System Application of Genetic Algorithm for Scheduling Optimization Study Using Java

(Case Study : Department of Computer System UNTAN)

Fatma Agus Setyaningsih[1]

Abstract—***Scheduling of lecture in a university is a college activities into the work place and time that there is room in such a way so as to minimize the violation of the scheduling problems of a lecture. Problems that are often referred to as the University Timetabling Problem (UTP) is, requires a lot of consideration include the number of students, number of teachers who are not proportional to the number of courses, the amount of space used, as well as lecture predetermined time. Scheduling lectures with the automation system is very important because it can save hours of work and provide optimum solutions in a short time, which can increase productivity, the quality of teaching and learning, and quality of service. One method that can be used to complete the course scheduling problem is a genetic algorithm approach. Genetic algorithm is an optimization tool that models the principles of evolution. Genetic algorithm is able to find a globally optimum solution in the search space is very complex. By using an initial population of solutions is encoded and selected based on its quality, and then used to create a new population using crossbreeding and mutation process of the initial individuals. Evaluation function is used to calculate the hard constraints and soft constraints can be met.This study discusses the scheduling college located in Department of Computer Systems UNTAN. The purpose of study is to optimize the college scheduling and to make easier the scheduling job with make a college scheduling softwar.***

Keywords—College scheduling, Schedule, Optimization, Genetic Algorithms.

I. INTRODUCTION

Arrangement of a lecture schedule may look easy, but when existing resources become more limited resulting a long time in process. Arrangement of schedule activities related to the various requirements that must be fulfilled so that requires a lot of consideration to support the implementation of these activities.

Schedule of lectures at a university is a complex thing. The problems are often referred to as the University Timetabling Problem (UTP), requires a lot of consideration include the number of students and teachers who are not proportional to the number of courses, the amount of classroom, as well as lectures predetermined time. Some resources are typically needed as the availability of the room, whether the room has sufficient criteria to hold a class, number of seats, and other support needs, such as laboratory room or hall room. Teachers problems are "what teachers are able to teach at all lecture time and what teachers also taught in other classes at the same time". The next problem is the class participants, whether they are also able to attend the class, or they may have other schedules at the same time.

The making of an application's scheduling lecture is expected to facilitate the user in managing the class schedules in accordance with the resources and constraints. This application is also expected to facilitate communication between the teacher and the class participants in the class schedule.

[1]Fatma Agus Setyaningsih is with Department of Computer System, University of Tanjungpura, Pontianak, Indonesia. E-mail: fatmasetyaningsih@gmail.com.

II. METHOD

A. Genetic Algorithms

Genetic algorithm is a computational algorithm that inspired Darwin's theory of evolution which states that the survival of an organism is affected rule that individuals high fitness value will survive while individuals low fitness value will die [1]. Darwin also stated that the survival of a creature can be maintained through the process of reproduction, crossover, and mutation. Of the theory is then adopted a computational algorithm to find a solution to an optimization problem, which is to get a value of an optimal solution to a problem that has many possible solutions. Genetic algorithm appeal lies in its simplicity and the ability to find a good and fast solution to a complex problem. Genetic algorithms are very useful and efficient for problems with the following characteristic [2]:

a. Problem scope is very large, complicated and difficult to understand.
b. Less or even no knowledge sufficient to represent the problem specifically.
c. Unavailability of adequate mathematical analysis.
d. When conventional models are not able to resolve the problems encountered.
e. The solution is not expected to be the most optimal, but quite "good" or acceptable.
f. There is a time limit, for example, in real-time systems or real-time systems.

Genetic algorithms are started from the set of randomly generated solutions called population. While each individual in the population is called chromosome which is a representation of each solution and evaluated the level of robustness (fitness) by a predetermined function. Through the process of natural selection on genetic operators, the genes of the two chromosomes (parent) is expected to produce a new chromosome with a higher fitness level as a new generation or progeny (offspring) next. Chromosomes will be recomputed

called generation (generation). A teach generation, the chromosomes are evaluated by a fitness function [3].

After several generations of the genetic algorithm will converge to the best chromosome, which hopefully is the optimal solution [1].

The main components of the genetic algorithm, to implement a genetic algorithm includes 6 main components present in the genetic algorithm [4], namely:

1. Coding techniques
   Coding is a technique to express the initial population as a potential solution to a problem in a chromosome as a key issue when using genetic algorithms [5]. This coding technique includes coding genes and chromosomes. The gene is part of a chromosome that can be represented in the form string bit, trees, array estate numbers, list of rules, elements permutation, program elements, or other representation that can be implemented for genetic operators.
2. Initialization procedure
   Initialization procedure (generating initial population) is randomly generating a number of individuals or through certain procedures. Population size depends on the problem to be solved and the type of genetic operators are implemented. Once the population size is determined, then the initialization must be performed on the chromosomes present in the population. Chromosome initialization is done randomly, however, must still consider the solution domain and the constraints of the problem.
3. Function Evaluation
   There are 2 things that must be done to evaluate chromosomes, namely: evaluation objective function (objective function) and the conversion function objective into fitness function. In general, the fitness function derived from objective function with no negative values. If it turns out objective function has a negative value, it is necessary to add a constant C so that the fitness value becomes negative form.
4. Selection
   This selection aims to provide greater opportunities for reproduction of the most fit members of the population.
5. Genetic operators
   There are two genetic operators, namely:
   a. Operator to perform recombination, which consists of: binary valued recombination (crossover).
   b. Mutation, mutation binary value.
      Mutation is done by giving gaussian noise to one or more genes which is a real number. Initially, selecting genes from a random case and then the value is given different values to the previous values. Mutation is done to expand the search area so that cases can achieve optimal performance value.
6. Determination of parameters.
   The parameter is the genetic algorithm control parameters, namely population size (pop size), chances of crossover (Pc), and the chances of mutation (Pm). The parameter values are based on the value of the ability to be used for selecting individuals.
   Each individual who marked will be evaluated for its ability to look for value. This value is used to make the selection of individuals in a population in which individuals who value his ability does not meet the requirements will be removed from the population.

B. Schedule Arrangement by Genetic Algorithms Method

1. Determination of hard constraints and soft constraints.
2. Chromosome representation/candidate solutions encoded using integers into the matrix of the lecture time slot.
3. Initial Population initialization is set/collection of candidate solutions/chromosomal form of lecture time slot matrix.
4. Determination of Fitness Functions to calculate any violations of hard constraints and soft constrains on potential solutions will increase by 1.

$$\text{Fitness Funtion} = \frac{1}{1 + \sum_{i=1}^{n} \omega_i c_i(p)} \qquad (1)$$

   The resulting value indicates how optimal solutions are obtained, in other words in the scheduling of the smaller number of violations lectures produced the resulting solution, the better.
   There are three process inputs, four main processes, the two branches, and the output issued in this application.
5. Selection.
   Selection is done on the evaluation of the fitness value by using roulette wheel selection. As a result, fitness has a good quality of most chromosomes have likely selected in the generation or next iteration greater.
6. Crossover randomly determined one gene or index of each of the two chromosomes or prospective parent solutions, then do an exchange only on gen or course schedule index.
7. Mutation is done by selecting a random chromosome or solution is the best candidate, determine the two genes or chromosomes index of or potential solutions and then change the bit that is in the genes or index.
8. Condition is complete, if after a few generations of iterations in a row the best fitness value does not change back. Thus, taken chromosome or potential solutions with the best fitness value as a solution and put it into a matrix of time slots to establish a schedule of lectures.
9. Manufacture Lecture Schedule.
10. Final Analysis

C. Application of Theory

To simplify the structure of chromosomes in a genetic algorithm, chromosome genes is structured as follows:

1. Each gene will contain components: faculty, schedule. For instance (D, J) →Dis a lecturer, J is the schedule.
2. Subgen consists of lecturers: lecturers, course, class.
3. Subgen schedule consists of components: day, time, space. Each of correlation lecturer with the schedule given presence degrees with scale 1-9. Degree 1

means that lecturers cannot attend, 9 degrees means professors can present with certainty on the schedule.

4. Value presence degrees of correlation-schedule lecturers contribute to the fitness value of the chromosome. For example, the number of genes in a chromosome there are 10, then the desired fitness value is 9*10 = 90, this means that all teachers can attend for sure at any given schedule. Lowest fitness value is 9 * 1 = 9, which means that all teachers cannot attend a lecture given on schedule.

Here are some steps for the implementation of genetic algorithms scheduling in Department of communications System UNTAN:

1. One teacher can teach any subject any class (A, B, C).
2. College days are Monday to Saturday. With hours of lecture starting at 7:00 until 16:00 (no slight modifications at college).
3. One credit course duration 1 hour.
4. There are 4 rooms available.
5. The lecturer should not be on the schedule for the two time/same hour.
6. Not associated with taking courses by students who normally varies.
7. For a class schedule, a teacher may not be able to attend so that the schedule should be avoided, prioritized on a schedule that lecturers can certainly present.

III. Results and Discussion

Implementation in detail the course scheduling application system is:

1. Lecture Room
   There are 2 classrooms used: A1, A2 (case in point of fact there are 4 lecture halls).Figure 2 is the main menu which will display the course scheduling system.
2. Subjects of teaching lecturer.
   Process scheduling, it should take care about the subjects given to lecturers. In this subject comprise as some of the names of courses and majors that will refer to several class, the number of credits and preference quota of teaching by a lecturer.
3. Data of Lecturer
   Lecturer at the data contains a list of lecturer of subjects relating to hours and the room will be used for lectures.
4. Scheduling Process
   Assumed 3 credits each course, so there are 1's class schedule each space, namely: 07:00 to 10:00, 10:00 to 13:00, and 13:00 to 4:00 p.m. (real case every 1 Credit lasted 60 minutes).
   The table 1 is only Monday and Tuesday alone, the actual schedule until Saturday so the code up to 24. The following table scheduling system using a genetic algorithm.
5. Chromosome Structure
   Chromosome structure be simple but covers all components. Forming genes chromosome consists of two components: teaching lecturer and schedules. Number of genes needed for a chromosome will be equal to the number of combinations between lecturer, subjects, and classes. In the example above table there are 12 combinations, then to a chromosome will consist of 12 genes. So that a chromosome describes the scheduling conditions that occur. For example: if there are 12 combinations as above, a chromosome can be formed with the pattern {(D1,J1), (D2,J2), (D3,J3), (D4,J4), (D5,J5), (D6,J6), (D7,J7), (D8,J8), (D9,J9) , (D10,J10), (D11, J11), (D12, J12)}
   Making random chromosome to subgen schedule, while lecturers do not teach.
   For example:
   {(1,10), (2,3), (3,9), (4,18), (5,20), (6,24), (7,4), (8,1), (9,15), (10,12), (11,17), (12,19)}
   Which must be considered in the form of randomized schedule paired with faculty teaching should not be used twice. Because of the schedule contains: day, time, and room. Probably not days, hours, and same room used two classes at once.
   Figure 3 are the results of the preparation course scheduling system using a genetic algorithm.

IV. Conclusion

With the help of Genetic Algorithms, creation and scheduling lectures can be optimized. Applications can seek a solution of population based case given the resources and given constraints. In addition, the application can be prevent overlapping schedules lecturer or class and also minimize the incision class that has the same semester or one academic year lapse.

References

[1] D. E. Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning, USA: Addison-Wesley, 1989.

[2] Suyanto, Algoritma Genetika, Yogyakarta: Andi Offset, 2005.

[3] Gen, Eng; and Cheng, R.,, Genetic Algorithms and Engineering Optimization, United States of America: John wiley and Son, 2000.

[4] S. Kusumadewi, Artificial Intelligence: Teknik dan Aplikasinya, Yogyakarta: Graha Ilmu, 2003.

[5] A. Desiani, Konsep Kecerdasan Buatan, Yogyakarta: Andi Offset, 2006.

[6] Edmund, B.;E. David, and W. Rupert;, "A Genetic Algorithm Based University Timetabling System," Department of Computer Science, University of Nottingham, University of Park, Nottingham, NG& 2 RD.

[7] J. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbon, Michigan, 1975.

[8] M. Aria, "Aplikasi Algoritma Genetik untuk Optimasi Penjadwalan," Universitas Komputer Indonesia, 2006.

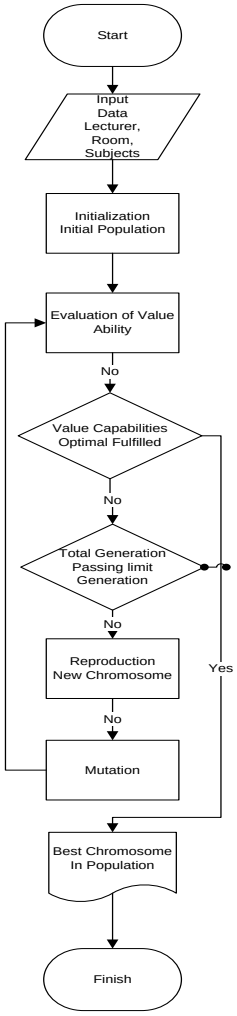

Figure 1. Flowchart of Genetic Algorithm

Figure 2. Main Form of Menu System Application Scheduling Genetic Algorithms Using Java

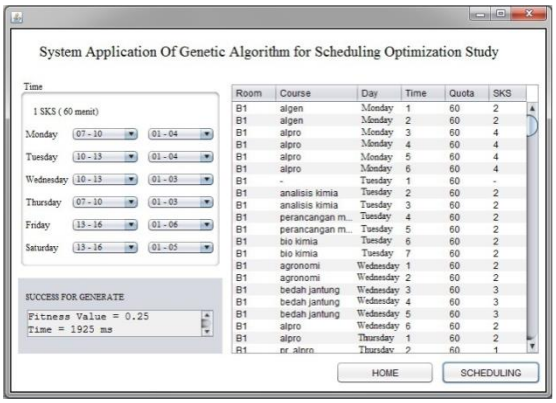

Figure 3. ResultsCreation of Schedule Genetic Algorithm Using Java

TABLE 1.
SCHEDULING SYSTEM OF GENETIC ALGORITHM

Room	Course	Day	Time	Quota	Semester Credit Units
B1	algen	monday	1	60	2
B1	algen	monday	2	60	2
B1	alpro	monday	3	60	4