

Genetic Algorithm Optimization for Automatic Scheduling in the System at State Junior High School Four Binjai

Muhammad Ilham Fadilla ^{1*}, Akim Manaor Hara Pardede ², Magdalena Simanjuntak ³

^{1,2,3}STMIK Kaputama

fadilailham105@gmail.com^{1*}, akimhmp@livi.com², magdalena.simanjuntak84@gmail.com³

Abstract

Binjai State Junior High School faces challenges in developing an optimal class schedule each semester. The manual process currently in place is often time-consuming, error-prone, and inflexible in adapting to sudden changes such as teacher changes or school policy changes. Furthermore, scheduling conflicts frequently occur, where a teacher is scheduled to teach two different classes at the same time, or a class has two subjects in one session. This scheduling process must consider various constraints, such as teacher availability, class size, subject matter, and limited learning space. The manual scheduling process is often time-consuming, error-prone, and difficult to adapt to sudden changes such as teacher changes or curriculum changes. To address these challenges, an automated system is needed that can generate schedules efficiently and optimally. Genetic Algorithms are a method in artificial intelligence that can solve optimization problems by mimicking biological evolutionary mechanisms such as selection, crossover, and mutation. By implementing Genetic Algorithms in the scheduling system, it is hoped that more optimal schedules can be produced by reducing schedule conflicts and increasing time efficiency in the scheduling process.

Keywords: Genetic Algorithm Optimization, Scheduling, Junior High School

1. Introduction

Genetic Algorithms are an evolution-based optimization technique that has proven effective in solving various scheduling problems. These algorithms mimic the process of natural selection by applying operations such as selection, crossover, and mutation to find the best solution for developing an optimal schedule. Shows that the application of Genetic Algorithms in scheduling optimization can reduce waiting time and increase operational efficiency [1].

Shows that the combination of Genetic Algorithm with other search methods can produce more optimal solutions in task scheduling in cloud computing environments [2]. Shows that this method can avoid schedule conflicts, ensure a balanced distribution of teaching, and increase efficiency in preparing lecture schedules [3]. By implementing a similar approach at Binjai State Junior High School, it is hoped that an automated scheduling system can be created that is more optimal, flexible, and adaptive to change.

This research aims to develop an automated scheduling system based on a genetic algorithm at the Fourth State Junior High School in Binjai. This approach is expected to reduce scheduling conflicts, accelerate the scheduling process, and improve the effectiveness of teacher and student time management. The application of genetic algorithms in educational settings can be an innovative solution to address complex scheduling challenges.

2. Literatur Riview

2.1. Optimization

Optimization is the process of finding the best solution from a group of alternatives that meets certain objectives, such as minimizing conflict or maximizing resource utilization [4]. In the context of scheduling, optimization is used to increase efficiency by considering various constraints, such as the availability of space, time, and human resources [5].

2.2. Automatic Scheduling

Automated scheduling is the process of systematically preparing schedules using algorithms or software to optimize resource use and minimize scheduling conflicts [6]. In education, this system is widely used in scheduling lectures, exams, and other academic activities. Automated scheduling considers two types of constraints: hard constraints and soft constraints. Hard constraints include prohibitions on

lecturer schedule conflicts, classroom capacity limitations, and the overlap between courses taken by the same student. Soft constraints, meanwhile, include lecturer teaching time preferences and the equitable distribution of courses throughout the week. The advantages of automated scheduling include time efficiency, increased accuracy, flexibility to changes, and transparency in the distribution of schedule information.

2.3. Genetic Algorithm

A genetic algorithm (GA) is an optimization technique inspired by the selection process. It is a population-based search algorithm that utilizes the concept of survival of the fittest [7]. A new population is generated by the iterative application of genetic operators on individuals in the population. Chromosome representation, selection, crossover, mutation, and fitness function computation are the key elements of GA. The procedure of GA is as follows. A population (Y) consisting of n chromosomes is randomly initialized. The fitness of each chromosome in Y is calculated. Two chromosomes, say C1 and C2, are selected from the population Y based on their fitness values. A one-point crossover operator with crossover probability (C_p) is applied to C1 and C2 to produce an offspring, say O. After that, a uniform mutation operator is applied to the resulting offspring (O) with mutation probability (M_p) to produce O'. The new offspring, O', are placed in a new population. The selection, crossover, and mutation operations will be repeated on the current population until the new population is complete. GA dynamically changes the search process through crossover and mutation probabilities to reach an optimal solution. GA can modify encoded genes. GA can evaluate multiple individuals and generate multiple optimal solutions. Therefore, GA has superior global search capabilities [8].

2.4. Initializing The Population

The initial process in a genetic algorithm is population initialization, where available genes are randomly assigned to form a population. However, the population formed at this initialization stage does not yet meet the desired objectives and may still contain conflicts or incompatibilities. However, by performing this initialization process, the algorithm can save time and increase efficiency in finding optimal solutions [9].

2.5. Crossover Operator

Crossover, also known as crossbreeding, is a crucial step in genetic algorithms where two genes are randomly selected from the population and used as parents. The parents' chromosomes are then exchanged with each other, producing a new, improved gene that combines the traits inherited from both parents [10].

2.6. Mutation

Mutation or mutation is the next stage in the Genetic Algorithm which functions to randomly change one or more genes in a chromosome, creating a new, more adaptive solution in the new generation of the population [11].

2.7. Fitness

The fitness process is carried out to evaluate the value of offspring in each new generation. Genes with higher fitness values are maintained, and those that do not achieve the desired fitness value are manipulated in the next stage [12].

2.8. Scheduling System at State Junior High School Fourth Binjai

Scheduling is a crucial process to ensure that all activities, especially learning activities, can run efficiently and orderly. Good scheduling takes into account not only time but also various other factors such as classroom availability, teacher availability, and student needs. In this context, optimal scheduling will help reduce scheduling conflicts and maximize the use of existing resources. Typically, scheduling in schools is done manually or using simple software. Manual methods involve setting schedules by hand using paper or spreadsheets, which can be very complex and time-consuming, especially for large schools with many subjects and teachers.

2.9. Optimization In Scheduling

To overcome the limitations of manual methods, many schools are turning to software or automated scheduling systems. Automated scheduling systems can help speed up the schedule creation process, reduce errors, and allow for easier schedule adjustments.

3. Analysis And Design

3.1. Research Methodology

The method used in this study is a quantitative method with an experimental approach. The main focus of this study is to optimize a genetic algorithm for an automated academic scheduling system at the Four Binjai State Junior High School, which involves various interrelated parameters, such as the number of teachers, the number of classes, subjects, classroom availability, and the time available for each student and teacher.

3.2. System Requirement Analysis

To achieve the desired results, the system must meet certain conditions. To meet these conditions, an analysis of functional and non-functional requirements is performed.

a. Functional Requirements

These requirements include the elements required by the system to function properly, in accordance with its intended purpose. The following are some of the functional requirements of the system being developed:

1. The system must be able to generate automatic schedules based on the school's academic regulations.
2. The system must be able to manage teacher, class, subject, and classroom data.
3. The system must be able to avoid schedule conflicts between teachers and classes.
4. The system must support flexible schedule changes or improvements.
5. The system must provide a user interface for viewing and managing schedules.

b. Non-Functional needs

Non-functional requirements cover system elements that are not directly related to its function, but generally relate to the system's performance. Here are some of the system's non-functional requirements:

1. The system must perform well in generating schedules within a reasonable timeframe.
2. The system must be easy to use for school operators.
3. The system must support expansion for future changes in academic regulations.
4. The system must operate stably without interruptions or major errors.

3.3. Model Implementation

The automated scheduling system developed aims to address the challenges of managing academic schedules at Binjai State Junior High School 4. Currently, the scheduling process is still done manually, which often leads to clashes between subjects, limited classroom space, and an imbalance in teaching time allocation for teachers. By applying a genetic algorithm (GA), this system is able to generate an optimal schedule that takes into account various constraints such as the number of classes, teacher availability, classroom capacity, and the distribution of subjects throughout the week.

Genetic algorithms work on the principle of natural selection to find the optimal solution from a range of possible schedules.

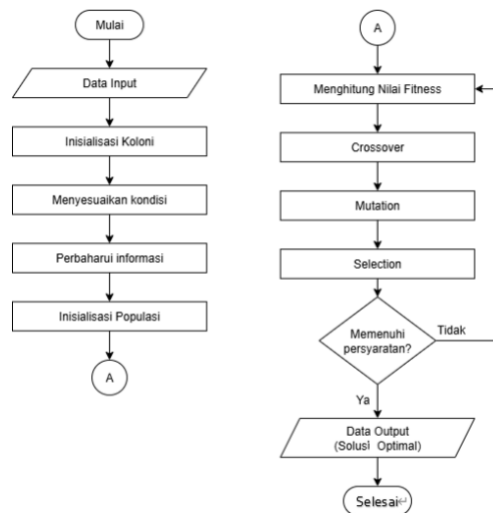


Fig. 1 : Flowchart

Fig. 1 The flowchart of the genetic algorithm illustrates how the system works in generating an optimal schedule using a genetic algorithm, from initialization to evaluation of the best results.

3.3.1. Database Model

The database model is designed to ensure the integrity, efficiency, and flexibility of data management required by a Genetic Algorithm (GA)-based automated scheduling system. A good database structure will not only simplify the management of large and complex data but will also support the algorithm's performance in generating optimal scheduling solutions and addressing various issues that may arise in the manual scheduling process. The database structure in this automated scheduling system includes several main tables that represent important entities in the scheduling process. These tables are designed to store the information required by the algorithm to perform its functions efficiently. The following is a further explanation of these main tables:

Table 1: Database structure that includes

Table Type	Fill in the columns and data types	Information
Teacher Table	teacher_id INT (Primary Key), teacher_name VARCHAR(100), subject_name VARCHAR(100), number_of_hours INT.	Unique ID for each teacher, Full name of teacher, Subject taught, Number of teaching hours per week

Class Table	id_kelas INT (Primary Key), nama_kelas VARCHAR(50), tingkat INT, jumlah_siswa INT	Unique ID for each class, Class name (eg: 7A, 7B, 8A), Class level (eg: 7, 8, 9), Number of students in the class
Subject Table	subject_id INT (Primary Key), subject_name VARCHAR(100), level INT, number_of_hours_per_week INT.	Unique ID for each subject, Name of the subject, Grade levels that can take this subject, Number of class hours in a week.
Classroom Table	id_ruang INT (Primary Key), nama_ruang VARCHAR(50), capacity INT, facility TEXT	Unique ID of the classroom, Name or number of the classroom, Maximum student capacity, Facilities available (AC, Projector, etc.).
Schedule Table	id_schedule INT (Primary Key), id_guru INT (Foreign Key), id_kelas INT (Foreign Key), id_mapel INT (Foreign Key), id_ruang INT (Foreign Key), hari ENUM('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'), jam_mulai TIME, jam_selelais TIME	Unique schedule ID, Linking to teacher table, Linking to class table, Linking to subject table, Linking to class_room table, Day of the week, Lesson start time, Lesson end time.

3.4. Design

User interaction with the system is simple:

- Data input: Users enter data via a form (teacher, subject, class, room, and teaching schedule).
- Scheduling control: Users press a button to start the process and save the schedule results.
- Review results: Users can view the schedule.

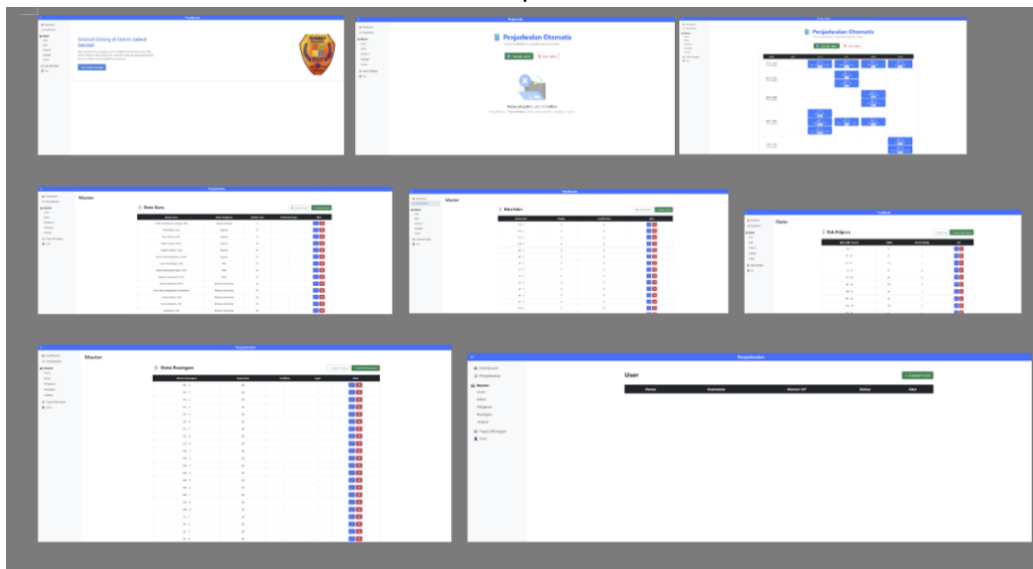


Fig. 2 : Design System

The entire system's interface is designed to be simple and easy for schools to operate. Furthermore, the interface design takes into account ease of navigation to ensure a fast and efficient scheduling process.

4. Discussion And Implementation

The automated scheduling system developed in this research features a key advantage: the implementation of a genetic algorithm optimized to efficiently address scheduling conflicts. Compared to previous manual or semi-automatic methods, this system is able to generate schedules without conflicting teachers, subjects, and spaces more quickly.

4.1. Program Test Results

The trial was conducted using real-world data from Binjai State Junior High School 4:

- 5 teachers
- 1 subject
- 3 classes; with 9 levels for grade VII, 9 levels for grade VIII, and 8 levels for grade IX
- 26 rooms

The system successfully generated a schedule without conflicts for all classes with a single subject. The scheduling process time was 0.088 seconds per population, with the system generating 410 populations, with a maximum generation of 50.

Table 2: Displaying scenario test results

Scenario	Test Conditions	Results	Status
Complete input, default parameters	There is no clash	Valid schedule	Passed
Teacher data based on teaching class	Schedule clashes arise	Error notification appears	Passed

4.2. Success Analysis

Success analysis is carried out to evaluate the extent to which the developed automatic scheduling system is able to produce an optimal schedule and meet the criteria without conflict.

a. Test Graph

In the test graph, we display the fitness value development from the genetic algorithm execution results. Data is taken from the genetic_log.jsonl file. Each point on the graph represents an individual solution in a generation.

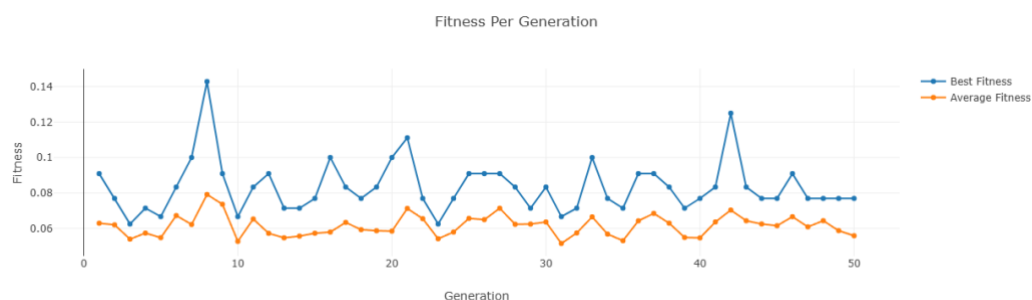


Fig. 2 : Graph of fitness values per individual tested.

Each individual in the early generations exhibits a relatively low fitness value, ranging from 0 to 0.3. Some individuals show an increase in fitness approaching 0.1429 as the number of generations increases. The graph pattern shows an evolutionary process, where solutions improve (increase in fitness) from generation to generation.

5. Conclusion and Recommendations

5.1. Conclusion

Based on the research and implementation of an automated scheduling system using a genetic algorithm conducted at State Junior High School 4 Binjai, the following conclusions can be drawn:

1. The developed scheduling system is capable of generating automatic schedules with a low level of conflict. Test results show a fitness value of 0.3 for individuals in the first generation, with an even distribution of schedules across classes, teachers, subjects, time, and space.
2. The implemented genetic algorithm successfully processed schedule data efficiently at a junior high school scale with the following data:
 - 5 teachers
 - 1 subject
 - 3 classes: with 9 levels for grade VII, 9 levels for grade VIII, and 8 levels for grade IX
 - 26 rooms

The system successfully generated schedules without conflicts for all classes with one subject. The scheduling processing time was 0.088 seconds per population, with the system generating 410 populations, and a maximum generation of 50.

3. Algorithm optimization through parameter settings (population, crossover probability, mutation probability) affected the quality of the resulting schedule. The right parameter values can accelerate the achievement of optimal fitness.

5.2. Recommendations

As a follow-up and further development of this research, the following are recommended:

1. The system needs to be equipped with stricter input validation to prevent data conditions that make it impossible to generate a schedule (e.g., insufficient number of teachers or space).
2. It is recommended to add a schedule visualization feature in the form of a heatmap to make it easier for schools to analyze scheduling results.
3. Testing with more generations is recommended to maximize the evolution process and find more optimal solutions.
4. The system can be further developed with a real-time schedule adjustment feature for sudden changes (e.g., teacher leave, space unavailable).
5. System testing is needed in other schools with larger data sets or different characteristics to measure system scalability.

References

- [1] Rivera, G., Cisneros, L., Sánchez-Solís, P., Rangel-Valdez, N., & Rodas-Osollo, J. (2020). Genetic algorithm for scheduling optimization considering heterogeneous containers: A real-world case study. *Axioms*, 9(1). <https://doi.org/10.3390/axioms9010027>
- [2] Velliangiri, S., Karthikeyan, P., Arul Xavier, V. M., & Baswaraj, D. (2021). Hybrid electro search with genetic algorithm for task scheduling in cloud computing. *Ain Shams Engineering Journal*, 12(1). <https://doi.org/10.1016/j.asej.2020.07.003>
- [3] Nasien, D., & Andi, A. (2022). Optimization of Genetic Algorithm in Courses Scheduling. *IT Journal Research and Development*. <https://doi.org/10.25299/itjrd.2022.7896>
- [4] Wasudeorao, K. A., & Prasad, K. (2024). Evolutionary Optimization of Deep Learning Models. *Advances in Nonlinear Variational Inequalities*, 27(2), 632–648.
- [5] Selvam, G., & Tadepalli, T. C. M. (2019). Genetic Algorithm-Based Optimization for Resource Leveling Problem. *International Journal of Construction Management*, 19(3), 1–12. DOI: 10.1080/15623599.2019.1641891
- [6] Nuraisyah, N., Permana, I., & Salisah, F. N. (2017). Sistem Penjadwalan Otomatis Tempat Khutbah Jum' At Mubaligh. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 3(1).
- [7] Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs. In *Genetic Algorithms + Data Structures = Evolution Programs*. <https://doi.org/10.1007/978-3-662-03315-9>
- [8] Liu, D. (2019). Mathematical modeling analysis of genetic algorithms under schema theorem. *Journal of Computational Methods in Sciences and Engineering*, 19(S1). <https://doi.org/10.3233/JCM-191019>
- [9] Ba, M., Hu, Y., Xu, C., & Zhong, Y. (2019). A course scheduling algorithm in secondary vocational school based on genetic algorithm. 2019 6th International Conference on Systems and Informatics, ICSAI 2019. doi:10.1109/ICSAI48974.2019.9010352
- [10] Xu, J. (2021). Improved Genetic Algorithm to Solve the Scheduling Problem of College English Courses. *Complexity*, 2021. doi:10.1155/2021/7252719
- [11] Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., & Prasath, V. B. S. (2019). Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach. *Information (Switzerland)*, 10(12). doi:10.3390/info10120390
- [12] Jatoth, C., Gangadharan, G. R., & Buyya, R. (2019). Optimal Fitness Aware Cloud Service Composition using an Adaptive Genotypes Evolution based Genetic Algorithm. *Future Generation Computer Systems*, 94. doi:10.1016/j.future.2018.11.022