Implementation of a Web Application Based on Genetic Algorithms for the Preparation of Academic Schedules in an Educational Unit

Georgette Alexia Guevara Sánchez[1], Jorge Aldair Sánchez Rojas[1*], Igor Aguilar-Alonso[1,2]

[1]Escuela Profesional de Ingeniería de Sistemas, Universidad Nacional Tecnológica de Lima Sur
[2]Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos
Email: 1823010277@untels.edu.pe

Currently, the management of academic schedules in educational institutions faces significant challenges, as traditional approaches often prove to be inefficient and lead to conflicts in resource allocation. To address this issue, the development of a web application that utilizes genetic algorithms as the foundation for creating academic schedules is proposed. Through a literature review, key elements were identified, including scheduling approaches, web technologies, and user interface components. The conclusion highlights the suitability of genetic algorithms as the primary approach, supporting their implementation in a web application based on modern technologies. Features such as efficient resource allocation, conflict management, and the generation of optimal schedules are proposed. This approach has the potential to significantly enhance academic schedule management in educational institutions, promoting efficiency and satisfaction among all involved stakeholders.

Keywords: Genetic Algorithms, Scheduling, Web Application.

1. Introduction

Currently, there is a significant concern in the education sector due to the constant delay in the scheduling process, attributed to the increasing number of students. Additionally, resources must be allocated properly to avoid economic losses; hence, greater attention needs to be given to this process.

According to Zaqaibeh's research [1], scheduling preparation for the responsible departments is challenging due to the growing constraints imposed by Jadara University. In the Philippines, manual schedule preparation is a complicated task, as outlined by Austero [2]. They face numerous situations and possibilities to find the optimal schedule and examining all these alternatives is often impractical.

In some situations, the complexity of this problem arises because institutions are too large, requiring consideration of numerous variables and constraints. Handling this manually consumes a significant amount of time in the academic field, as mentioned by Marwah regarding universities in Saudi Arabia [3].

In countries like Iraq, partially optimal results are achieved despite various preliminary efforts. Due to the enormity of the task, the most optimal schedule is often not attained [4]. Peru faces similar challenges in schedule creation, as evidenced by the case of the Private University of the North. Despite the importance of class schedule generation in achieving strategic objectives, the most optimal solution has not yet been found due to the large number of students, classrooms, and courses opened each academic semester [5].

The methods commonly used for schedule generation are mostly manual, resulting in a lengthy process that does not guarantee the most optimal result. This can lead to monetary and resource losses, limiting institutions in their objectives.

The research focuses on the current situation of schedule creation in a Private Educational Institution. The objective is to implement a web application that produces the most optimal schedule in terms of resources and time, using genetic algorithms.

The structure of this paper have 5 sections: Section (I) introduces the problem of schedule creation in institutions. Section (II) defines the background of the research, explaining concepts related to the research proposal. Section (III) includes the literature review of academic articles related to the research topic. Section (IV) describes the results found in the previous section. Section (V) analyzes the proposals and responds to research questions. Finally, Section (VI) presents the conclusions.

2. Background

A. Scheduling.

Author Yasuki [6] points out that scheduling is an integral part of the efficiency of an entire system, and the more constraints and variables there are, the more it becomes complex and extremely difficult. Yunus [7], regarding scheduling in universities and institutions, considers the requirements and limitations, aiming to optimize the resources they have (classrooms, professors, students) in this process.

B. Genetic Algorithms.

Author Tawfiq [8], mentions that a genetic algorithm is a solution inspired by nature and demonstrates a significant advantage in generic environments. In Kumar's research [9], it is described as an intelligent research technique used to solve NP-hard problems, mimicking the process of natural selection by starting with a population of genes and selecting the most optimal one over generations.

C. Web Application.

Freyre [10], defines it as software that runs in a web browser. Companies need to share information and provide services remotely, and that's why they use web applications to communicate with clients securely when necessary. Among the common functions of websites are shopping carts, product search and filtering, instant messaging, and social media feeds. They share the same design as web applications.

3. Literature Review

In this article, the primary databases utilized were Science Direct, Scopus, and IEEE. The literature review methodology proposed by Kitchenham [11], was employed for this research, consisting of three phases: Planning the review, developing the review and review results.

D. Planning the Review

The research development questions are as follows:

Q1: What programming languages are used for scheduling with genetic algorithms?

Q2: What types of variants of genetic algorithms are employed for scheduling?

Q3: What approaches do genetic algorithms take in academic scheduling?

TABLE I. CRITERIA TABLE

Selection Criteria	Exclusion Criteria
Articles published from 2018 to 2023.	Articles published outside the range 2018 to 2023.
Articles related to the topic of genetic algorithms for scheduling.	Articles published on websites or non-indexed journals, videos.
Publications made in conferences and indexed journals.	Articles unrelated to the research topic

E. Developing the Review

The literature found has been obtained based on the selection criteria from the previous step. To make the selection, it was necessary to conduct a preliminary review to determine if the content is relevant to the research.

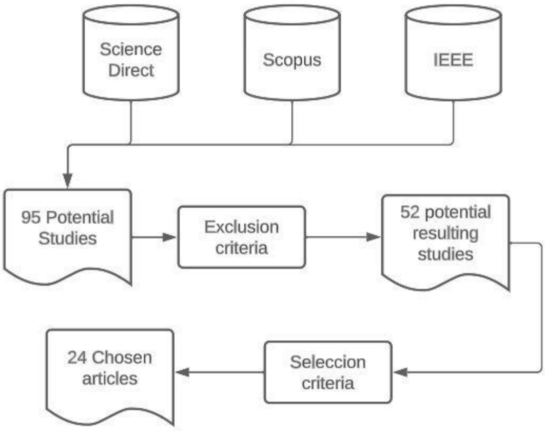


Fig. 1 Review Development

F. Review Results

The findings derived from the search for relevant information for the research were obtained from various studies, of which a total of 20 were selected. Table II presents the number of studies retrieved from the databases.

TABLE II. NUMBER OF STUDIES

Database	Potential eligible studies	Selected studies
Science Direct	12	6
Scopus	32	14
IEEE	8	4
Total	52	24

4. Found Results

G. Programming Languages for Developing Schedules Based on Genetic Algorithms.

Alansari [3], designed the genetic algorithm as an Application Programming Interface (API), written in the JAVA programming language. This facilitated the integration of services that each university possesses. Hedar [12], used various frameworks to create the genetic algorithm but primarily employed the C++ programming language, significantly improving precision.

Gupta & Sinha [13], in their research on schedule creation, investigated formulating a mathematical model and complementing it with a simulation system developed in the C language, incorporating genetic algorithms to obtain solutions under uncertainties.

Albedhalim [14], conducted tests for his genetic algorithm personally on a computer, using Java 6 as the programming language and PostgresQL 9.3 as the database management system.

Alomari [15], implemented the proposed genetic algorithm using the MySQL database and the PHP programming language, allowing for the coverage of many students and courses as the algorithm's population.

Arias [16], programmed the genetic algorithm solution in MATLAB, while the factorial design was developed in the statistical software Minitab, reaching a solution for the class schedules of the academic periods 2018-1 and 2018-2 for the Industrial Engineering program at UIS.

Oktavia [17], proposed a genetic algorithm to solve the scheduling problem at her university and opted for Python to display the results of the teachers' schedules on a specific day and for a specific class.

Ying & Kutty [18], assisted in efficiently creating schedules through a genetic algorithm. For this, they created a web application that functions to create academic schedules using the Java programming language.

Reid [19], in his research, created a genetic algorithm to generate schedules for the National University of Ireland, considering the availability of teachers, students, or locations. The genetic algorithm runs efficiently on a Java application server, presented through a web API to control the algorithm's execution.

Aminu [20], in his article indicates that due to the problems encountered in manually creating schedules because of the significant time and resource effort involved, he aims to design an efficient automated means to eliminate this tedious process. He designed a genetic algorithm in Java with graphical user interfaces (GUI) and used Swing as the database.

H. Types of Most Used Genetic Algorithm Variants for Educational Institutions' Scheduling

The study by Matías, Fajardo & Medina [21], presents a hybrid variant of the genetic algorithm to address scheduling and teaching load management. This algorithm combines the global search of genetic algorithms with the adjusting capability provided by local search methods. Global search is performed with the genetic algorithm, while local refinement is carried out through local search methods. Additionally, penalty techniques and flexible constraints are incorporated to ensure solution feasibility and compliance with problem requirements, including penalties for workload overload, regulation of teacher preparation, and control of classroom usage. Together, this hybrid variant of the genetic algorithm ensures feasible and optimal solutions that adapt to the needs of scheduling in the educational context.

In his article [22], Wang presents an adaptive and hybrid genetic algorithm to address course scheduling in academic contexts. The variant of the genetic algorithm presented in this article combines adaptive flexibility with the effectiveness of local search. It generates initial solutions and improves them in real-time, considering multiple objectives such as daily adequacy, course combination, and faculty satisfaction, to provide optimal solutions. In summary, Wang's approach is a powerful combination of adaptive genetic algorithms and local search to solve challenges in course scheduling in the academic field.

Saptarini [23], propose an Adaptive Genetic Algorithm (AGA) to address the complex problem of scheduling in secondary schools. The AGA uses a flexible mutation rate adjusted according to the current population's fitness to effectively explore the solution space and avoid local optima. Its fitness function evaluates the quality of generated schedules, considering both hard and soft constraints. Schedules that violate hard constraints are discarded, and those violating soft constraints are penalized proportionally. This approach aims to maximize fitness, resulting in schedules that meet all problem constraints, achieving high-quality solutions.

According to Abayomi [24], proposes an effective hybrid variant of the genetic algorithm to solve the university exam scheduling problem. This system combines genetic algorithms with tabu search memory and the "course sandwiching" technique. Genetic algorithms generate initial solutions and evolve the solution population, while tabu search memory is used to explore the solution space and improve the current solution's quality. The "course sandwiching" technique is employed to address constraints related to course conflicts and classroom availability.

According to Badoni [25], a hybrid variant of the genetic algorithm is proposed in combination with local search and tabu search to solve university course scheduling problems. The genetic algorithm is described as an optimization technique based on natural selection and evolutionary theory, finding the optimal solution to the complex problem of scheduling. In this study, a population of candidate solutions is created, and genetic operators such as selection, crossover, and mutation are applied to generate new solutions. The fitness of each solution is then evaluated, and the best ones are selected for the next generation. This process is repeated until a satisfactory solution is found or an iteration limit is reached.

In the article by Hafsa [26], the multi-objective variant MOEA (Mult objective evolutionary algorithm) of genetic algorithms is employed to address a professional course scheduling problem at Mandarine Academy. This multi-objective optimization problem involves five objectives and 18 constraints (both hard and soft). The objectives include minimizing schedule conflicts, maximizing schedule quality, reducing overtime hours, decreasing working days, and minimizing classroom changes. The article also introduces custom genetic operators and compares them with classical genetic operators to assess their effectiveness.

The article by Kakkar [27], mentions genetic algorithms as an intelligent technique to address hard problems. The article focuses on using genetic algorithms to generate class schedules that meet specific constraints and objectives, highlighting their effectiveness in solving complex scheduling problems. Previous related research, such as the use of hybrid variants combining genetic algorithms with simulated annealing and tabu search, is also mentioned, showing promise in this field. Overall, the article underscores the potential of evolutionary algorithms, such as genetic algorithms, to address challenges in class scheduling and suggests future improvements in this area.

Vrielink [28], in their article on scheduling research, rely on the best individual solution in the population space. The article also mentions that memetic algorithms are a hybrid variant of genetic algorithms and are mainly based on local search methods. The article notes that finding the right balance between exploiting local search and exploring through population-based methods in the search space is a challenge.

Arias & Mora in their article [16], provide a comprehensive solution to the university exam scheduling problem that can easily adapt to different contexts. The proposed solution uses a genetic algorithm and a conflict matrix to develop the schedule. The conflict matrix is employed to identify courses that are often taken together, and this is used in the data mining process to create the exam schedule. The genetic algorithm is used to optimize the schedule by generating a population of possible solutions and selecting the best ones based on their fitness. Additionally, the proposal also uses a memetic algorithm, forming a hybrid variant of genetic algorithms together with hill climbing, for additional schedule optimization. The solution is evaluated based on its compatibility with the actual schedule developed for the current period.

5. Approaches of Genetic Algorithms in Academic Scheduling.

In the research conducted by Siang [29], which aimed to review all methods used to solve high school scheduling problems, genetic algorithms are mentioned to focus on heuristic optimization and efficiently utilize resources.

Thepphakorn & Pongcharoen [30], propose a genetic algorithm to solve scheduling problems, emphasizing the search and resource utilization approach that can be obtained compared to other algorithms.

Diveev [31], indicates that a genetic algorithm allows finding a solution close to the optimal one, and its precision increases the longer the genetic algorithm is executed. Furthermore, it can provide more than one optimal solution in programming, indicating its heuristic optimization nature.

In the article developed by Ruji [2], a genetic algorithm is implemented to optimize how teachers' work is allocated and how study rooms are used, optimizing resources (rooms, teachers, money, courses), and improving time efficiency by 89%.

Tawfiq [31], aimed to reduce the costs associated with exam scheduling at Al-Hussein Bin Talal University in Jordan. The conclusion is that a higher number of enrolled students in the semester does not negatively impact, effectively utilizing the allocated resources.

In the research conducted by Hosny & Al-Olayan [32], it is explained that there are many constraints considered when creating academic schedules, depending on the university. In response to this situation, a classic genetic algorithm is proposed to optimize two key points: room allocation and supervisor assignment, both being university resources. As a result, the algorithm positively affects the creation of exam schedules at the university.

6. Analysis of Results

Based on the results of the literature review, this section provides an analysis of the findings and answers the questions formulated in Chapter IV.

I. Programming Languages for Genetic Algorithm-Based Scheduling.

The obtained results are detailed below in Fig. 2, where the following is identified:

Java proves to be the most widely used, representing 50% of the programming languages.

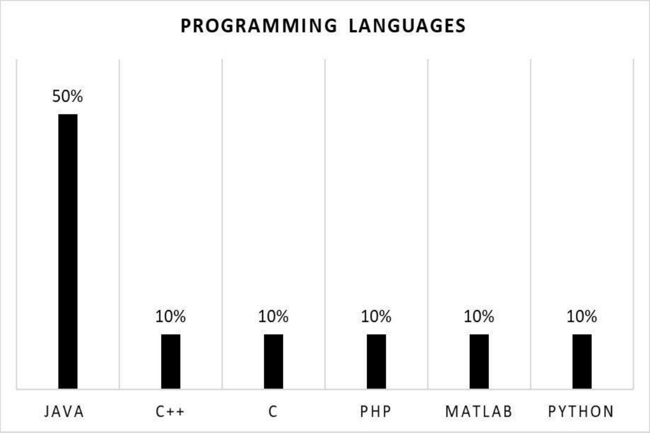


Fig. 2 Results of programming languages to create schedules based on genetic algorithms

J. Types of Genetic Algorithms Used for Scheduling

The obtained results are detailed below in Fig. 3, where the following is identified:

The most used genetic algorithm variant is the hybrid, representing 63.64% of all variants found in the literature review.

Another result we found was that the hybrid variant of the genetic algorithm is most frequently combined with a local search algorithm.

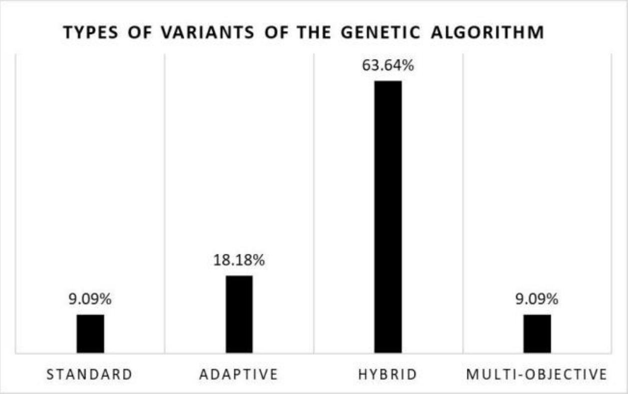


Fig. 3 Results of the types of variants of genetic algorithms used in the preparation of schedules

K. Approaches of Genetic Algorithms in Academic Scheduling.

The obtained results are described below in Fig. 4, where the following is identified:

The most used approach is the "use of resources," representing 62.50% of the identified approaches.

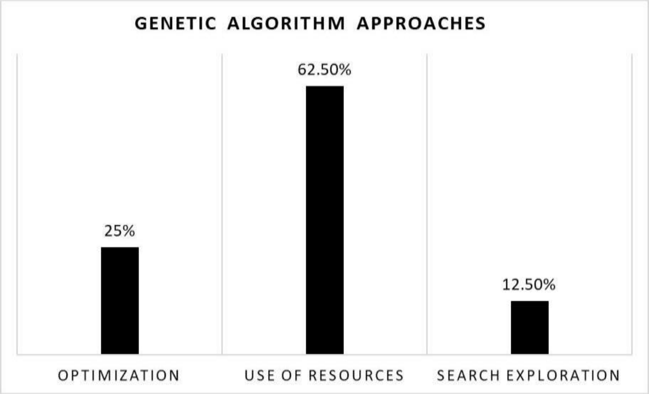


Fig. 4 Results of genetic algorithm algorithm approaches when drawing up academic schedules

7. Architecture Proposal

As a proposal, we have designed a web-based application using a genetic algorithm capable of creating schedules based on the specified generations, as shown in Fig. 5.

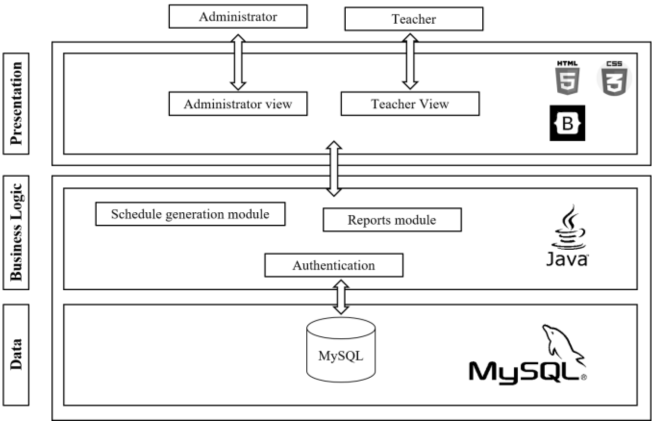


Fig. 5 Web application architecture

In the presentation layer, the interaction of the user (administrator or teacher) with the various views provided for the use of web applications is shown. To create the views, we will use HTML, CSS, and JavaScript.

In the business logic layer, user interactions are received from the view and coordinate actions between the view and the model. Additionally, it contains the logic to perform operations (Authentication, Scheduling Module, Reports Module) in the model according to the view's requests. For this, we will use the Java programming language.

In the data layer, the model is primarily located, representing the data structure. It will also handle any interaction with the MySQL database.

To implement this web application, the following table provides a description of the system.

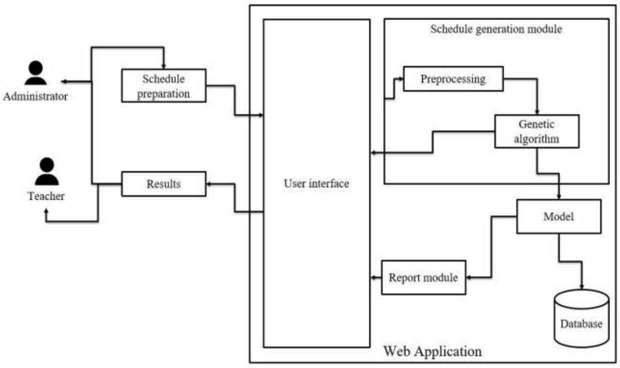
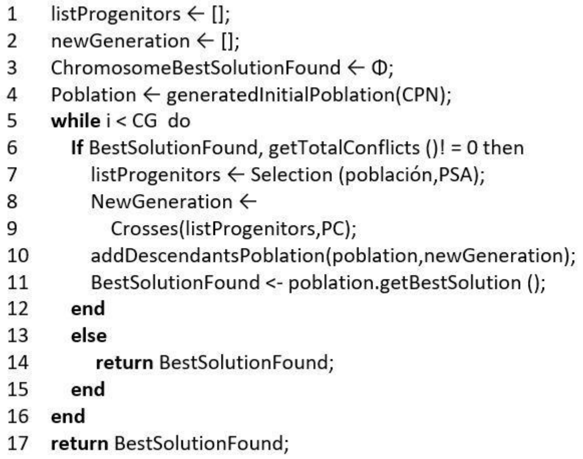


Fig. 6 Application Description

In Fig. 6, it is explained that only the administrator can generate schedules through the user interface, and through the scheduling module, the genetic algorithm will be used to return the generated schedule. It is observed that the database provides information to both the genetic algorithm and the reporting module. In the latter, results are displayed through the user interface to the administrator and the teacher.

For the genetic algorithm, we will use the following code shown in Fig. 7.



```
listProgenitors ← [];
newGeneration ← [];
ChromosomeBestSolutionFound ← Φ;
Poblation ← generatedInitialPoblation(CPN);
while i < CG  do
  If BestSolutionFound, getTotalConflicts ()! = 0 then
    listProgenitors ← Selection (población,PSA);
    NewGeneration ←
      Crosses(listProgenitors,PC);
    addDescendantsPoblation(poblation,newGeneration);
    BestSolutionFound <- poblation.getBestSolution ();
  end
  else
     return BestSolutionFound;
  end
end
return BestSolutionFound;
```

Fig. 7 Genetic Algorithm Code

We will employ the function of a hybrid algorithm, adding to the code shown a basic search algorithm that aids in facilitating schedule creation. This genetic algorithm is focused on optimizing the use of resources such as time, classrooms, and teacher availability to distribute them efficiently.

8. Conclusion

In conclusion, genetic algorithms are effective in optimizing school schedules by minimizing conflicts and satisfying the preferences of students and teachers. They use genetic representation and simulate evolutionary processes to enhance the quality of schedules and reduce human errors.

Genetic algorithms are versatile and can adapt to various constraints and optimization criteria, improving the satisfaction of all parties involved in education. Furthermore, empirical results support their effectiveness, demonstrating significant improvements compared to traditional approaches. This not only increases efficiency but also enhances the educational experience.

In summary, genetic algorithms are a valuable tool in educational management for optimizing school schedules and promoting future research in artificial intelligence applied to educational planning.

References

1. B. Zaqaibeh, "An Enhanced Population Selection Algorithm for Timetabling System," Information Sciences Letters, vol. 12, no. 1, pp. 325–334, Jan. 2023, doi: 10.18576/isl/120127.
2. L. D. Austero, R. P. Medina, A. M. Sison, and J. B. Matias, "Optimizing Faculty Workloads and Room Utilization using Heuristically Enhanced WOA Graduate Programs Technological Institute of the Philippines Quezon City, Philippines," IJACSA) International Journal of Advanced Computer Science and Applications, vol. 13, no. 11, 2022, Accessed: Oct. 12, 2023. [Online]. Available: www.ijacsa.thesai.org
3. M. M. Alansari, "Optimized Automatic Course Timetabling Service Architecture for Integration with Vendor Management Systems," IJACSA) International Journal of Advanced Computer Science and Applications, vol. 13, no. 10, 2022, Accessed: Oct. 12, 2023. [Online]. Available: www.ijacsa.thesai.org
4. I. A. Abduljabbar and S. M. Abdullah, "An evolutionary algorithm for solving academic courses timetable scheduling problem," Baghdad Science Journal, vol. 19, no. 2, pp. 399–408, Jan. 2022, doi: 10.21123/BSJ.2022.19.2.0399.
5. N. Salvador, L. Asesor, M. Rolando, and J. B. Beltrán, "Algoritmo genético en la asignación de horarios de un instituto de educación superior," Universidad Privada del Norte, Mar. 2021, doi: 10.18687/LACCEI2020.1.1.518.
6. Y. Shima, R. A. Kadir, and F. H. Ali, "A Novel Approach to the Optimization of a Public Bus Schedule Using K-Means and a Genetic Algorithm," IEEE Access, vol. 9, pp. 73365–73376, Jan. 2021, doi: 10.1109/ACCESS.2021.3080508.
7. Y. Demır and C. Çelık, "Hierarchical mathematical modelling approach for a case study in university timetabling," Sigma Journal of Engineering and Natural Sciences, vol. 37, no. 2, pp. 323–339, Jan. 2019.
8. T. Alrawashdeh, K. G. Al-Moghrabi, and A. M. Al-Ghonmein, "A profiling-based algorithm for exams' scheduling problem," International Journal of Electrical and Computer Engineering, vol. 13, no. 5, pp. 5483–5490, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5483-5490.
9. M. K. Kakkar, J. Singla, N. Garg, G. Gupta, P. Srivastava, and A. Kumar, "Class Schedule Generation using Evolutionary Algorithms," in Journal of Physics: Conference Series, IOP Publishing Ltd, Aug. 2021. doi: 10.1088/1742-6596/1950/1/012067.
10. F. M. Juan, C. P. Carolina, C. O. Patricia, and P. V. Carlos, "Collaborative desing in web aplication development to improve tuberculosis diagnostic," Indonesian Journal of Electrical Engineering and Computer Science, vol. 30, no. 3, pp. 1821–1828, Jun. 2023, doi: 10.11591/ijeecs.v30.i3.pp1821-1828.

11. B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," Information and Software Technology, vol. 51, no. 1. pp. 7–15, Jan. 2009. doi: 10.1016/j.infsof.2008.09.009.
12. A.-R. Hedar, W. Deabes, M. Almaraashi, and H. H. Amin, "Evolutionary Algorithms Enhanced with Quadratic Coding and Sensing Search for Global Optimization," Mathematical and Computational Applications 2020, Vol. 25, Page 7, vol. 25, no. 1, p. 7, Jan. 2020, doi: 10.3390/MCA25010007.
13. S. Gupta and S. Sinha, "Academic Staff planning, allocation and optimization using Genetic Algorithm under the framework of Fuzzy Goal Programming Academic Staff planning, allocation and optimization using Genetic Algorithm under the framework of Fuzzy Goal Programming," Procedia Comput Sci, vol. 172, pp. 900–905, 2020, doi: 10.1016/j.procs.2020.05.130.
14. E. A. Abdelhalim and G. A. El Khayat, "A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA)," Alexandria Engineering Journal, vol. 55, no. 2, pp. 1395–1409, Jun. 2016, doi: 10.1016/J.AEJ.2016.02.017.
15. K. Alomari, O. Almarashdi, A. Marashdh, and B. Zaqaibeh, "A New Optimization on Harmony Search Algorithm for Exam Timetabling System," Journal of Information and Knowledge Management, vol. 19, no. 1, 2020, doi: 10.1142/S0219649220400092.
16. J. Arias-Osorio and A. Mora-Esquivel, "A solution to the university course timetabling problem using a hybrid method based on genetic algorithms," DYNA (Colombia), vol. 87, no. 215, pp. 47–56, 2020, doi: 10.15446/dyna.v87n215.85933.
17. M. Oktavia et al., "Class Schedule Generation using Evolutionary Algorithms," J Phys Conf Ser, p. 12067, 1950, doi: 10.1088/1742-6596/1950/1/012067.
18. H. Kutty Mammi and L. Ying Ying, "Timetable Scheduling System using Genetic Algorithm for School of Computing (tsuGA)," International Journal of Innovative Computing, vol. 11, no. 2, pp. 67–72, Oct. 2021, doi: 10.11113/ijic.v11n2.342.
19. A. R. East, "Timetable Scheduling via Genetic Algorithm," 2019.
20. A. Aminu et al., "Design and Implementation of An Automatic Examination Timetable Generation and Invigilation Scheduling System Using Genetic Algorithm," Proceedings of the 2019 2nd International Conference on Applied Engineering, ICAE 2019, Oct. 2019, doi: 10.1109/ICAE47758.2019.9221700.
21. J. B. Matias, A. C. Fajardo, and R. P. Medina, "A hybrid genetic algorithm for course scheduling and teaching workload management," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018, Jul. 2019, doi: 10.1109/HNICEM.2018.8666332.
22. Wang W-J., "Improved adaptive genetic algorithm for course scheduling in colleges and universities," International Journal of Emerging Technologies in Learning, vol. 13, no. 6, pp. 29–42, 2018, doi: 10.3991/ijet.v13i06.8442.
23. N. G. A. P. H. Saptarini, P. I. Ciptayani, N. W. Wisswani, and I. W. Suasnawa, "Adaptive Genetic Algorithm for High School Time-Table," in Journal of Physics: Conference Series, 2020. doi: 10.1088/1742-6596/1569/3/032095.
24. A. Abayomi-Alli, S. Misra, L. Fernández-Sanz, O. Abayomi-Alli, and A. R. Edun, "Genetic algorithm and tabu search memory with course sandwiching (Gats_cs) for university examination timetabling," Intelligent Automation and Soft Computing, vol. 26, no. 3, pp. 385–396, 2020, doi: 10.32604/iasc.2020.013915.
25. R. P. Badoni et al., "An Exploration and Exploitation-Based Metaheuristic Approach for University Course Timetabling Problems," Axioms, vol. 12, no. 8, Aug. 2023, doi: 10.3390/axioms12080720.

26. M. Hafsa, P. Wattebled, J. Jacques, and L. Jourdan, “Solving a multiobjective professional timetabling problem using evolutionary algorithms at Mandarine Academy,” International Transactions in Operational Research, 2023, doi: 10.1111/itor.13276.
27. M. K. Kakkar, J. Singla, N. Garg, G. Gupta, P. Srivastava, and A. Kumar, “Class Schedule Generation using Evolutionary Algorithms,” in Journal of Physics: Conference Series, IOP Publishing Ltd, Aug. 2021. doi: 10.1088/1742-6596/1950/1/012067.
28. R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, “Practices in timetabling in higher education institutions: a systematic review,” Ann Oper Res, vol. 275, no. 1, pp. 145–160, Apr. 2019, doi: 10.1007/s10479-017-2688-8.
29. J. Siang Tan, S. L. Goh, G. Kendall, and N. R. Sabar, “A survey of the state-of-the-art of optimisation methodologies in school timetabling problems,” Expert Syst Appl, vol. 165, p. 113943, 2021, doi: 10.1016/j.eswa.2020.113943.
30. T. Thepphakorn and P. Pongcharoen, “Performance improvement strategies on Cuckoo Search algorithms for solving the university course timetabling problem,” 2020, doi: 10.1016/j.eswa.2020.113732.
31. A. I. Diveev, O. V. Bobr, D. E. Kazaryan, and O. Hussein, “Some Methods of Solving the NP-difficult Problem of Optimal Schedule for the University,” Procedia Comput Sci, vol. 150, pp. 410–415, Jan. 2019, doi: 10.1016/J.PROCS.2019.02.071.
32. M. Hosny and M. Al-Olayan, “A mutation-based genetic algorithm for room and proctor assignment in examination scheduling,” Proceedings of 2014 Science and Information Conference, SAI 2014, pp. 260–268, Oct. 2014, doi: 10.1109/SAI.2014.6918199.