Journal of Information Technology and Computer Science
Volume 3, Number 2, 2018, pp. 113-119
Journal Homepage: www.jitecs.ub.ac.id

# Invigilator Examination Scheduling using Partial Random Injection and Adaptive Time Variant Genetic Algorithm

Maulidya Larasaty Seisarrina[1], Imam Cholissodin[2], Heru Nurwarsito[3]
[1,2,3]Faculty of Computer Science, Computer Science, Brawijaya University, Malang, Indonesia
*Corresponding author, [1]maulidyalarasaty@gmail.com, {[2]imamcs*, [3]heru}@ub.ac.id

Received 25 June 2018, accepted 24 October 2018

**Abstract**. Examination for every semester is a routine activity for faculties to do. Academic division of faculty responsible to make the schedule for every subject that is going to be tested, and prepare rooms for the test. Meanwhile, coordinators of invigilator committee responsible to make the schedule in *Fakultas Ilmu Komputer Universitas Brawijaya* (FILKOM UB). This research focuses on scheduling the invigilator's schedule in FILKOM UB. Scheduling with conventional method or manual takes much time because it needs to consider many rules on scheduling it. That is the reason why we need a system to schedule it. The purpose of making this system is to help the committee to schedule their invigilator's time line. This research offers a concept of solution from using genetic algorithm. Genetic algorithm is an algorithm to find the optimum solution. The system of scheduling that use this genetic algorithm method can produce invigilator's schedule that is having the least troubles on the arrangement. The data that is used in this research is the final test's schedule of the odd semester in 2015/2016, lecturer and the employee's data of FILKOM UB. The optimal genetic parameter that is obtained from the test consists of 900 population, 3000 generations, and a combination of crossover rate and mutation rate value which are 0,4 and 0,6. The system that is built in making this invigilator's schedule is close to the optimum point with 0,877 fitness value.
**Keywords:** scheduling, invigilator, partial random injection, adaptive time variant genetic algorithm.

## 1 Introduction

Schedule is a table of activity that is arranged based on time and place in order to do the activity, according to Collins Concise Dictionary [1]. An activity usually makes more than one person in a particular place. The determination of where and when they meet is arranged by using schedule. Schedule in education is part of class scheduling that happens in education institution or university. Scheduling pays attention in some parameter that is related to the implementation of an activity, such as examination in university.

Scheduling examination is a routine activity in every semester that has to be done by academic division and committee in a university [2]. In scheduling, there are some limitations that cause difficulties in making correct and good schedule [3]. Based on an interview that the author does, scheduling examination in *Fakultas Ilmu Komputer Universitas Brawijaya* (FILKOM UB) includes two kinds of scheduling; subjects' schedule and inviligator's schedule. Academic division of faculty responsible to make the schedule for every subject that is going to be tested, and prepare rooms for the test.

Meanwhile, coordinators of invigilator committee have responsible to make the schedule in FILKOM UB. This research focuses on scheduling the invigilator's schedule in FILKOM UB.

This research focuses in scheduling invigilator in FILKOM UB. It has complicated rules. Some of important rules are a lecturer and a staff cannot invigilate one more subject in one same session and the lecturers of particular subject should invigilate subject that they teach and also other rules. Not infrequently in making this scheduling invigilator is very difficult, and still done manually.

Scheduling by conventional/manual method takes a long time because considering the many rules in the scheduling. While the density of committee exams that also comes from lecturers to make the preparation of the exam this semester exam becomes difficult. Therefore, a system is required to perform scheduling of the exam supervisor. Problems that occur in the scheduling can be solved by performing appropriate scheduling calculations, eg with optimization techniques. Conducting research scheduling exams using genetic algorithms [4]. This research shows that genetic algorithm is a good method to solve the scheduling problem, which is usually done by conventional method which takes a big time. Then research on lecture scheduling using genetic algorithm [5]. This study proves that complex college scheduling problems in universities can be solved by genetic algorithm.

Genetic algorithms are best suited for solving problems that have complex mathematical models to accomplish [6]. This study offers the concept of a solution obtained from the search process using a genetic algorithm. The system built is expected to help the coordinator of the semester exam committee in making the semester exam supervisory schedule. This study establishes a scheduling system that can generate a semester exam supervisory schedule with the least amount of violation of the constraints.

## 2 Method

### 2.1 Invigilator Examination Scheduling

"Supervisor" according to Dictionary means a person who guard, look carefully, observe and control the behavior pattern of something or someone [7]. While "Invigilator" can be interpreted someone who keep, look carefully, pay attention and control the state of a test. Almost all exams require one or more exam supervisors to make the exams conducive. In addition, the existence of exam supervisors aims to minimize cheating in the exam.

In the scheduling, there are some restriction/constraint terms which means the rules for preparing a schedule to form a good schedule. The formulation of a scheduling constraint has many variations, depending on the problems and decisions of stakeholders [8]. Type of constraint is divided into 2 namely:

1. Hard constraint: general rules that must be fulfilled in the scheduling and if violated will make the schedule unusable.
2. Soft constraint: Additional rules that if violated are still okay but are encouraged or expected to minimize soft constraint violations in order to maximize the quality of schedules [9].

This study discusses the problem of "Invigilator Examination Scheduling" in FILKOM UB. In general, this exam is supervised by one employee and one lecturer in one room. The criteria used in modeling this scheduling are the courses, the execution time, the course class, and the room.

### 2.2 Genetic Algorithm

The genetic algorithm is one of the most famous scientific branches of evolution algorithms (EA). John Holland (1975) from the University of Michigan, United States, discovered the idea of an inspired genetic algorithm from genetic principles and natural selection [10]. The genetic algorithm begins by encoding the solution of a problem to be solved into a sequence of chromosomes which then calculates its fitness value and is selected which individual has the best fitness value. Encoding technique to encode or map the value of genes (chromosomes), to make individuals. In general, the process in genetic algorithm consists of initialization, reproduction, evaluation, and selection. The procedure in the genetic algorithm is shown in Fig. 1 [11].

```
procedure GeneticAlgorithm
begin
       t = 0
       initialization P(t)
       while (not terminated condition) do
              reproduction C(t) from P(t)
              evaluation P(t) and  C(t)
              selection P(t+1) from P(t) and C(t)
              t = t + 1
       end while
end
```

**Fig. 1 Pseudocode genetic algorithm**

Crossover-rate (*cr*) and Mutation-rate (*mr*) values to determine the number of individuals produced in the reproductive process [12]. The chromosomes in each generation will go through the evaluation process. The evaluation process is the process used to calculate the fitness value of each chromosome. Fitness in biological sciences means the value of the quality of a chromosome that expresses reproductive efficiency. In a genetic algorithm, the quality of a chromosome is represented by a fitness value, if the chromosome fitness value is high, then the quality of the chromosome is also high and to be a potential solution.

### 2.3 Propose Method 1st: Partial Random Injection (PRI)

```
procedure PartialRandomInjection
begin
      t = 0
      initialization P(t)
      while (not terminated condition) do
            reproduction C(t) from P(t)
            evaluation P(t) and C(t)
            selection P(t+1) from P(t) and C(t)
            if (t mod g = 0)
                  repairing few gene from n chromosome
            end
            t = t + 1
      end while
end
```

**Fig. 2 Pseudocode PRI or Atomic Random Injection (ARI)**

in any case of optimization problems with multiple resources, many constraints, and multiple purposes, a chromosome may become infeasible or not possible to be a potential solution. Infeasible chromosomes can be reused by a process of repairing gene values.

In this case, the PRI makes genes improvements to minimize the conflicting supervisors in the same session and other conflicting. In hard constraint, the most important thing is "every employee and lecturer can not supervise more than one exam in the same session (clash)". Based on Fig. 2, PRI is performed in the same way as random injection in the Fig. 1, ie fixing $n$ individuals, where $n = 0.2 \times$ popsize. PRI to repair of the worst individuals as much as $n$ individuals from the selection process. Step by step of the PRI is to check for any genes that contain conflicting supervisors, then look for replacement supervisors for conflicting genes. Based on Fig. 2, the pseudocode part "if ($t$ mod $g$ = 0)" indicates that the PRI process is executed after the selection process, and only in every $g$ generation.

### 2.4 Propose Method 2nd: Adaptive Time Variant Cr and Mr (ATVCM)

Crossovers and mutations influence the genetic algorithm in getting the optimal solution. Crossover, greater influence on the ability of exploitation and mutations affect the ability of exploration. Good exploration and exploitation techniques are key to the success of genetic algorithms [13]. If the *cr* value is too high and the *mr* value is too low it will make the genetic algorithm produce more offspring crossover, the offspring crossover will be similar to the parent, causing the diversity of the population to decrease so that the early convergence will happen quickly [14]. So, in this research we have proposed solution that is Adaptive time variant *cr* and *mr*.

Based on previous research [15], assuming in an early generation, *cr* values are low and will continue to increase until the last generation. In contrast, *mr* values are assumed to be high in the early generations and will continue to decline until the last generation. Based on these circumstances, ATVCM offers the concept of arithmetic series to get out of local optimum conditions. The arithmetic array function is shown in Eq. 1.

$$U_n = a + b(n-1) \tag{1}$$

Thus, based ATVCM concept, each generation of *cr* and *mr* values will increase and decrease following Eq. 2 and 3.

$$Cr = CrMin + \left[ (t-1)\left( \frac{CrMax - CrMin}{t_{\max} - 1} \right) \right] \tag{2}$$

$$Mr = MrMax + \left[ (t-1)\left( \frac{MrMin - MrMax}{t_{\max} - 1} \right) \right] \tag{3}$$

## 3 Results and Discussion

### 3.1 Testing of Partial Random Injection (PRI) and Without PRI

Based on Fig. 3, the PRI method has proven to have improved the genetic algorithm in finding the optimal solution due to a significant increase in average fitness values. This is because PRI works to reduce the violation of hard constraint 1st. At each iteration mod 50 = 0, the system applies a PRI method that improves fitness of 20% worst individual of the popsize from selection.

By using PRI methods it will improve the bad individuals, so the average fitness value of a population will increase. And the most important is the PRI method will reduce the cases of clashes that occur on the schedule.

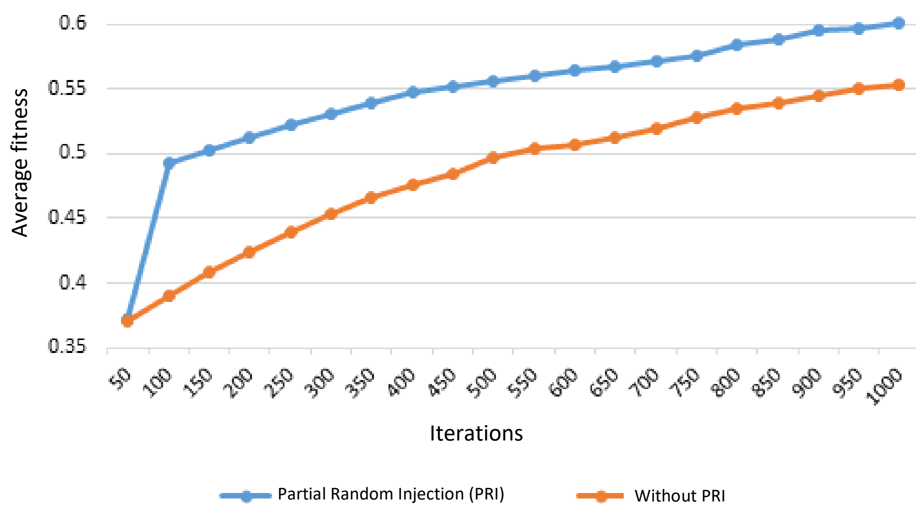


**Fig. 3 Test Result Partial Random Injection (PRI) versus Without PRI**

### 3.2 Testing of Adaptive Time Variant Crossover-rate and Mutation-rate (ATVCM) and Without ATVCM

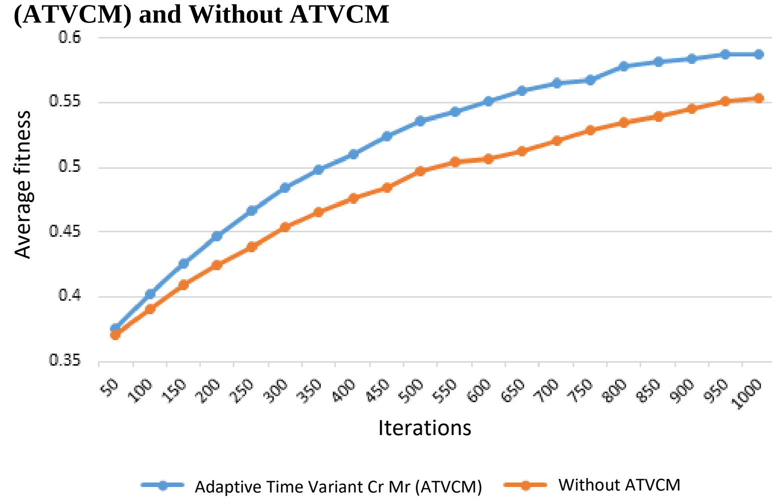


**Fig. 4 Test Adaptive Time Variant Crossover-rate and Mutation-rate (ATVCM) versus Without ATVCM**

In Fig. 4 obtained the result of fitness value, which amount is directly proportional to the amount of generation for each method, that is ATVCM and without ATVCM. But significantly, ATVCM gives excellent fitness value results. This is because the method applies the concept of exploration and exploitation dynamically based on the amount of iteration value when the system is run, referring to previous research [16]. At the beginning of the iteration, the ATVC for the initial value of *cr* is to be set with small value, while the ATVM for initial value of *mr* tend to set with large value, but when at iteration values are increasing, the *cr* tend to be large value, and *mr* tends to be small value. This is in the hope that the system is able to focus on the exploration process when at iteration beginning, and when iterations approach to the maximum iteration, the system will be more focused on the exploitation process. So that this can avoid to optimum local and early convergence conditions.

## 4 Conclusion

Partial Random Injection has shown that this method is very effective to achieve higher fitness and Adaptive Time Variant Crossover-rate and Mutation-rate have also proven that can get the result of fitness value also tends to be better than using Crossover-rate value and Mutation-rate with constant value. Therefore, the Partial Random Injection and ATVCM-GA Methods are well suited for solving highly complex optimization problems with multiple constraints, that allowing a very large search area, which is the greatest probability of the optimal solution being found, being in a small and limited area. So if using a standard method will be very difficult to find the optimal solution. However, the proposed method is very easy to achieve such a small and limited area. And although sometimes solutions at small and limited areas are not feasible to use. This it can be solved by using the PRI method that allows individual solutions to improve by modifying and repair on few of the genes in which indications are not feasible to use to be a feasible condition. And in the end, every individual in the population stable in good condition, which is feasible as the best solution.

## References


1. Collins Concise Dictionary. (2018). “Definition of 'schedule'”. https://www.collinsdictionary.com/dictionary/english/schedule.
2. N. K. Mawaddah and W. F. Mahmudy. (2006). "Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika," Kursor, vol. 2, no. 2, pp. 1-8.
3. W. A. Puspaningrum, A. Djunaidy and R. A. Vinarti. (2013)."Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS," JURNAL TEKNIK POMITS, vol. 2, no. 1, pp. 127-131.
4. S. K. Jha. (2014). "Exam Timetabling Problem Using Genetic Algorithm," IJRET : International Journal of Research in Engineering and Technology, vol. 03, no. 05, pp. 649-654.
5. Jain, A., Jain, S. & Chande, P., (2010). “Formulation of Genetic Algorithm to Generate Good Quality Course Timetable”. International Journal of Innovation and Technology, 1(3), pp. 248-251.
6. W. F. Mahmudy. (2013). Algoritma Evolusi, Malang: Universitas Brawijaya.
7. Pusat Bahasa Departemen Pendidikan Nasional Republik Indonesia. (2008). "KBBI (Kamus Besar Bahasa Indonesia) Daring". [Online]. Available: http://badanbahasa.kemdikbud.go.id/kbbi/. [Accessed 24 Desember 2015].
8. A. Qoiriah. (2014). "Penjadwalan Ujian Akhir Semester dengan Algoritma Genetika (Studi Kasus Jurusan Teknik Informatika UNESA)," Jurnal Manajemen Informatika, vol. 03, no. 02, pp. 33-38.
9. R. Perzina and J. Ramik. (2013). "Self-Learning Genetic Algorithm for a Timetabling Problem with Fuzzy Constraints," International Journal of Innovative Computing, Information and Control, vol. 9, no. 11, pp. 4565-4582.

10. T. Sutojo, E. Mulyanto and V. Suhartono. (2011). Kecerdasan Buatan, 1st ed., Yogyakarta: ANDI.
11. W. F. Mahmudy. (2013). Algoritma Evolusi, Malang: Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya.
12. W. F. Mahmudy, R. M. Marian and L. H. S. Luong. (2012). "Solving Part Type Selection and Loading Problem Flexible Manufacturing System Using Real Code Genetic Algorithms - Part I : Modeling," World Academy of Science, Engineering and Technology, vol. 69, pp. 773-779.
13. W. F. Mahmudy. (2014). "Optimasi Penjadwalan Two-Stage Assembly Flowshop Menggunakan Algoritma Genetika yang Dimodifikasi," STMIK Dipanegara Makassar.
14. W. F. Mahmudy, R. M. Marian and L. H. S. Luong. (2014). "Hybrid Genetic Algorithms for Part Type Selection and Machine Loading Problems with Alternative Production Plans in Flexible Manufacturing System," ECTI TRANSACTIONS ON COMPUTER AND INFORMATION TECHNOLOGY, vol. 8, no. 1, pp. 81-93.
15. Cholissodin I., Khusna R.A., Wihandika R.C., (2016). Optimization Of Equitable Distributions Of Teachers Based On Geographic Location Using General Series Time Variant PSO. 2nd International Symposium on Geoinformatics (ISyG).
16. H. Y. B. W. G. e. a. Chen. (2011). "A Novel Bankruptcy Prediction Model Based on an Adaptive Fuzzy K-Nearest Neighbor Method," College of Computer Science and Technology, Jilin University, Changchun 130012, China.