Comparison of genetic algorithms and Particle Swarm Optimization (PSO) algorithms in course scheduling

D R Ramdania1,*, M Irfan1, F Alfarisi1 and D Nuraiman2

1 Informatics Engineering, Universitas Islam Negeri Sunan Gunung Djati Bandung, Bandung, Indonesia
2 Mathematics, Universitas Islam Negeri Sunan Gunung Djati Bandung, Bandung, Indonesia

*diena.rauda@uinsgd.ac.id

Abstract. Making lecture schedules is a complicated matter because it involves many parties and resources. In order to arrange the schedule optimally, a method is needed that can produce the best optimization value. In this paper, we will discuss the comparison of Genetic Algorithms and Particle Swarm Optimization to design lecture schedules. Once implemented, then comparative analysis of the results of the course scheduling process is carried out by comparing the fitness value and execution speed of the two algorithms. The results showed that in the 25th iteration the Genetic algorithm succeeded in compiling a lecture schedule with the best fitness value of 0.021 and 9.36 second execution time compared to the PSO algorithm which produced a fitness value of 0.099 and execution time of 61.95 seconds in the same iteration. This proves that the PSO fitness value outperforms the Genetic Algorithm, but the Genetic Algorithm execution time is faster than the PSO algorithm.

1. Introduction

The scheduling process is one of the important things in the teaching and learning process. Lecturer and student activities depend on the existing schedule, so it must be properly compiled and improved at the beginning of the academic year, so that later it does not interfere with teaching and learning activities between lecturers and students [1]. There are several aspects that must be considered in making a schedule, there should be no conflicting schedules, for example there is a lecturer who is scheduled to teach 2 different classes at the same time or students are scheduled to study in 2 different subjects at the same time, availability of teaching time for temporary lecturers, limited space, and differences in credits, for example in 2 credits, 100 minutes and 3 credits 150 minutes [2].

Because of its complexity, the scheduling process often takes a little time. A system is needed that can generate schedules automatically to be more efficient. Several studies have applied algorithms to answer this problem, including: scheduling lectures using the PSO algorithm [1], scheduling subjects using Genetic Algorithms [3,4], outsourcing scheduling using Genetic Algorithms [2], scheduling job [5], and hydro generation scheduling using the Genetic Algorithm, PSO, and Different Evolutional Algorithm [6]. Most existing research uses Genetic Algorithms or PSO.

The Genetic Algorithms are algorithms that use techniques and processes that are inspired by biological evolution to solve complex optimization problems, there are several selections including selection, crossover, and mutation applied to obtain the optimal solution [2] and the Particle Swarm
Optimization (PSO) algorithm is one search techniques in computer science based on a population. The PSO initialization process starts with a random population called particles, each PSO particle has a velocity component [4]. So far no studies have compared the Genetic Algorithm and PSO Algorithm performance. That's why we built a scheduling system that uses both algorithms. This study has an algorithm that has better performance in managing schedules.

2. Genetic algorithm
The Genetic Algorithm was discovered at the University of Michigan, the United States by John Holland through a study and popularized by one of his students, David Goldberg [7]. Genetic algorithms are search algorithms that mimic the mechanisms of natural genetics. This algorithms fall into the Evolutionary Algorithm group. Genetic algorithms are based on the principles of genetics and natural selection, the basic elements of natural genetics are reproduction, crossover, and mutation. These elements are used in GA [8]:

![Genetic Algorithm Diagram](image.png)

**Figure 1.** Illustration of genetic algorithm [9].

2.1. Initialize the initial population
Initialization of the initial population is a method for producing the initial chromosomes. The number of individuals in the initial population is input from the user. After the number of individuals in the initial population is determined, initialization of the chromosomes is found in the population. Initialization is done randomly, but still shows the solution domain and the existing problem constraints [2].

2.2. Evaluation function
An individual or chromosome is evaluated based on a particular function as a measure of its performance. The function used to measure a matching value is called the fitness function. The process of evaluating this fitness function will continue until the criteria for stopping are fulfilled [10].
2.3. Selection
The selection process is a process for determining which individuals will be chosen to do a crossover. There are several types of selection methods commonly used, among others: Methods that mimic the roulette-wheel game where each chromosome occupies a circle piece on the roulette wheel proportionally according to its fitness value. Ranking Selection The process starts by ranking or sorting chromosomes in the population based on their fitness then giving a new fitness value based on the order [11].

2.4. Crossovers
Crossovers or crossover is a process that forms a new chromosome from two parent chromosomes by combining the information part of each chromosome. Crossover produces a new chromosome called the child chromosome (offspring) [12].

2.5. Mutation
The process of mutation is to change one or more several genes from a chromosome. This process serves to replace lost genes from the population due to a selection process that allows the reappearance of genes that do not appear in population initialization [13].

3. Particle Swarm Optimization (PSO) algorithm
The PSO algorithm was introduced by Dr. Eberhart and Dr. Kennedy in 1995, was an optimization algorithm that mimicked the processes that occur in the lives of populations of birds and fish in survival. Since it was first introduced, the PSO algorithm has developed quite rapidly, both in terms of application and in terms of the development of the methods used in the algorithm [14]. This is because, the PSO algorithm is an optimization algorithm that is easy to understand, quite simple and has work that has proven to be reliable [15].

In the case of lecture schedules, particles are a class schedule consisting of several classes. There are 3 stages of the PSO method for lecture scheduling problems, namely as follows:

3.1. Generation of position and speed of particles
The particle changes its position from an iteration to another position based on the update velocity. First position X\_0^i and particle velocity X\_0^i from a collection of particles that are randomly processed using the upper limit (X\text{max}) and lower limit (X\text{min}), like the following equation:

\[ x_0^i = x_{\text{min}} + \text{rand} (x_{\text{max}} - x_{\text{min}}) \]

\[ v_0^i = x_{\text{min}} + \text{rand} (x_{\text{max}} - x_{\text{min}}) \]

Where :
- \( X_0^i \) = Initial position
- \( V_0^i \) = Initial speed
- \( X_{\text{min}} \) = Lower limit
- \( X_{\text{max}} \) = upper limit
- Rand = random value between values 0 and 1

Position and speed are represented in vector form where n-dimensional vectors represent the number of particle variable designs, with superscript and subscript denoting particles to i at time k. With this initialization process, the collection of particles can be distributed randomly in a design space [1].

Vector as shown below:

\[ x_k^i = (x_k^{i1}, x_k^{i2}, ..., x_k^{in}) \]

\[ v_k^i = (v_k^{i1}, v_k^{i2}, ..., v_k^{in}) \]
3.2. Velocity update (speed update)
Update the velocity (speed) for all particles at 1 + k using the objective function or the fitness value of the current particle position in design space at time k. From the fitness value can be determined which particle has the best global value in the current swarm, \( p^g_k \), and also can be determined the best position of each particle at all times now and before, \( p^i \). The formulation of the update velocity uses the two information for all particles in the collection with the effect of the current displacement \( v^i_k \), to give the search direction \( v^i_{k+1} \), for the next generation. The update velocity formulation includes several random parameters, \( rnd \), to get good coverage on design space, three parameters that affect the search direction, namely inertia factor \( (w) \), self-confidence \( (c1) \), swarm confidence \( (c2) \) will be combined in one presentation, as shown in the following equation [1]:

\[
v^i_{k+1} = w * v^i_k + c1 * rnd * (p^i - x^i_k) + c2 * rnd * (p^g_k - x^i_k)
\]

With a range of \( w = 0.4 \text{ - } 1.4, c1 = 1.5 \text{ - } 2.0 \) and \( c2 = 2.0 \text{ - } 2.5 \)

3.3. Position update
Update the position of each particle with a velocity vector, as shown in the following equation:

\[
x^i_{k+1} = x^i_k + v^i_{k+1}
\]

The three steps above will be repeated until the convergence criteria are met, the convergence criteria are very important in avoiding the addition of the evaluation function after the optimum solution is obtained, but convergence criteria are not always absolutely necessary, the determination of the maximum iteration can also be used as the stopping condition of the algorithm.

4. Research method
We made this system uses the method of developing prototype software because this method facilitates the process of building software and customers can interact with each other during the system creation process [15].

Product development is carried out with the following prototype process: First, we meet Deputy head of the Curriculum to the objectives, desired needs and description of the parts that will be needed, and interview the problems in the preparation of the schedule. After that, we design and represents all aspects of the known software, as well as gathering the needs that will be used in the system. At the modelling quick design stage, we represents aspects of the software that can be seen by the Deputy head of the Curriculum, such as the application interface design and database design. Modeling Quick Design tends to make prototypes. The prototype that has been made by the author will be submitted to the Deputy Head of the Curriculum, to be evaluated, then they will provide feedback that will be used to revise the software requirements that will be built. Repetition of this process continues until all needs are met.

5. Result and discussion

5.1. Display the generate schedule page
Interface implementation is an implementation of a system that has been created containing the appearance of system applications, such as login display, lecturer data, course data, space data, time data, lecture data, and generate schedules. Figure 2 is a page view of the lecture schedule, on this page to create a class schedule based on the data previously added. To make a schedule, first, enter the algorithm parameters to be able to calculate the data.
5.2. **Testing**
The test shows some results of testing systems for Genetic Algorithms and Particle Swarm Optimization (PSO) algorithms by varying input parameters.

5.2.1. **Genetic Algorithm.** This performance test is carried out by experiment with 7 times testing, with user input parameters.

| Cromosom | Generation | Crossover Rate | Mutation Rate | Fitness | Time   | Result |
|----------|------------|----------------|---------------|---------|--------|--------|
| 10       | 25         | 75             | 25            | 0.024   | 9.25'  | Success|
| 15       | 30         | 77             | 26            | 0.022   | 16.02' | Success|
| 20       | 35         | 75             | 25            | 0.023   | 25.31' | Success|
| 25       | 40         | 75             | 25            | 0.023   | 36.14' | Success|
| 30       | 45         | 80             | 30            | 0.022   | 49.38' | Success|
| 35       | 50         | 85             | 40            | 0.022   | 62.45' | Success|
| 40       | 55         | 90             | 50            | 0.022   | 75.56' | Success|

5.2.2. **Particle Swarm Optimization algorithm.** This performance test is carried out by experiment with 7 times testing, with user input parameters.

| Iteration | C1 | C2 | W  | Fitness | Time   | Result |
|-----------|----|----|----|---------|--------|--------|
| 25        | 1  | 2  | 0.6| 0.082   | 54.50' | Success|
| 30        | 3  | 2  | 0.6| 0.164   | 59.61' | Success|
| 35        | 0.7| 1.4| 0.4| 0.083   | 66.33' | Success|
| 40        | 3  | 1  | 0.5| 0.499   | 72.96' | Success|
| 45        | 2  | 1  | 0.8| 0.083   | 71.40' | Success|
| 50        | 2  | 0.5| 0.9| 0.083   | 83.04' | Success|
| 55        | 4  | 1  | 0.5| 0.166   | 85.81' | Success|
5.2.3. **Comparison of Genetic Algebraic and Particle Swarm Optimization (PSO) algorithms.** The results of this comparison use 40 scheduling data, using the default value parameter value, this test is done by comparing the fitness value obtained by the two algorithms with the same generation and iteration.

| **Table 3. Testing the comparison results of GA and PSO algorithms.** |
|----------------------------------|------------------|-------|
| **Parameter**                    | **Comparison**   |       |
| Generation/Iteration to achieve Fitness value | GA | 25 |
| Optimom fitness value            | GA | 0.021 | PSO | 0.099 |
| Speed                            | GA | 9.36' | PSO | 61.95' |

Testing result shows Genetic Algorithm successfully composes the lecture schedule with the best results of fitness 0.021 in the 25th generation, with execution time of 9.36 seconds. While the particle swarm optimization (PSO) algorithm is fitness 0.099 in 25 iterations, with execution time 61.95 second. From several trials, it was obtained that the PSO fitness value outperformed GA so that the PSO has a tendency to a fitness value that is better than GA. But the speed of execution of GA has a fast time compared to PSO.

6. **Conclusion**

A course schedule management system has been built using genetic algorithms and Particle Swarm Optimization. This system is able to manage a schedule more quickly and efficiently. The results showed that the fitness value obtained by PSO outperformed the Genetic Algorithm. But the speed of execution of GA has a fast time compared to PSO. In the future, these two algorithms can be integrated with other algorithms and applied to other scheduling processes.

**References**

[1] Ariani D, Fahriza A and Prasetyaningrum I 2011 Optimasi Penjadwalan Mata Kuliah di Jurusan Teknik Informatika Pens dengan Menggunakan Algoritma Particle Swarm Optimization (PSO) EEPIS Repos. 1–11

[2] Janata A and Haerani E 2015 Sistem Penjadwalan Outsourcing Menggunakan Algoritma Genetika ( Studi Kasus : PT. Syarikatama ) J. CoreIT 117–24

[3] Handoyo T, Rachmawati A K and Prasetyo E 2015 Sistem Penjadwalan Mata Pelajaran di SMA Muhammadiyah 1 Kota Magelang Dengan Algoritma Genetika Transformasi 1 114–9

[4] Hariyadi H P, Widiyantyayas T, Arifin M Z and Sendari S 2016 Implementation of Genetic Algorithm to academic scheduling system 2016 IEEE Region 10 Conference (TENCON) pp 2013–6

[5] Aprianto R 2018 Pengembangan Aplikasi Web Mobile Penjadwalan Tugas Aparatur Desa Untuk Meningkatkan Layanan Masyarakat 01 81–6

[6] Choubey S 2017 Genetic algorithm, particle swam optimization and differential evolution algorithm for optimal hydro generation scheduling 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) pp 486–91

[7] Mauluddin S, Ikbal I and Nursikuwagus A 2018 Optimasi Aplikasi Penjadwalan Kuliah Menggunakan Algoritma Genetik vol 2

[8] Santosa B and Ai T J 2018 Buku Pengantar Metaheuristik – Implementasi Dengan Matlab _ Penerbit Guna widya (Surabaya: ITS Tekno Sains)

[9] Darmawan E 2018 C4.5 Algorithm Application for Prediction of Self Candidate New Students in Higher Education J. Online Inform. 3 22

[10] Hijriana N 2015 Penerapan Metode Algoritma Genetika Untuk Permasalahan Penjadwalan
Perawat (Nurse Scheduling Problem) *Info Tek.* 16 61–74

[11] Studi P, Jurusan M, Matematika F, Ilmu D A N, Alam P and Dharma U S 2007 Implementasi Algoritma Genetika Untuk Masalah Penjadwalan Job-Shop

[12] Laksono A T, Utami M C and Sugiarti Y 2016 Sistem Penjadwalan Kuliah Menggunakan Metode Algoritma Genetika (Studi Kasus: Fakultas Kedokteran Dan Kesehatan Universitas Muhammadiyah Jakarta) *J. Sist. Inf.* 9 177–88

[13] Rahman A, Yuniarno E mulyanto and Ketut Eddy Purnama I 2018 *Optimasi Penjadwalan Perkuliahan Menggunakan Metode Harmony Search* vol 2

[14] Haupt R L and Haupt S E 2004 *Practical Genetic Algorithm*

[15] Rosa A S and M Salahuddin 2015 *Rekayasa Perangkat Lunak* (Bandung: INFORMATIKA)