

University Course Timetabling with Genetic Algorithm: A Case Study

Toha Ardi Nugraha, Karisma Trinanda Putra, Nur Hayati

Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta
Kampus Terpadu UMY, Jl. Lingkar Selatan, Kasihan, Bantul, DI Yogyakarta 55183
e-mail: toha@ft.umsida.ac.id

Abstract – *University Course Timetabling Problems is a scheduling problem to allocate some lectures with some constraint, such as the availability of lecturers, number of classrooms and time slot in each day. The schedule of courses is one of important factors before start the semester in order to manage the study process. Generally, the university course scheduling in some universities are usually created manually through administration office. It needs to synchronize for all schedules from all departments in faculty of the university. In addition, the limitations of classroom and timeslot can make collision of the courses, lecturers and also incompatibility between the room capacity and the number of students whom take the course in the class. This paper proposes the university course time tabling systems. Based on some simulations with 93 courses, 18 lecturers and up to six classrooms, the result is that the system will get the best violation if the system adds more number of iteration. This situation also happens in the result of the scheduling lectures, the system will get the best percentage when the number of iteration sets as maximum. Copyright © 2017 Universitas Muhammadiyah Yogyakarta- All rights reserved.*

Keywords: *University Course Timetabling Problems, Genetic Algorithm, Scheduling,*

I. Introduction

The university timetabling problem is one of the cases of timetabling and common scheduling problems []. The aim of university course timetabling is to allocate the availability of resources under predefined constraints so that it maximizes the possibility of allocation or minimizes the violation of constraints.

One of the solutions to solve the scheduling problem is by using genetic algorithm. Genetic algorithm is a method which can be generated from evolutionary computation in the optimization technique. Genetic algorithm is classified as heuristic algorithm (not only exact solution but near is also acceptable) and stochastic method (using random process and selection criterion). Genetic algorithm has been used for machine learning and NP-complete, for example it is used for course timetabling problem in some university.

In Indonesia, university course scheduling in some universities are usually created manually through administration office of each faculty or department. Therefore, this process can take a long time because the officer still needs to synchronize for all schedules from all departments in that faculty. In addition, the limitations of classroom and timeslot can make collision of the courses, lecturers and also incompatibility between the room capacity and the number of students whom take the course in the class.

This paper investigates the university time tabling in a real-world problem from a major of private university in Indonesia. In this case, all of the classrooms have maximum capacity for fifty students and a single lecture's duration is run based on its credits. One credit is equal to fifty minutes of course. Therefore, the available time slots are different between each day and the others, depend on the total credits of all course running on the day.

The rest of this paper is organized as follows. Related works are described in the next section.

Section III is the university course timetabling problem description. Genetic algorithm is presented in Section IV. Result and discussion are described in section V and finally our conclusions are drawn in Section VI.

II. Related Works

The first research on timetabling for scheduling problems was from Gotlieb in 1962 [1]. After that, many researchers have introduced various timetabling approaches, as example by using graph methods, ant-colony optimization, population-based approaches (e.g., genetic algorithms (GAs), and etc. A comprehensive paper review and recent research directions in timetabling problems could be found in [2].

There are many researchers proposed many different techniques used to solve the University Course Timetabling Problems, but the paper [5] informs that genetic algorithm is powerful for very complex timetabling problems, but it still needs to be some modified. That paper also proposed the improvement of genetic algorithm for solving the University Course Timetabling Problems without crossover part. Therefore, this paper implements the improvement of the genetic algorithm (informed genetic algorithm) to solve the University Course Timetabling Problems in the real case in a university.

III. University Course Timetabling Problem

Regarding to [4,5], University Course Timetabling Problems is an NP-hard problem with purpose to set all the courses/lectures into available time slots such that there is no conflict within lecturer's schedules, student's schedules and to satisfy some other constraints, such as maximum number of lecture in a day for each lecturer, or to fulfill requested schedules. The large amount of lecture sessions with limited time slot and available rooms needs an effective and efficient method to obtain an optimum schedule where the manual approach will suffer for this.

Generally, in the first step of solving UCTP is defining the constraints that should be satisfied during the searching of optimum solution. The constraints are usually separated into two parts, namely hard constraints and soft constraints. Hard

constraints absolutely should not be violated because it prevents the conflict of lecturer schedules, class/ student schedules, and rooms to happen. On the other hand, an optimum solution should minimize the violation of soft constraints. However, for some cases with very complex condition, e.g. very large number of lecture with very limited time or room slots, the minimum number of violation of soft constraints is difficult to achieve.

There are a lot of algorithms that have been proposed by many researchers to contribute a better solution for UCTP. Most approaches are in the basis of Evolutionary Computations (ECs) and Swarm Intelligence. Some papers that used Evolutionary Computation (EC), especially Genetic Algorithm (GA) [3-7], have shown that optimum timetable solution could be produced efficiently using GA, even various modification to the standard algorithm can guide to much better solution. Swarm Intelligence (SI), especially Particle Swarm Optimization (PSO) is also popularly used as an alternative technique beside GA [8-10].

Table 1. Timetable Parameters

| Parameters | Value |
|-----------------------------------|---------|
| Number of Courses | 93 |
| Number of room/class | Up to 6 |
| Number of Lectures | 18 |
| Maximum student in the room/class | 50 |

This research investigates a real-world problem timetabling in a university consists of 93 courses from a major of private university in Indonesia with total of 18 lecturers and up to 6 classrooms. All the classrooms have the same maximum capacity of 50 students. A single lecture's duration is run based on its credits. One credit is equal to 50 minutes of lecture. Therefore, the available time slots are different between a days to the others depends on the total credits of all lectures running on that day.

Table 1. Data of lectures

| Number | Course | Lecturer | Class |
|--------|----------|----------|-------|
| 1 | TEI-2211 | ROH | A |
| 2 | TEI-2211 | ROH | B |
| 3 | TEI-4411 | ANN | A |
| 4 | TEI-4411 | ANN | C |
| 5 | TEP-2211 | TOH | D |
| | | | |
| 93 | TEP-2212 | RAM | C |

There are five hard constraints (HC) and eleven soft constraints (SC) with details as follow:

- **Hard Constraints (HC):** HC1. No time slot conflict; HC2. No lecturer conflict; HC3. No room conflict; HC4. No student/class conflict; and HC5. The lecturer must be run on the time slot with suitable duration.
- **Soft Constraints (SC):** SC1. Maximum number of lecture for each lecture in a day; SC2. Maximum number of lecture for each student/class; SC3. Maximum number of sequential lecture for each lecturer; SC4. The number of lecture run on night for each lecturer; SC5. The number of lecture run on night for each student/ class; SC6. Room of sequential lecture for each lecturer; SC7. Maximum number of lecture on the weekend; SC8. Maximum credits in a day for each lecturer; SC9. Minimum lecture in a day for each lecturer; SC10. Time constraint for lecturers who teach in the morning to not having schedule till night; SC11. Optional lectures conflict with regular lectures.

Table 2. Hard and Soft Constraint

| Constraint (c) | | Penalty/Violation Cost (p) |
|-----------------|------|----------------------------|
| Hard Constraint | HC1 | 100 |
| | HC2 | |
| | HC3 | |
| | HC4 | |
| Soft Constraint | SC1 | 10 |
| | SC2 | 10 |
| | SC3 | 10 |
| | SC4 | 20 |
| | SC5 | 20 |
| | SC6 | 20 |
| | SC7 | 20 |
| | SC8 | 30 |
| | SC9 | 30 |
| | SC10 | 30 |
| | SC11 | 30 |

IV. Genetic Algorithm

An Informed Genetic Algorithm has several steps to produce an optimum (minimum) fitness solution as informs in a schematic diagram in Figure 1.

- **Initialization chromosome** is formed using many gens which mean many lectures, for this research starts from $1 \leq N \leq 93$. The construction of chromosomes that satisfy all hard constraints and soft constraints, are achieved by adding or removing appropriate events (lectures) from the table based on room availability.
- **Genetic Operation:** The genetic operator for this research is defined as follows:
 - **Selection:** this process guides the evolutionary algorithm to the optimal solution by preferring chromosomes with low violations (penalty). The chromosomes change through the selection methods, which select n -gens (from timetables)

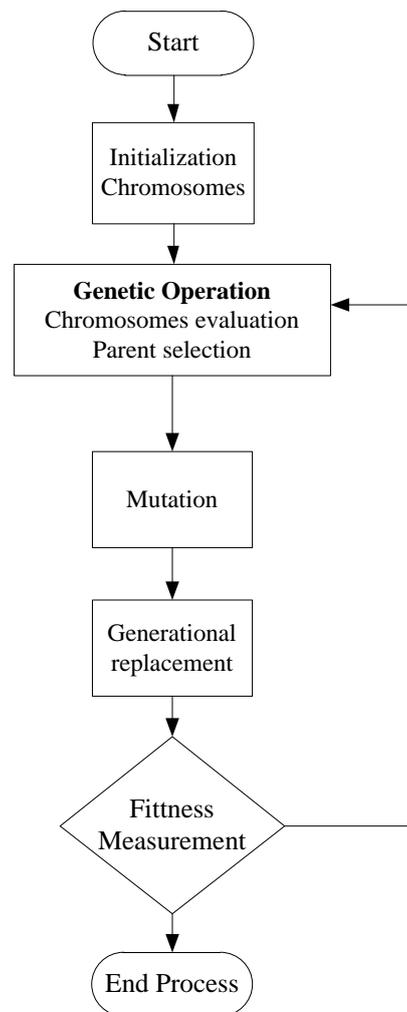


Figure 1. Schematic diagram of Genetic Algorithm

- Mutation: this is a background operator which allowing searches chromosomes and gens to find which the best chromosomes. Mutation points are selected by probability of mutation, P_m will set 0.5 in our implementations.
- Generational replacement: after mutation operations, the resulting chromosome should become infeasible or outside the search space.
- Fitness/violation measurement: this genetic algorithm is intended to minimize the violation score, i.e. the number of violation multiply by the defined penalty. The fitness function is formulated as [3].

$$f = T * \sum_i^n p_i - \sum_i^n c_i * p_i \quad (1)$$

where T is the total number of events/lectures, c_i and p_i represent total violation of i -th constraint and its penalty respectively. A timetable with violation of all constraints will have $f = 0$. But, a scheduling without violation/penalty will have the maximum fitness.

In this research, all those steps, basically, it can be modified with various settings and scenarios depend on the problem faced.

V. Result Discussions

This system was programmed by using NetBeans IDE 8.1 software and simulations were performed on the Intel Core i3 1.3GHz and 3 GB of RAM.

This system investigates several different numbers of iterations and initiate populations in order to evaluate which best as minimum violations/percentage of success in a case study as explained in Section II. As mentioned before, the resulting of some cases may have different mount of violations since there are given hard constrain and so many given of soft constraints. The best violation (minimum penalty) is that when there are no hard constrains happen on the system as Eq.1.

This system is capable to create some case of university timetable problems. Based on the real case study as mention in above, the results show that the system will get the best violation if the

system using high number of iterations. For example, if the system sets number of iteration in 100, the penalty cost is about 2000. If the system sets with 200 and 1000 iterations, the penalty cost are about 1000 and about 500 respectively.

Table 3. Number of iteration with penalty cost and percentage of success measurements

| Number of Iteration | Penalty Cost |
|---------------------|--------------|
| 50 | 93502 |
| 75 | 4215 |
| 100 | 2510 |
| 125 | 2360 |
| 150 | 1515 |
| 175 | 1325 |
| 200 | 1020 |
| 225 | 1015 |
| 250 | 1130 |
| 275 | 1055 |
| 300 | 925 |
| 400 | 835 |
| 500 | 525 |
| 600 | 715 |
| 700 | 735 |
| 800 | 730 |
| 900 | 450 |
| 1000 | 625 |

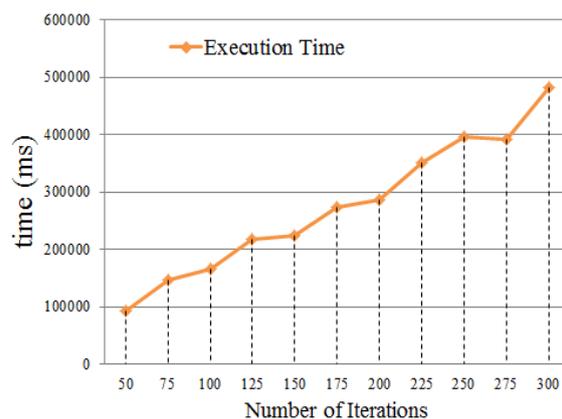


Figure 2. Time consuming with several number of iteration (50 – 300 with 25 steps each)

Based on the simulations from different number of iteration, execution times will longer if the system adds more number of iterations as shown in

Figure 2 and 3. The system sets with 1000 iterations will have execution time about 1500 seconds.

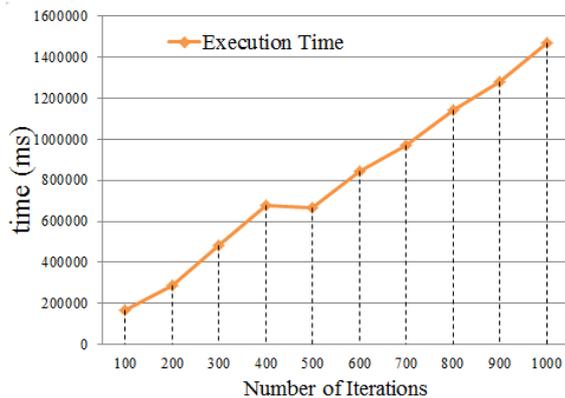


Figure 3. Time consuming with several number of iteration (100 – 1000 with 100 steps each)

This situation also happens for getting result of the scheduling lectures, the system will get the best penalty cost when the number of iteration sets as maximum.

VI. Concussion

This paper investigates university timetabling system using genetic algorithms. The research focuses on plotting the lecture automatically and testing the performance of the timetabling system. This system is programed to help some case of university timetable problems. Based on the real case study as mention in above, examined by using 93 courses, 18 lecturers and up to 6 classrooms. The result is that the system will get the best violation if the system adds more number of iteration. This situation also happens in the result of the scheduling lectures, the system will get the best percentage when the number of iteration sets as maximum.

Acknowledgments

Many thanks to LP3M Universitas Muhammadiyah Yogyakarta (UMY) for the financial support, and to all colleagues at Department Electrical Engineering, UMY and also to Muhammad Fachrie for the guidance and useful suggestions.

References

[1] Shengxiang Yang, Sadaf Naseem Jat, Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) (Volume: 41,

Issue: 1, Jan. 2011), pp. 93 - 106.
 [2] R. Qu, E. K. Burke, B. McCollum, and L. T. G. Merlot “A survey of search methodologies and automated system development for examination timetabling,” J. Sched., vol. 12, no. 1, pp. 55–89, 2009
 [3] Suyanto, An Informed Genetic Algorithm for University Course and Student Timetabling Problems, 2010, ICAICS, Springer Verlag, pp. 229-236
 [4] Jaya Pandey, A K Sharma, 2016, Survey on University Timetabling, Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on, 16-18 March 2016, New Delhi, India, pp. 160-164
 [5] A. SCHAERF, A Survey of Automated Timetabling, Artificial Intelligence Review 13: 87–127, 1999., Netherlands.
 [6] Hamdy M. Mousa, Ashraf B. El-Sisi, Design and implementation of course timetabling system based on genetic algorithm, Computer Engineering & Systems (ICCES), 2013 8th International Conference on, 26-28 Nov. 2013, Cairo, Egypt.
 [7] Shara S. A. Alves, Saulo A. F. Oliveira, Ajalmar R. Rocha Neto, A Novel Educational Timetabling Solution through Recursive Genetic Algorithms, Computational Intelligence (LA-CCI), 2015 Latin America Congress on, 13-16 Oct. 2015, Curitiba, Brazil.
 [8] Amir Hossein Karami, Maryam Hasanzadeh, University Course Timetabling Using a New Hybrid Genetic Algorithm, Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference on, 18-19 Oct. 2012, Mashhad, Iran.
 [9] Ho Sheau Fen Irene, Deris Safaai, Mohd Hashim Siti Zaiton , University Course Timetable Planning using Hybrid Particle Swarm Optimization, GEC'09, June 12–14, 2009, Shanghai, China. pp. 239-245.
 [10] Ho Sheau Fen Irene, Deris Safaai, Mohd Hashim Siti Zaiton , A Study on PSO-based University Course Timetabling Problem, dvanced Computer Control, 2009. ICACC '09. International Conference on, 22-24 Jan. 2009, Singapore, Singapore. pp. 648-651.
 [11] Ruey-Maw Chen and Hsiao-Fang Shih, Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search, Algorithms 2013, 6, 227-244.

Authors' information



Toha Ardi Nugraha, he received the B.E. degree in telecommunication engineering from Telkom Institute of Technology (Telkom University), Indonesia, in 2011 and the M.S. degree in IT Convergence Engineering from Kumoh National Institute of Technology, South Korea, in 2014. He worked at

R&D Center PT.Telekomunikasi, Indonesia, for three years, as a research assistant. Since 2016, he is working at Electrical Engineering Universitas Muhammadiyah Yogyakarta, Indonesia, as a lecturer and researcher. His research interests include wireless and mobile networks.



Karisma T. Putra, born in Bondowoso on June 19, 1990. He graduated from elementary to senior high school in Bondowoso until 2008. Studied bachelor degree program in Surabaya, precisely in Electronics Engineering Polytechnic Institute of Surabaya (EEPIS) until 2012. He got scholarship program to continue

master degree in Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Now, he is a lecturer at electrical engineering, faculty of engineering, Universitas Muhammadiyah Yogyakarta. The main focus of research is the intelligent systems and controls. He engaged in joint research related to the development of food commodity tracking systems and integrated intelligent systems. He was involved in several competitions in developing smart devices. Pursue the field of electronics and software development since college. Mr. Putra joined in Indonesian's engineer union organization (PII) in 2016. Mr. Putra is also active in writing publications on IEEE society.



Nur Hayati currently is a lecturer and researcher in Universitas Muhammadiyah Yogyakarta. She graduated from telecommunication engineering major in EEPIS-ITS Surabaya and continued her master degree in electrical engineering major Universitas Indonesia Depok. She is

member of senior instructor in Cisco Networking Academy Program and Unity of Indonesian Engineers (Persatuan Insinyur Indonesia)