2017 International Conference on Sustainable Information Engineering and Technology (SIET)

Modified Genetic Algorithm for High School Time-Table Scheduling with Fuzzy Time Window

Ruth Ema Febrita, Wayan Firdaus Mahmudy
Master of Computer Science
Universitas Brawijaya
Malang, Indonesia
Corresponding email to author: ruthemaf@gmail.com

***Abstract*— Time-table scheduling in an educational institution is a very complex problem. This is because of many regulations that must be considered, which are often referred as hard and soft constraints. Unlike in college whose students can be at school only if they have any classes, in junior and senior high schools the schedule amounted to remain a full set of periods every day, so each student must follow a series of all learning process from morning till afternoon. The crowded schedule can make students too tired and distracted during the learning process. The purpose of this research is to make time-table scheduling within the right time window due to the consideration that each subject requires a different thinking portion. Considering the proper time window and the hardness of manual scheduling, we proposed a modified genetic algorithm with a fuzzy time window to solve this problem. Genetic algorithm with modified mutation operation has been implemented to find the optimum solution for this time-table scheduling. The modified mutation operator provides a guarantee in producing offspring that has better fitness value compared to the parent because this operator utilizes fuzzy values as a reference in the exchange of genes. With these guarantees, most offspring generated from the mutation process can be a feasible offspring and can save time to avoid repair mechanisms, so the proposed method can get the optimum solution faster.**

Keywords*— *fuzzy time window, genetic algorithm, time-table scheduling

I. INTRODUCTION

The time-table scheduling problem is a complex problem and classified as NP-Hard problems [1][2]. That means, to find an optimum solution can take a very long time. The complexity of the scheduling problems determined by some variables associated with it, such as the number of rooms, the number of teachers, as well as the number of classes available. The number of classes, limited rooms, and teachers will increase the complexity of the scheduling problem. There are two kinds of constraint in time-table scheduling are a hard constraint and soft constraint. Hard constraints associated with the restrictions that should not be violated, like a room should not be used by more than one learning activity at the same time, a class may not be scheduled to follow two learning activities at the same time. While the soft constraint is a non-essential restriction, for example, the maximum number of exact scheduling lessons in one day. When the soft constraint is fulfilled, it will increase the satisfaction and the comforts of the teachers and students.

Several previous research applied genetic algorithms to solve scheduling problems [3][4]. Genetic algorithms are used because of its ability to search for optimal solutions in large search space. For example, Ghasemi [2] used a genetic algorithm to improve solutions feasibility integrated with the neighborhood in the bee colony algorithm to solve time-table scheduling problem. Previous studies have suggested that genetic algorithms are used successfully to find the optimum solution for scheduling problems.

Most of the previous research has solved the problem of subject scheduling at the college level. However, the time-table scheduling problem in junior and senior high schools is different from time-table scheduling at the college level. At the college level, a student may get a different number of courses each day. For example, they can have 3 courses on Monday, but there are only 2 courses on Tuesday, and there is no class on Wednesday. The schedule allows students to come to school only when they have classes to follow. Unlike in the university, the time-table scheduling in junior and senior high schools includes the same number of periods for each day. Students are required to attend a series of lessons from morning till afternoon. The wrong scheduling may lead the students to feel bored and be easily distracted. For example, students will feel bored when all of the subjects that they have in a day are exact subjects, like Math, Physics, and Biology. On the other hand, students can be easily distracted when all of the subjects in a day are the fun subjects, like arts, physical education, which don’t require much thinking portion. The exact subjects need more thinking portion than social subjects because in the exact subjects students will have many equation problems to solve, which need logical and computational thinking. This kind of subjects needs to be placed in the morning when students still have a fresh time to think so they can follow the lesson well.

Recognizing that each subject requires a different thinking portion, this study aims to solve the scheduling problem by considering the right time window for each subject. For example, physical education will have higher satisfaction level for students and teachers when placed in the first period than the last period. This is due to environmental conditions (the air and the sunlight) in the morning is very good for the body. When sports lessons are placed in the last period, students may have been tired and have less power to do physical activity because they have followed all the lessons and activities in the past few hours, so it will become inappropriate and less efficient. Fuzzy logic can be used to measure how effective and satisfy a subject is when it is put in a certain period. So in this research, we proposed a modified genetic algorithm with a fuzzy time window to solve this problem.

II. Literature Review

A. Genetic Algorithm (GA)

GA is a meta-heuristic algorithm which has the ability to search within a large space area to find a near-optimum solution [5]. GA is one of the evolutionary algorithms which inspired by evolution theory, which searches for a better solution for the next generation. In GA, a solution will be encoded to a string or an array value called chromosomes. While each variable that influences the solution called the gene. There are some types of chromosome representations, such as binary, real-coded, integer, and permutation representation. There is also some operation of the genetic algorithm that has to do in an iteration, such as initialization, reproduction, evaluation, and selection.

Initialization is generating some random solution and store in population. The population itself is a collection of solutions that will be forwarded to the next iteration. Reproduction is producing some other solution through genetic operations, like crossover and mutation. Crossover needs genes from at least two parents. The crossover operation usually produces more than one child solution. Mutation is modifying a parent solution to become another solution. If reproduction processes are done, then the algorithm will evaluate all existing solutions using the fitness function. A fitness function is a function that will show how good a solution is. From this evaluation, the GA will select some solutions that have better fitness value to be processed in the next iteration.

B. Fuzzy Time window

Fuzzy is a method to deal with uncertainty and ambiguity variables in the real world [6]. A fuzzy time window usually implemented in solving the vehicle routing problem [7][8] to find the most efficient routing schedule for a few vehicles. In such cases, time windows do not always have to be strictly obeyed. In the field of transportation and distribution, the schedule sometimes makes the vehicle arrive earlier or even later from the specified time window. The deviation between arrival time and service time (time window) can affect customer satisfaction. The bigger the time difference between arrival time and the specified time window, will decrease the level of customer satisfaction.

In Tang [7], the fuzzy membership function is used to determine customer's satisfaction level when time window violation occurs. The vehicle routing problem with fuzzy time window (VRPTW) then split into two subproblems are vehicle routing problem with time window (VRPTW) and the modification service of the problem. This research applied two kinds of the membership function, linear and concave, and found that the linear membership function can be solved within finite iterations while the concave membership function can be solved using a subgradient-based algorithm.

III. Research Methodology

A. Dataset

The data used in this study is subject and teacher data, which are taken at a private school in Malang. This school implements a full day school system (five school days) with seven periods for each day.

B. Fuzzy Logic for Fuzzy Time window

All subjects given in a school will be divided into two groups based on its acquirement of thinking portion, are exact subjects and non-exact subjects. Mathematics, Physics, and Biology are categorized in the group of exact subjects. The best time window for exact subjects is at 1st period until 4th period. The best time window for Physical Education also at 1st period until 4th period. Thus the non-exact subjects can be placed in the remaining periods. The fuzzy time window is shown in Fig 1.

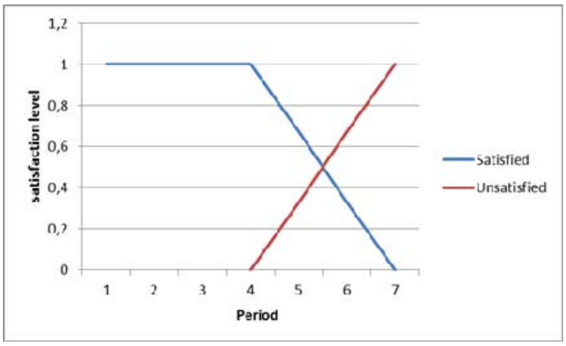

Fig 1. Fuzzy Time window for Exact Subjects and Physical Education

A non-exact subject can be scheduled at any time and always have the satisfaction level=1 for any period. The level of satisfaction schedule for an exact subject and physical education are calculated by the following fuzzy membership function (1):

$$\mu_{satified(z)} \begin{cases} 1, & z \leq 4 \\ \frac{7-z}{7-4}, & 4 < z < 7 \\ 0, & z = 7 \end{cases} \quad (1)$$

This fuzzy membership function is then used to calculate fitness function and as the reference for mutation operation.

C. Finding Optimal Solution with Genetic Algorithm

This research is set for any school with 5 school days and 7 periods in a day.

- Chromosome representation. In this research, each chromosome consists of 210 genes, which are arranged in a matrix form measuring 35 x 6. The number of 35 shows all periods remaining in a week (7 periods x 5 days), while 6 shows the number of classes. Each gene is represented using an integer number. This integer number is the key to distinguish every job-teacher who teach in each different class. For example, Teacher A teaching all classes by two hours each class, then the teacher will have 6 integer numbers, which the number should only appear 2 times for each class.

- Fitness function in this scheduling problem will be shown at (2)

$$fitness(x) = \frac{1}{(\mathrm{HC} * 10) + (SC * 3) + 1} \quad (2)$$

where HC and SC is a hard constraint and soft constraint.

- Crossover and mutation operation. According to Liu [9], the most suitable crossover operation for matrix chromosome is the two-dimensional graphical crossover. This method establishes four dots randomly as the boundaries of the sub-matrix. The sub-matrix in parentA will be exchanged with the sub-matrix in parentB with the same index positions. Fig 2 will show the two-dimensional crossover.

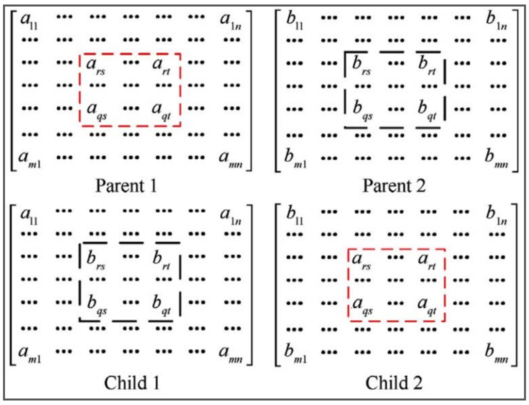

Fig 2. Two-dimensional graphical crossover

- The proposed mutation operation used in this research is the modification of reciprocal exchange mutation. This method specifies 2 points randomly for each class to be exchanged, where the first point is a gene that has a fuzzy value less than 1 (unsatisfied time window) and the second point is a gene which has a non-exact subject which is placed in the 1^{st}-4^{th} period. So, for each chromosome, there are 12 points to be selected in this mutation operation.
- The selection operation used in this research is the replacement selection mechanism is used in this study, where the offspring whose fitness is better than its parent will replace the parent.

IV. Result and Discussion

For the experiments, several parameters of the genetic algorithm are tested, such as pop size testing, crossover rate, and mutation rate, as well as the maximum number of iteration testing. Each test is done by 10 times. We also compared the result of the proposed method (GA with the proposed mutation operation) with the ordinary GA with the reciprocal exchange as the mutation operation.

A. Result of Parameter Testing

1) Pop Size Testing

Pop size testing is done to find the best pop size to achieve the optimum solution. Pop size testing will be shown in Fig 3.

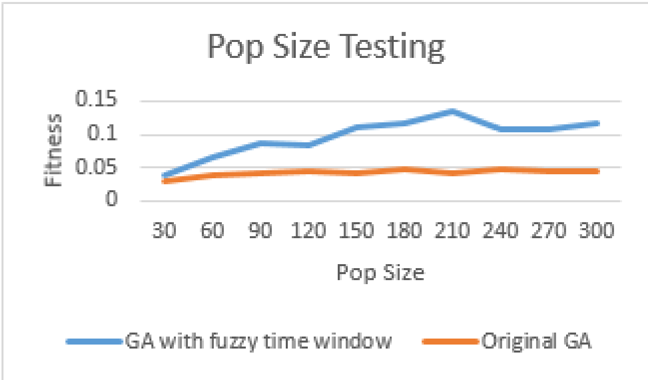

Fig 3. Pop size testing

Based on the test results in Fig 3, GA with fuzzy time window obtained its best fitness when the pop size is at 210 solutions, with the fitness value is 0.1363, while the original GA tend to be convergent from the beginning.

2) Crossover and mutation rate

In examining the crossover rate and mutation rate, there are 9 test scenarios that will be presented in Table 1 by using the best pop size number. Based on the test results shown in Fig 4, the best fitness was successfully obtained when the crossover rate is set at 0.7 and mutation rate set at 0.3 for the GA with a fuzzy time window. The original GA obtain the best fitness when crossover rate at 0.8 and the mutation rate at 0.2.

TABLE I. CROSSOVER AND MUTATION RATE SCENARIOS

Scenario id	Crossover rate (cr)	Mutation rate (mr)
1	0.1	0.9
2	0.2	0.8
3	0.3	0.7
4	0.4	0.6
5	0.5	0.5
6	0.6	0.4
7	0.7	0.3
8	0.8	0.2
9	0.9	0.1

3) Maximum iteration testing

Testing the maximum iteration aims to see the extent of this proposed algorithm in finding a near-optimum solution before convergence condition occurred. This test uses the best pop size number and crossover rate and mutation rate that has been found in the previous testings. Fig 5 will show the result of maximum iteration testing.

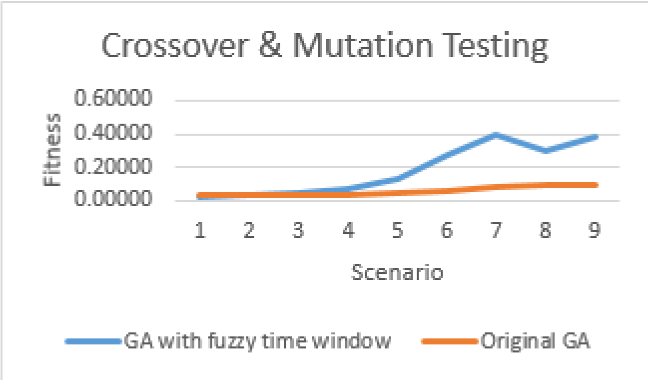

Fig 4. Crossover rate and mutation rate testing result

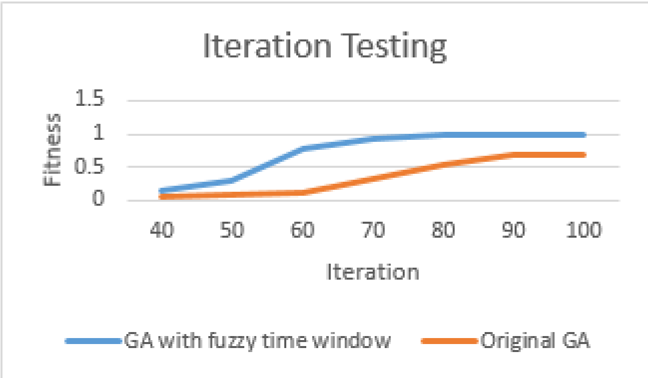

Fig 5. Maximum iteration testing result

Based on Fig 5, the proposed algorithm will obtain its maximum fitness function at the 80th iteration and there is no improvement for several next iterations. The computation time for generating 80 iterations is 8 seconds by using the GA with a fuzzy time window. Meanwhile, the original GA needs 10 seconds for executing 80 iterations, although the fitness value obtained is much less than the proposed method.

B. Discussion

Based on the best parameters achieved from parameter testings, the near-optimum solution is shown in Table II.

The resulting solution guarantee that each class receives the same amount of hours for each subject and each teacher only teaches one class at a time. In the solution shown in Table II, there is still exact subjects or PE are placed above the 4th period, that are in the 5th periods. This may happen because when processing mutation and selection operations, the algorithm will choose the priority so that no hard constraint is violated. But the fuzzy output used in the mutation process will ensure that the exact subject placement will not be too far from the 4th period. However, the solution has a lack in terms of regulating the amount of frequency of a subject in a day. For example, class 8A must follow Indonesian lessons in 3 different periods in Friday, and of course, this can make the students bored and saturated.

Meanwhile, the crossover operations play an important role in maintaining the diversity of solutions during the process of the genetic algorithm. Meanwhile, the mutation operation proposed in this case has a major role in modifying a solution so the solution has an offspring with better fitness value. The novelty of this research is the usage of fuzzy output (satisfaction level) is not only for calculating the fitness value, but also used as a reference to modify the chromosome in the mutation operation. This modified mutation operation assists in finding solutions with better fitness values faster than the original reciprocal exchange mutation operation. This is because the modified mutation operator directly selects some points that have great potential in making the solution not feasible and violates many soft constraints then exchanges it with the satisfied points. However, the crossover and mutation processes will probably produce an unfeasible offspring. An unfeasible offspring occurs when there is at least one code which frequency of occurrence exceeds or is less than the specified in a class. Fig 6 will show the illustration of an unfeasible solution.

TABLE II. NEAR-OPTIMUM SOLUTION

Day	P	7A	7B	8A	8B	9A	9B
Monday	1	SS	Bio	Math	SS	Comp	Math
	2	PE	Math	SS	Indo	Civic	Math
	3	Math	SS	Bio	ELL	Bio	Civic
	4	Music	Math	Phy	ELL	Math	Phy
	5	Civic	Bio	Math	PE	Art	Comp
	6	Bible	Indo	Comp	ELL	Comp	SS
	7	Indo	Civic	Art	Comp	Bible	SS
Tuesday	1	ELL	Math	Phy	Math	Bible	Comp
	2	Bio	Indo	Music	ELL	Art	Bio
	3	Indo	SS	PE	Civic	Math	ELL
	4	Math	Bible	SS	Bible	Bio	PE
	5	Indo	Bible	Phy	SS	PE	Phy
	6	SS	Indo	Comp	Indo	ELL	Bible
	7	SS	ELL	Civic	Music	ELL	SS
Wednesday	1	Cha	Cha	Cha	Cha	Cha	Cha
	2	Bio	Art	Music	Indo	PE	Music
	3	ELL	Bio	Bio	Math	Bible	SS
	4	Bible	Math	Bio	PE	Math	Bio
	5	Math	Music	Math	Phy	Music	Math
	6	SS	Comp	SS	Art	ELL	Bible
	7	Art	Bible	ELL	Comp	SS	ELL
Thursday	1	Comp	Math	Art	Bio	Math	Bio
	2	Math	PE	PE	Music	Music	Math
	3	Phy	ELL	Bible	Bio	Phy	PE
	4	Bio	Phy	ELL	Math	Phy	Art
	5	Phy	ELL	Bible	Phy	Indo	Art
	6	ELL	Art	ELL	Bible	Indo	ELL
	7	Art	Music	SS	Art	ELL	Bible
Friday	1	Comp	Phy	Math	Phy	Phy	Music
	2	ELL	Phy	Math	Bio	Indo	Math
	3	Math	SS	ELL	Math	SS	Indo
	4	Phy	PE	Bible	Math	Bio	Phy
	5	PE	ELL	Indo	Bible	Math	ELL
	6	Bible	Comp	Indo	SS	SS	Indo
	7	Music	SS	Indo	SS	SS	Indo

Based on the illustration shown in Fig 6, the code 1 supposed to appear only 2 times in a class. Child 2 (offspring) is an unfeasible solution because the code 1 appears more than two times, so we need to repair this solution using the repair mechanism. The repair mechanism introduced by the following pseudocode at Algorithm 1.

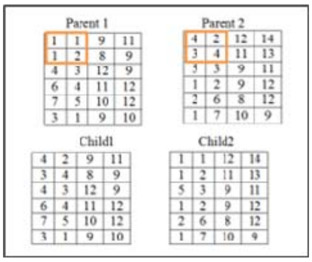

Fig 6. Unfeasible solution illustration

Algorithm 1. Repair Procedure

```
procedure repair
TYPE less[1:34] is scalar array;
TYPE over[1:34] is scalar array;
TYPE class, subject, freq IS SCALAR;

class=0;
do while class < numberOfClasses
  subject=0;
  do while subject < numberOfSubjectCode
  freq =calculate freq of a code in a class
     if freq < supposedToBe[subject] then
        less[subject]=supposedToBe[subject]-freq;
     else
        over[subject]=freq-supposedToBe[subject];
     end if;
  end do;
  change some points which code is listed on the
  less with a code which is listed in over.
 end do;
 end.
```

Based on the Algorithm 1, there is an array that records the supposed number of occurrences for every subject code. The arrays "less" and "over" are used to record the number of deficiencies and excess appearance of all codes. For each code whose frequency of occurrence exceeds than it should be, will be replaced by a subject code that still lacks the frequency of occurrence. Thus the number of each code will correspond to the expected occurrence frequency.

V. CONCLUSION

Genetic algorithm combined with the fuzzy time window has been successfully applied to solve time-table scheduling problems. In this study, the fuzzy time window is not only used to calculate the fitness value of a solution, it is also used as a reference for mutation operation. Based on several testing treatments, the best population size achieved when the pop size is set at 210 solutions, while crossover rate and mutation rate set at 0.7 and 0.3. With this arrangement, the optimal solution was found in 80^{th} generation with the computing time needed is 8 seconds.

For further research, the algorithm can be modified so it can address more complex scheduling problems, such as, setting the subject's maximum frequency of occurrence in one day, as well as ensuring the schedule of a subject close together if the subject appears more than one period in one day required. The adaptive GA can also be tried to solve time-table scheduling problem [10].

ACKNOWLEDGMENT

We especially thank to Charis National Academy who has shared the scheduling data for this research.

REFERENCES

[1] [1] R. Lewis and J. Thompson, "Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem," *European. Journal of Operation Research*, vol. 240, no. 3, pp. 637–648, 2014.

[2] [2] E. Ghasemi, P. Moradi, and M. Fathi, "Integrating ABC with genetic grouping for university course timetabling problem," *International Conference on Computer and Knowledge Engineering (ICCKE)*, 2015.

[3] [3] V. N. Wijayaningrum and W. F. Mahmudy, "Optimization of Ship ' s Route Scheduling Using Genetic Algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 2, no. 1, pp. 180–186, 2016.

[4] [4] A. El Amraoui, M.-A. Manier, A. El Moudni, and M. Benrejeb, "A genetic algorithm approach for a single hoist scheduling problem with time windows constraints," *Engineering Application of Artificial Intelligence*, vol. 26, no. 7, pp. 1761–1771, 2013.

[5] [5] W. S. AlSharafat and M. S. AlSharafat, "Adaptive steady state genetic algorithm for scheduling university exams," *ICNIT 2010 - 2010 International Conference on Networking and Information Technology*, pp. 70–74, 2010.

[6] [6] M. Vadiati, A. Asghari-Moghaddam, M. Nakhaei, J. Adamowski, and A. H. Akbarzadeh, "A fuzzy-logic based decision-making approach for identification of groundwater quality based on groundwater quality indices," *Journal of Environmental Management*, vol. 184, pp. 255–270, 2016.

[7] [7] J. Tang, Z. Pan, R. Y. K. Fung, and H. Lau, "Vehicle routing problem with fuzzy time windows," *Fuzzy Sets and Systems*, vol. 160, pp. 683–695, 2009.

[8] [8] J. Xu, F. Yan, and S. Li, "Vehicle routing optimization with soft time windows in a fuzzy random environment," *Transportation Research Part E*, vol. 47, no. 6, pp. 1075–1091, 2011.

[9] [9] W. Liu, H. Zhu, Y. Wang, S. Zhou, Y. Bai, and C. Zhao, "Topology optimization of support structure of telescope skin based on bit-matrix representation NSGA-II," *Chinese Journal of Aeronautics*, vol. 26, no. 6, pp. 1422–1429, 2013.

[10] [10] M. Z. Sarwani, W. F. Mahmudy, and A. Naba, "Cost Optimization of Multi-Level Multi-Product Distribution Using an Adaptive Genetic Algorithm," *Journal of Information Technology and Computer Science*, vol. 1, no. 2, pp. 53–64, 2017.