optimization. A new algorithm was produced to take care of the timetabling problem in a university. The proposed of heuristics approach will prompt a superior utilization of the accessible classroom space for a given time table of courses at the university. Genetic Algorithm through Java programming languages were used in this study and aims at reducing the conflicts and optimizes the fitness. The algorithm considered the quantity of students in each class, class time, class size, time accessibility in each class and lecturer who in charge of the classes.

1. Introduction

Scheduling of timetable is an important thing that should be done in any academic institution or even un-academic institution also. This to make sure syllabus of the lesson can be completed on time or the scheduling project or activity done in the given time [11]. According to dictionary, scheduling can be defined as a plan of procedure, usually written, for a proposed objective, especially with reference to the sequence of and time allotted for each item or operation necessary to its completion. According to Bethel et al. in 1956 [1], scheduling is the phase of production control which rates the works in order of its priority and then provides for its release to the plant at the proper time and in the correct sequence. Based on Jha (2014) [2], the meeting of people will be the events and the timetable must specify the people need to be met, the location and also the time they need to have the meeting. The most and famous scheduling timetable is educational timetable. While, according to Lansburgh and Spriegel (1995) [3], scheduling is about completion of any work and also about the time each element of the work will start or the order of work. In 2007, based on Burke *et al.* [4], it can happen that probably, educational timetabling is the most widely studied. Burke also said, operational research and artificial intelligence research communities has an interest in the field of timetabling for almost more than 40 years. There are many types of scheduling problem such as job-shop scheduling (Kacem, 2003) [5], transportation scheduling (Sauer & Appelrath, 2000) [6], staffing scheduling (Labidi *et al.*, 2014)

[7], sport scheduling (Ribeiro, 2012) [8], home health care scheduling (Yuan & Fugenschuh, 2003) [9] and operation theatre scheduling (Sufahani & Ismail) [10].

For this paper, a general problem of timetable problem in an academic institution will be solved. To schedule the timetable many constraints should be considered such as the size of class with the capacity of students, time for each slots and class, number of class in a day, subject involved and the lecturer as well. The most suitable approach to solve problem is a genetic algorithm (GA). Genetic algorithm is such a popular algorithm technique used to solve the scheduling problem. This genetic algorithm, was first invented from University of Michigan which is Prof. John Holland in 1975. Java programming is used to solve the problem.

2. Timetable Problem Description

There are many timetable problem in this world such as transport, nurse-rostering, sport, job shop and also timetable of student. To make a best timetable, all the conflicts arise should be well managed. The example of conflicts are the rooms, the lecturer or instructors, the size of room or capacity of the rooms, the subject, the class size, and sometimes the number of subject in a day also can be one of the conflict. Some students consider only 4 class in a day while other consider 5. Usually the conflicts can be divided into two which are hard constraints and soft constraints.

2.1 Problem definition

Following are the participant in the timetable scheduling problem:

I is an instructor or lecturer $\boldsymbol{i_{1,} i_2, i_3, \ldots, i_8}$ and

MT is a Meeting Time $\boldsymbol{mt_1, mt_2, mt_3, \ldots, mt_8}$ and

C is a course $\boldsymbol{c_1, c_2, c_3, \ldots, c_{11}}$ and

R is a room $\boldsymbol{r_1, r_2, r_3, \ldots, r_5}$ and

D is a department of ***dept1*** and ***dept2.***

I is the lecturer involve in the timetable. The meeting time and the day of the class has been set in the program. The course involved also until 11 course and the available room is only 5 that has been set the capacity of each room. While the department is only 2.

2.2 Constraints Involved

In the timetable scheduling, there are a constraints should be followed which are:

- No two subjects in one class at the same time.
- Instructor having no more than one class in the same time
- No room should be double book
- All the allocated rooms must can hold the number of student

To get the feasible solution, the constraints should be followed. This paper should satisfy as many soft constraints as possible so that, the good quality of timetable can be obtained. The soft constraints generally is just to get the best timetable that can make all the students and also instructors satisfy and happiest.

This paper has constructed a lecturer timetable by using the genetic algorithm techniques. A natural chromosomes representation was chosen and genetic operators was also be build. A chromosomes is made up of groups as genes.

3. Genetic Algorithm

Genetic algorithm is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solution. The best point in the population approaches an optimal solution. A population is maintained and the fittest timetables are selected to form the basis of the next generation or iteration. In GA there are three basic operator which are selection, crossover and mutation. This three operator is applied to get the best results. The initialization of a population, the evaluation, and the genetic operator were implemented and controlled by using programming. The GA will assign course in what class, at what time and the instructor involved. Initial population is generated randomly. In the figure 1 below shows the genetic algorithm cycles.

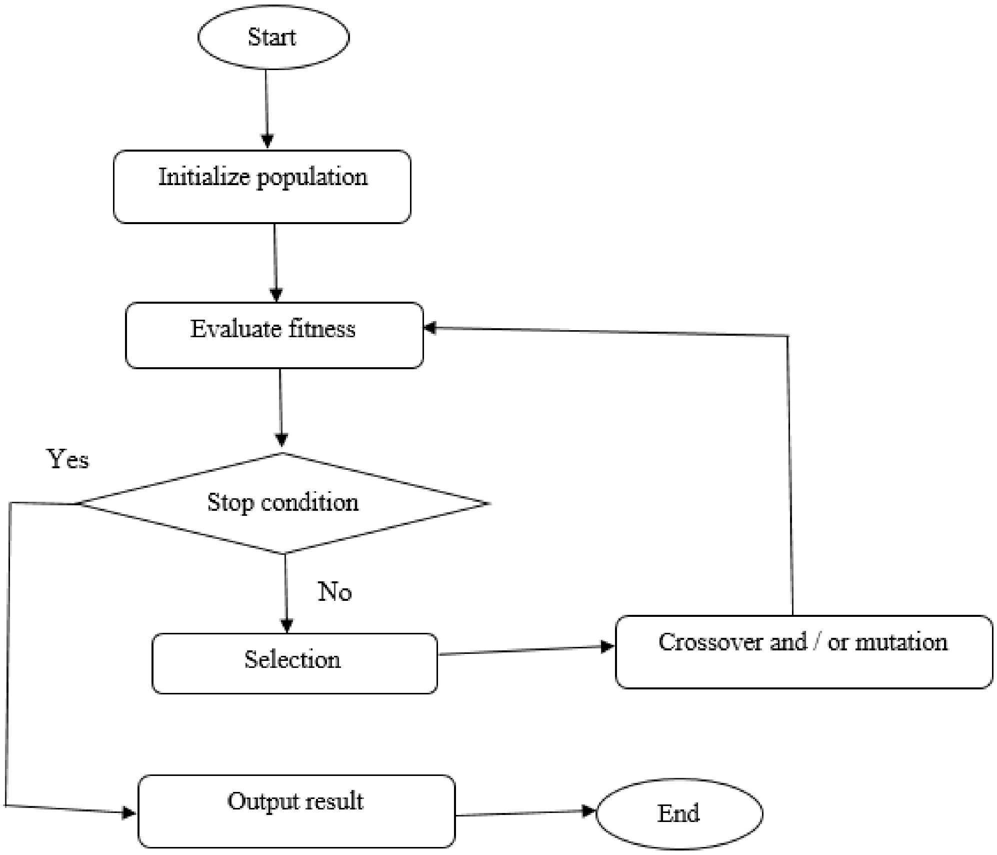

Figure 1. The cycle of GA.

3.1 Chromosome Representation

The chromosome is usually representations as a bit string. All the possible information should be contains in the chromosome such as;

$$f : I \times MT \times C \times R \times D \rightarrow \{0,1\} \tag{1}$$

Where *f (i, mt, c, r, dept) = 1* if and only if course c has I has an instructor in the given class and time. A gene in this representation may also be considered as an element with 5-dimensional matrix, with an allele value of 0 if false and 1 if true.

3.2 Initialize Population

A population of a solution is initialize randomly. For a timetable to be produced, the core course should be considered first. The course is selected in random order, and each course is assigned to a randomly chosen meeting time with the number of students in the course with the available room capacity without violated the constraints.

3.3 Evaluate Fitness

Fitness function is an objective function of problem. Fitness will give the value that then will specify the solution is the best solution or not. This evaluation of fitness also will make sure the looping process in genetic algorithm to be stopped or not. In others word, fitness will control and maintain the process flow of genetic algorithm. The option of the next generation will be control and maintain and will not go further from it. The fitness function for this timetable problem is the inverse of the number of students with class conflicts. This give a meaning, the lesser the number of students with class conflicts, the more fit the class is. The fitness function can be;

$$f = \frac{1}{1+x} \tag{2}$$

Where, x is a submission of conflicts or constraints.

$$x = I + MT + C + R + D \tag{3}$$

I is an instructor, the instructor will assign value of 0 in number if there is only *i* instructor at meeting time *mt* in class course *c* in room *r* from department *dept*. The value will be equal to 1, if there is clash of any meeting time *mt*, class course *c*, room *r*, department *dept* or even with other lecturer. This situation is same with other 4 criteria which are meeting time, class course, room and department. There should have no clash and the value is 0, if there are clash, value of 1 will be assign, show there is a conflict. Therefore, for the value of *x*, it can be equal to 0 if there is no conflict happen, while it will assign a value of 1,2,3,4, or 5 if there is a conflict happen, depend on the how many conflict arise.

3.4 Selection

Selection is one of the Genetic Algorithm operators and it's usually will be applied first. According to Jha (2014) [2], reproduction or selection usually will be the first operator being applied on a population. After the fitness value is calculated, the most fitter will be selected.

3.5 Crossover and Mutation

Crossover is a phase where there is a recombination of two string to get a better string. The crossover process being done to vary a chromosome from one generation to the next generation. Such an example of crossover is;

Parent 1: 1 1 1 0 0

Parent 2: 0 1 0 1 1

Choose a crossover point

Parent 1: 1 1 | 1 0 0

After crossover

Offspring 1: 1 1 0 1 1

Offspring 2: 0 1 1 0 0

At this point, the fitness value will be calculated, and if it's not fit enough, the mutation process will go through. Mutation is a process of altering one or more gene values in chromosomes from its initial state. This phase will be used to maintain genetic diversity from one generation. Example of mutation process is:

Before: 1 1 0 0 1

After: 1 1 0 1 1

After the mutation process, the fitness value will be calculated, if its fitter, then it will stop and get the best solution, but if it's not fitter, the process will begin from evaluate to selection, crossover and mutation until it satisfy stopping criteria or objective function achieved.

3.6 GA Implementation

The timetabling is perform in Java programming for this paper of GA. From this paper, the timetable produced is based on the best fitness value which has a low in conflicts.

4. Computational Results

The result of timetable can be represented as a class, dept, course, room, instructor, and meeting time. In the table 1, its shows the fitness value and conflicts of the program after some iteration. Its gives a fitness of 1.0000 and 0 conflicts. Therefore, the iteration stop as it's satisfy objective function. While in table 2, showing the timetable with more proper arrangement and more easy to be seen with its separation. The separation is made into course with number and maximum of students, room with capacity, the instructor's name and the meeting time as well.

While in table 2, showing the timetable with more proper arrangement and more easy to be seen with its separation. The separation is made into class, department course with number and maximum of students, room with capacity, the instructor's name with the Id and the meeting time as well which has been set the day and time of meeting.

Table 1. Solution showing fitness value and conflicts.

Schedule	Classes [dept,class,room,instructor,meeting-time]	Fitness	Conflicts
0	[STAT,C1,R5,I1,MT7],[STAT,C2,R5,I2,MT8],[STAT,C5,R3,I5,MT8],[STAT,C6,R3,I6,MT6],[STAT,C7,R3,I1,MT4],[STAT,C10,R1,I8,MT4],[MATH,C3,R5,I3,MT1],[MATH,C4,R3,I4,MT2],[MATH,C8,R5,I7,MT5],[MATH,C9,R4,I7,MT1],[MATH,C11,R3,I3,MT3]	1.00000	0
1	[STAT,C1,R1,I1,MT3],[STAT,C2,R1,I2,MT7],[STAT,C5,R2,I5,MT5],[STAT,C6,R4,I6,MT7],[STAT,C7,R5,I1,MT4],[STAT,C10,R3,I8,MT1],[MATH,C3,R3,I3,MT5],[MATH,C4,R3,I4,MT3],[MATH,C8,R3,I7,MT6],[MATH,C9,R2,I7,MT6],[MATH,C11,R4,I3,MT2]	0.50000	1
2	[STAT,C1,R5,I1,MT8],[STAT,C2,R1,I2,MT3],[STAT,C5,R2,I5,MT4],[STAT,C6,R3,I6,MT5],[STAT,C7,R3,I1,MT1],[STAT,C10,R1,I8,MT1],[MATH,C3,R3,I3,MT3],[MATH,C4,R5,I4,MT2],[MATH,C8,R2,I7,MT8],[MATH,C9,R2,I7,MT8],[MATH,C11,R2,I3,MT1]	0.33333	2
3	[STAT,C1,R4,I1,MT4],[STAT,C2,R2,I2,MT1],[STAT,C5,R3,I5,MT2],[STAT,C6,R2,I6,MT3],[STAT,C7,R2,I1,MT1],[STAT,C10,R5,I8,MT7],[MATH,C3,R5,I3,MT7],[MATH,C4,R4,I4,MT3],[MATH,C8,R4,I7,MT1],[MATH,C9,R1,I7,MT6],[MATH,C11,R4,I3,MT6]	0.33333	2
4	[STAT,C1,R1,I1,MT2],[STAT,C2,R3,I2,MT6],[STAT,C5,R2,I5,MT7],[STAT,C6,R2,I6,MT1],[STAT,C7,R2,I1,MT1],[STAT,C10,R1,I8,MT7],[MATH,C3,R5,I3,MT2],[MATH,C4,R5,I4,MT7],[MATH,C8,R2,I7,MT8],[MATH,C9,R3,I7,MT6],[MATH,C11,R3,I3,MT1]	0.33333	2
5	[STAT,C1,R2,I1,MT2],[STAT,C2,R3,I2,MT6],[STAT,C5,R5,I5,MT4],[STAT,C6,R2,I6,MT1],[STAT,C7,R4,I1,MT1],[STAT,C10,R4,I8,MT8],[MATH,C3,R5,I3,MT6],[MATH,C4,R4,I4,MT1],[MATH,C8,R5,I7,MT8],[MATH,C9,R1,I7,MT3],[MATH,C11,R4,I3,MT6]	0.33333	2
6	[STAT,C1,R3,I1,MT5],[STAT,C2,R4,I2,MT7],[STAT,C5,R5,I5,MT5],[STAT,C6,R2,I6,MT8],[STAT,C7,R1,I1,MT2],[STAT,C10,R3,I8,MT5],[MATH,C3,R5,I3,MT1],[MATH,C4,R4,I4,MT2],[MATH,C8,R2,I7,MT3],[MATH,C9,R5,I7,MT2],[MATH,C11,R3,I3,MT5]	0.25000	3
7	[STAT,C1,R1,I1,MT1],[STAT,C2,R4,I2,MT1],[STAT,C5,R4,I5,MT4],[STAT,C6,R1,I6,MT8],[STAT,C7,R1,I1,MT1],[STAT,C10,R2,I8,MT6],[MATH,C3,R1,I3,MT1],[MATH,C4,R5,I4,MT5],[MATH,C8,R5,I7,MT3],[MATH,C9,R1,I7,MT5],[MATH,C11,R1,I3,MT4]	0.20000	4
8	[STAT,C1,R3,I1,MT5],[STAT,C2,R5,I2,MT7],[STAT,C5,R5,I5,MT5],[STAT,C6,R5,I6,MT4],[STAT,C7,R3,I1,MT5],[STAT,C10,R3,I8,MT2],[MATH,C3,R1,I3,MT4],[MATH,C4,R3,I4,MT5],[MATH,C8,R4,I7,MT3],[MATH,C9,R3,I7,MT4],[MATH,C11,R5,I3,MT5]	0.16667	5

Table 2. The timetable.

Class	Dept	Course (number, max # of students)	Room (capacity)	Instructor (Id)	Meeting time (Id)
01	STAT	PS (C1, 50)	R5 (60)	Dr A (I1)	16:00 - 18:00 (MT7)
02	STAT	S&K2 (C2, 50)	R5 (60)	Dr B (I2)	16:00 - 18:00 (MT8)
03	STAT	PPS (C5, 50)	R3 (60)	Dr E (I5)	16:00 - 18:00 (MT8)
04	STAT	KTP (C6, 50)	R3 (60)	Dr F (I6)	14:00 - 16:00 (MT6)
05	STAT	SK (C7, 50)	R3 (60)	Dr A (I1)	10:00 - 12:00 (MT4)
06	STAT	AP (C10, 50)	R1 (60)	Dr H (I8)	10:00 - 12:00 (MT4)
07	MATH	PL (C3, 50)	R5 (60)	Dr C (I3)	08:00 - 10:00 (MT1)
08	MATH	PPB (C4, 50)	R3 (60)	Dr D (I4)	08:00 - 10:00 (MT2)
09	MATH	PI (C8, 50)	R5 (60)	Dr G (I7)	14:00 - 16:00 (MT5)
10	MATH	PP (C9, 50)	R4 (60)	Dr G (I7)	AS 08:00 - 10:00 (MT1)
11	MATH	TG (C11, 50)	R3 (60)	Dr C (I3)	IR 10:00 - 12:00 (MT3)

5. Conclusions

In this paper it concentrated on a scheduling of timetable. The suitable techniques being chosen in order to solve the problem. This problem of scheduling being solve by using a genetic algorithm and it shown that genetic algorithm is the best choices to solve a problem regarding of student timetable. Then, the java programming languages being applied to get the result. In this paper, the meeting and day has been set. The GA operators which are selection, crossover and mutation also is very helping in getting the best solutions and efficient. Selection, crossover and also mutation give the best solution and variation.

Reference

1. Lawrence L. & Bethel, (1956), *Industrial Organization and Management*, Mc Graw-Hill Publisher.
2. Jha S. K., (2014), Exam timetabling problem using genetic algorithm, *International Journal of Research in Engineering and Technology*, 3(**5**), 649-654.
3. Burke E. K., McCollum B., Meisels A., Petrovic S., & Qu, (2007), R. A graph-based hyper-heuristic for educational timetabling problem, *European Journal of Operational Research*, **176**, 177-192.
4. Lansburgh, R. & Spriegel, W. (1995), *Industrial Management*, Wiley and Sons Publisher, New York.
5. Kacem I., (2003), scheduling flexible job-shops: a worst case analysis and an evolutionary algorithm, *International Journal of Computational Intelligence and Applications*, 3(**4**), 437-452.

6. Sauer J., Appelrath H.-J. (2000), Integrating transportation in a multi-site scheduling environment, *published in the Proceedings of the Hawai'i International Conference On System Sciences*, Maui.
7. Labidi M., Mrad M., Gharbi A., and Louly M. A. (2014), Scheduling IT staff at a bank: a mathematical programming approach, *The Scientific World Journal.*
8. Ribeiro C. C. (2012), Sports scheduling : problems and applications, *International Transactions In Operational Research*, 19 (**1-2**), 201-226.
9. Yuan Z., Fugenschuh (2003), Home health care scheduling: A case study, *European Journal of Operational Research*, 151(**3**), 447-460, 6.
10. Sufahani, S. & Ismail, Z. (2014), A real scheduling problem for hospital operation room, *Applied Mathematical Sciences,*8 (**113-116**), 5681-5688
11. Sufahani, S.F. & Ahmad, A. (2012), A comparison between normal and non-normal data in bootstrap, *Applied Mathematical Sciences*, 6 (**89-92**), 4547-4560