

# adaptive

*by Adaptive Adaptive*

---

**Submission date:** 13-May-2023 05:18PM (UTC+0700)

**Submission ID:** 2092046590

**File name:** adaptive\_genetic.pdf (546.08K)

**Word count:** 2687

**Character count:** 13762

PAPER · OPEN ACCESS

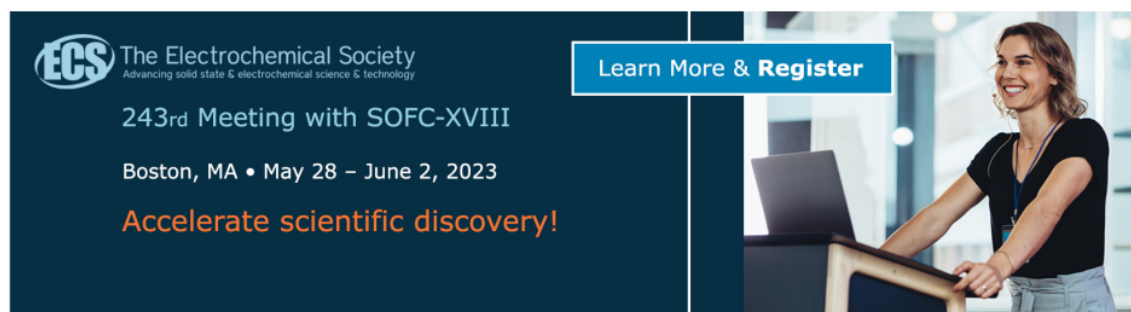
## Adaptive Genetic Algorithm for High School Time-Table

To cite this article: N G A P H Saptarini *et al* 2020 *J. Phys.: Conf. Ser.* **1569** 032095

View the [article online](#) for updates and enhancements.

### You may also like

- 6** - [A comparative study on agarose acetate and PDLA scaffold for rabbit femur defect regeneration](#)  
Rufang Zhao, Zunkai Xu, Bing Li et al.
- 16** - [MASS ECLIPSING BINARIES IN THE INITIAL KEPLER DATA RELEASE](#)  
J. L. Coughlin, M. López-Morales, T. E. Harrison et al.
- 6** - [AGA-GRU: An Optimized GRU Neural Network Model Based on Adaptive Genetic Algorithm](#)  
Chenyao Bai



**ECS** The Electrochemical Society  
Advancing solid state & electrochemical science & technology

**243rd Meeting with SOFC-XVIII**

Boston, MA • May 28 – June 2, 2023

**Accelerate scientific discovery!**

**Learn More & Register**

The advertisement features a dark blue background on the left with white and orange text. On the right, there is a photograph of a woman with blonde hair, wearing a black top and light-colored pants, smiling and looking at a laptop on a desk in a modern office setting.

## Adaptive Genetic Algorithm for High School Time-Table

N G A P H Saptarini<sup>1\*</sup>, P I Ciptayani<sup>1</sup>, N W Wisswani<sup>1</sup> and I W Suasnawa<sup>1</sup>

<sup>1</sup>Electrical Engineering Department, Politeknik Negeri Bali, Raya Uluwatu No.45, Jimbaran, Kec. Kuta Sel., Kabupaten Badung, Bali 80361, Indonesia

Corresponding author <sup>\*</sup>: ayu.harry@pnb.ac.id

**Abstract.** Time-table scheduling in senior high school is complex combinatorial problem. The scheduling process needs much time and susceptible to human error. This human error can lead to the violation of hard constraints. Beside the hard constraints, the good schedule must minimize the violation of soft constraint to ensure the students' and teachers' preference can be satisfied. The aim of this study was to build the automated scheduling process using adaptive genetic algorithm (AGA). The AGA implements in this study to flexible the probability of mutation (pm) based on the fitness. The minimum pm was set to 0.01, while the maximum pm was 0.2. The experiment shows that the AGA give the better fitness compared to the original GA. The best fitness reached by AGA was 0.054 with the average of maximum generation was 402. The original GA with fixed pm 0.1 resulted best fitness 0.045 and the average generation was 387.

### 1. Introduction

The process of build a schedule between rooms, subjects, students and teachers to avoid the constraints violations is called time-table problem (TTP) [1]. TPP is one of the NP-hard problems, so it is need much time to compute the solution [2, 3]. The senior high school time-table can be more complex by the limitation of resource such as teacher. In senior high school time-table, there are some constraints to be satisfied, the hard and soft constraints. The hard constraint is a constraint that cannot be violated, for example one teacher must teach one class at certain time, a teacher must teach the subjects related to their field. While the soft constraint related to the satisfaction of teacher and student, for example equitable distribution of teaching hours, a student may not take the subject more than 3 hours a day. The less violation of soft constraint, the better quality of schedule resulted.

There are many studies in time-table problem. Birbas et al [4] used integer programming to solve the school timetabling. Other study done by Mittal et al [5] using genetic algorithm to build automated time-table. Genetic algorithm is one of meta-heuristic algorithm that suit for combinatorial problem such as time-table scheduling. Perzina and Ramik [6] use a self-learning genetic algorithm and heuristic local search operators. The fitness function is calculated by using fuzzy set. This fuzzy set is used to define the soft constraint violation. The combination of genetic algorithm and fuzzy time window were studied by Febrita and Mahmudy [7]. The study modified the mutation operator in genetic algorithm.

Genetic algorithm has genetic operator such as probability of crossover (pc), probability of mutation (pm). In regular genetic algorithm, the value of pc and pm are fixed in each iteration. The improved of genetic algorithm, introduce a flexible of genetic operator, called adaptive genetic algorithm (AGA). There are many study using AGA to solve combinatorial problem in real world. Taoshen and Zhihui [8] use AGA in multiple QoS Anycast Routing. The study calculate the pc and pm based on the fitness in each iteration. This flexible pc and pm resulted a diversity population and



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

satisfied the constraints. Ying et al [9] also implement adaptive genetic algorithm combine with the Proportional Integral Derivative (PID) to control the temperature in fermentation process. The study resulted the simulation that shows the AGA and PID was effective. Chang et al [10] proposed AGA to tune the PID controller. The AGA used in the study successfully control the activated sludge aeration process. The study about AGA, also conducted by Lin [11]. Study [11] proposed the adaptive on probability of mutation (pm) based on the diversity of population in in certain iteration. Using the adaptive pm proposed in the study, the premature convergence can be avoided and the global optimum can be reach. Study [12] use adaptive pc and pm in Knapsack problem. The study showed that AGA is faster, and have a higher robustness compared to general GA.

This study aims to solve time-table in senior high school using AGA. All of the study mentioned above, use the original GA to solve the time-table. As shown by study [8-12], the AGA has better performance than the original GA, so this study observed how the AGA on probability of mutation influence the convergence and the fitness value.

## 2. Methodology

Genetic algorithm is one of the meta-heuristic techniques that find the solution by randomly generate initial solutions and then do iterations to find the better solution until it is reach the maximum iteration or steady. In genetic algorithm, the solution is called a chromosome. A set of chromosome is called a population. Each iteration in GA will resulted a new generation with the better solution. The quality of a solution /chromosome will define by the fitness function. The iteration will stop if it is reach the maximum generation defined by the user or it is state to be convergence. Convergence happened when the best fitness in each iteration is the same. The adaptive genetic algorithm (AGA) proposed in this study, define the flexible probability of mutation (pm). The flowchart of original GA is shown in Figure 1, while the adaptive part, shown in Figure 2.

### 2.1. Chromosome Representation

The chromosome in genetic algorithm represents the solution. The chromosome in this study represent by two dimensional arrays. The first dimension represent the course session in each class, while the second dimension represent the class (a group of students). The value of each cell is an integer represents the unique code of teacher on certain subject. For example code 01 for teacher X on subject A, and the code 02 for teacher X on subject B.

### 2.2. Fitness Function

The fitness function measures the quality of the chromosome. The fitness function is related to the goal. The goal here was to avoid the hard constraint and minimize the soft constraint. The hard constraint does not involved in fitness function because it the violation must be zero, while the soft constraints will derive the fitness function such as defined in Equation 1.

$$f = \frac{1}{\text{soft constraints violation} + 1} \quad (1)$$

### 2.3. Selection Method

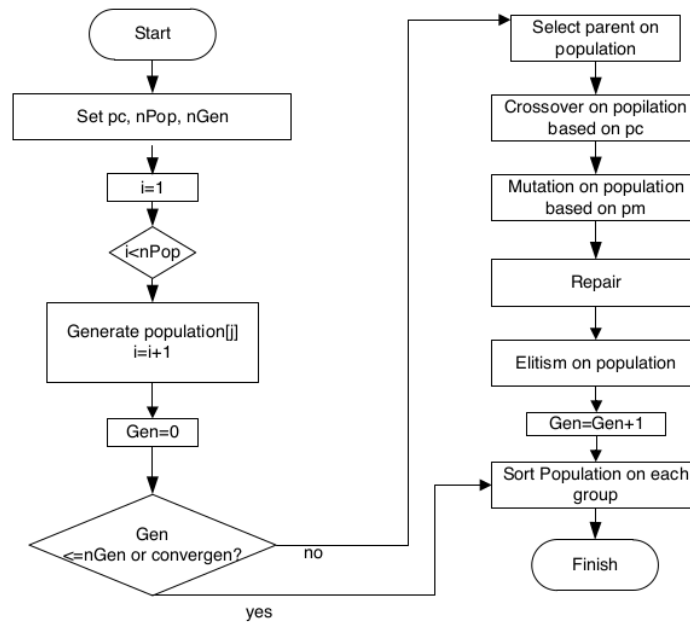
The selection method is important operator to select the parent for crossover. The selection method used in this study was roulette wheel selection, which select the chromosome based on their fitness. The better the fitness, the higher chance for the chromosome to be selected. Here is the pseudo-code of the roulette wheel method [13]:

```
RW_pseudo_code
{
  • Calculate the sum  $S = \sum_{a=1}^n f(a)$ ;
  • For each chromosome  $1 \leq a \leq n$  do {
    o Generate a random number  $r \in [0, S]$ ;
    o Sum = 0; b = 0;
```

```

    ○ do{
      ■ Sum ← Sum+f(b);
      ■ b ← b+1;
    ○ } while (Sum < r and b < n)
    ● select the chromosome b;}
  }

```



**Figure 1.** The genetic algorithm flowchart

#### 2.4. Crossover

The crossover plays the important role in GA. This step will produce offsprings in each iteration. The crossover was conducted using the chromosome selected at selection step. The crossover technique used was one-point crossover. Here each two parents will produce two offsprings. The pseudocode for one-point crossover is as follow:

```

OP_pseudo_code
{
  ● Generate random cut_point
  ● Offspring1 = copy (Parent1 Gen[1] to Gen[cut_point])
  ● Offspring1 = Offspring1 + copy (Parent2 Gen[cut_point+1] to Gen[n])
  ● Offspring2 = copy (Parent2 Gen[1] to Gen[cut_point])
  ● Offspring2 = Offspring2 + copy (Parent1 Gen[cut_point+1] to Gen[n])
}

```

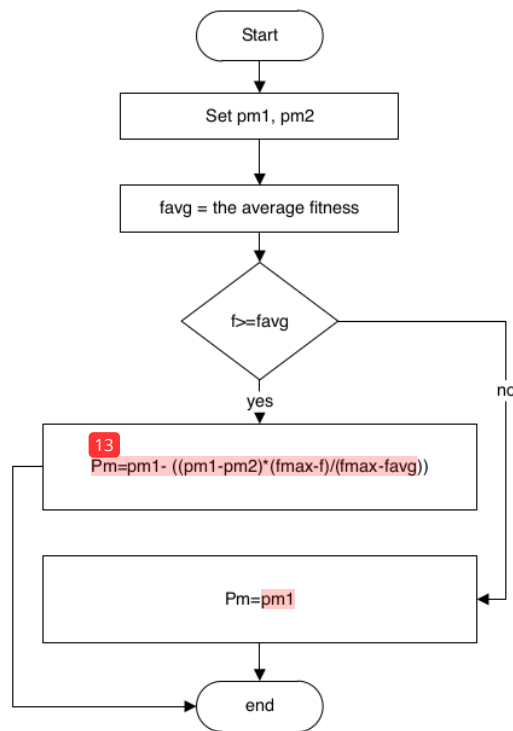
#### 2.5. Mutation

Mutation is one of mechanism in nature. It is happened in small possibility. Based on that fact, the probability of mutation is set to be small. The mutation play a role to make chromosome vary, so more

solution is explore and the local optima can be avoided. This study use adaptive genetic algorithm to define the flexibility of pm of each chromosome based on their fitness. If the fitness of is small than average population, the pm is set to be the biggest pm, while if the fitness is bigger or equal to average of population member the pm is set based on Equation 2 [8].

$$pm = \begin{cases} pm1 - \frac{(pm1-pm2)(fmax-f)}{(fmax-favg)} & \text{if } f \geq favg \\ pm1 & \text{if } f < favg \end{cases} \quad (2)$$

Where pm1 maximum is the pm, pm2 is the minimum pm, fmax is the maximal fitness value, favg is the average fitness and the f is the fitness of chromosome. The flowchart of pm is shown in Figure 2.



**Figure 2.** The pm calculation flowchart

### 2.6. Repair

Repair is one of mechanism to avoid the hard constraint violation after the crossover and mutation were conducted. This step will check if there are hard constraint violation in a chromosome and will fixed it by change the gene with a valid value that not violate the hard constraint. Using this step, the violation of hard constraint will be guaranteed not to occur.

### 2.7. Elitism

This is the last step in GA. Each offspring will be compared to the population member. If the offspring has better fitness compared to at least one population member, then the offspring will be include in the

population to change the chromosome with the worst fitness. This step is done to ensure only the good quality of offspring will be included to form the new generation.

### 3. Result

The case used in this study was taken from SMA Surya Wisata, Jalan Wagimin, Kediri, Tabanan. There are total 27 classes and each classes has maximum 11 session each day. Here each class will be using the same classroom every day. The number of teacher was 61. The parameter used in this study was shown in Table 1.

**Table 1.** The parameters in this study

Parameter	Number
Number of individu in population	50
Maximal number of generation	1000
Probability of crossover (pc)	0.8
Probability of mutation maximum (pm1)	0.2
Probability of mutation minimum (pm2)	0.01

Based on the flowchart in Figure 1, the first step was to generate the initial population by randomly generate 50 chromosomes. The crossover was conducted based on the pc, but the selection was conducted first. The mutation was then conducted based on the adaptive pm, where the minimal pm was set to be 0.01 and the maximum pm was 0.2. All of the steps were conducted until the maximal generation was reached or the convergence state is occurred. After 30 experiments with the same parameter, the average best fitness of all experiment was 0.054 and average of maximum generation was 402. The original GA set with the same parameter, except the pm 0.1. The original GA result the best fitness of 0.045 and the average generation was 387.

### 4. Conclusion

This study used adaptive genetic algorithm to solve time-table scheduling in senior high school. The adaptive GA implement to the flexible probability of mutation (pm), with the minimum pm 0.01 and the maximum pm 0.2. The adaptive pm based on the chromosome fitness. This adaptive pm aims to keep the diversity in population in order to avoid the algorithm stuck in local optima. The experiment shows that the AGA gives the better fitness compare to the original GA. The AGA reach the best fitness 0.054 and the maximum generation was 402. For further study, the adaptive can be implement in probability of crossover and the population size.

### 5. References

- [1] A Nanda, M P Pai and A Gole, 2014, An algorithm to automatically generate schedule for school lectures using a heuristic approach, *International Journal of Machine Learning and Computing* 2 (4), 2012, pp 492-495
- [2] R Lewis and J Thompson, 2014, 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
- [3] E Ghasemi, P Moradi, and M Fathi, 2015, Integrating ABC with genetic grouping for university course timetabling problem, *International Conference on Computer and Knowledge Engineering (ICCKE)*.
- [4] T Birbasa, S Daskalakib and E Housosa, 2009, School timetabling for quality student and teacher schedules, *Journal of Scheduling*, Volume 12 Issue 2, pp. 177 - 197

- [5] D Mittal, H Doshi, M Sunasra and R Nagpure, 2015, Automatic timetable generation using genetic algorithm, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, pp. 245-248
- [6] R Perzina and J Ramik, 2013, Self-learning genetic algorithm for a timetabling problem with fuzzy constraints, International Journal of Innovative Computing, Volume 9, Number 11, pp. 4565 - 4582
- [7] R E Febrita and W F Mahmudy, 2017, Modified genetic algorithm for high school time-table scheduling with fuzzy time window, 2017 International Conference on Sustainable Information Engineering and Technology (SIET), Batu, East Java, Indonesia, pp 88-92
- [8] L I Taoshen and G E Zhihui, 2009, A multiple QoS anycast routing algorithm based adaptive genetic algorithm, 2009 Third International Conference on Genetic and Evolutionary Computing, pp 89 - 92
- [9] Z Ying, Z Lei, W Gui-li and Y Qian, 2009, A nonlinear PID controller based on an adaptive genetic algorithm, 2009 Second International Conference on Intelligent Networks and Intelligent Systems, pp. 98 - 91
- [10] P Zhang, M Yuan and H Wang, 2006, Self-tuning PID based on adaptive genetic algorithms with the application of activated sludge aeration process, Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China, pp. 9327 - 9330
- [11] C Lin, 2009, An adaptive genetic algorithm based on population diversity strategy, 2009 Third International Conference on Genetic and Evolutionary Computing, pp 93-96
- [12] Y Ma, 2012, The modified hybrid adaptive genetic algorithm for 0-1 knapsack problem, 24th Chinese Control and Decision Conference (CCDC), Taiyuan, 2012, pp. 326-329
- [13] K Jebari, 2013, Selection methods for genetic algorithms, International Journal of Emerging Sciences, 3, pp 333-344



# adaptive

---

## ORIGINALITY REPORT

---

15%

SIMILARITY INDEX

12%

INTERNET SOURCES

10%

PUBLICATIONS

8%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1	<a href="http://www.cjche.com.cn">www.cjche.com.cn</a> Internet Source	1%
2	Submitted to University of Warwick Student Paper	1%
3	Submitted to Federal University of Technology Student Paper	1%
4	Submitted to Midlands State University Student Paper	1%
5	Submitted to University of Adelaide Student Paper	1%
6	<a href="http://nlist.inflibnet.ac.in">nlist.inflibnet.ac.in</a> Internet Source	1%
7	Assembly Automation, Volume 33, Issue 1 (2013-02-02) Publication	1%
8	<a href="http://aggietranscript.ucdavis.edu">aggietranscript.ucdavis.edu</a> Internet Source	1%
9	<a href="http://ijrar.com">ijrar.com</a> Internet Source	1%

---

10	pdfs.semanticscholar.org Internet Source	1 %
11	repository.essex.ac.uk Internet Source	1 %
12	Sarimveis, H.. "A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms", Computers and Chemical Engineering, 20040115 Publication	1 %
13	Shang, W.. "A flexible tolerance genetic algorithm for optimal problems with nonlinear equality constraints", Advanced Engineering Informatics, 200907 Publication	1 %
14	Submitted to Universiti Tenaga Nasional Student Paper	1 %
15	vdel.me.cmu.edu Internet Source	1 %
16	core.ac.uk Internet Source	1 %
17	Hong Wang. "Self-Tuning PID Based on Adaptive Genetic Algorithms with the Application of Activated Sludge Aeration Process", 2006 6th World Congress on Intelligent Control and Automation, 2006 Publication	<1 %

18	<a href="http://ieomsociety.org">ieomsociety.org</a> Internet Source	<1 %
19	<a href="http://j.mecs-press.net">j.mecs-press.net</a> Internet Source	<1 %
20	<a href="http://nipesjournals.org.ng">nipesjournals.org.ng</a> Internet Source	<1 %
21	<a href="http://www.ijcic.org">www.ijcic.org</a> Internet Source	<1 %
22	Rushil Raghavjee, Nelishia Pillay. "An informed genetic algorithm for the high school timetabling problem", Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '10, 2010 Publication	<1 %

Exclude quotes Off  
Exclude bibliography Off

Exclude matches Off