Solving University Course Timetabling Problem Using Memetic Algorithms and Rule-based Approaches

M A Nugroho and G Hermawan*
Department of Informatics Engineering, Faculty of Engineering and Computer Science, Universitas Komputer Indonesia, Bandung, Indonesia

*galih.hermawan@email.unikom.ac.id

Abstract. In this paper, we present the optimization of a university course timetabling (UCT) using memetic algorithms and rule-based approaches. The method will optimize the course timetables adjustment which includes: lecturers, classrooms, time slots, and courses in one week of activity for one semester. Some hard and soft constraints are also applied to test the performance of the algorithm. The results of the experimental tests prove that memetic algorithms combined with some rule-based approaches can be used optimally for UCT. As for the occurrence of multiple violations of soft constraints, it can be handled by manually correcting the course timetables that have been generated. From some experiments conducted also known that the system has been able to produce satisfactory course timetables, which is to adjust the distribution of timetables needs between lecturers, classrooms, time slots, and courses.

1. Introduction
The timetabling problem is difficult to solve due to its huge search space and is highly constrained [1]. A lot of researchers has proposed some solver method for the timetabling problem. The memetic algorithm (MA) is the expansion of the genetic algorithm. The advantages of MA is on how it works in parallel. The MA work in the search space that uses many individuals at the same time so that the possibility of MA to be trapped in the local extreme is smaller than the other methods [1]. Heuristic method is one method solutions that use the concept of the approach. Some heuristic algorithms have been developed among others MA. The MA is have the process of local search that able to resolve the problem of debt rescheduling. According to Albar M A [2] on cases that have time slots numbered large memetic algorithm with good resolve the problem. According to Albar M A, Jat S N, and Yang S [2, 3] local search process that is owned by MA is also very helpful in the search for optimal solution closest to improve the quality of the individual. According to Poonam G [4], MA is better than the genetic algorithm in this space search the solution. And according to Ray Pillarrichie and Suyanto [5], there are differences in the quality of the chromosomes that is produced is seen from the value of fitness centers from every generation. In recent years, some research in the field of university course timetabling problem is still underway, as is done by Havas J [6], Chen R-M [7], and Perera M T M [8]. In addition, memetics algorithms utilization is also continuously developed for optimization purposes. As is done by Simon D [9] and Islam M M [10]. The timetabling problem is the problem of the complex lecture due to various components that consists of elective courses, lecturers, the classroom and lecture time slots with attention to the limitations and certain conditions that must be kept or met. In the
implementation, when many lecturers of courses that must be scheduled and rescheduling process requires a long time and high precision.

In this paper, we consider the timetabling problem with the master timetables approach, adapted to the requirements and characteristics of the Department of Informatics Engineering of Universitas Komputer Indonesia. In this case study, a semester is the academic time unit. The course timetables and student sectioning must be conducted in the beginning of every semester.

2. Research Methods
This study begins with the collection of the course timetables in the scope of the study program of Informatics Engineering, UNIKOM, in the even semester of the academic year 2015/2016. Then, the obtained dataset is converted into the database.

Furthermore, some rules apply to the database to match the needs of the campus course. The dataset will be used in the optimization process by the memetic algorithm. For testing purposes, different soft constraints are applied. The detail methods is shown in Figure 1.

![Figure 1. The research procedure](image)

The explanation in figure 1 is as follows. The first part is the process of entering the data related to the timetables, among which are the data of lecturers, courses, classrooms, time slots, time availability of lecturers. After all data has been entered, the system can generate timetables. The second part is the process of memetics algorithms. The first stage is to generate the population according to the input data from the user which will be the solution. In these populations there is a chromosome number of courses to be scheduled and there are 4 genes in each chromosome. Each gene represents the data of the lecturer of the course, hour, day, space. The next step is to calculate the fitness value of each individual, then the rank-based selection method of each individual is ranked based on the greatest fitness value to the smallest. And then the parent will be randomly selected based on the previous ranking process. At the crossover stage using a one-point crossover, at this stage two mains of the selection process will be exchanged based on the probability of crossover. If the random value generated is less than the probability, then a crossover occurs. After conducted the mutation process, where the new individual results from the crossover process will be randomly changed the gene. Then the last stage of the process of local search. Local searches are used to locate which individuals have the greatest penalty and fix the individual. The process is repeated constantly until there is an optimal solution (no constraint has been
violated) or not found the right solution when the generation has reached the maximum. The third part is the schedule of results from the second stage process (if solution is found).

3. Results and Discussion
In the creation of the course timetables, there are some limitations that should be noted. That the course timetables is created can be used. Where the limits of is not to be there that is broken or called hard constraint. There is also the limits of that be broken or called soft constraint.
The Hard constraint (hard) restrictions in timetabling is as follows:
a. Clashed pool and teaching time, when the hours of the day or pool fully charged in the same time.
b. Clashed lecturer, when a lecturer teaching at the same hour the same day or the same lecturer.
c. Clashed with the Friday prayer.
d. Clashed time availability of special lecturer.
Soft constraint (limitations software) is a limitation in which if broken will not damage the course timetables, will only reduce the quality of the course timetables. Soft constraint is obtained based on the limitations of a government agencies. Soft constraint effect on the course timetables, when the number of soft constraint that violated the more small then the quality of the course timetables the better. While the Soft constraint (limitations software) there is clashed time availability of lecturers.

The flow diagram of the memetics algorithm is shown in Figure 2.

![Flow Diagram](image)

**Figure 2.** The memetics algorithms flow diagram

The algorithms parameter testing was conducted to observe how the influence of the crossover probability and mutation probability against the number of generations and the time required to find the solution. The parameters that will be used in accordance with the test plan. The parameters will be tested to achieve the value of fitness centers reached 1 (no constraint being violated). The tests are performed as much as 10x on each of the parameters value. With the purpose to get what size parameter that matches to find the optimal course timetables solution.

3.1. Crossover parameter testing
The crossover parameter testing was conducted as much as 10x and obtained the results in Table 1.
Table 1. Testing results of the crossover parameters.

| Parameters             | The value of P crossover |
|------------------------|--------------------------|
|                        | 0.9                      | 0.95                     |
| Needed time (minutes)  | 188.7 minutes            | 158.2 minutes            |
| Number of generations  | 16822                    | 12029                    |
| Average fitness value  | 0.3075                   | 0.3164                   |

The results of the test obtained that crossover probability with the value of 0.95 better compared with 0.90 in the side of the time until it reaches the generation of the optimal solution and the average fitness value produced.

3.2. Testing the parameters of mutation

Based on the test from the previous crossover probability, then the value of the crossover probability used is 0.95. Mutation parameters testing done as much as 10x and obtained the results in Table 2.

Table 2. Testing results of the mutation parameters.

| Parameters             | The value of P mutation |
|------------------------|-------------------------|
|                        | 0.05                    | 0.1                      |
| Needed time (minutes)  | 140.7 minutes           | 118 minutes              |
| Number of generations  | 11325                   | 9971                     |
| Average fitness value  | 0.3164                  | 0.3118                   |

The results of the test is obtained that the probability of mutation with the value of 0.10 better compared with 0.05 in the side of the time, and generation to achieve optimal solution, but not with the average value of fitness centers that produced. Because of the difference in the average fitness centers, that is not too far then taken 0.10 to the value of the probability of mutation.

3.3. Testing the soft constraints

The used of soft constraints is the availability of regular lecturer, where applied to the lecturers gradually, ranging from 1 to 5 lecturers. All lecturers have not prepared as much as 100 slot in the slot time. Testing using parameter with the value of the crossover probability 0.95, probability of mutation 0.10, generations as much as 1000000, and the population as much as 18. The results of the experiment is shown in Table 3.

Table 3. Testing results of the soft constraints.

| Experiment number | Lecturer number | Constraint violation number |
|-------------------|-----------------|-----------------------------|
| 1                 | 1               | 2                           |
| 2                 | 2               | 5                           |
| 3                 | 3               | 8                           |
| 4                 | 4               | 15                          |
| 5                 | 5               | 17                          |

It can be seen the value of soft constraint being violated every experiment has increased. Then the conclusions can be made when the number of lecturers who have soft constraint more then soft constraint being violated will be more. This means that the timetables that formed there are still soft constraint so that the timetables has not been fully optimum said.
3.4. Overall Test results
Based on the results of the tests are shown in Table 1 and Table 2 obtained the optimal parameters to complete the scheduling problems with as much data 325 namely the value of the crossover probability of 0.95 and probability of mutation 0.10. With the population of 18 and generation as much as 1000000000. And based on the table 3 obtained that the more the number of lecturers who have soft constraint then the more the number of soft constraint is breached and this makes the course timetables less than fully optimize.

The difference between our research and Jat S N and Yang S [3] is that we apply some rule-based approach in this method. Some of the rule-based we apply is to determine the relationship between practicum courses and particular classrooms practice. As for Havas J [6], Chen R-M [7], and Perera [8] use different methods.

4. Conclusion
Based on the results, memetic algorithm with a rule-based combination has been able to solve the course timetables problem. In some soft constraint tests, we still found violations of soft constraint applied. Local search process is very helpful once in the improvement of the quality of the individual or increase the value of fitness each his generation. Local search process can also overcome the optimal local values that often occurs in the genetic algorithm when the data his input omitted numbered very much. In testing parameters, it is obtained that the value of the crossover probability of 0.95 and mutations 0.10 will optimize the search process optimal solution. The number of soft constraints applied also affects the number of soft constraint violations. And of course this will affect the quality of the resulting course timetables.

5. Acknowledgements
We would like to thanks to Universitas Komputer Indonesia for funding our research.

References
[1] Rossi-Doria O and Paechter B 2004 A memetic algorithm for University Course Timetabling Comb. Optim. 56, pp.54 – 70.
[2] Albar M A 2013 Algoritma Genetik Tabu Search Dan Memetika Pada Permasalahan Penjadwalan Kuliah Seminar Nasional Teknologi Informasi dan Multimedia 8(2), pp 45–50
[3] Jat S N and Yang S 2008 A Memetic Algorithm for the University Course Timetabling Problem 2008 20th IEEE International Conference on Tools with Artificial Intelligence 9(3), pp 427–33
[4] Poonam G 2009 Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard Algorithm International Journal of Network Security & Its Applications 1(34), pp.1-10.
[5] Pillarrichie R, Suyanto 2012 Algoritma Genetika Dengan Local Search Untuk Penjadwalan Kuliah Theses (Bandung: Telkom University), 8(2), pp. 15 – 30.
[6] Havas J, Olsson A, Persson J and Schierscher M S 2013 Modeling and optimization of university timetabling-a case study in integer programming (Göteborg), 9(3), pp.7 – 20.
[7] Chen R-M and Shih H-F 2013 Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search Algorithms 6, pp.227–44
[8] Perera M T M and Lanel G H J 2016 A Model to Optimize University Course Timetable Using Graph Coloring and Integer Linear Programming IOSR J. Math. 12, pp.13–8
[9] Simon D, Wang Y, Tiber O, Du D, Filev D and Michelini J 2017 Trajectory optimization with memetic algorithms: Time-to-torque minimization of turbocharged engines 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings 3(2), pp 983–988
[10] Islam M M, Singh H K, Ray T and Sinha A 2017 An Enhanced Memetic Algorithm for Single-Objective Bilevel Optimization Problems Evol. Comput. 25, pp.607–42