International Journal of Artificial Intelligence and Neural Networks – IJAINN
Volume 4 : Issue 4 [ISSN 2250 – 3749]
Publication Date : 27 December, 2014

An Automatic Scheduling of Academic Time Table by using Genetic Algorithms

Chinnakrit Sittiha, Panuwat Ketwong and Sittichai Bussaman

***Abstract*— Time Table Scheduling is one of optimization problems. It use a skillful human and take a long time for each constraint. Many of researchers try to use computers to manage them automatically. There are two main goals in this research: to schedule the academic time-table of Surin Poly-technical College automatically by using Genetic Algorithms and to find the best condition of Genetic operators for solving the scheduling problem. The best condition was found from 24 experiments; two point crossover followed by position base mutation is the best condition sequence. Having got the sequence, a computer program was implemented by using that condition. The application was useful after testing with the constraints of Surin Poly-technical College.**

***Keywords*— Genetic Algorithm, Time Table Scheduling, NP-complete, Optimization.**

I. Introduction

The timetabling problem (TTP) is one of challenging topic for computational intelligent research, because it is NP-complete [1]. A typical timetable instance requires several days of work for a manual solution [2], depending on the complexity of constraint. Academic time-table scheduling is a typical planning of meeting between teachers and students in a period of time. Several techniques have been developed to automatically solve the problem. A comprehensive review and recent research directions in the timetabling can be found in [3], [4], and [5].

Genetic Algorithms (GAs) are very well known, having several applications to general optimization and combinatorial optimization problems. GA is based on the controlled evolution of a structured population, and is considered as an evolutionary algorithm. It handles a population of possible solution. Each solution is represented through a chromosome. Several researchers such as [6], [7] and [8] used GAs to solve a university time-tabling problem.

The objectives of this work is to develop a computer program that can schedule an academic (college) time-table automatically and to find the best condition which appropriate for Surin Poly-Technical College constraint. This paper is organized as follows: after the introduction, Section 2 presents the problem formulation that make up a problem instance and the form of a solution. Section 3 describes the genetic algorithm implemented. Finally, Section 4 demonstrated the experiments carried out with the algorithm on problem instances, and Section 5 is a conclusion.

Chinagrit Sittihar
Department of Computing Technology for Education, Faculty of Science, Rajabhat Mahasarakham University, Thailand.

Panuwat Ketwong
Department of Computer Technology, Faculty of Science, Rajabhat Mahasarakham University, Thailand.

Sittichai Bussaman
Department of Computer Science, Faculty of Science, Rajabhat Mahasarakham University, Thailand.

II. Problem Formulation

The constraint was gained from the time-table scheduling principles of the college. The schedule is from Monday to Friday within 8 periods a day. The requirements or constraints include hard- and soft constraints. Hard Constraints are not allowed to be happened. While Soft Constraints are ones, that can be accepted within minimization of frequency. Both of them are focused in this paper.

A. *Hard Constraint*

- A teacher is not allowed to teach in the same period of a day. This constraint can be formulated as equation (1).

$$F1 = \min(\sum_{t=1}^{Nt} \sum_{p=1}^{Np} Countteacher(p)) \quad (1)$$

- A room is not allowed to be occupied by more than one subject per period. This constraint can be formulated as equation (2).

$$F2 = \min(\sum_{r=1}^{Nr} \sum_{p=1}^{Np} Countroom(p)) \quad (2)$$

B. *Soft Constraint*

- There are at least one free day for a teacher in a week. This constraint can be formulated as equation (3).

$$F3 = \min(\sum_{t=1}^{Nt} \sum_{d=1}^{Nd} Countfreeday(d)) \quad (3)$$

- A teacher is not assigned to teach in both morning and afternoon periods. This constraint can be formulated as equation (4).

$$F4 = \min(\sum_{t=1}^{Nt} \sum_{b=1}^{Nb} Counttime(b)) \quad (4)$$

C. *Fitness Function*

Having got the constraints, Fitness function was conducted to measure appropriateness of timetabling and search to find the optimal one. A modeling of fitness function can be written as shown in equation (5).

$$F = \min(\sum_{i=1}^{4} W_i F_i) \quad (5)$$

W_i is a weight of constraint, hard constraint and soft constraint are heterogeneous.

III. GA Modeling

GA is a method that use genetic as its model of problem solving. It is a search technique to find approximate solutions. Each solution is represented through a chromosome. Coding solutions into a chromosome is the first part. After that, a set of reproduction operators has to be determined. Reproduction operators are applied directly on the chromosomes.

A. *Problem encoding*

Encoding is a process of representing individual genes. The problem in this work was encoded to be a chromosome as shown in figure1.

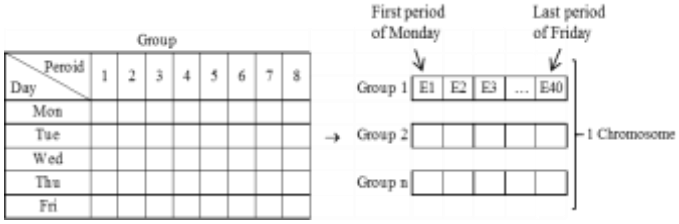

Figure 1. Encoding Scheme.

B. *Reproduction Operators*

Process of Genetic Operators in timetabling will be run inside each string of chromosomes. In this work, several classical genetic operators were selected to be candidates to finding the best condition for the constraint. These operators are as follows:

- One point crossover
- Two point crossover
- Position Base crossover
- Inversion Mutation
- Center Inversion Mutation
- Regeneration Mutation

C. *GA process*

After encoding step, GA process was occupied for some steps as follows:

1. Produces 10 parent-chromosomes randomly.
2. Introduces Genetic Operations process to produce 10 children-chromosomes.
3. Now mating pool has 20 chromosome, examines fitness value of each chromosome with the ranking by fitness function.
4. Select the new 10 parent-chromosomes for next generation.
5. Terminate GA process when the fitness value converges toward the optimal solution.
6. Go to step 2.

IV. Result

According to the objective of this work, there are two categories of result.

A. *Condition finding*

For the step of condition finding, the genetic operators was tested for 24 experiments from the data of 1/2012 semester. The result can be divided into 4 tables. TABLE I and TABLE II shown results on the conditions of either crossover or mutation. TABLE III and TABLE IV shown results on the condition of combination between crossover and mutation.

Each of experiments was executed 5 times. The best condition, which was found on TABLE III, was the sequence of genetic operators that come first by the Two-point crossover and then followed by the Regeneration mutation. It yield average 241.2 of fitness value and terminated at 8929.8 generation.

TABLE I. CROSSOVER

Crossover Type	*Fitness Value*	*Generation*
One point	8800.6	247.8
Two point	9123.2	1544
Position Base	10063.6	206.8

TABLE II. MUTATION

Mutation Type	*Fitness Value*	*Generation*
One point	16478	43
Two point	15894	69.8
Position Base	1696	4565

TABLE III. CROSSOVER AND MUTATION

Crossover Type	*Mutation Type*	*Fitness Value*	*Generation*
One Point	Inversion	7126.4	2003.8
One Point	Center Inversion	6192	2573.2
One Point	Regeneration	683.6	8754.8
Two Points	Inversion	7121.8	2169.8
Two Points	Center Inversion	8798	1901.4
Two Points	Regeneration	241.2	8929.8
Position Base	Inversion	6231.6	1890.6

Crossover Type	*Mutation Type*	*Fitness Value*	*Generation*
One Point	Inversion	7126.4	2003.8
One Point	Center Inversion	6192	2573.2
Position Base	Center Inversion	5944.6	2394.4
Position Base	Regeneration	442.8	8982.6

TABLE IV. MUTATION AND CROSSOVER

Mutation Type	*Crossover Type*	*Fitness Value*	*Generation*
Inversion	One Point	4914.2	3930.4
Center Inversion	One Point	5350.8	4954.4
Regeneration	One Point	977.8	8102
Inversion	Two Points	5750.4	6222.6
Center Inversion	Two Points	6213.2	2165
Regeneration	Two Points	442.6	8751.8
Inversion	Position Base	4685.6	5589.6
Center Inversion	Position Base	3437.2	6458
Regeneration	Position Base	697.6	8506.4

B. *Implemented Program*

Having got the best condition, a computer program was implemented. The program was shown as follows:

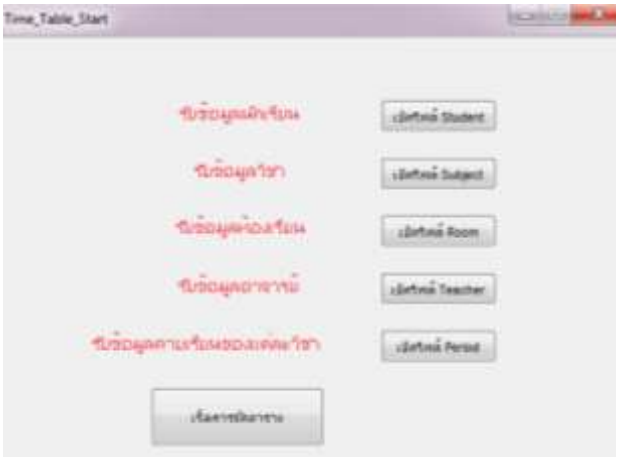

Figure 2. First window when enter the program.

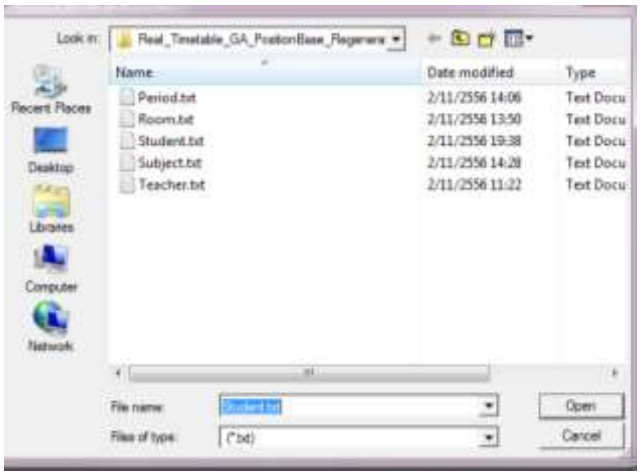

Figure 3. The example window for data loading.

Figure 4. Example of the complete time-table

v. Conclusion

By conclusion, this paper present the computer program that automatically schedule an academic time-table for Surin Poly-Technical College constraint by using GA-based technique. Several classical genetic operators was selected to be a candidate to finding the best condition for the constraint. The program has been worked well with no hard-constraints appear.

References

[1] S. Even, A. Itai, and A. Shamir, "On the Complexity of Timetable and Multicommodity Flow Problems," Siam Journal of Computing, vol. 5, No. 4, pp. 691–703, December 1976.

[2] de Werra D., "An introduction to timetabling," European Journal of Operations Research, pp 151–162, No.19,1985 .

[3] M. W. Carter and G. Laporte, "Recent developments in practical course timetabling,"inProc.2ndInt.Conf.Pract.TheoryAutomatedTimetabling, (Lecture Notes in Computer Science), vol. 1408, pp. 3–19, 1998.

[4] R. Lewis, "A survey of metaheuristic based techniques for university timetabling problems," OR Spectrum, vol. 30, no. 1, pp. 167–190, 2008.

[5] R. Qu, E. K. Burke, B. McCollum, and L. T. G. Merlot, "A survey of searchmethodologiesandautomatedsystemdevelopmentforexamination timetabling," J. Sched., vol. 12, no. 1, pp. 55–89, 2009.

[6] P. Adamidis and P. Arakapis. "Weekly lecture timetabling with genetic algorithms". Proceedings of the 2nd International Conference on the

Practice and Theory of Automated Timetabling, University of Toronto, Canada, 1997. 4.

[7] J.P. Caldeira and A.C Rosa. School timetabling using genetic search. Proceedings of the 2nd International Conference on the Practice and Theory of Automated Timetabling, University of Toronto, Canada, 1997.

[8] S. Abdullah and H. Turabieh, "Generating university course timetable using genetic algorithm and local search," in Proc. 3rd Int. Conf. Hybrid Inform. Tech., 2008, pp. 254–260.