Senior high school course scheduling using genetic algorithm

NGAPH Saptarini*, I W Suasnawa, P I Ciptayani
Electrical Engineering Department, Politeknik Negeri Bali, Jalan Kampus Bukit Jimbaran, Badung - 80364, Bali, Indonesia

Corresponding email*: ayu.harry@pnb.ac.id

Abstract. Manual course scheduling can be very complex and take a long time, even sometimes violate hard and soft constraints. Soft constraints usually relate to teacher and student preferences as part of the schedule. The purpose of this study was to apply genetic algorithms (GA) to prevent the violation of hard constraints and minimize the violation of soft constraints. The GA in this study distributing population in some groups. The distributed GA generate groups of population and then after each iteration, the migration between groups will be conducted based on given probability of migration. The distributed GA applied to avoid the premature convergence that could occur in original GA. The probability of migration observed in this study was 0, 0.1, 0.2, 0.3, 0.4, and 0.5. The study shown that the distributed GA succeed to prevent violation of hard constraints, minimize the soft constraints violation and avoid the premature convergence.

1. Introduction

Senior high school will certainly arrange the lesson schedule at the beginning of each semester. The process of setting up a high school schedule can be a complex and time-consuming problem. The process of preparing the schedule also requires high precision in order to avoid clashes between the resources involved in it. On scheduling in general, there are several resources that need to be organized including: subjects, students, teachers, study rooms. In high school, students are usually incorporated into specific classes, so the student arrangements are packaged rather than individual. The room can be fixed or in turn. The classroom will be used permanently by a particular class, while the laboratory room will be used in turns according to the practice session schedule for a particular lesson. The process of setting up a schedule can be very complex because of the various limitations, such as no teacher may teach two classes in the same hour, the laboratory room should not be used by more than two classes simultaneously, certain subjects should not be exercised over the clock 10, the students should not get the same subjects in the adjacent day, and others. Manual scheduling by relying on human accuracy will be long and vulnerable to errors that violate defined limitations. Sometimes the limited amount of resources makes arranging the scheduling process even more difficult.

Some of algorithms used to solve scheduling problem, such as local search [1], [2], simulated annealing[3], [4], cultural algorithm that focused mainly on genetic and natural selection concepts[5] and genetic algorithm. Genetic algorithm is one of the most popular meta heuristic method for combinatorial problem. Genetic algorithm able to solve combinatorial problems such as scheduling in a faster time and satisfactory results. Genetic algorithms generate initial solutions randomly, then step
by step through steps called selection, crossover, mutation and elitism will improve the solution repeatedly until the final conditions are met. The use of genetic algorithms in the settlement of scheduling cases has been widespread. Pattanayak and Kumar [6] used a binary genetic algorithm using elitism and adaptive mutations to solve the scheduling case of multi-user broadcasting with multiple inputs and outputs. The result of this study is that the genetic algorithm used successfully accelerates the scheduling process significantly with the schedule results close to the optimal value of the actual schedule. Kaiafa and Chassiakos [7] used a genetic algorithm to construct resources on a project and compare the results with paid project scheduling software. The results of this study indicate that the genetic algorithm is able to produce a more balanced schedule and optimize predetermined goals. While Nanda et al in [8] use heuristic algorithm to solve timetabling problem in school. In that study, heuristic algorithm was used to complete school scheduling from the teacher point of view. The resources set out in the scheduling are batch of students, subjects and teachers. Sbeity et al. [9] combines the Analytical Hierarchy Process (AHP) with a genetic algorithm to complete the scheduling process at the school. AHP is used to assess teachers according to the criteria provided by the school. The value will be an input to the genetic algorithm in determining the schedule according to the limitations given and satisfy each teacher. Pappu et al used parallel genetic algorithm to solve the exam scheduling problem. The parallel GA was used to reduce the computation time for yielding the solution. This kind of GA is operated in parallel system. The experimental results showed that PGA produces good exam timetables with good parallel efficiency[10]. Mahto and Kumar [11] used genetic algorithm to schedul the exam and implemented that GA using Mathlab. Timilsina et al [12] used 3D structure to represent the chromosome. This 3D structure was used to represent the real representation of timetabling. The used of GA in timetabling can be found in [13], [14], [15] and [16].

An improvement using parallel or distributated population in GA was found in study [17]. The study applied parallel GA(PGA) in travelling salesman problem. The study found that the PGA that using migration approach gave the best result. The migration is the operator that guides the exchange of individuals among demes in a DGA [18]. Based on [18], there was a migration rate that determining the number of individuals that undergo migration in every exchange. This value can be expressed as a percentage of the population or else as an absolute value. In any case, it is not clear which is the best value for the migration rate. The study suggested low percentage should be used (from 1% to10% of the population size) but the study also stated that many studies have been conducted and no clear conclusions can be drawn about the best migration rate. It is depend on the case to be solved. The study [19] suggested the new migration technique called the selective migration model. This scheme allows migration among demes only if the individuals meet certain criteria at both the source and the destination. Experiments show that this model improves the performance by offering faster convergence in large population setups and better solutions in time-constrained small population setups. Study [20] used adaptive migration in DGA to prevent the premature convergency and to reduce the difficulty in determining the parameters. The individuals for migration are chosen automatically according to the average quality of the subpopulations.

This study will be carried out using distributed genetic algorithms (DGA) that is based on the previous research was the improvement of GA. The DGA will generate several groups of solutions that will follow each genetic algorithm process in each group, and at the end of the iteration there will be an exchange of solutions from each group. As found in [18] that it is not clear which is the best value for the migration rate, so this study aims to observed best migration rate (probability of migration / pmig) in order to avoid local optimum conditions. Instead of use adaptive migration such as done in [20], here we observed six value of pmig, because when use adaptive migration, we have to found the good formula to define the value of pmig. This study result complete scheduling with distributed genetic algorithms regardless of laboratory limits used interchangeably. Here we have to manage many classes with weekly subject and the classes will divide into 2 time slots, morning and afternoon, and each timeslot will have sub time slot. Data ini this study was taken from SMA Surya Wisata, Jalan Wagimin, Kediri, Tabanan.
2. Methodology

Genetic algorithm is a heuristic search technique that is widely used to solve the problems of combinatorics[21]. Genetic algorithms start to find a set of solution randomly called population. While each individual in the population called chromosomes, that are a representation of the solution. Chromosomes evolved into an ongoing and iterative process called generation. In every generation, the chromosomes are evaluated by an evaluation function and produce a fitness value to indicate the degree of resilience of the chromosome in adapting to the problem. After several generations, the genetic algorithm will converge on the best chromosome, which is expected to be the optimal solution.

The DGA used a group of population set. Each group will conduct their own GA and at the last of each iteration, the migration between group is conducted. Therefore, the population’ member of each group will be vary and by this scenario we hope the local optimum will be avoided. The DGA we used in this study was conducted as shown in Figure 1.

2.1. The Chromosome Representation

In GA, the solution is represented by the chromosome. The chromosome will be randomly generated. In this study, we represent the chromosome as Figure 2. The chromosome was represent using two-dimesional representation (2D). The rows represents section (day and time), while the columns represents class name. Each class consists of eleven time slots and six days, so there will be 55 section (rows) for each class. Here, the DGA will generate the value in each cell. The value assigned will code the teacher and subject, for example 21 represents teacher Mr John for subject English. In the example Mr John will teach English 2 time slot on Monday. The code of each teacher and subject will be saved in database before the algorithm run.

Figure 1. The flowchart of modified DGA
2.2. Fitness Function

Before we define the fitness function, we have to define the hard and soft constraints for this problem. The hard constraints are the constraint that can be violated by the solution provided, while the soft constraints are the constraint that can be violated, but will decrease the value of the solution. So the objective of this study is to minimize the violation of soft constraint. The lists of hard constraints are shown in Table 1.

| No | Hard Constraint                                                                 |
|----|-------------------------------------------------------------------------------|
| 1  | Each subject should be taught by the appropriate teacher in that field         |
| 2  | No teacher may be placed on the same section in a different class (no clash)   |
| 3  | There can be no lesson on Monday first section (because there will be the national ceremony) |
| 4  | No class can take the same subjects for more than 3 hours in a day.            |

The soft constraints will be ordered based on their priority. The most important criteria will be given the higher score. The soft constraints and their score for this case are shown in Table 2.

| No | Soft Constraint                                                                 | Penalty |
|----|-------------------------------------------------------------------------------|---------|
| 1  | Total teacher’ section in a week exceed 50 hours or more                      | 12      |
| 2  | Total teacher’ section in a week exceed 45 hours or more                      | 11      |
| 3  | Total teacher’ section in a week exceed 40 hours or more                      | 10      |
| 4  | Total teacher’ section in a week exceed 35 hours or more                      | 9       |
| 5  | Total teacher’ section in a week exceed 30 hours or more                      | 8       |
| 6  | Distribution of uneven teaching hours for teachers in one area                | 8       |
| 7  | Total teacher’ section in a week exceed 25 hours or more                      | 7       |
| 8  | Total teacher’ section in a week exceed 20 hours or more                      | 6       |
| 9  | A class get 3 hours of the same lesson on the same day                        | 6       |
| 10 | The morning lessons are completed over the 9 sections                         | 5       |
| 11 | The morning lessons are completed over the 8 sections                         | 3       |
| 12 | The afternoon section completed over 9                                        | 5       |
| 13 | The afternoon section completed over 8                                        | 3       |
| 14 | Lesson hours with 2 section/week load broken down on different days           | 2       |
| 15 | There are interlude of lessons in a class                                     | 1       |

The fitness function of this case is as follow:
The selection of parent to conduct crossover will be done by roulette wheel method. This method will select the parent based on the fitness they have. The greater the fitness, the more likely it is to be selected. This is done to provide a greater opportunity for superior individuals to lower the genes they have in shaping the next generation.

2.3. Selection
The selection of parent to conduct crossover will be done by roulette wheel method. This method will select the parent based on the fitness they have. The greater the fitness, the more likely it is to be selected. This is done to provide a greater opportunity for superior individuals to lower the genes they have in shaping the next generation.

2.4. Crossover
The crossover technique used in this research is one-point-crossover. This technique is chosen because it is quite simple and does not take too long. Thus it is expected that the timing of the formation of the schedule will not take too long. In doing crossover will use parameters in the form of crossover probability (pc). The higher the pc the more likely it will be the crossover.

2.5. Mutation
Mutations are performed with the probability of mutations (pm) that tend to be small. Mutation in this research will be done with change and swap technique. This will depend on the following conditions: (a) change will be done if there are other teachers whose lessons are less than other teachers in their field or (b) swap will be done for other conditions.

2.6. Elitism
Elitism is done by eliminate the worst chromosome in a population and replace it with the better child produced by the crossover. This is aim to gain the better schedule from one generation to next generation.

2.7. Migration Between Groups
This is the important part of the DGA. We proposed our own technique here. This step will generate a random number. If the random number is below than the probability of migration then the algorithm will randomly select the chromosome to swap between one group to other. This step is done to prevent the algorithm to stuck at a local optimum that lead to premature convergence. Here we used six kind of migration rate to observe the effect of migration rate in order to prevent the premature convergence.

3. Result
We use the scheduling case in SMA Surya Wisata, Jalan Wagimin, Kediri, Tabanan. In this case study, we managed 9 classes (group of students) of first year students, 9 clasess of second year students and 9 classes of third year students. All of the classes will be divided into two time slots: morning and afternoon timeslot. Each time slot will divided into sub timeslot. Each sub timeslot will have maximal 11 section. Each class has their fix classroom. Here we have to distributed 61 teachers for certain subject and sub timeslot. The parameters used in this study was shown in Table 3.

| Table 3. The parameters in this study |
|--------------------------------------|
| Number of population group | 4 |
| Number of individu in population | 50 |
| Maximal number of generation | 1000 |
| Probability of crossover (pc) | 0.8 |
| Probability of mutation (pm) | 0.1 |
| Probability of migration (pmig) | 0, 0.1, 0.2, 0.3, 0.4 and 0.5 |
The first step in this study was generating 50 chromosomes randomly in each group. The next step was doing selection and crossover based on the pc and mutation based on the pm. After doing the the elitism procedure, the migration will be conducted based on the pmig. This process will be conducted repeatedly until 1000 generation or until all of the group reach their convergence state. Convergence state will be happen when the best fitness in the group is the same for 100 generation. In this study we investigate six kind of pmig. The pmig 0 is same with original GA because there is no migration and because this is original GA, we just generate one population. We do this to investigate the effect of DGA over GA and the effect of pmig to the best and mean fitness. To gain the valid data, we conduct the experiment 30 times for each pmig.

In Table 4, we can see that the maximum generation for original GA (pmig=0) was 354. That mean the original GA reach the convergence state very fast. The implement of DGA using low pmig 0.1 reach the convergence state in the 756th generation. While the bigger pmig resulted the maximum generation number is reach before the convergence state.

| pmig | Max gen |
|------|--------|
| 0    | 354    |
| 0.1  | 756    |
| 0.2  | 1000   |
| 0.3  | 1000   |
| 0.4  | 1000   |
| 0.5  | 1000   |

4. Conclusions
This study using distributed genetic algorithm (DGA) to solve the timetabling problem in senior high school. We managed the teacher and subject into different class and timeslot. DGA was used to prevent the algorithm from stuck at the local optimum. By implementing DGA, we generate four groups of population and apply the probability of migration to control the migration between one group to other group. We compare the original GA and various probability of migration to investigate the influence of DGA and pmig to convergence state of GA. Based on the experimental result, the DGA with pmig more than 0.1 avoid the premature convergence. For further work, we suggest to invetigate the various crossover technique for timetabling that give the better fitness.

5. Acknowledgment
We sincerely thank to the Ministry of Research, Technology and Higher Education Republic of Indonesia for the research fund through the agreement with Politeknik Negeri Bali. This research is funded by contract no : 023/SP2H/LT/DRPM/IV/2017 dated on April 12nd, 2017.

6. References
[1] Schaefer, A., 1999, Local Search Techniques for Large High-School Timetabling Problems, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 29(4) pp 368-377
[2] Savimiec, L., Constantino, A. A., 2017, Effective local search algorithms for high school timetabling problems, Applied Soft Computing Journal vol 60 pp 363-373
[3] Basir, N., Ismail, W., Norwawi, N. M., 2013, A Simulated Annealing for Tahmidi Course Timetabling, Procedia Technology vol 11 pp 437 – 445
[4] Sadegheih, A., 2006, Scheduling problem using genetic algorithm, simulated annealing and the
effects of parameter values on GA performance, Applied Mathematical Modelling vol 30 pp 147–154

[5] Sozaa, C., Becerrab, R. L., Riff, M. C., Coelloc, C. A. C., 2011, Solving timetabling problems using a cultural algorithm, Applied Soft Computing vol 11 pp 337–344

[6] Pattananayak, P., Kumar, P., 2015, A Computationally Efficient Genetic Algorithm for MIMO Broadcast Scheduling, Applied Soft Computing vol 36 pp 545-553.

[7] Kaiafa, S., Chassakiokos, A. P., 2015, A Genetic Algorithm for Optimal Resource-Driven Project Scheduling, Procedia Engineering vol 123 pp 260-267.

[8] Nanda, A., Pai, M. P., Gole, A., 2012, An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach, International Journal of Machine Learning and Computing 2 (4) pp 492-495.

[9] Sbeity I., Dbouk, M., Kobeissi, H., 2014, Combining the Analytical Hierarchy Process and The Genetic Algorithm to Solve the Timetable Problem, International Journal of Software Engineering & Applications 5 (4) pp 39-50.

[10] Pappu, S., Talele, K. T., Mandiwala, M. 2012, Application of Parallel Genetic Algorithm for Exam Timetabling Problem, International Journal of Scientific & Engineering Research 3(9) pp 1-4

[11] Mahto, M., Kr., Kumar, L. 2015. Exam Time Table Scheduling using Genetic Algorithm, International Journal of Enhanced Research in Management & Computer Applications 4(8) pp 31-35

[12] Timilsina, S., Negi, R., Khurana Y., 2015. Genetically Evolved Solution to Timetable Scheduling Problem, International Journal of Computer Applications 114 pp 12-17

[13] Mittal, D., Doshi, H., Sunasra, M., Nagpure, R., 2015, Automatic Timetable Generation using Genetic Algorithm, International Journal of Advanced Research in Computer and Communication Engineering 4(2) pp 245-248

[14] Gu, J., Hu, J. Zhao, T., Sun, G., 2012, A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment, JOURNAL OF COMPUTERS 7(1) pp 42-52

[15] Jain, A., Jain, S., Chande, P.K., 2010, Formulation of Genetic Algorithm to Generate Good Quality Course Timetable, International Journal of Innovation, Management and Technology, 1(3 ) pp 248-251

[16] Abdelhalim, E. A., El Khayat, A. A., 2016, A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA), Alexandria Engineering Journal vol 55 pp 1395–1409

[17] Wang L., Maciejewski, A. A., Siegel, H. J., Roychowdhury, V. P., 2005, A Study of Five Parallel Approaches to a Genetic Algorithm for the Traveling Salesman Problem, Intelligent Automation and Soft Computing, 11(4) pp 217-234

[18] Alba, E., Troya, J.M., 1999, A survey of parallel distributed genetic algorithms, Complexity 4(4) pp 31-52

[19] Eldos, T., 2015, A New Migration Model For Distributed Genetic Algorithms, Proceedings of the International Conference on Scientific Computing (CSC’ 06), Las Vegas, NV pp 128–134

[20] Lee, H., Oh, B., Yang, J., Kim, S., 2009, Distributed genetic algorithm using automated adaptive migration Evolutionary Computation, 2009 IEEE Congress on Evolutionary Computation pp 1835-1840

[21] M. Gen, and R. Cheng, Genetic algorithms and engineering optimization, New York : A Wiley-Interscience Publication, 2000