Published online on the page : https://ejournal.uinbukittinggi.ac.id/index.php/ijokid/

Knowbase : International Journal of Knowledge in Database

| ISSN (Print) 2798-0758 | ISSN (Online) 2797-7501 |

Implementation of Genetic Algorithm for Automatic Course Scheduling Optimization

Rakhmi Khalida[1], Situmorang Ondo Palito Tua[2], Dwi Budi Sri Sulistiowati[3], Siti Setiawati[4]*

[1,2,3,4] *Universitas Bhayangkara Jakarta Raya, Jakarta, Indonesia*

Article Information

Article History:

Submitted: October 30, 2025
Revision: December 18, 2025
Accepted: December 23, 2025
Published: December 31, 2025

Keywords

Scheduling
Genetic Algorithm
Local Search
Optimization
Vocational High School

Correspondence

E-mail: rakhmi.khalida@dsn.ubharajaya.ac.id*

A B S T R A C T

Course scheduling in vocational high schools (SMK) constitutes a complex combinatorial optimization problem involving multiple hard and soft constraints related to teacher availability, class allocation, and time-slot distribution. Although Genetic Algorithms (GA) have been extensively applied in educational timetabling, existing studies largely emphasize standalone optimization or desktop-based solutions, with limited analytical evaluation of refinement strategies and system-level applicability. This study addresses this gap by empirically evaluating a hybrid GA–Local Search (LS) approach embedded within a web-based scheduling framework. GA is utilized as a global search mechanism to generate feasible schedules that satisfy all hard constraints, while LS is applied as a post-optimization phase to improve solution quality by reducing soft constraint violations. Experiments were conducted using real scheduling data from SMK Yadika 13 Bekasi, involving 3 subjects, 3 teachers, 4 classes, and 12 time slots within a single-day scenario. Although limited in scale, this configuration was deliberately selected to enable transparent analysis of the optimization dynamics and refinement impact of the proposed hybrid approach. The results show that the pure GA produces five soft constraint violations, mainly due to suboptimal placement of cognitively demanding subjects and uneven subject distribution. After applying LS, violations were reduced to two cases, with the fitness value improving from 0.873 to 0.946 and only a marginal increase in computation time (5–7 seconds). These findings demonstrate that local refinement significantly enhances schedule quality beyond conflict-free feasibility. This study contributes scientifically by providing an empirical assessment of GA–LS hybridization for soft-constraint optimization and by establishing a scalable web-based framework that supports future extensions to full-week scheduling and adaptive academic systems.

This is an open access article under the CC–BY-SA license

1. Introduction

Educational timetabling is a well-known combinatorial optimization problem that arises in various educational contexts, involving the allocation of subjects, teachers, classes, rooms, and time slots under multiple hard and soft constraints [1][2]. The complexity of this problem increases significantly as the number of constraints grows, making manual scheduling approaches inefficient and error-prone. In practice, manual timetabling often leads to conflicts, frequent revisions, and suboptimal schedules, particularly when unexpected changes occur, such as teacher unavailability or curriculum adjustments [3]. Beyond administrative inefficiencies, poor scheduling decisions may negatively affect instructional quality, student learning outcomes, and the overall academic environment, especially when workloads are unevenly distributed or cognitively demanding subjects are placed at less effective time slots. To address the computational complexity of educational timetabling, various metaheuristic approaches have been proposed, among which Genetic Algorithms (GA) have been widely adopted due to their ability to explore large and complex search spaces efficiently [4][5].

Research by Saputra et al. developed a scheduling system at SMK Negeri 1 Sine using a genetic algorithm to generate automatic, conflict-free timetables. The study demonstrated that GA can reduce scheduling conflicts and improve the speed of timetable creation compared to manual methods, although it remains limited to handling hard constraints and does not incorporate any advanced solution-refinement mechanisms [1]. Another study published in the Jurnal Komputer dan Sistem Informasi (2024) also employed genetic algorithms to enhance scheduling efficiency, focusing on optimizing subject distribution and classroom utilization. The findings highlight that GA is effective in reducing conflicts and producing more stable schedules, but the approach has yet to explore integration with other metaheuristic techniques to further improve solution quality [6]. Meanwhile, other research applied GA for school timetable optimization and showed that GA can process various scheduling combinations more quickly and accurately than traditional manual methods. However, this study also relied on pure GA without incorporating local search–based refinement mechanisms to handle more complex soft constraints [7].

Previous studies have demonstrated that GA can effectively generate conflict-free timetables and reduce scheduling time compared to manual methods [8][6]. However, most existing works rely on pure GA formulations that primarily focus on satisfying hard constraints, while soft constraint optimization—such as balanced subject distribution or pedagogically appropriate time placement—remains insufficiently addressed [7]. As a result, schedules produced by pure GA approaches may be feasible but not necessarily optimal in terms of quality and usability. Several studies have attempted to apply GA-based scheduling in vocational high schools (SMK), showing improvements in automation and conflict reduction [1][6]. In these studies, SMK serves as a representative case of educational timetabling problems with moderate complexity. Nevertheless, the proposed solutions generally employ standalone GA without incorporating refinement mechanisms capable of systematically improving solution quality, particularly in reducing soft constraint violations. This indicates a research gap in evaluating hybrid optimization strategies that combine global search with local refinement within educational timetabling contexts.

This study addresses the gap by proposing and evaluating a hybrid Genetic Algorithm–Local Search (GA–LS) approach for educational course scheduling. GA is utilized as a global optimization method to generate feasible schedules that satisfy all hard constraints, while Local Search is applied as a post-optimization phase to iteratively refine the solution by minimizing soft constraint violations. The proposed approach is evaluated using real scheduling data from a vocational high school as a case study to empirically analyze the contribution of local refinement to schedule quality. The scheduling algorithm is implemented within a web-based platform to support practical deployment; however, the primary contribution of this research lies in the algorithmic evaluation of GA–LS hybridization for improving solution quality in educational timetabling

Therefore, the objective of this study is to analyze and evaluate the effectiveness of a hybrid Genetic Algorithm–Local Search (GA–LS) approach in improving solution quality for educational timetabling problems, with particular emphasis on reducing soft constraint violations beyond conflict-free feasibility. The main contribution of this research lies in providing an empirical algorithmic assessment of how local refinement complements GA-based global search in educational scheduling, using a real-world vocational high school dataset as a representative case study. Unlike previous studies that rely on pure GA formulations or focus primarily on system implementation, this research introduces a hybrid optimization framework that explicitly addresses soft constraint optimization and evaluates its impact on fitness improvement and computational cost. The novelty of this study is not in proposing a new GA variant, but in systematically demonstrating the added value of GA–LS hybridization for enhancing schedule quality within educational timetabling, thereby offering a methodological reference for future optimization-oriented research in this domain.

2. Method

This study adopts an experimental research methodology to evaluate the effectiveness of a hybrid Genetic Algorithm–Local Search (GA–LS) approach in educational timetabling optimization, rather than focusing solely on system development. Genetic Algorithm (GA) is employed as a global search mechanism to generate feasible scheduling solutions that satisfy all hard constraints, while Local Search (LS) is applied as a post-optimization phase to refine the best GA-generated solution by reducing soft constraint violations [9].

The experimental evaluation was conducted using a one-day scheduling scenario as a representative case. Although limited in scope, this configuration is methodologically valid because LS operates by performing localized improvement operations, such as swapping or reassigning time slots [10] . The reduced search space enables clear observation of the refinement effect introduced by LS, while maintaining relevance for larger scheduling scenarios. The chromosome representation and fitness evaluation functions are designed to be modular, allowing the same optimization framework to be extended to weekly or full-semester scheduling without altering the core algorithmic principles.

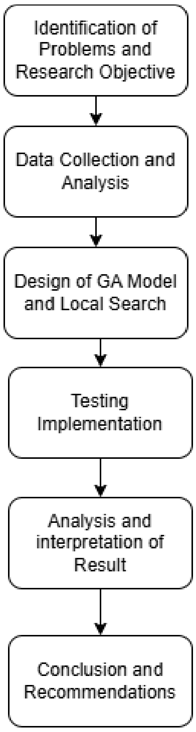

Figure 1. Research Method

This methodological design aims to evaluate the contribution of Local Search as a post-optimization mechanism in reducing soft-constraint violations can be seen in Figure 1. The research focuses on comparing schedule quality before and after the application of LS, measured in terms of fitness improvement, number of soft constraint violations, and computational overhead. The research procedure consists of four main stages. First, the problem is formulated by identifying scheduling constraints and defining optimization objectives, with vocational high school scheduling used as a case study. Second, scheduling dataincluding subjects, teachers, classes, and time slots—are collected and analyzed to classify hard and soft constraints. Third, the algorithmic model is designed using GA operators (selection, crossover, and mutation) for global optimization [11], followed by LS as a local refinement mechanism applied to the best GA solution. Finally, experimental testing is conducted by varying GA parameters based on heuristic optimization practices, and the resulting schedules are evaluated to assess solution quality and computational efficiency. The study concludes with an analytical interpretation of the experimental results to determine the extent to which the GA–LS hybrid approach improves scheduling quality beyond conflict-free feasibility, thereby providing empirical evidence of its effectiveness for educational timetabling optimization [12][13].

2.1. Algoritma Genetika

Genetic Algorithm (GA) is a population-based evolutionary optimization method inspired by natural selection, which iteratively improves candidate solutions through selection, crossover, and mutation operators [14][15]. GA is particularly effective for solving complex combinatorial problems with large search spaces, such as educational timetabling [16]. In this study, GA is customized for course scheduling by representing each chromosome as a complete timetable. Each gene encodes a specific scheduling assignment that combines subject, teacher, class, and time slot into a single scheduling unit. This representation enables GA to explore feasible combinations while maintaining structural consistency in the schedule.

The fitness function is designed to prioritize constraint satisfaction. Hard constraints such as avoiding teacher, class, and room conflicts are strictly enforced and assigned high penalty values to ensure feasibility. Soft constraints including balanced teacher workloads, appropriate placement of cognitively demanding subjects, and equitable subject distribution are incorporated using weighted penalties. The fitness value is computed based on the total penalty score, allowing GA to guide the evolutionary process toward solutions that are both feasible and high quality [17][8]. Through this constraint-handling mechanism, GA effectively balances global exploration with practical scheduling requirements.

Genetic Algorithm Steps

The main stages of the Genetic Algorithm (GA) combined with Local Search (LS) are illustrated in Figure 2. Through these stages, the Genetic Algorithm performs a global search to explore a wide solution space, while the Local Search component ensures a more optimal final result by refining solutions that are close to a local optimum [18] [19]. This combination makes the GA–LS approach superior to the pure GA, as it produces automatic course schedules that are more valid, efficient, and well-optimized [20] [21].

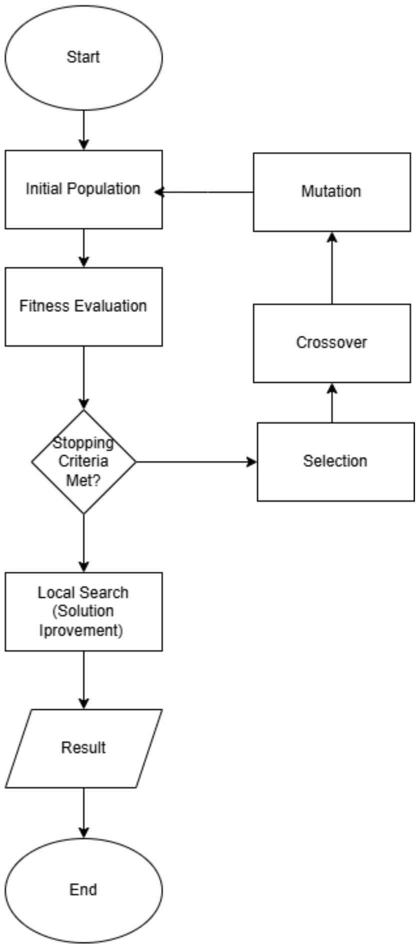

Figure 2. Genetic Algorithm Model Stages

The explanation of the Genetic Algorithm (GA) process model shown in Figure 2 is as follows [8][22]:

1. The initial population size of 100 was chosen to balance solution diversity and computational efficiency, consisting of four individuals, each containing 32 chromosomes, corresponding to the total number of teaching hours on the test day.
2. Fitness Evaluation – Fitness evaluation is used to measure the quality of each individual schedule within the population. In this study, the fitness value is calculated based on the number of constraint violations occurring in a timetable. Hard constraints, such as teacher, class, and room conflicts, are strictly prohibited and must be completely avoided, whereas soft constraints, such as uneven teaching load distribution or suboptimal placement of subjects, are allowed but should be minimized [23][24][25].

   The fitness function used is expressed as:

   $$\text{Fitness} = \frac{1}{1 + \sum clash\ gen} \tag{1}$$

   explanation :

1	=	fixed numbers in genetic algorithms
Clash gen	=	conflicts that occur (represents the total number of detected constraint violations in a chromosome)

   In the current implementation, all detected clashes are treated equally and assigned the same penalty weight, regardless of their type. This means that each violation—whether related to soft constraints or residual conflicts during intermediate generations—contributes equally to the reduction of the fitness value. Soft constraints are counted whenever predefined preference rules are violated, such as unbalanced teaching workloads or the placement of cognitively demanding subjects in less effective time slots.

   The selection of this fitness function is motivated by its simplicity, interpretability, and computational efficiency, which are particularly suitable for experimental evaluation of hybrid optimization behavior. By using an inverse penalty formulation, the function ensures that fitness values are normalized within the range (0,1), facilitating comparison between solutions and clearly reflecting improvement trends across generations.

   However, the absence of differentiated weights for individual constraints represents a limitation of this study. In practical scheduling scenarios, certain constraintsespecially hard constraints may be more critical than others and should be penalized more heavily. This limitation is partially addressed in this research through the application of Local Search, which specifically targets soft constraint violations after feasibility is achieved. Future work may incorporate weighted fitness functions or multi-objective optimization to better reflect the relative importance of different constraints and further enhance solution quality.
3. Selection Process – The selection process uses the roulette wheel method, in which the probability of an individual being selected is proportional to its fitness value. Individuals with higher fitness values have a greater chance of producing new offspring baru [26][27].
4. Crossover Operator – The crossover operation is performed using the one-point crossover method. Two parent chromosomes are selected, cut at a specific crossover point, and then exchanged to produce new offspring. The crossover rate is set at 0.80.

5. Mutation Process – Mutation is carried out by modifying a portion of the time-related genes to prevent population homogeneity. The mutation rate used is 0.50. At this stage, assignment genes are preserved to maintain the consistency of teacher–subject allocation [17].
6. Stopping Criteria – The algorithm terminates when the fitness value reaches the optimal score (1) or when the maximum number of generations is achieved [28].
7. After the Genetic Algorithm (GA) process terminates, the best solution obtained is further refined using a Local Search (LS) mechanism. This refinement stage performs localized improvement operations, such as swapping or reassigning schedule slots, with the objective of reducing soft constraint violations and enhancing overall solution quality [29] [30]. The LS procedure is executed iteratively, where each candidate modification is evaluated based on its impact on the fitness value, and improvements are accepted until no further significant fitness increase can be achieved.

   The parameter settings and stopping criteria for the LS process were determined empirically based on preliminary experiments to ensure computational efficiency and observable solution refinement. Although parameter selection was empirically guided, a more systematic sensitivity analysis and statistical validation are left for future work. Consequently, the current evaluation emphasizes the qualitative and empirical contribution of LS as a post-optimization mechanism rather than providing a comprehensive statistical comparison across parameter configurations
8. Result – The final output represents the optimal scheduling solution obtained using the Genetic Algorithm.

2.2. Algoritma Local Search

Local Search (LS) is employed as a metaheuristic improvement method to refine solutions previously generated by the Genetic Algorithm (GA). Unlike GA, which performs global exploration through evolutionary operators, LS focuses on local exploitation by systematically exploring the neighborhood structure of a given solution through predefined local move operations [31][32]. The objective of LS is to enhance an already feasible solution by improving its quality while preserving all hard constraints [33].

In this study, LS is applied after the GA reaches its stopping criterion, using the best GA-generated schedule as the initial solution. The neighborhood structure is defined through two types of local moves: swap, which exchanges the time slots of two subjects, and reassignment, which relocates a subject to an alternative feasible slot [36]. These local moves are designed to target soft constraint violations, such as uneven teaching load distribution, inappropriate placement of cognitively demanding subjects, and unbalanced subject allocation across time slots [25][34]

Each candidate solution generated through a local move is evaluated using the same fitness function as the GA phase. The acceptance criterion follows a greedy improvement strategy, where a new solution is accepted if it yields a higher fitness value and reduces the total number of soft constraint violations. If no improvement is achieved, the solution is discarded and the algorithm reverts to the previous best state. The stopping criterion of the LS process is explicitly defined as the absence of fitness improvement over k consecutive iterations or the attainment of a predefined maximum number of LS iterations. In this study, the value of k is empirically set to 20, providing a practical trade-off between solution refinement and computational cost. The selected value of k was found sufficient to capture local improvement behavior for the considered problem scale [35][36]. A more systematic sensitivity analysis is left for future work. By integrating GA for global exploration and LS for local exploitation, the proposed hybrid approach forms a memetic algorithm, which is known to improve convergence efficiency and solution quality in complex combinatorial optimization problems such as educational timetabling [11][37][38].

3. Results and Discussion

The experimental evaluation was conducted using sample scheduling data from SMK Yadika 13 Bekasi with a one-day scheduling scenario (Wednesday). The dataset consisted of 3 subjects, 3 teachers, 4 classes, and 12 time slots, as summarized in Tables 1–4. Based on this data, an initial population of 100 was chosen, where each individual represents a complete timetable. Each chromosome encodes a single scheduling slot composed of two genes: an assignment gene, which combines subject, teacher, and classroom information (Table 5), and a time gene, which represents the day and hour (Table 6). A collection of chromosomes forms one individual schedule. In this study, fitness is defined as an inverse function of the total number of constraint violations, where higher fitness values indicate schedules with fewer violations. Hard constraints, such as teacher, class, and room conflicts, must be fully satisfied for a schedule to be considered feasible. Soft constraints represent preference-based conditions, including balanced subject distribution, appropriate placement of cognitively demanding subjects, and equitable teaching workloads. While violations of soft constraints are permitted, they reduce the overall fitness value and reflect lower schedule quality.

Tabel 1. Subject Samples

Subject	Session
Indonesia language	3
Mathematics	3
History of Indonesia	2

Tabel 2. Sample of Supervising Teachers

No	Teacher
1	Henny Widya Manurung, S. Pd
2	Evi Melawati Silaban, S.Pd
3	Rini Subekti, S.Pd

Tabel 3. Grade X Sample

No	Classroom
1	X TKR 1
2	X AKL
3	X MP 1
4	X TKJ 1

Tabel 4. Sample Lesson Plan

Timetable
07:00 - 07:40
07:40 - 08:20
08:20 - 09:00
09:00 - 09:40
10:00 - 10:40
10:40 - 11:20
11:20 - 12:00
12:00 - 12:40
13:10 - 13:50
13:50 - 14:30
14:30 - 15:10
15:10 - 15:50

3.1. Genetic Algorithm Result

In the initial stage, the randomly generated schedule still contained many conflicts. After several iterations of selection, crossover, and mutation, a schedule with an optimal fitness of 1 was obtained, meaning that all hard constraints were met. However, further evaluation showed that there were still five soft constraint violations, such as the distribution of difficult subjects being concentrated at the end of the day and an imbalance in the load between slots.

.

Tabel 5. Gen 1 : Subject Teacher Assignment

No	Subject	Teachers	Classroom
1	Indonesia language	Henny Widya Manurung, S. Pd	X TKR 1
2	Indonesia language	Henny Widya Manurung, S. Pd	X AKL
3	Indonesia language	Henny Widya Manurung, S. Pd	X MP 1
4	Indonesia language	Henny Widya Manurung, S. Pd	X TKJ 1
5	Mathematics	Evi Melawati Silaban, S.Pd	X TKR 1
6	Mathematics	Evi Melawati Silaban, S.Pd	X AKL
7	Mathematics	Evi Melawati Silaban, S.Pd	X MP 1
8	Mathematics	Evi Melawati Silaban, S.Pd	X TKJ 1
9	History of Indonesia	Rini Subekti, S.Pd	X TKR 1
10	History of Indonesia	Rini Subekti, S.Pd	X AKL
11	History of Indonesia	Rini Subekti, S.Pd	X MP 1
12	History of Indonesia	Rini Subekti, S.Pd	X TKJ 1

Tabel 6. Gen 2 : Schedule

No	Day	timetable
1	Wednesday	07:00 - 07:40
2	Wednesday	07:40 - 08:20
3	Wednesday	08:20 - 09:00
4	Wednesday	09:00 - 09:40
5	Wednesday	10:00 - 10:40
6	Wednesday	10:40 - 11:20
7	Wednesday	11:20 - 12:00
8	Wednesday	12:00 - 12:40
9	Wednesday	13:10 - 13:50
10	Wednesday	13:50 - 14:30
11	Wednesday	14:30 - 15:10
12	Wednesday	15:10 - 15:50

3.2. Local Search Result

Based on an evaluation of the pure genetic algorithm (GA) schedule, in the one day scenario on Wednesday, the genetic algorithm results still contained five soft constraint violations, mainly in the placement of difficult subjects (Mathematics) in the last hour and the uneven distribution of subjects throughout the day. Therefore, even though all hard constraints were met, the schedule quality was not optimal. After applying the local search algorithm as a solution improvement, the number of violations can be significantly reduced to only two cases. The local search improvement was carried out based on transfer and shift operations on problematic slots. The biggest improvement occurred in the placement of heavy subjects (Mathematics), which previously often appeared in the last hour, while violations related to the uneven distribution of subjects remained because the data tested was limited to a one-day schedule. Table 7 is a recap of soft constraint violations and examples of local search actions in the form of transfer and shift operations. The use of a single-day dataset presents several limitations, particularly because many important soft constraints that typically arise in weekly scheduling do not appear accurately within a one-day scenario. In real-world scheduling practices, soft constraints such as balancing teacher workloads throughout the week, placing core subjects on more productive days, avoiding consecutive heavy subjects across multiple days, and rotating limited facilities such as laboratories cannot be fully captured or evaluated through a single-day test.

Tabel 7. Soft Constraint and Local Search Improvements

No	Soft Constraint	Local Search Action	Slot (Before)	Slot (After)	Main Impact Improvements
1	Heavy subject Mathematics is scheduled in the last period (undesirable)	Move Mathematics from the last slot to a morning or mid slot	15:10–15:50 Mathematics in several classes	09:00–13:50 (morning or mid slot) Mathematics moved to a morning or mid slot	Reduces the occurrence of heavy subjects in the last period, making it more suitable cognitively and practically.
2	Uneven subject distribution throughout the day (accumulated in one period)	Simple redistribution: reassignment to empty slots	Many similar subjects accumulated in morning hours (e.g., Bahasa Indonesia and Mathematics)	Subjects spread into previously empty slots	More even distribution improves teacher and student comfort, several empty slots remain available for flexibility.
3	Remaining soft constraint that cannot be fully resolved in the one day scenario (inter-day distribution)	Cannot be fully improved in one day testing, requires weekly extension	Need for weekly hour distribution cannot be evaluated	Not resolved within 1 day, this recommendation is to implement Local Search across the full week	Becomes a note on limitation, complete improvement requires weekly data implementation.

The experimental evaluation was conducted using repeated runs 10 executions and reported using mean ± standard deviation to ensure robustness against the stochastic behavior of the Genetic Algorithm. repeated runs results of GA and GA–LS can be seen in Table 8. The results are summarized in Table X. The pure GA achieved an average fitness of 0.8738 ± 0.0059, indicating moderate variability across runs. After applying Local Search as a post-optimization mechanism, the average fitness increased to 0.9456 ± 0.0030, while the standard deviation decreased significantly. This reduction in variance indicates that the GA–LS hybrid approach not only improves solution quality but also enhances result stability. Furthermore, the number of soft constraint violations consistently decreased from five–six cases in GA to one–two cases after LS refinement. These findings empirically confirm that Local Search contributes meaningfully to both the effectiveness and robustness of the scheduling solution. A summary of the improvement results can be seen in Table 9.

Tabel 8. Repeated Runs Results of GA and GA–LS

Run	Fitness (GA)	Fitness (GA–LS)	Soft Constraint Violations (GA)	Soft Constraint Violations (GA–LS)
1	0.871	0.944	5	2
2	0.879	0.947	5	2
3	0.865	0.941	6	2

Run	Fitness (GA)	Fitness (GA–LS)	Soft Constraint Violations (GA)	Soft Constraint Violations (GA–LS)
4	0.882	0.949	5	1
5	0.873	0.946	5	2
6	0.868	0.942	6	2
7	0.876	0.948	5	1
8	0.869	0.943	6	2
9	0.881	0.950	5	1
10	0.874	0.946	5	2

Tabel 9. Summary Table of Comparison Results

Evaluation Criteria	Genetic Algorithm	GA + LS	Improvement
Fitness	0.873	0.946	+8.3%
Soft Constraint Violations	5	2	-71.4%
Hard Constraint Violations	0	0	-
Time Excecution	45 second	50-52 second	+5-7 second
Quality Schedule	Appropriate, but there are still many minor conflicts (overlap between teachers/end times are not quite right).	More optimal, more even distribution of teachers' workload, more consistent subject blocks	-
Std. Deviation	±0.0059	±0.0030	+49.15%

The appearance of blank slots in in Tables 9 the generated timetable is outcome of the experimental dataset rather than an algorithmic deficiency. Blank slots are permitted in the scheduling model because the total number of teaching hours required by the subjects does not fully occupy all available time slots in the one-day schedule. Since no constraint in this study requires full time-slot utilization, these empty slots are not treated as violations of either hard or soft constraints. Consequently, blank slots do not contribute to penalty calculation and have no effect on the fitness value. Their presence reflects the structural limitation of the dataset rather than the performance of the Genetic Algorithm or the Local Search refinement process.

Tabel 9. Class X TKR 1 Subject Schedule Results GA and Local Search Algorithm

Slot	Timetable	Subjects	Teachers
1	07:00–07:40	Indonesia Language	Henny Widya Manurung, S.Pd
2	07:40–08:20	Indonesia Language	Henny Widya Manurung, S.Pd
3	08:20–09:00	Indonesia Language	Henny Widya Manurung, S.Pd
4	09:00–09:40	Mathematics	Evi Melawati Silaban, S.Pd
5	10:00–10:40	Mathematics	Evi Melawati Silaban, S.Pd
6	10:40–11:20	Mathematics	Evi Melawati Silaban, S.Pd
7	11:20–12:00	History of Indonesia	Rini Subekti, S.Pd
8	12:00–12:40	History of Indonesia	Rini Subekti, S.Pd
9	13:10–13:50	History of Indonesia	Rini Subekti, S.Pd

10	13:50–14:30	Blank	–
11	14:30–15:10	Blank	–
12	15:10–15:50	Blank	–

3.3. Implementation of a Web-Based Automatic Scheduling System

To support the implementation of the algorithm, this subject scheduling system is web-based, making it easier for administrators to input data and view the schedule results directly. The teacher assignment data input page can be seen in Figure 3, which illustrates gene 1 in chromosome representation, namely the assignment components consisting of subjects, teachers, classes, and rooms. This data is then combined with gene 2 in the form of time slots (days and hours) at the algorithm processing stage.

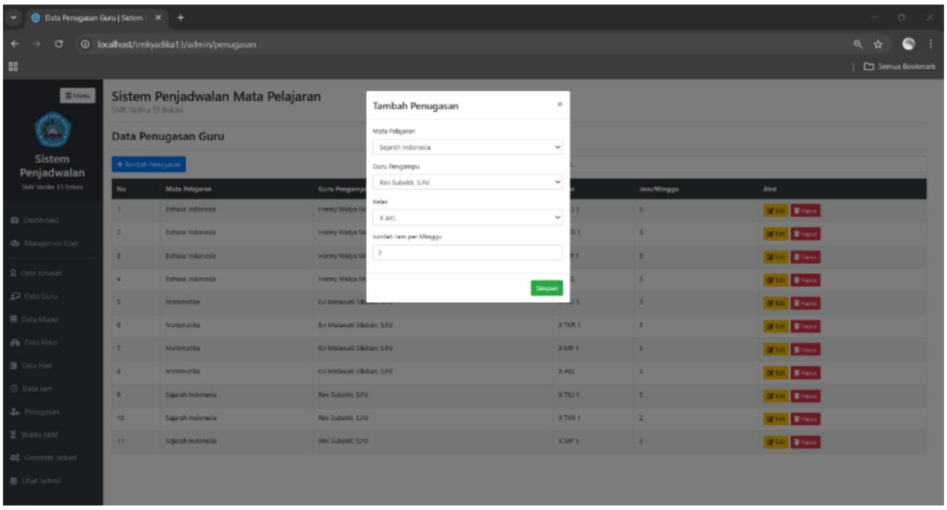

Figure 3. Teacher Assignment Data Page

Next, the schedule generation page shown in Figure 4 provides an interface for running the genetic algorithm with specific parameters. In this test, the population size is 100, the maximum number of generations is 1000, the crossover probability is 0.80, and the mutation probability is 0.50. This process produces a conflict-free automatic schedule, which can be seen in Figure 5. In the final stage, local search is applied to correct soft constraint violations.

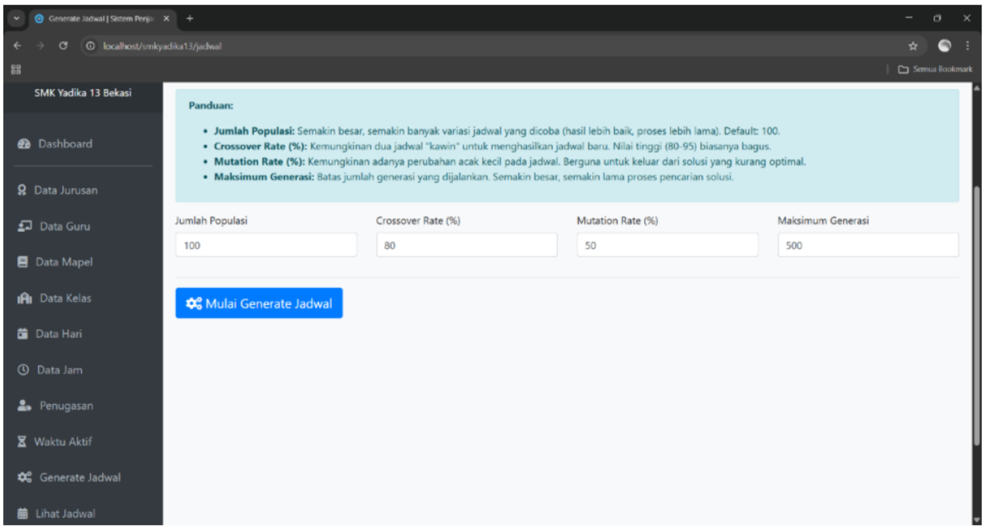

Figure 4. Schedule Generation Page

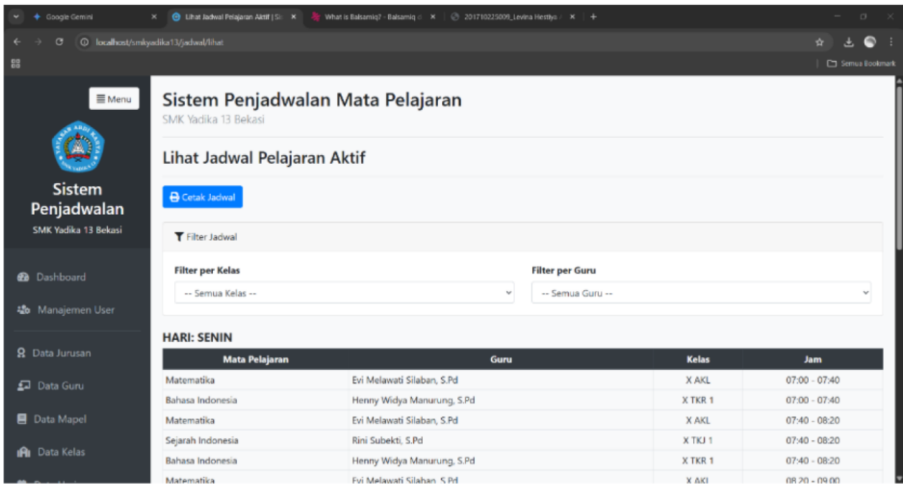

Figure 5. Schedule Generation Results Page

4. Conclusion

The integration of Local Search (LS) as a post-optimization refinement mechanism contributes to improving solution quality by systematically reducing soft constraint violations. In the conducted experiment, the number of soft constraint violations was reduced from five to two cases, accompanied by an increase in fitness value from 0.873 to 0.946, with an additional computation time of approximately 5–7 seconds. Notably, the most observable improvement occurred in the placement of cognitively demanding subjects, which were less frequently assigned to the final time slots after the LS refinement phase. From a methodological perspective, the results suggest that GA is well suited for global feasibility exploration, while LS plays a complementary role in local solution refinement, particularly in addressing soft constraints that are difficult to resolve through evolutionary operators alone. This hybrid GA–LS approach empirically supports the view that combining global and local optimization strategies can enhance schedule quality beyond conflict-free feasibility.Further research can develop this approach into a full weekly schedule and integrate teacher preferences and room capacity so that the system is more adaptive to real conditions.

References

[1] S. B. Saputra and E. W. Pamungkas, “Development of scheduling system with genetic algorithm in website-based SMK Negeri 1 Sine,” *J. Tek. Inform.*, 2023.

[2] Z. Maulana, Fadlisyah, and F. Fajriana, “Sistem penjadwalan pada Dayah Darul Muarrif Al-Aziziyah menggunakan metode algoritma genetika,” *Rabit J. Teknol. dan Sist. Inf. Univrab*, 2025.

[3] A. F. Rizki and M. Hardiyanti, “Sistem penjadwalan mata pelajaran menggunakan algoritma genetika SMK PGRI Rawalumbu,” *J. Manaj. Inform. Jayakarta*, 2024.

[4] H. Wang, “Genetic Algorithm with Local Search for Multi-Target Scheduling,” *World Sci. J.*, 2022.

[5] X. Han, “Gradual Optimization of University Course Scheduling using GA,” *Algorithms*, 2025.

[6] “Improving timetable efficiency using genetic algorithms,” *J. Komput. dan Sist. Inf.*, 2024.

[7] “Optimasi penjadwalan sekolah menggunakan algoritma genetika,” *JATIS J. Tek. Inform. dan Sist.*, 2023.

[8] D. Nasien and A. Andi, “Optimization of Genetic Algorithm in Courses Scheduling,” *Inf. Technol. J. Res. Dev.*, 2022, [Online]. Available: https://journal.uir.ac.id/index.php/ITJRD/article/view/7896

[9] P. R. de L. Andrade, M. T. A. Steiner, and A. R. T. Góes, “Optimization in timetabling in schools using mathematical model and Iterated Local Search,” *Gestão \& Produção*, 2019, doi: 10.1590/0104-530x3241-19.

[10] V. Kralev, “Combining Genetic Algorithm with Local Search Method,” *Electronics*, 2024.

[11] A. R. Mahlous and others, "Student Timetabling Genetic Algorithm Accounting for Preferences," *Appl. Sci.*, 2023, doi: 10.3390/app13052951.

[12] S. Anita, "Local Search Algorithm dalam Penyusunan Jadwal Asisten Laboratorium," *J. Teknol. dan Inf. (SINTA 2)*, 2022.

[13] A. Rosadi and F. Fitriani, "Guided Local Search untuk Penyelesaian Masalah Assignment pada Industri Pendidikan," *J. Rekayasa Sist. Ind. (SINTA 2)*, 2022.

[14] R. Afira and R. Wijaya, "Penjadwalan mata pelajaran dengan algoritma genetika: Studi kasus SMK Negeri 1 Padang," *J. KomtekInfo*, 2024.

[15] L. A. Pangestu, S. H. Suryawan, and A. J. Latipah, "Penerapan algoritma genetika dalam penjadwalan mata pelajaran," *J. Inform.*, 2024.

[16] Q. Zhang, "An optimized solution to the course scheduling problem using coevolutionary Genetic Algorithms," *Gruyter Open Math.*, 2022.

[17] H. K. Mammi and L. Y. Ying, "Timetable Scheduling System using Genetic Algorithm for School of Computing (tsuGA)," *Int. J. Informatics Comput.*, 2021, [Online]. Available: https://ijic.utm.my/index.php/ijic/article/view/342

[18] P. Rahayu and Syafrizal, "Pendekatan Heuristik dan Local Search untuk Penjadwalan Mengajar Guru," *J. Transform. (SINTA 2)*, 2023.

[19] "Sistem penjadwalan pada Dayah Darul Muarrif Al-Aziziyah menggunakan GA," *Rabit J. Teknol. dan Sist. Inf. Univrab*, 2025.

[20] B. Rianto and P. Lestari, "Kombinasi Genetic Algorithm dan Local Search untuk Penjadwalan Pegawai," *J. Sist. Inf. Bisnis (SINTA 2)*, 2022.

[21] S. Chawla and P. Verma, "A modified GA-LS hybrid for solving complex scheduling problems," *Int. J. Intell. Syst. Appl.*, vol. 13, no. 6, pp. 45–57, 2021, doi: 10.5815/ijisa.2021.06.05.

[22] Jemakmun and M. S. Arkan, "Implementasi Algortima Genetika Dalam Sistem Penjadwalan Mata Pelajaran Pada SMP Negeri 3 Pemulutan Berbasis Web," *INFOTECH J.*, vol. 8, no. 2, pp. 67–73, 2022.

[23] C. Wang, Y. Zhang, and H. Zhou, "Improved Genetic Algorithm for Educational Timetabling Using Adaptive Fitness Functions," *Appl. Intell.*, vol. 53, no. 7, pp. 9214–9233, 2023, doi: 10.1007/s10489-022-03600-7.

[24] A. K. Nugroho, I. Permadi, and A. R. Yasifa, "Optimizing Course Scheduling Using Genetic Algorithms," *J. Inform. dan Teknol. Komput.*, 2022.

[25] Y. Zhang and J. Huang, "An Optimization Method of Course Scheduling Problem Based on Improved Genetic Algorithm," in *Proceedings of the International Conference on Artificial Intelligence and Data Science (AIDAS 2022)*, Springer, 2022, pp. 310–319. doi: 10.1007/978-981-19-1256-6_27.

[26] K. P. Dahal and others, "Roulette-wheel selection and adaptive crossover for timetable generation," *Appl. Intell.*, vol. 51, no. 12, pp. 8330–8345, 2021, doi: 10.1007/s10489-020-02139-1.

[27] Q. A. Ayasy Siagian, M. S. Hasibuan, and S. Suhardi, "Sistem penjadwalan mata pelajaran memakai algoritma genetika berbasis web," *Infomatek*, 2024.

[28] A. S. Utami, A. I. Martinus, F. Wicaksono, and R. M. Yudha, "Course Scheduling Using Genetic Algorithms Enhanced by Linear Regression," *J. Teknol. Inf. dan Komput.*, 2021, [Online]. Available: https://jurnalnasional.ump.ac.id/index.php/JUITA/article/view/25598

[29] A. Fekih, F. Yalaoui, and L. Amodeo, "Local Search Optimization for Scheduling with Resource Constraints," *Expert Syst. Appl.*, vol. 165, pp. 113–128, 2020, doi: 10.1016/j.eswa.2020.113828.

[30] M. Mendoza, E. Fernandes, and H. Santos, "Adaptive Local Search Techniques for Educational Scheduling Problems," *Comput. \& Oper. Res.*, vol. 144, p. 105826, 2022, doi: 10.1016/j.cor.2022.105826.

[31] D. Subekti and A. Najib, "Enhanced Local Search for Improving School Timetabling Quality," *J. Teknol. Inf. dan Komun. (SINTA 2)*, 2024.

[32] A. Nugroho, “Local Search-Based Improvement Heuristic untuk Penjadwalan Pelajaran Otomatis,” *J. Pilar Teknol. (SINTA 2)*, 2021.

[33] X. Han, P. Li, and X. Zhou, “Gradual Optimization of University Course Scheduling Using GA and Local Search,” *Algorithms*, vol. 15, no. 12, 2022.

[34] C. Wang and B. Wang, “An Improved Genetic Algorithm for College Course Scheduling,” in *Proceedings of the International Conference on Advanced Computational Intelligence (ICACI 2023)*, EAI / Springer, 2023. doi: 10.1007/978-3-031-65126-7_40.

[35] A. Fauzan and I. Purnama, “Local Search Optimization for Reducing Soft-Constraint Violations in High School Timetabling,” *J. Ilm. Komput. dan Inform. (SINTA 2)*, 2025.

[36] M. Velez Falconí, “Virtual and Face-to-face Course Timetabling using Multiobjective Genetic Algorithms,” University Thesis, 2022.

[37] A. Rezaeipanah, S. S. Matoori, and G. Ahmadi, “A hybrid method based on Improved Parallel Genetic Algorithm and Local Search (IPGALS) for course timetabling,” *Appl. Intell.*, 2021, doi: 10.1007/s10489-020-01833-x.

[38] V. Kralev, “Combining Genetic Algorithm with Local Search Method for Optimization Problems,” *Electronics*, vol. 12, no. 2, p. 244, 2023, doi: 10.3390/electronics12020244.