Design of Rescheduling of Lecturing, using Genetics-Ant Colony Optimization Algorithm

C F Palembang

Magister Sistem Informasi, Universitas Komputer Indonesia, Indonesia

citra_fathia82@yahoo.co.id

Abstract. The purpose of this research is to produce a new scheduling system by minimizing impact and error rate of a new schedule. The algorithm used in this research is the Genetics-Ant Colony algorithm combination with research data taken at the University Computer Indonesia (UNIKOM) and using PHP programming language. Results obtained from this research is the creation of a new scheduling system with success rate above >50% without error and clash. Based on the results of this research, it is expected to contribute the management of scheduling system at a university in finding solutions to find replacement lecturers automatically if there is absence of lecturers who are unable to attend without disturbing other schedules.

1. Introduction

Course Schedule is a list that contains name of the courses, lecturer so each course, time, the classroom and so forth. Build a Course Schedule for a large study program especially for a university is indeed a complex job. Arrangement of schedule often clashing either clashing against lecturer schedule or clashing lecture room, so required requirement to overcome [1]. Course scheduling can be described as containing the following factors: courses offered, teacher expertise and preferences, minimum teaching hours required by the school, the limit of overtime hours, avoiding conflict in teacher schedules, availability of classrooms (laboratories), and the continuity of each course should be taken into account [2]. An admissible schedule will have to satisfy a set of hard and soft constraints imposed on jobs and resources. So, a scheduling problems can be seen as a decision making process for operations starting and resources to be used.

Unexpected problem arises when the course has been scheduled, The lecturer who is suddenly unable to attend in the middle of the semester runs for a long time, make the schedule disturbed, so it should be rescheduled soon in order not to interfere with other schedules. This research is intended to provide solutions to design a re-scheduling system by combining Genetics Algorithm (GA) and Ant Colony Optimization (ACO) at the Indonesian Computer University (UNIKOM) Bandung academic year 2015/2016 with success rate above >50% without error and clash.

In a previous study, a new algorithm which is called Hybrid GA-Bees was proposed [3]. The authors in [3] have tested the Hybrid GA-Bees algorithm to solve a university’s course timetabling problem, and concluded that it produced in a better schedule, the genetic algorithm has better performance than the ant colony optimization algorithm in solving the case of course scheduling [4]. The purpose of this research is to produce a new scheduling system by minimizing impact and error rate of a new schedule. The algorithm used in this research is the Genetics-Ant Colony algorithm combination with research data taken at the University Computer Indonesia (UNIKOM) and using PHP programming language.

2. Experimental method

For testing GA and ACO in the Department of Communication Science, Indonesian Computer 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.

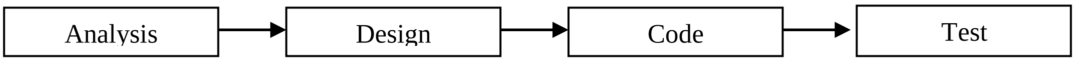

Figure 1. Waterfall model [4].

Such as previous research [4] 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:

- Hard Constraints
  - There should be no lecture room used at the same time
  - There should be no lecturers who teach at the same time
  - There should not be the same class who take the classes at the same time
- Soft Constraints
  - There should be no clash with the lecturers' time preference
  - There should be no clash with space usage time preference

2.1. Genetic algorithm

Genetic Algorithm (GA) is a computational method designed to simulate the evolution processes and natural selection in organism [5, 6], which follows the sequence as generating the initial population, evaluation, selection, crossover, mutation, and regeneration [5, 7]. The initial population becomes important as this represents the solution and it is normally randomly generated. Using a problem specific function, the population is then evaluated. GA will select some of them, based on certain probability, that will mate in the next process. Crossover and Mutation will be performed to them to get a new and better ones. The idea of GA is that the new generation of solution should be better than the previous one. This process is repeated until some stopping criteria are reached [5].

In addition, this algorithm is possible to find the global optimum and well adapted to the problem [5, 8]. On the other hand, GA is easy to fall into premature convergence that makes the best solution difficult to be achieved; and GA also needs longer processing time for problem with large data are considered as its disadvantages [5, 8, 9].

2.2. Ant colony optimization

Ant Colony Optimization (ACO) is a computational method that is inspired from the way of ant colony seeking the shortest path from the food resource to the nest without visual aid [10]. In their searching, ants deposit a certain amount of pheromone while walking to form a line and communicate with other ants. Those that could not smell the pheromone, they keep travelling at random route. The pheromones of certain path is enhancing when more ants are attractively tracking on it to obtain the shortest one. [5] The ant algorithm is started by spreading out the ants randomly to every city to be set as the initial city for the respective ant. Such ant will select the next city based on certain probability. This probability is a function of pheromone matrix, distance matrix, and parameters. This selection is repeated until each ant visited every city one time. That is the first cycle in the algorithm, and the cycle is carried on until reaching the stopping criteria. In each cycle the overall route is changing methodically as the pheromone matrix is updated.

2.3. Genetic algorithm-ant colony optimization

Some previous researchers who have combined GA and ACO are [11] Proposed to hybrid GA and ACO as a new algorithm to solve TSP, [12] GA, in such work, is benefited to initiate the matrix of pheromone in ACO and to recombine the route from ACO. They claimed the hybrid algorithm is more effective compare to the GA and ACO, [13] the author claimed this hybridization will simplify in selecting the adjustable parameter in ACO where human experience is needed and in most cases depending on coincidence. Jin-Rong et al. developed two sub-algorithms based on GA and ACO hybridization that is covering up the weaknesses both algorithms, [14] ACO is employed to help GA to eliminate the appearance of invalid tour, while GA is used to overcome the dependency on the matrix of pheromone in ACO. Al-Salami developed a combination of GA and ACO by incorporating each basic method as a sub-solution generator and followed by selecting the better sub-solution as a new population in the next iteration.

The main idea of this paper is to define an appropriate approach to hybridize GA and ACO to find course scheduling solution by which constructs the concept combining certain steps in GA and ACO to perform GACO. Hybridization is also applied to some parameters and variables of GA or ACO that share same characteristics in the computation, i.e. population size in GA and number of ants in ACO, number of generations in GA and number of cycles in ACO, and chromosome in GA and in ACO. The proposed technique in this paper is introducing the evolution steps of GA into the computation step of ACO as shown in the Figure 2.

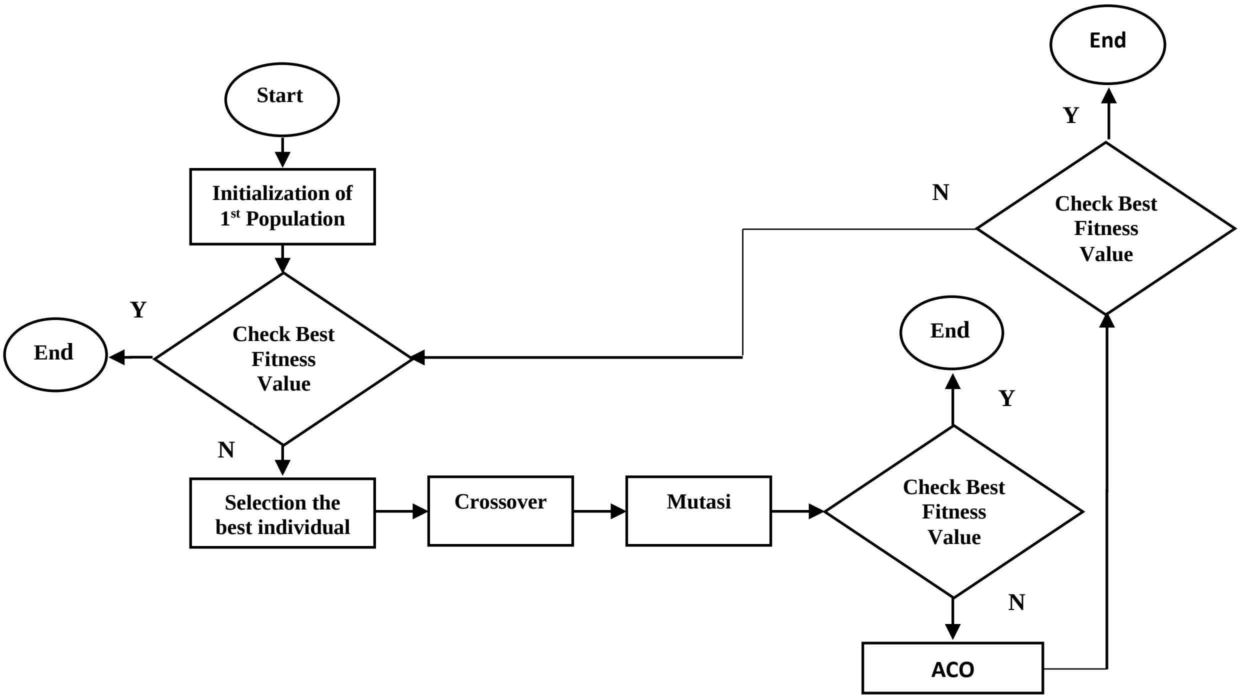

Figure 2. GA-ACO algorithm.

The following is the procedure for completing the steps of combining genetic algorithms and ant colonies, namely:

- Genetic Algorithm :
  a. Initialization
  b. Check all populations and the best individuals (the best fitness value)
  c. Crossover
     When the probability value is less than the exchange probability value then the schedule is exchanged.
  d. Mutations

Determine which courses and classes to be randomized or replaced, Change the time, Change the day, changing space, changing lecturers time.
After it is checked again, the best candidate from the mutation if the hard and soft constrain is fulfilled then finished. However, if the hard constrain is fulfilled but soft constrain is not met, it will be continued with ACO.

- Ant Colony Optimization :
  a. Generating a neighborhood route from a solution formed by genetic algorithms
  b. The value of eta (heuristic value) is worth 1 if feasible, is 0 if (violates hard constrain), but the inclusion will not occur clashed then the value given = 1
  c. Initial TA value (Parameter Pheromone)
  Each iteration evaporates (evaporation), if TAU initial value will be added 5% then formed pheromone trail of solution that has been made by previous genetic algorithm.
  d. Calculate the probability value of Pij. The probabilities that are in node i take route to j.
  i = Class
  j = Time, Space, Lecturer
  e. After it is formed its individual and remove the new fitness value, then checked back to the previous stage whether the hard and soft constrain it fulfilled or not. If fulfilled then get the solution, but if not met it will be the process of search again from the beginning to the next genetic generation.

3. Results and discussion

To determine the parameters of the GA-ACO used for data processing, first experiments on parameters to see the best combination of parameters. When the program is run by changing the parameters, it will get different outputs, this is where the program is verified.

Experiment 1:
The first experiment will be tested the method of GA-ACO to 2 substitute lecturers where lecturer A taught 3 courses and lecturer B teaching 1 course.

From the search results of the replacement schedule, found four different substitute lecturers for each course without disturbing the old schedule that has been there before. The substitute lecturers are C, F, K and L.

Which can be illustrated in the following Table 1 and Table 2:

- Previous Schedule

Table 1. Previous schedule experiment 1.

Lecturer	Time	M	T	W	TH	F	S
A	**09.15-11.30**		✓			✓	
A	**12.15-13.45**						✓
B	**13.00-15.15**				✓		
A	**13.45-16.00**					✓	

- Replacement Schedule

Table 2. Previous schedule experiment 1.

Lecturer	Time	M	T	W	TH	F	S
C	**07.00-09.15**					✓	
F	**09.15-11.30**	✓					
K	**15.15-17.30**				✓		
L	**16.00-17.30**					✓	

The results obtained on 2 replaced lecturers are 4 substitute lecturers with each lecturer teaching 1 course (class) in different time and space from 5 times iteration with 100% success rate without clash.

Experiment 2:
Rescheduling 2 lecturers who will be replaced with limited time replacement lecturer can only be scheduled in the afternoon starting at 15:15. The lecturer replaced in this case are X and Z, which has two courses with different classes. The substitute lecturers are MX1, Q X2, SZ1 and WZ2.

Which can be illustrated in the following Table 3 and Table 4:

- Previous Schedule

Table 3. Previous Schedule experiment 1.

Lec-turer	Time	M	T	W	TH	F	S
Z1	**09.15-10.45**					✓	
X1	**10.00-12.15**		✓				
X2	**13.45-16.00**						✓
Z2	**15.15-17.30**			✓			

- Replacement Schedule

Table 4. Previous Schedule experiment 1.

Lect-urer	Time	M	T	W	TH	F	S
MX1	**13.00-15.15**				✓		
QX1	**15.15-17.30**		✓				
S Z1					✓		
WZ2	**16.00-17.30**			✓			

The results obtained in 2 lecturers replaced and started at 15.15-17.30 are 4 substitute lecturers with each lecturer teaching 1 course (class) in different time and space, from 10 times iteration with 80% success rate. Which one at a time does not match the request that can be viewed in table 2 which is bold in red. From two experiments, it can be concluded that the combination of GA-ACO can produce better performance than just using genetic algorithms or ant colonies.

4. Conclusion

In this paper, we describe the combined algorithm using GA and ACO for the rescheduling problem in the middle of the running semester. The advantages of GA and ACO are that they can reduce clashes and minimize errors in substitute lecturers' rescue mid-term. To test the feasibility of the combined GA and ACO, three different examples of rescheduling problems are applied, the results show that our new combined algorithm can provide a new schedule without disrupting the old schedule with a small error rate and few clashes.

For later researchers to explore again to increase the number of searches on a large scale at universities from various majors.

References

[1] Wang Y Z 2002 "An application of genetic algorithm methods for teacher assignment problems," *Expert System with Application* **22** 295-302.

[2] Ana M, Carlos R and Silvio D C M S 2003 "Using genetic algorithms for dynamic scheduling," *In 114th Annual Production and Operations Management Society Conference (POMS)* **3**(2) 16-18.

[3] Nguyen B P, Nguyen T T M K and Tran T H N 2011 "A New Hybrid GA-Bees Algorithm for a Real-world University Timetabling Problem," *Proceedings of the International Conference on Intelligent Computation and Bio-Medical Instrumentation (ICBMI)* 321-326.

[4] Imam A A, Much A M and Alamsyah 2016 "Comparison Performance of Genetic Algorithm and Ant Colony Optimization in Course Scheduling Optimizing," *Scientific Journal of Informatics* **3** 29-39.

[5] Zainudin Z and Irving V P 2013 "A Hybrid Optimization Algorithm based on Genetic Algorithm and Ant Colony Optimization," *International Journal of Artificial Intelligence & Applications (IJAIA)* **4** (5) 167-189.

[6] Raza Z, Vidyarthi D P 2009 "A computional grid scheduling model to minimize turnaround using modified GA," *Int. J. of Artificial Intelligence* **3** **(**A9) 86-106.

[7] Banga V K, Singh and Kumar R. 2007 “Simulation of Robotic Arm using Genetic Algorithm & AHP,” *J. Engineering and Technology* **25** 95-101.

[8] Qing-Bao and Shuyan C 2007 “A new ant evolution algorithm to resolve TSP problem,” *In IEEE Proc. Int. Conf. Machine Learning and Applications* 9 210-224.

[9] Yingzi W, Yulan H and Kanfeng G. 2007 “Parallel Search Strategies for TSP’s using a Greedy Genetic Algorithm,” *In IEEE Proc. Int. Conf. Natural Computation* **5** 78-99.

[10] Dorigo M and Colorni A 1996 “The Ant System: Optimization by a colony of cooperating agents,” *IEEE J. Systems, Man, and Cybernetics* **26 (**1) l-13.

[11] Shang G, Xinzi J and Kezong T 2007 “Hybrid Algorithm Combining Ant Colony Optimization Algorithm with Genetic Algorithm,” *In Proc. of the 26th Chinese Control Conf.* **2** 68-77.

[12] Duan H and Yu X 2007 “Hybrid Ant Colony Optimization Using Memetic Algorithm for Traveling Salesman Problem,” *In Proc. IEEE Int. Symposium. Dynamic Programming and Reinforcement Learning* **6** 28-37.

[13] Jin-rong X, Yun L, Hai-tao L and Pan L 2008 “Hybrid Genetic Ant Colony Algorithm for Traveling Salesman Problem,” *J. Computer Applications* **28** (8) 2084-2112.

[14] Al-Salami N M A 2009 “Ant Colony Optimization Algorithm,” *J. UbiCC* **4** (3) 823-826.