A Novel Approach of Genetic Algorithm for Solving University Timetabling Problems: a case study of Thai Universities

NAWAT NUNTASEN [1], SUPACHATE INNET [2]
Computer Science and Information Technology Department
Faculty of Science and Technology, Rajabhat Mahasarakham University
Mahasarakham 44000 THAILAND[1]
Informatics Research Laboratory
School of Engineering, University of the Thai Chamber of Commerce
Bangkok 10400 THAILAND[2]
Tel. +662-697-6708, Fax: +662-275-4892

Abstract:- University timetabling problems have been interested by many researchers for more than a decade. However, there is no appropriated solution or computation model available to solve these problems successfully. This is because of many different version of timetabling problems. In this paper, a noval approach of Genetic Algorithm (GA) for solving educational timetabling problem is proposed, including the constraints statements, the definition of a hierarchical structure for the fitness function, and the generalized genetic operators, which can be applied to matrices representing timetables. The paper focuses on lecturing timetables only, but not the examinational timetabling, and a Thai university is employed for a case study. The crossover rate and mutation rate were varied to conduct effective results and they shows that the given appropriate crossover rate of 50% and mutation rate of 50% is the best ratio to solve university timetabling problems which significantly can be able to control the maximum one period separation between subjects and decrease loading of computer resources for GA process.

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

1 Introduction

Course (school or university) timetabling problems have been interested by many researchers for more than four decade. This is because of different forms of timetabling problem. Usually, these problems are often solved by leveraging human resource in universities or institutes. This requires a couple weeks or more to be done and the result is often not satisfied. Although, there are different problems, there exists many problem solving methods to save a lot of man-hours work, which usually concepts of employ optimization algorithms. Such methods include Genetic Algorithms [1]-[9], Tabu Search [10], Simulated annealing [11], and Constraints Logic Programming [12]. However, there is currently no appropriate computer tool.

To obtain great quality university timetable, optimal constraints satisfaction and optimization of the timetable's objectives at the same time is introduced [4]. Problems of university timetabling are NP-complete for reaching high quality solutions [2][3][5], so that solving it satisfactorily is often hard. 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, and the constraints employed to gain effectiveness are identified. Then, the development of GA for solving university timetabling problems are discussed. In addition, the subsection includes a discussion of proposed model of chromosomes, required fitness function, and the employed genetic operators, i.e., crossover and mutation. The results session is then provided following with the conclusion session.

2 Problems Formulation

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.

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 (30 periods are the maximum for timetabling in a week). They are explained in details in the next subsection.

2.1 Hard Constraint*s*

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.

2.2 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 New Approach Genetic Algorithm

3.1 Developed Genetic Algorithms

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 ranking by fitness function.*
4. *Mutation or crossover was Randomly chosen*

    4.1. In case of crossover, the two best chromosomes that give lowest fitness value to gain parents chromosomes of next generation (Fig.1).

    4.2 In case of mutation (Fig.2), one best chromosome that gives lowest fitness value following with comparison with

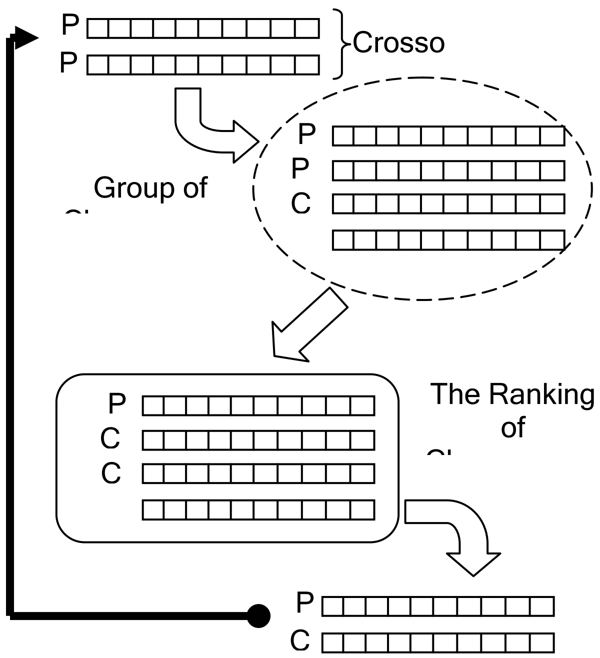


Fig.1. Crossover was randomly selected for timetabling

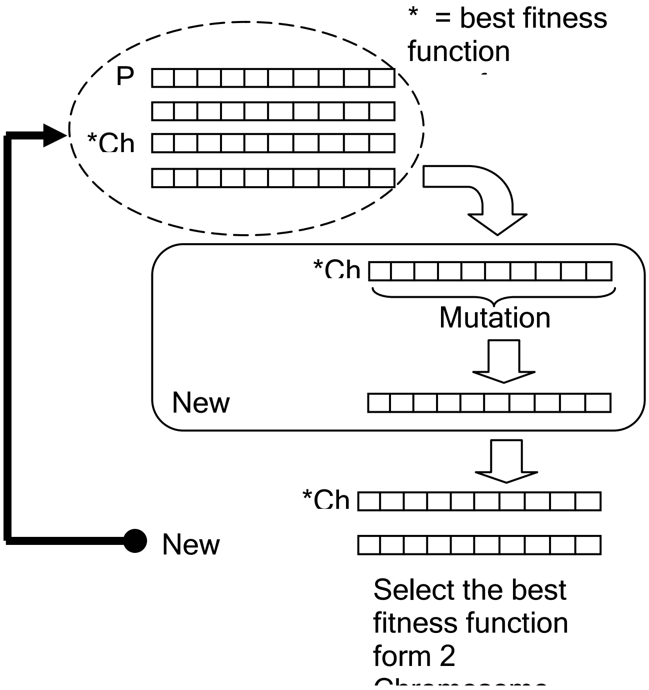


Fig.2. Mutation was randomly selected for timetabling

previous values to be selected mutation under the criteria of lesser or equal of previous minimize value. 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.

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 = $\{L_1, L_2, L_3, \ldots, L_n\}$, which willrepresent 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 = $\{S_1, S_2, S_3, \ldots, S_m\}$and R = $\{R_1, R_2, R_3, \ldots, R_n\}$, 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.

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

Table 1. Types of phenomena happened in timetabling

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

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

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. We use the same fitness function as same as our previous work [13]. It can be rewritten in a simple form as shown in equation (2) below.

$$\text{weight} \times \sum \text{hard constraints} \times \sum \text{soft constraint} \quad (2)$$

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:

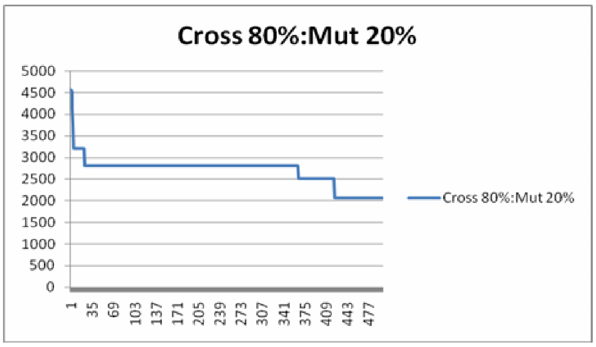
1. Random two positions in a group of students to run mutation in selected chromosomes.
2. At the right position from the random, elements were examined. Then two following steps were processed.
   2.1 At positions, which randomly bordered to each other, elements were examined.
   2.2 In case of such 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.

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					

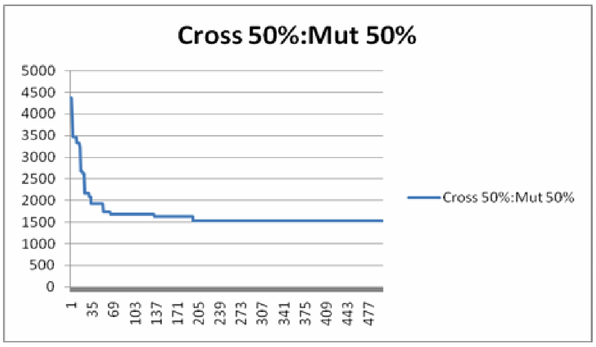
Fig.7. Chromosome random and selection For Mutation

4 Results

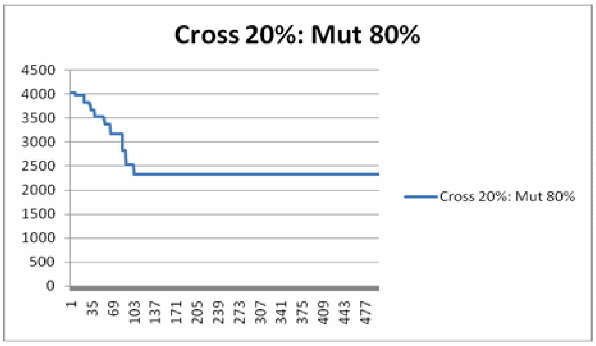
This trail uses data for timetabling of School of Engineering, the University of the Thai Chamber of Commerce. They comprise 80 subjects, 26 rooms, 42 lecturers and 25 groups of students.500 generations of GA application were conducted via 3 different proportions between crossover : mutation which are 80%:20%,50%;50% and 20%:80% . They are shown as Graph 1-3 as follows;



Graph 1. Fitness function of GA under crossover rate of 80% and mutation rate of 20%



Graph 2. Fitness function of GA under crossover rate of 50% and mutation rate of 50 %



Graph 3. Fitness function of GA under crossover rate of 20 % and mutation rate of 80 %

Three ratios of crossover and mutation were presented to show the desired proportion of crossover and mutation which shows tendency of effectiveness via 500 generations of processing. Graph 1-3 show how fitness function values have been improved by generations with variations of ratio of crossover rate and mutation rate.

The results from the 3 graphs show that fitness function values have tendency to descend until each of stable lines is shown. Selective mutation yields that every time the chromosomes have been produced by mutation process, the value of fitness function can be decreased or same level. From Graph 1-3, the results show that the fitness value has no fluctuation. This can give the tendency of prediction. Graph 2. Can be the best predictable line of fitness function reduction among the all the three.

The results in Table 2, it also shows that the more rate of crossover will give better fitness function value but need more generation of processing while the more mutation rate will give the opposite results. The comparison among the 3 different proportions shows that the lowest fitness function value with minimize number of generations reachable to stable line of GA under crossover rate of 50% and mutation rate of 50 % are the most satisfactory one.

Ratio of GA Operator	Best Generation	Fitness Function value
Cross 80% : Mut 20%	423	2070
Cross 50% : Mut 50%	197	1520
Cross20% : Mut 80%	103	2318

Table.2. Ratio of GA Operator, Best Generation and Fitness Function value

According to the loading of computer resource as shown in Fig. 8, the program use only 13 % of CPU loading and 15,816 Kb of memory. It is acceptable because very small loading of computer resource is required.

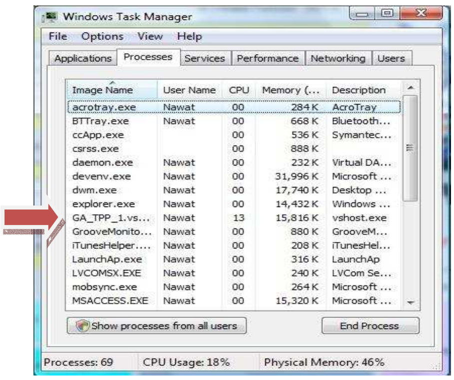


Fig.8. Computer resource utilization

In addition, more than one period of separation between the subjects was not found after using this program under crossover rate of 50% and mutation rate of 50 % which is one of distinctiveness. It can be shown in Fig. 9 as follow;

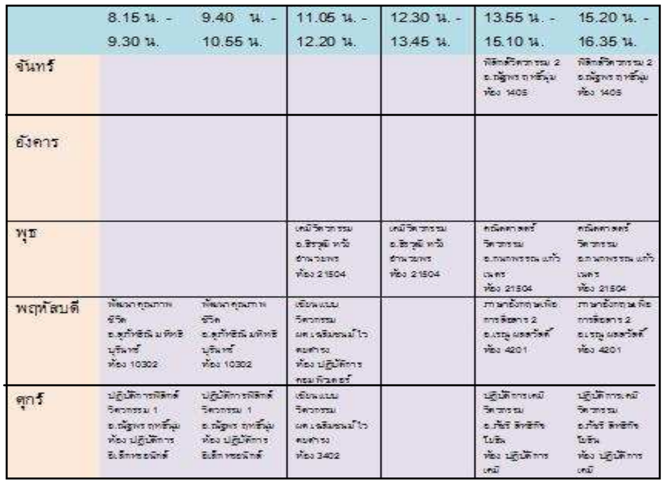


Fig. 9. Timetabling results from GA applications under the satisfied trail.

5 Conclusion

This study represents genetic algorithm approach by use generation of living things which has construct chromosome group of answer and construct cycle of new genetic algorithm, which in constructing schedule of the course that is rapid collecting and using resource of the system. That has efficient ability to control cross over and mutation which has no affect to schedule desire in learning continuity more than one period. However, this approach should improve the process and be able to response of various conditions in the schedule which is appropriate and the environment of the problem occurred in constructing a class schedule which is different in each education institute.

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.

[10] A. Herth, Tabu Search for Large Scale Timetabling Problem, *European Journal of Operational Research*, 1992.

[11] A. Abramson, Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms, *Management Science*, 1991.

[12] C. Gueret, N. Jussien, P. Boizumault, C. Prins, Building University Timetables Using Constraint Logic Programming. In: Burke, E, Ross, P. (Eds.) Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers), *Vol. 1153. Springer-Verlag, Berlin Heidelberg New York, 1996*

[13] N. Nuntasen, S. Innet, Application of Genetic Algorithm for Solving University Timetabling Problems: a case study of Thai University*, 7th WSEAS International Conference on Simulation, Modelling, and Optimization, Beijing, China, 2007.*