Application of Genetic Algorithm for Solving University Timetabling Problems: a case study of Thai Universities

NAWAT NUNTASEN [1], SUPACHATE INNET [2]
Informatics Research Laboratory
School of Engineering, University of the Thai Chamber of Commerce
Bangkok 10400 THAILAND
Tel. +662-697-6708, Fax: +662-275-4892

Abstract:-There are many problems in university timetabling derived from various limitations and constraints. They comprise congruence and relation among group of students, subjects, lecturers, rooms, periods and time space. Genetic Algorithm (GA) was applied to deal with these problems. This paper shows the innovation of GA operator in level of elements. Fitness functions of chromosomes, GA operators (crossover and mutation) are run through the process of this development. Appropriate value testing to search for the most effective point was conducted under variation of proportion between crossover and mutation. The result of this research displayed that the given appropriate value of crossover 93% and mutation 7% of GA operator is the best rate to resolve the problem of university timetabling.

Key-Words: Genetic Algorithm, Automated Timetabling Problems, Evolutionary Computation

1 Introduction

Automated timetabling is a task of importance as it saves a lot of man-hours work and provides optimal solutions with constraints satisfaction within minutes. This boosts productivity, quality of education, quality of services, and finally quality of life. To produce high quality university timetable, optimal constraints satisfaction and optimization of the timetable's objectives at the same time is suggested [4]. Problems of university timetabling are NP-complete for reaching high quality solutions [2][3][5]. It is a searching of solution to reach the most appropriate timetabling which does not enable within the appropriate time. A Genetic Algorithm (GA) is a powerful algorithm to find optimized solution; hence, is employed to solve these problems [3][1].

Genetic algorithm is developed from evolution of living things to survive on the Earth under such changeable environment [6]. Parents' chromosomes and GA operation constructs evolution in better consequence of the next generation. Many parents' chromosomes were constructed to solve problems and gained groups of desired answers [8] [9]. This task needs huge computer's memory. In the past, many researchers have studied applications of GA for solving universities timetabling problems [4][8][9]. They are mostly interesting in element-level conditions. However, nobody yet has totally considered separation of period between the same subjects. This problem occurs when; for example, a subject requires two connected periods which one period may be set in one place and the other may be located in the other. The algorithms is developed to eliminate this problem as well as to reach high efficient solution with less computation memory employment.

The paper is organized as follows. After the Introduction, the problems of universities timetabling is stated in details. Then, the steps of GA application for timetabling are discussed, and the constraints employed to reach the effective results are identified. Next, the model of chromosomes is proposed. Then the constraints stated in the previous session are transforming to Fitness Function. After that, the Genetic Operators, i.e., Crossover and Mutation are explained. The results session is then provided following with the conclusion session.

2 Timetabling Problems: a University Perspective

University Timetabling is arranging to reach congruence and relation among students, subjects, lecturers, rooms, periods and time space. To gain effective timetable, problems have to be solved. Such problems are able to be revealed by constraints. They can be considered into two groups, Hard Constraints and Soft Constraints [2] [3] [4].

Hard Constraints are timetabling unacceptable problems that need compulsory treatment. These problems were not allowed to be happened. For example, a lecturer cannot teach different subjects in the same period of day. While Soft Constraints are ones, that can be accepted within minimization of frequency. They will maximize the perfectiveness of timetabling, e.g., there should not be time space more than two to three periods between the nearby subjects of students' schedule. Both of them are focused in this paper.

3 Problem Solving Procedure

In this section, a genetic algorithm is applied to solve university timetabling problems under some certain constraints. The algorithm can be explained as follows:

1. *Produces two prototypes (called parents chromosomes: P1, and P2)*
2. *Introduces crossover process to produce two children chromosomes (denoted as Ch1 and Ch2).*
3. *Examines fitness value of each chromosome with the fitness function, and then selects two best chromosomes that give highest fitness value.*
4. *Decides whether to use crossover process or mutation process by randomizing a probability value.*
   - *If crossover process is chosen, do 3 - 4 again.*
   - *If mutation process is employed, examine fitness value of each chromosome with the fitness function, and then select only the chromosome that gives highest fitness value to do the mutation process. Then put it back to the pool and continues 4 again.*
5. *Continues the process until the fitness value is met the requirement or the user end the process.*

The proposed genetic algorithm use only four chromosomes in the mating pool to be employed in the crossover process or mutation process. This reduces utilization of computation memory. Fig.1. shows how to produce desired chromosomes from their parents. Fig.2 shows the process of mutation.

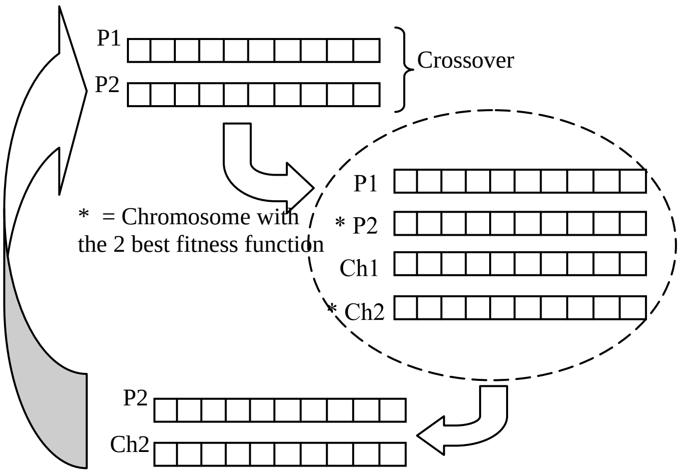

Fig.1. Steps of GA application for timetabling

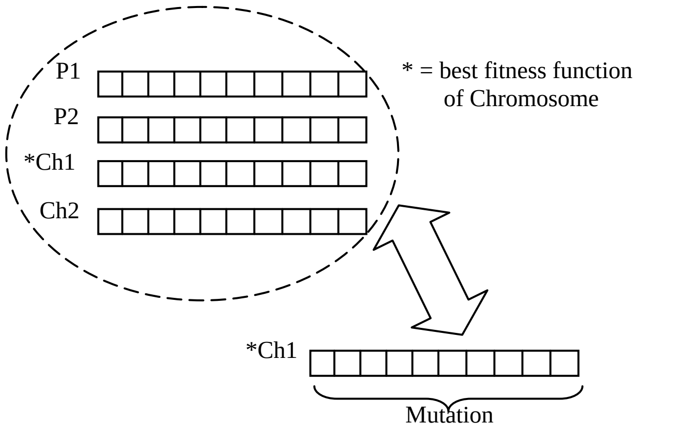

Fig.2. The best chromosome selection for Mutation

3.1 Constraint Identification.

The constraint identification was gained from the timetabling principles of the University of the Thai Chamber of commerce, which applied for bachelor degree of engineering regular course, year 2004-2006. The schedule is from Monday to Friday within 6 periods a day. They are explained in details in the next subsection.

3.1.1 Hard Constraints

1 Lecturers are not allowed to teach different subjects in the same period of a day.

2 A room is not allowed to be occupied by different subjects in the same period of a day.

3.1.2 Soft Constraints

1 There should not be time space more than two to three periods between the nearby subjects of students' schedule.

2 In each day, period 3 or period 4 should be vacant for students' lunch space.
3 There should not be more than 4 continuing periods occupied for students in a day.
4 There should not be more than 4 continuing periods occupied for lecturers in a day.

These constraints are conducted into this paper to gain the desired effectiveness.

3.2 Timetabling Chromosomes applied by GA

Timetabling Chromosomes applied by GA are constructed into tuple or group of element (E) to be spaces for selected input [6][7][8][9]. According to this paper, elements comprise three types of data, which are lecturer (L), Subject (S) and Room (R). In each element, it can be presented in a form of E = {L, S, R}. A set of E will represent as constituent elements of chromosomes. So, E = {E1,…, En} will represent as the elements of input to timetabling while each of E(1-n) comprise of consequent set of L, S, and R. Each of them has subset of each own. It can be shown as L = {L_1, L_2, L_3,…, L_n}, which will represent lecturers in that semester. In the same way of a set of Subject and a set of Room in that semester will do as S = {S_1, S_2, S_3,…, S_m}and R = {R_1, R_2, R_3,…, R_n}, respectively. To enter the timetabling by elements, it can be happened in three types of phenomena. They are a period for an element, several periods for an element, and a blank period or more for no element as there is no class in the schedule as shown in Table.1.

Table 1. Types of phenomena happened in timetabling

	1	2	3	4	5	6
Mon	E1	E 1				
Tue			E2	E2 {L_1,S_4,R_2}		
Wed	E7 {L_9,S_{12},R_5}				E9	E9
Thurs						
Fri				E21	E21	

A chromosome will be stringed following the sequences of periods that yield the length of that chromosome as shown in equation (1)

$$lChrom = G * P(n) \qquad (1)$$

where, lChrom: the length or number of bits of each chromosome G: Group of students who register the subjects following the plan in each major and P(n): number of periods in the timetabling.

Bit	1	2	…	…	9	9	…	17
Chro	E1	E1	…	…	E2	E2	…	E9

Fig.3. Timetabling chromosomes

Fig.3 shows timetabling chromosomes resulting from equation (1).

3.3 Fitness Function

Fitness function is used to measure appropriateness of timetabling and search to find the best one, which capable to be the raised problems. Fitness function constructed from timetabling conditions. It can be shown as the following equations;

$$\begin{aligned}
& w_1 \sum_{G=1}^{G=n}\sum_{P=1}^{P=40}\sum_{S=0}^{S=n}\sum_{L=0}^{L=n} \text{BoundLecturerClash}\left[\left(\left(L_L, S_S\right), P_P\right), G_G\right] \\
& + w_2 \sum_{G=1}^{G=n}\sum_{P=1}^{P=40}\sum_{S=0}^{S=n}\sum_{R=0}^{R=n} \text{BoundRoomClash}\left[\left(\left(R_R, S_S\right), P_P\right), G_G\right] \\
& + w_3 \sum_{G=1}^{G=n}\sum_{P=1}^{P=40}\sum_{S=0}^{S=n} \text{Break not more than 2 period}\left[\left(S_S, P_P\right), G_G\right] \\
& + w_4 \sum_{G=1}^{G=n}\sum_{P=3}^{P=4}\sum_{S=0}^{S=n} \text{Break Period 3 or Period 4}\left[\left(S_S, P_P\right), G_G\right] \\
& + w_5 \sum_{G=1}^{G=n}\sum_{P=1}^{P=40}\sum_{L=0}^{L=n} \text{LecturerLoad}\left[\left(L_L, P_P\right), G_G\right] \\
& + w_6 \sum_{G=1}^{G=n}\sum_{P=1}^{P=40}\sum_{S=0}^{S=n} \text{Student Load}\left[\left(L_L, P_P\right), G_G\right] \qquad (2)
\end{aligned}$$

where weight of condition for each constraints are as follows:

- w_1 *: weight of condition which a lecturer is not allowed teaching different subjects in the same period.*
- w_2 *weight of condition which a room is not allowed to be used to teach different subjects in the same period.*
- w_3 *: weight of condition which a group of students should not have more than two periods free before any subjects in timetabling schedule.*

- w_4: *weight of condition which the period 3-4 should be free for lunch time in a day.*
- w_5: *weight of condition which a lecturer student should not have classes more than 4 continuing periods.*
- w_6: *weight of condition which a group of students should not have classes more than 4 continuing periods.*

In addition, the value of each function can be explained as seen below:

$$Bound\ Lecturer\ Class\ [((L_L, S_S), P_P), G_G$$

- *Equal 0: when P are as L_0 (no lecturer) or there is the same L_L and the same S_S in the same P*
- *Equal 1: when the same L_L happens in the same P but different S_S*

$$Bound\ Room\ Class\ \{((R_R, S_S), P_P), G_G$$

- *Equal 0: when as others*
- *Equal 1: when the same R_R happens in the same P but different S_S*

$$Break\ not\ more\ than\ 2\ Period\ [(S_S, P_P), G_G]$$

- *Equal 0: when as others*
- *Equal 1: when S_S happens in P and P-4 or S happens in P and P+4*

$$Break\ P3\ or\ P4\ [(S_S, P_P), G_G]$$

- *Equal 0: when S_S happens in P_3 or when S_S does not happen in P_3 and P_4*
- *Equal 1: when S_S happens in P_3 and P_4*

$$Lecturer\ Load\ [(L_L, P_P), G_G]$$

- *Equal 0: when P happens in L_L while $P \leq 4$*
- *Equal 1: when P happens in L_L while $P > 4$*

$$Student\ Load\ [(S_S, P_P), G_G]$$

- *Equal 0: when S_S happens in P of each G while $S_S \leq 4$*
- *Equal 1: when S_S happens in P of each G while $S_S > 4$*

The Best answers of timetabling by GA application comprise the minimum target function, which is not allowed to have the hard constraints (w_1, w_2) and the one which should yield minimize soft Constraint (w_3, w_4, w_5, w_6)

3.4 Genetic Operator

A GA application is a process to improve and develop chromosomes of the best answers by selection the chromosome which has the best fitness function value. Then such chromosome takes into genetic process, which comprise of two genetic operators: Crossover and Mutation. This paper shows development of genetic cycle. It begins from the two initial chromosomes called parents, and produces subsequent ones following the genetic evolution, including process development of Crossover and Mutation. Data of L, S, and R were examined in level of element for timetabling.

Process of Crossover and Mutation in timetabling will be run inside each string of chromosomes, which occupied by groups of students. It can be shown as the following picture of chromosome Fig.4.

	Group 1					Group 2				
Bit	1	2	3	..	30	31	32	33	..	n
period	1	2	3	..	30	31	32	33	..	n
Ch1	E1	E1		..	E10		E9	E9		En
	Genetic Operator					Genetic Operator				
Ch2		E7	E7	..		E25		E9	..	En

Fig.4. Chromosome grouping for genetic process

3.4.1 Crossover

Crossover is an exchange element of two parents' chromosomes. Its process comprises:

1. In the first chromosome of parents, Random position to run crossover.
2. At the right position from the random, elements were examined by the 2 following steps:
   - 2.1 From the random position, elements, which bordered to each other were examined
   - 2.2 In case of such positions have equal value, both of them will be saved and put in the first child Chromosomes. In case of not equal, one from the random will be selected. It can be shown as the following picture of chromosome Fig.5.

Bit	1	2	3		30	31	32	33	…	n
Period	1	2	3		30	31	32	33	…	n
Parent 1	E1	E1		E5	E10		E9	E9		En
	*				*			*		
Ch1	E1	E1			E10		E9	E9		

Fig.5. Chromosome random and selection for crossover

3. At the second chromosome of parents,delete every element , which same with the random position in the first chromosome of parents and put the residual elements in to the next space of the chromosome of the first child.
4. Using the same principles to the second chromosome of parents to yield the chromosome of the second child.

Bit	1	2	3	…	30	31	32	33	…	n
period	1	2	3	…	30	31	32	33	…	n
Parent 2	~~E5~~	~~E1~~	E1	…		~~E9~~	~~E9~~	E2		En
Ch1	E1	E1	E5		E10	E2	E9	E9		

Fig.6. Delete and Selected Element from the Second Parents of Chromosome

3.4.2 Mutation

The process of chromosome mutation was done by altering the elements randomly. This can be explained in detail as follows:

1. Random two positions in a group of students to run mutation in selected chromosomes.
2. At the right position from the random, elements were examined. Then two following steps were processed.
   - 2.1 At positions, which randomly bordered to each oth]er, elements were examined.
   - 2.2 In case of such positions have equivalent value, both of them will be saved to insert at the first position, which makes the sequent shifting of previous chromosomes to the next position. It can be shown as the following picture of chromosome Fig.7.

4 Results

This trail used data for timetabling of School of Engineering, the University of the Thai Chamber of Commerce. They comprise 80 subjects, 26 rooms, 42 lecturers and 25 groups of students.

Bit	1	2	3	4	5	…	27	28	29	30
period	1	2	3		30	31	32	33	…	n
Ch1	E1	E1		E5	E10		E9	E9		En
	*						*	*		
Ch1	E9	E9	E1	E1	E5					

Fig.7. Chromosome random and selection for Mutation

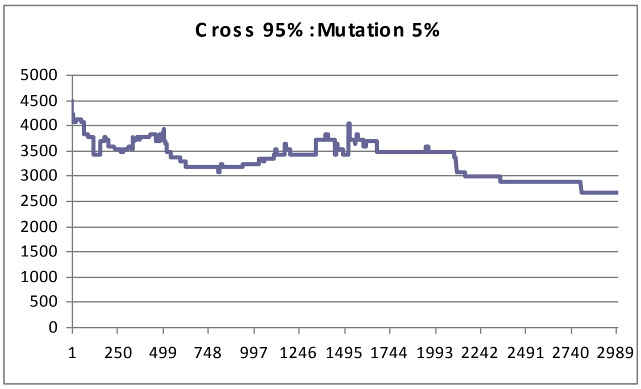

Fig.8. Fitness function of GA under crossover rate of 95 % and mutation rate of 5 %

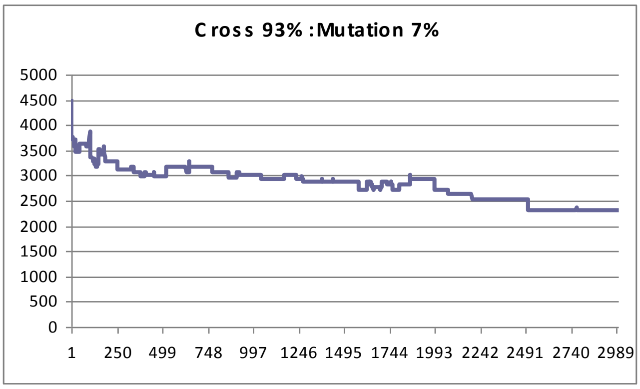

Fig.9. Fitness function of GA under crossover rate of 93 % and mutation rate of 7 %

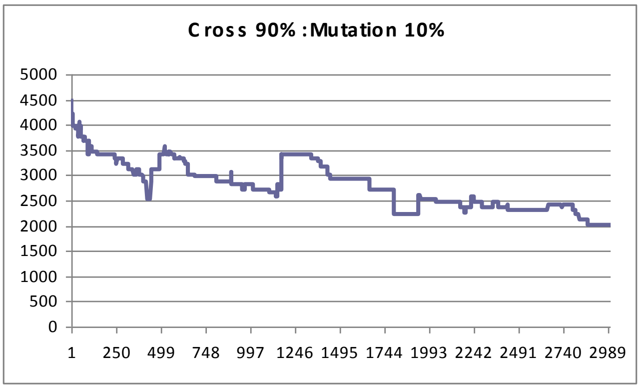

Fig.10. F Fitness function of GA under crossover rate of 90 % and mutation rate of 10 %

Three ratios of crossover and mutation were presented to show the desired period, which shows tendency of effectiveness via 3,000 generations of processing. Fig. 8 – 10 show how the values of fitness functions have been improved by generations with variations of ratio of crossover rate and mutation rate.

The results from the figures show that fitness function have tendency to descend until period of desired proportion is shown in Fig.8 - 10. However, every time the chromosomes have been produced by mutation process, the value of fitness function can be either increased or decreased. From Fig. 8 and Fig. 10, the results show that the fitness value is fluctuated. This cannot give the tendency of prediction. Fig.9 can be the best predictable line of fitness function reduction among the all the three.

The result tabulated in Table 2 also shows that the proposed algorithm solves the problem of which one subject requires connected period.

Table.2. Example of university timetable after using the model of GA with crossover rate of 93% and mutation rate of 7%

Period

Day

12	12	0	0	28	28
20	20	0	0	19	19
17	17	21	21	0	0
0	0	0	0	14	14
22	22	0	0	0	0

5 Conclusion

This paper presents an alternative model of GA, which has two points distinctiveness. The first one is the way to get the best couple of chromosomes by using fitness function among chromosomes inside the group to decrease memory units loading. The second is about a condition of no period separation of the same subject in the same group.

This paper yields two strong comments according to apply this model. Firstly, there should be the control of Mutation process which should allow Mutation only when fitness function is less than fitness function of the any chromosomes inside the group. Secondly, there is no rule for setting the ratio of Crossover rate and Mutation rate. As the circumstance changes, the ratio can be changed. Therefore, the ratio of Crossover rate and Mutation rate must be tested and selected to gain optimum results.

Reference:

[1] E. K Burke, D. G. Elliman and R. F. Weare, A Genetic Algorithm based University Timetabling System, *the 2nd East-West International Conference on Computer Technologies in Education*, 1994.

[2] S. Gyõri, Z. Petres, A. R. Várkonyi-Kóczy, Genetic Algorithms in Timetabling. A New Approach, 2001, *http://www.mft.hu/hallg/200107.pdf*.

[3] P. Srinin, P. Rojanavasu, E. Pin-nguen, The Automatic Timetabling Problem by using Genetic, *Joint Conference on Computer Science and Software Engineering (JCSSE2005)*, 2005.

[4] S. Kazarlis, Solving University Timetabling Problems Using Advanced Genetic Algorithms, *5th International Conference on Technology and Automation (ICTA'05)*, 2005.

[5] E. Ozcan, A. Alken, Timetabling using a Steady State Genetic Algorithm, *The 4th international conference on the Practice And Theory of Automated Timetabling*, Belgium, 2002.

[6] D. Abramson, J. Abela, A Parallel Genetic Algorithm for Solving the school Timetabling Problem, *15th Australian Computer Science Conference*, Hobart, 1992.

[7] A. Colorni, M. Dorigo, V. Maniezzo, Genetic Algorithms and Highly Constrained Problems: The Time-Table, *1st International Workshop on Parallel Problem Solving from Nature*, 1990.

[8] K. Paitoonwattanakij, K. Vongvipaporn, GA For Scheduling School Timetable, *J.Natl.Res.Council Thailand*, 2000.

[9] S. Junnatat, Multi Objective Genetic Algorithm For School Timetabling Problem, *The National Conference on Computing and Information Technology*, 2005.