Scientific Journal of Informatics
Vol. 3, No. 2, November 2016
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Comparison Performance of Genetic Algorithm and Ant Colony Optimization in Course Scheduling Optimizing

Imam Ahmad Ashari[1], Much Aziz Muslim[2], dan Alamsyah[3]
[1,2,3] Department of Computer Science, FMIPA, Semarang State University
Email: [1]imamahmadashari@gmail.com, [2]email: a212muslim@yahoo.com, [3]email: alamsyah@mail.unnes.ac.id

Abstract

Scheduling problems at the university is a complex type of scheduling problems. The scheduling process should be carried out at every turn of the semester's. The core of the problem of scheduling courses at the university is that the number of components that need to be considered in making the schedule, some of the components was made up of students, lecturers, time and a room with due regard to the limits and certain conditions so that no collision in the schedule such as mashed room, mashed lecturer and others. To resolve a scheduling problem most appropriate technique used is the technique of optimization. Optimization techniques can give the best results desired. Metaheuristic algorithm is an algorithm that has a lot of ways to solve the problems to the very limit the optimal solution. In this paper, we use a genetic algorithm and ant colony optimization algorithm is an algorithm metaheuristic to solve the problem of course scheduling. The two algorithm will be tested and compared to get performance is the best. The algorithm was tested using data schedule courses of the university in Semarang. From the experimental results we conclude that the genetic algorithm has better performance than the ant colony optimization algorithm in solving the case of course scheduling.

Keyword: Course scheduling, Genetic algorithm, Ant colony optimization algorithm, Metaheuristic algorithm, Performance.

1. INTRODUCTION

Scheduling problems can be classified into several types, such as scheduling academic level Higher Education, Primary and Secondary Schools scheduling, exam scheduling, transport scheduling, scheduling the sale or delivery of goods and others [1].

Scheduling problems at the university is a complex type of scheduling problems. The scheduling process should be carried out in every turn of the semester, it makes this work is exhausting and time-consuming. In a scheduling problem at the university each restriction should not be violated [2].

The core of the problem of scheduling courses at the university is that the number of components that need to be considered in making the schedule, some of the components was made up of students, lecturers, time and space with due regard to the limits and certain conditions so that no collision in the schedule such as mashed room, mashed lecturer and others [3].

Metaheuristic algorithm is an algorithm that is suitable to solve the problems of scheduling courses. Metaheuristic algorithm is an algorithm that has a lot of ways to solve the problems to the boundaries of the optimal solution [4, 5]. Some examples algorithm metaheuristic widely used are: Genetic Algorithm (GA), Ant Colony Optimization (ACO), Evolutionary Programming (EP), Particle Swarm Optimization (PSO), Differential Evolution (DE), Tabu Search (TS), Biogeography based Optimization (BBO), Simulated Annealing (SA), etc. [6, 7, 8, 9, 10, 11, 12, 13].

Do a comparison between the genetic algorithm with ant colony optimization algorithm to solve a scheduling problem subjects, genetic algorithm is an evolutionary methods that solve problems using a random way. Method is a method of natural evolution inspired by natural selection causes the variation to be collected in one particular direction that shows good results so as to resemble a deliberate process optimization [14]. The process begins with a genetic algorithm to produce an artificial chromosome population, then each individual in the chromosome is evaluated, the evaluation process of individual poor of the population will be in the process with the process of recombination (crossover and mutation) to produce better individuals [15]. The algorithm ant colony optimization or ant algorithms are algorithms that originally inspired by the behavior of ants, many developed that the ant species is very sensitive to noise, most species of ants communicate between individuals and individuals, or between individuals and their environment, ant species can produce a liquid chemical called pheromone, where the fluid can leave a trail of ants pheromone trail that can be followed by other ants such as during the process of foraging

ants [16]. Ant colony optimization algorithm using transition rules and pheromone update rules as a guide to select the next trail [17].

The data set will be used to compare the two algorithms is a schedule of courses in the Department of Computer Science in odd semester of the 2015/2016 academic year. There are hard constraints and soft constraints which need to be considered in scheduling courses in the computer science department. Hard constraints include mashed space, mashed lecturer and mashed class constraints and soft as there should be no lectures on at 12.00 on Thursday, there should not be lecturing at 11:45 s.d 13:00 on Friday, subjects the practice is in the laboratory and subjects the same theory and practice should be scheduled at the same time, the theory of continued practice.

2. METHOD

2.1. Data Collection Stage

Data collection is done to obtain data needed for research. In this study, there are two data collection techniques used are as follows.

1) Interview

Interviews were conducted to obtain the data in detail. Data that will be asked is data about the process of scheduling courses. The interview process conducted by face to face with the schedule-makers courses in the Department of Computer Science, State University of Semarang. During the interview process the researchers ask questions, ask for explanations and answers to the questions, and make notes about the data that has been obtained.

2) Study Documentation

Documentation study conducted by requesting the data necessary to conduct research in the Department of Computer Science, State University of Semarang, in the form of a data file / archive about scheduling that has been made in the odd semester academic year 2015/2016.

2.2. Systems Development Method

Development of implementation for comparison testing genetic algorithm and ant colony optimization algorithm in the Department of Computer Science, Semarang State University is using the Waterfall development model. Waterfall model is divided into 4 stages are interrelated and influence. The four stages, namely a needs analysis (analysis), design (design), coding (code) and testing (test). Waterfall models shown in Figure 1.

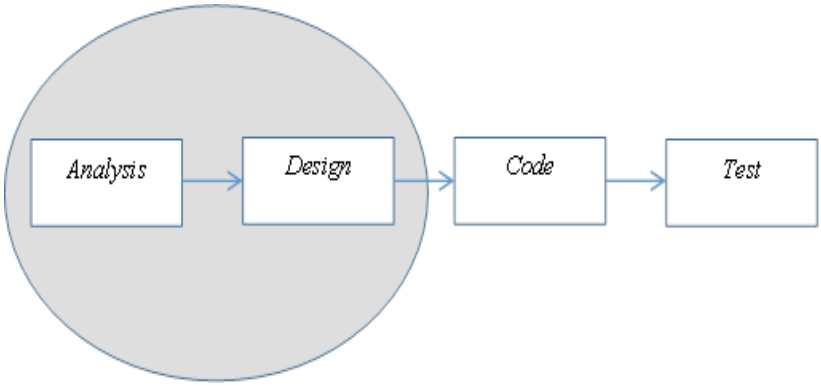


Figure 1. Waterfall model [18]

2.3. Constraint on Course Scheduling

There are hard constraints and soft constraints which need to be considered in scheduling courses in the department of computer science and genetic algorithm is as follows.

1) Hard constraints

1) Blend Lounge: There should be no lecture room used at the same time.

2) Mashed Lecturer: There should be no lecturers who teach at the same time, both lecturers and lecturers of the two.
3) Mash Class: There should not be the same class who take the classes at the same time.

2) Soft constraints

1) There should be no lectures from 10.00 s.d 13:00 on Thursday.
2) There should not be lecturing at 11.00 s.d 13:00 on Friday.
3) Adjust the lecture room, the courses are in the laboratory practices.
4) The course theory and practice of the same should be scheduled at the same time, the theory of continued practice.

3. RESULTS AND DISCUSSION

3.1. Course Scheduling With Genetic Algorithm

Steps undertaken to optimize the scheduling of subjects with a genetic algorithm is as follows:

1) Initialization Parameter

The parameters used for accomplishing the initial scheduling of subjects with genetic algorithms are: Length population, amount and probability generation crossover. Length population is used to generate random number of chromosomes which will then be calculated for each chromosome fitness value which has been raised.

2) Representation Chromosomes

Chromosome is represented by a real number of 0 to 24, with the length of each chromosome are 25 genes. Gene pool consists of three components, namely the data subjects, the data time and space. Subgen data is comprised of faculty lectures, courses and classes, subgen time data consists of days and hours of lectures and subgen space consisting of a lecture hall.

3) Evaluation

In the evaluation will be calculated fitness value of each chromosome is raised by equation (1).

$$Fitness = \frac{1}{(1+ ML + MC + MS)} \quad (1)$$

Information:
TD = Mash Lecturer
TK = Mash Class
TR = Mash Space

4) Selection

After the evaluation process by calculating the fitness value of the next process is the selection, the selection method used roulette wheel selection. In this method, each individual chromosomes to be mapped in sequence so that each individual will have the same size as the size of the fitness-owned. A random number generated and individuals who have a segment in the region of the random numbers will be selected. The process will be repeated until the best chromosome obtained.

5) Crossover

Process crossover or crossbreeding using single point crossover or crossing point. Will have two chromosomes at random and then will be crossing. Crossing will be carried out following a random position obtained.

3.2. Results Course Scheduling With Genetic Algorithm

The parameters used for an experiment genetic algorithm is as follows [17]:

- The length of each chromosome genes: 25
- Population size: 200
- The generation that is raised: 55

- Crossover Probability: 80%

The experiments were performed 10 times, the results can be seen in Table 1. To view the execution time of a genetic algorithm (Figure 2) and the memory used a genetic algorithm (Figure 3).

Table 1. The experimental results of genetic algorithm

Experiment	Time (s)	Memory (kb)	Output	
			Fitness	Generation
1	52,42	17.345,40	1	46
2	31,62	14.018,90	1	28
3	65,55	17.742,83	0.5	55
4	37,69	13.972,73	1	30
5	21,26	12.159,08	1	18
6	36,08	14.667,37	1	30
7	63,74	17.425,10	0.33	55
8	28,50	13.264,73	1	23
9	56,83	16.275,59	1	46
10	69,42	18.111,78	0.5	55

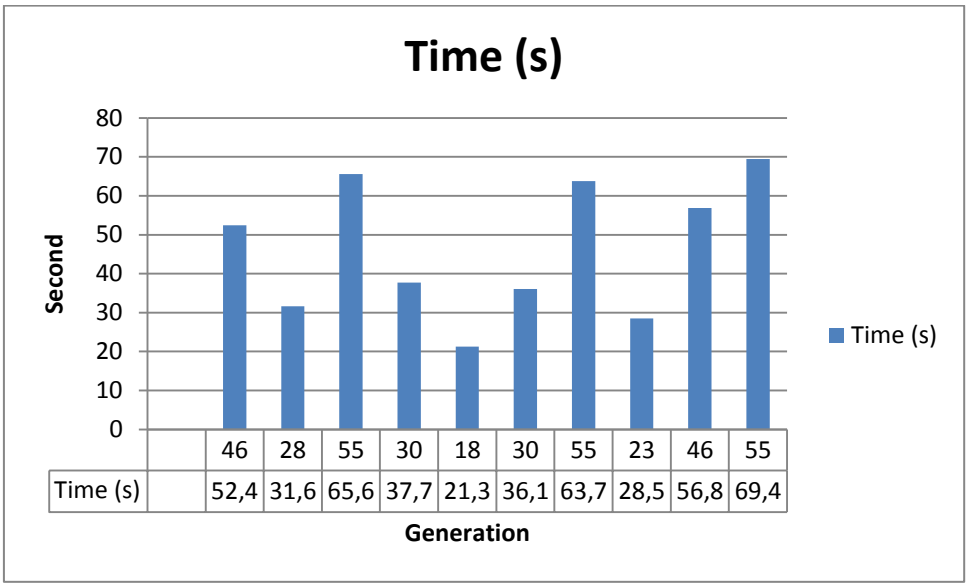
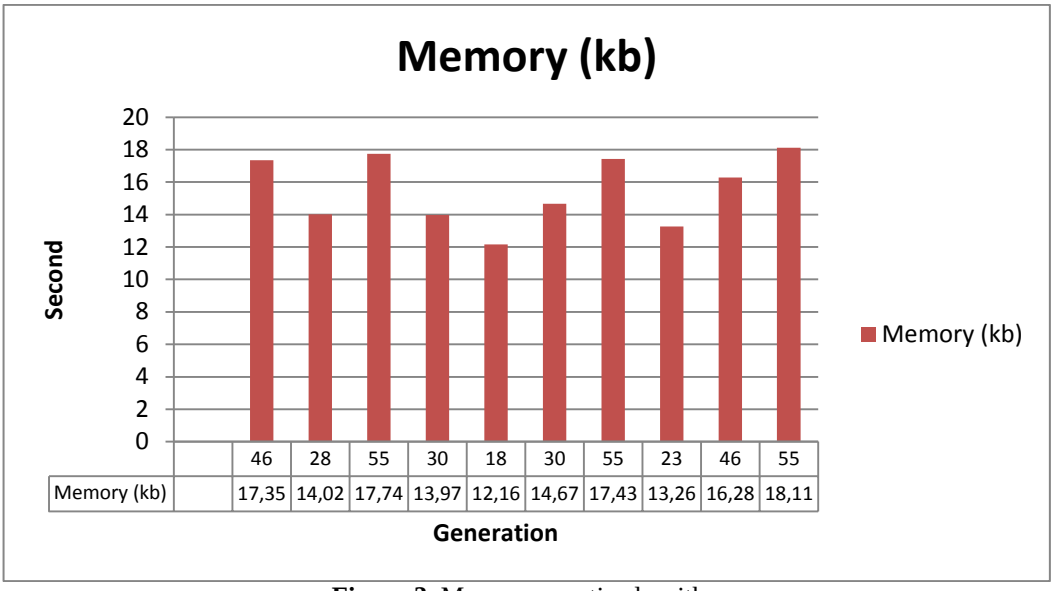


Figure 2. The execution time of genetic algorithm



	46	28	55	30	18	30	55	23	46	55
Memory (kb)	17,35	14,02	17,74	13,97	12,16	14,67	17,43	13,26	16,28	18,11

Figure 3. Memory genetic algorithm

The best performance in the genetic algorithm can an experiment to-5 is the execution time: 21,26 Second and memory used: 12.159,08 Kilo Bytes. Generation computing process stops at 18 with a fitness value = 1.

3.3. Course Scheduling Algorithm Ant Colony Optimization

Steps undertaken to optimize the scheduling of courses with ant colony optimization algorithm is as follows:

1) Initialization parameters

The parameters that is used for resolving scheduling subjects with algorithms ant colony optimization ism (number of cycles maximum), NC max (jumlah siklus maksimum), Q (constant cycle of ants), τ_ij (price Intensity footprints of ants) and ρ (constant evaporation trail of ants). m (Number of ants) used to generate random number of solutions m (Number of ants) which will then be evaluated to get the value Δτ _ij whereas forNC max is used to determine how many cycles of ants that will be raised to get the optimal solution.

2) Finding constraint each taboo list ($\Delta\tau_{ij}$)

Constraint each taboo list ($\Delta\tau_{ij}$) is calculated from the constraints obtained by each ant with the following formula by equation (2).

$$\Delta\tau_{ij} = \frac{Q}{(1+ \text{ML} + \text{MC} + \text{MS})} \quad (2)$$

Information:

ML = Mashed Lecturer MC = Mashed Class MS = Mashed Space

The fewer ants with a number of constraints then most likely the ants would be elected, it is best constrained if $\Delta\tau_{ij}$ = 1000.

3) Menghitung perubahan harga intensitas jejak kaki semut (*Update Pheromone*)

To calculate the price changes in the intensity of the ant legs for the next cycle using the following formula by equation (3).

$$\tau_{ij} = \rho \, . \, \tau_{ij} + \Delta\tau_{ij} \quad (3)$$

A shared value m (ant) will be replaced with new ones, the replacement will be done at random.

3.4. Scheduling Results Subjects With Ant Colony Optimization Algorithm

The parameters used for the experiment ant colony optimization algorithm is as follows [22]:
The number of ants (m) : 250

- Cycle: 1600
- τ_ij: 0.1
- ρ: 0.01
- Q: 1000
- $\Delta\tau_{ij}$: 1000

The experiments were performed 10 times, the results can be seen in Table 2. To view the execution time ant colony optimization algorithm (Figure 4) and the memory used ant colony optimization algorithm (Figure 5).

Table 2. Results of the experiments ant colony optimization algorithm

Experiment	Time (s)	Memory (kb)	Output	
			$\Delta\tau_{ij}$	Cycle
1	100,15	31.213,65	333.33	160
2	70,48	22.569,03	1000	64
3	95,46	31.189,01	333.33	160
4	92,45	31.165,32	500	160
5	99,18	31.246,43	500	160
6	69,11	21.674,48	1000	59
7	99,37	37.518,46	125	160
8	93,41	37.481,14	125	160
9	93,68	37.371,07	166.66	160
10	96,27	37.527,61	166.66	160

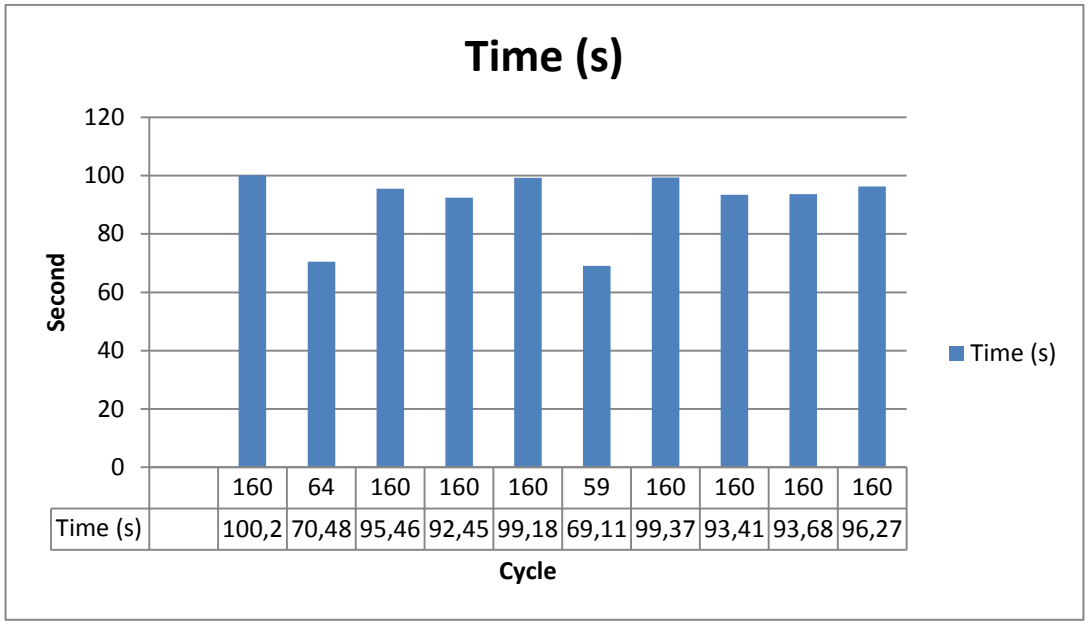


Figure 4. The execution time ant colony optimization algorithm

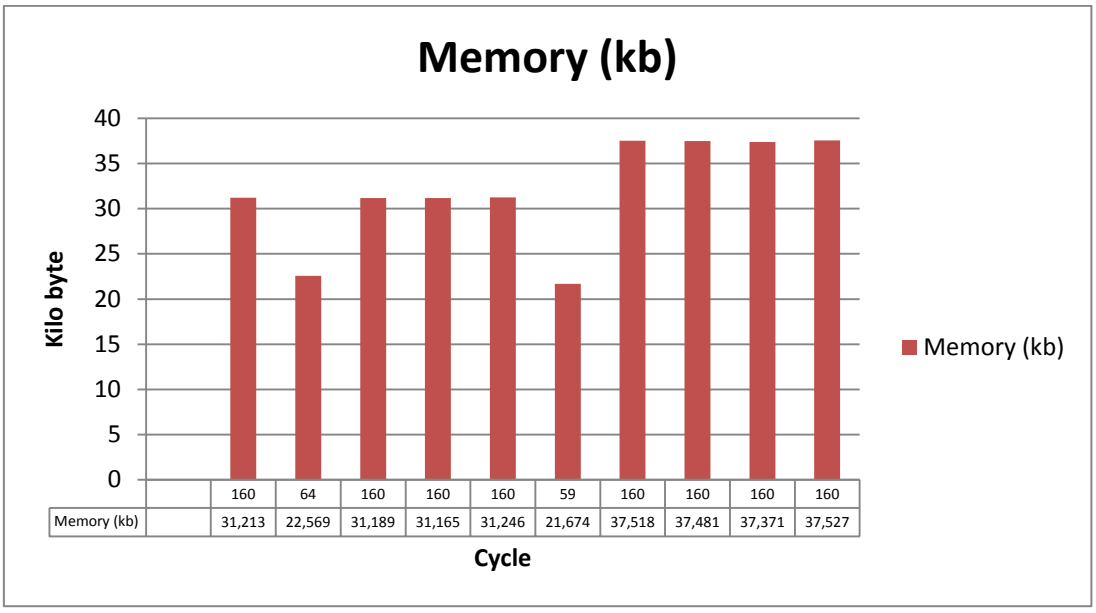


Figure 5. Memory ant colony optimization algorithm

The best performance in the ant colony optimization algorithm can an experiment 6th with time computing: 69,11 Second and memory used: 21.674,48 Kilo bytes. Computational process stops at 59 cycles with a value of $\Delta\tau_{ij}$ = 1000.

3.5. Comparison of the performance of the genetic algorithm and ant colony optimization algorithm

From the experimental results can be the best performance of the genetic algorithm and ant colony optimization algorithm can be seen in Table 3. Graph comparing the execution time can be seen in Figure 6 graph comparison of memory used in Figure 7.

Table 3. Best Performance genetic algorithm and ant colony optimization

Algorithm	Time (s)	Memory (kb)
Genetic	21,26	12.159,08
Ant Colony Optimization	69,11	21.674,48

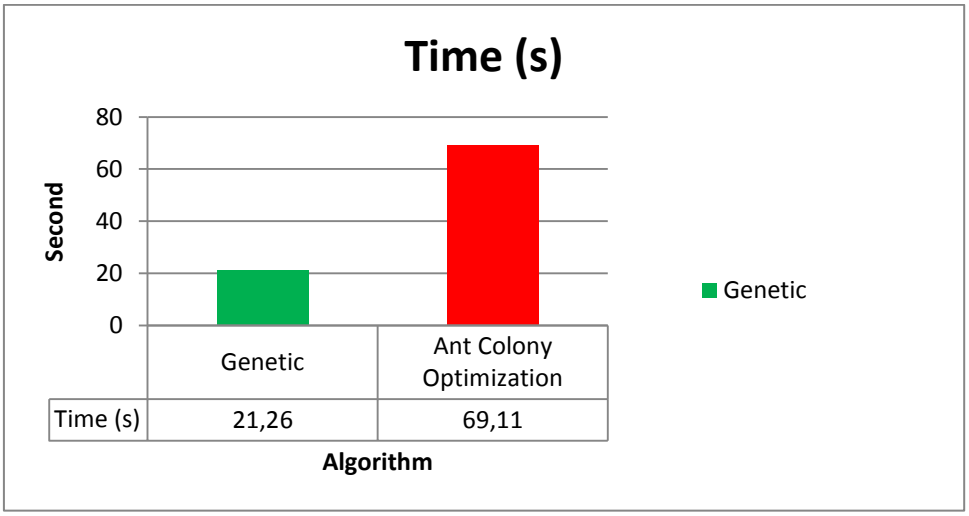


Figure 6. Chart of comparison of the time

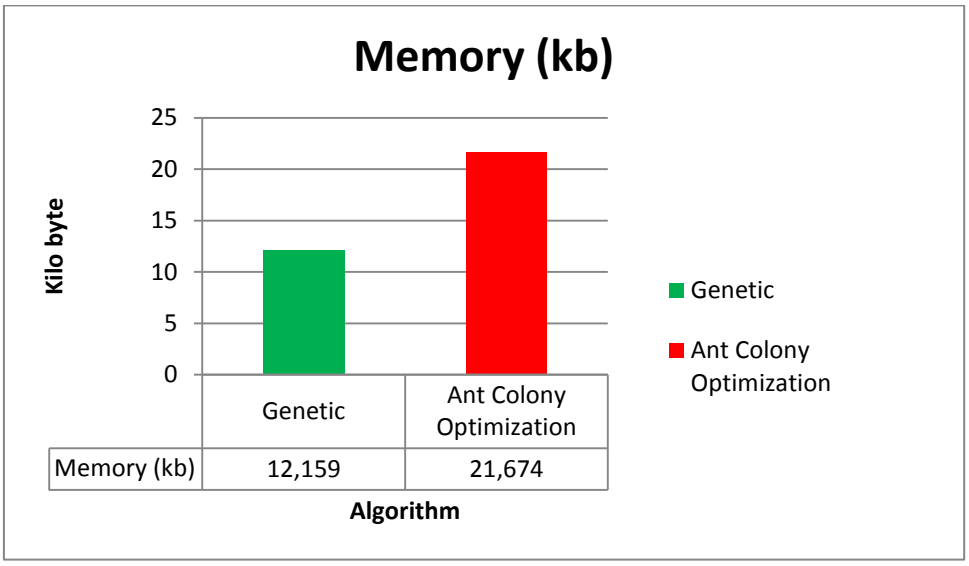


Figure 7. Chart comparison memory

From a number of attempts in this study proved that genetic algorithm has better performance than the ant colony optimization algorithm in case of optimization scheduling courses in the computer science department at the Semarang State University of with computational time: 21,26 Second and memory used: 12.159,08 Kilo bytes.

In performance testing system algorithms, the system is only implemented for memperoses scheduling courses in the semester were used. Required process add, edit and delete if will process the schedule of courses for different semesters.

4. CONCLUSION

The experimental results showed that the genetic algorithm has a better performance of the ant colony optimization algorithm. Genetic algorithm is suitable to solve scheduling msalah subjects. Various genetic operators such as selection and crossover can improve results in various aspects such as through the selection we can select the best chromosome fitness function on the basis of an chromosomes and also through the exchange of information crossover can schedule as needed.

5. REFERENCE

[1] Nugraha, D., & Kosala, R. 2014. A Comparative Study of Evolutionary Algorithms for School Scheduling Problem. *Journal Of Theoretical & Applied Information Technology*, 67(3).

[2] Babaei, H., Karimpour, J., & Hadidi, A. 2015. A survey of Approaches for University Course Timetabling Problem. *Computers & Industrial Engineering*, 86: 43-59.

[3] Alamsyah, Wardoyo, R. 2004. Optimalisasi Penjadwalan Multi Constraint menggunakan Logika Fuzzy = Multi Constraint Scheduling Optimization Using Fuzzy Logic. *Sains dan Sibernatika*, 17.

[4] Reddy, S. S., & Bijwe, P. R. 2016. Efficiency Improvements in Meta-heuristic Algorithms to Solve the Optimal Power Flow Problem. *International Journal of Electrical Power & Energy Systems*, 82, 288-302.

[5] Deb, K. (2001). *Multi-objective Optimization using Evolutionary Algorithms (Vol. 16)*. John Wiley & Sons.

[6] Osman, M. S., Abo-Sinna, M. A., & Mousa, A. A. 2004. A Solution to the Optimal Power Flow Using Genetic Algorithm. *Applied mathematics and computation*, 155(2), 391-405.

[7] Tiwari, P. K., & Vidyarthi, D. P. 2016. Improved Auto Control Ant Colony Optimization using Lazy Ant Approach for Grid Scheduling Problem. *Future Generation Computer Systems*, 60: 78-89.

[8] Yuryevich, J., & Wong, K. P. 1999. Evolutionary Programming Based Optimal Power Flow Algorithm. *IEEE Transactions on Power Systems*, 14(4), 1245-1250.

[9] Abido, M. A. 2002. Optimal Power Flow using Particle Swarm Optimization. *International Journal of Electrical Power & Energy Systems*, 24(7), 563-571.

[10] El Ela, A. A., Abido, M. A., & Spea, S. R. 2010. Optimal Power Flow using Differential Evolution Algorithm. *Electric Power Systems Research*, 80(7), 878-885.

[11] Abido, M. A. 2002. Optimal Power Flow using Tabu Search Algorithm. *Electric Power Components and Systems*, 30(5), 469-483.

[12] Bhattacharya, A., & Chattopadhyay, P. K. 2011. Application of Biogeography-based Optimisation to Solve Different Optimal Power Flow Problems. *IET generation, transmission & distribution*, 5(1), 70-80.

[13] Roa-Sepulveda, C. A., & Pavez-Lazo, B. J. 2003. A Solution to the Optimal Power Flow using Simulated Annealing. *International journal of electrical power & energy systems*, 25(1), 47-57.

[14] Squillero, G., & Tonda, A. 2016. Divergence of Character and Premature Convergence: A Survey of Methodologies for Promoting Diversity in Evolutionary Optimization. *Information Sciences*, 329: 782-799.

[15] Hoseini, P., & Shayesteh, M. G. 2013. Efficient Contrast Enhancement of Images using Hybrid Ant Colony Optimisation, Genetic Algorithm, and Simulated Annealing. *Digital Signal Processing*, 23(3), 879-893.

[16] Jiang, Z., Zhou, M., Tong, J., Jiang, H., Yang, Y., Wang, A., & You, Z. 2015. Comparing An Ant Colony Algorithm with a Genetic Algorithm for Replugging Tour Planning of Seedling Transplanter. *Computers and Electronics in Agriculture*, 113, 225-233.

[17] Aguilar-Rivera, R., Valenzuela-Rendón, M., & Rodríguez-Ortiz, J. J. 2015. Genetic Algorithms and Darwinian Approaches in Financial Applications: A survey. *Expert Systems with Applications*, 42(21), 7684-7697.

[18] Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.

[19] Hong, S. S., Lee, W., & Han, M. M. 2015. The Feature Selection Method based on Genetic Algorithm for Efficient of Text Clustering and Text Classification. *International Journal of Advances in Soft Computing & Its Applications*, 7(1).

[20] Sitarz, P., & Powałka, B. 2016. Modal Parameters Estimation using Ant Colony Optimisation Algorithm. *Mechanical Systems and Signal Processing*, 76, 531-554.