Seminar Report

On

Genetic Algorithm for University Course Scheduling

By

Sarvesh Dhapte

1032191225

Under the guidance of

Prof. Vandana Jagtap

MIT-World Peace University (MIT-WPU)
Faculty of Engineering
School of Computer Engineering & Technology

*** 2021-2022 ***

MIT-World Peace University (MIT-WPU)

Faculty of Engineering
School of Computer Engineering & Technology

CERTIFICATE

This is to certify that Mr._Sarvesh Dhapte_of B.Tech., School of Computer Engineering & Technology, Trimester – IX /X, PRN. No._1032191225_, has successfully completed seminar on

Genetic Algorithm for University Course Scheduling

To my satisfaction and submitted the same during the academic year 2021 - 2022 towards the partial fulfillment of degree of Bachelor of Technology in School of Computer Engineering & Technology under Dr. Vishwanath Karad MIT- World Peace University, Pune.

Prof.__Vandana Jagtap____

Seminar Guide

Prof. Dr. V.Y.Kulkarni

Head
School of Computer Engineering & Technology

- **List of Figures:**

- **List of Tables:**

Acknowledgement:

Prof. Vandana Jagtap, my seminar guide, provided invaluable assistance in completing this article and conducting the research that led to it. Her enthusiasm, competence, and meticulous attention to detail have inspired me and kept my work on track since the beginning. Learnt new things under her direction, and her extensive knowledge of genetic algorithms aided in the completion of this research.

- **Index:**

Abstract and keyword:

Systematic and efficient timetabling is required for the proper operation of any organization. The subject of university course scheduling is addressed in this study. The university frequently confronts difficulties in creating the timetable, as well as after it is completed. This is because as the number of students, professors, or courses increases, creating the timetable gets more difficult and time-consuming. As a result, the task is classified as an NP-hard problem. Unlike in schools, where a set of lectures must be attended every day for the same amount of time, the timetable at university is depending on the number of courses and types of courses taken by the student; for example, a student may have three courses on Monday but just one on Tuesday. As a result, effective timetabling is required. Our method presents a genetic algorithm-based approach for automatic university timetabling. A genetic algorithm belongs to evolutionary algorithm and is a type of adaptive algorithm that is used to address optimization problems. The proposed system is dynamic in nature, which aids in the resolution of hard and soft constraints, as well as the modification of these constraints later in the system. Any university can use this to adapt the system to their needs. The system also takes care of proper course day timings, such as which courses should the day begin with and which courses should it end with. The technology is unique because it employs a unique form of genetic operator combination. Thus, using a genetic algorithm, a near-optimal solution to this problem can be found.

Keywords: NP hard problem, Genetic Algorithm, Evolutionary Algorithm, Adaptive Algorithm, Optimization Problem.

1. INTRODUCTION:

Every semester, the university must create timetables for each panel/section of the corresponding department at the start of the semester. The university committee members must create timetables for the proper running of the courses allotted in the semester, taking into account the workload of teachers, the number of courses, the number of rooms, and other factors[1]. The more a university grows, the more difficult it becomes to manually create timetables. Making handwritten timetables may have an impact on the institution's ability to function properly. Automatic Timetabling/Scheduling was born out of the need to resolve efficient and complex timetables.

Genetic algorithm (GA) is an Adaptive Heuristic search method that belongs to the evolutionary algorithm family[2]. The GA is adaptive to environmental constraints, or in other words, to the amount and kind of parameters provided to the algorithm. GA is based on natural selection and genetics. GA is very useful for generating high-quality or optimal solutions to optimization problems. Scheduling is an NP-Hard[3] issue that can be addressed with GA but not with a classical approach. The table 1 below compares the GA to the classical algorithm:

Table 1. Comparison between Genetic Algorithm and Traditional Algorithm

Genetic Algorithm (GA)	Classical/Traditional algorithm
1. Genetics and Natural selection to solve optimization problem.	1. Step by step procedure to solve a given problem (finiteness)
2. Probabilistic rules	2. Fully deterministic rules
3. More advanced	3.Not advanced
4. Search on population of points	4.Search on a single point

The search space for a scheduling problem is far too large; many solutions exist, and only a few are realistic. Solutions that are feasible are those that satisfy the limitations. There are two types of constraints[4]: hard constraints, which require strict adherence to the restrictions, and soft constraints, which do not require strict adherence to the requirements. To arrive at an optimal solution, all hard constraints must be satisfied while the number of soft constraints must be minimised.

Some examples of Hard Constraints are as follows:

1. No two courses in the same room at the same time are allowed.
2.In the same time slot, no speaker or faculty member should be assigned to two separate rooms.
3.In the same time slot, no single room should be assigned to two different courses.

Some examples of soft constraints are as follows:

1. A student should take more than one course every day.
2. A student should take no more than two courses in a row.
3. The total time spent teaching should not exceed 6 hours.

The goal is to present a dynamic system that uses GA to address the scheduling problem. This dynamic approach allows the user to add or remove hard or soft constraints based on their university's requirements, resulting in ideal solutions or timetables. This would be accomplished through the right application of GA operators[5] like as encoding, cross-over, and mutation, as well as selection. The convergence test, often known as the termination condition, is crucial in determining the best solution.

2. LITERATURE SURVEY:

Priyanshi B. et al. [6] performed a state-of-the-art comparison of genetic and memetic algorithms on graph colouring. For chromosomes, the genetic algorithm used Integer Coding, whereas the memetic algorithm used a Binary Coding scheme. It not only shown that genetic algorithm utilizes Optimal Solution while memetic algorithm uses Local Solution, but it also demonstrated that the solution discovered in memetic algorithm is from a single run rather than multiple runs of k-coloring. Their research found that memetic algorithms are considerably superior to genetic algorithms since they are more efficient. To declare the final outcome, the authors should have evaluated more parameters.

The authors suggested a solution using the novel chromosomal representation to tackle the scheduling problem for two departments at their college in [7]. They studied the constraints and came to the conclusion of using the genetic algorithm to solve the problem. They employed a chromosome representation with 13 genes for each group, as well as the well-known 1 point crossover. XML, Java, and HTML were used to construct the system. There are still certain limitations to the system, such as the fact that it was only established for two departments of the university.

Shara S.A. Alves et al.[8] propose a unique, scalable, and parameterized model that uses recursive genetic algorithms to solve the timetable problem for several courses. Each recursion solves one course at a time until a global solution is found. To create feasible individuals for crossover, the permutation operator is employed. When the recursive model was compared to the iterative model, the recursive model outperformed the iterative model in terms of execution time. In repeated trials, the recursive model discovered more than one viable option. Rooms were not taken into account in the study.

R. E. Febrita;W. F. Mahmudy;[9] modified the genetic algorithm. Rather than University they proposed the method to solve high school timetable problem using fuzzy time window.The school follows a set of periods each day for a particular length; the major goal was to create an optimal timetable in which students would not become bored, worried, or exhausted due to a sequence of exact subjects(Physics, Biology,etc.) or non exact subjects(Art,Physical Education,etc). As a result, the hard constraints and suitable time window were bypassed using a modified mutation operator that used fuzzy values as a reference in gene exchange. For 80 rounds, the modified genetic algorithm took 8 seconds to compute, whereas the traditional genetic method took 10 seconds. However, the proposed method limited the maximum frequency of occurrence of the subjects in a single day.

J. Soyemi et al.[10] not only designed the Electronic Lecture Time Tabling Scheduler (ETTS) with a genetic algorithm, but also conducted a comparison study between Manual Time Tabling Scheduling (MTTS) and ETTS. The suggested system was created for a Nigerian institution, and the data included the number of lectures, locations, and capacities, student capacity per course, names of lecturers, courses, and units, as well as hours of lectures and practical sessions. A questionnaire with about 215 questions was distributed to students, teaching faculty, and committee members, and was based on the efficiency, time consumption, convenience, and overall performance of the two systems. According to the findings, respondents' perceptions of ELTS and

MTTS on delivery based on time, convenience, efficiency, and overall performance are 4.485 and 3.605, respectively. ELTS outperformed MTTS because it avoided many of the problems that the manual approach caused. The authors were unable to develop a fresh model to handle the challenge, but the proposed system successfully resolves the examined hard constraints, making the model stiff.

The authors of [11] propose utilizing graph colouring to discover the near optimal solution to the university scheduling problem. Numerous violations were decreased thanks to the three-dimensional chromosomal technology, in which an array of classes maintains information about the instructor, the room, and the time-slot. The roulette wheel is used as the selecting operator. The repair function helped strategies converge their algorithms at high speed for the best solution, overcoming the additional violations generated by the crossover operator. The mutation operator was unable to assist in the improvement of the solution. Though the crossover operator resolved some issues, it also created new ones.

Often, new or small institutions face the problem of allocating new teaching staff to different courses because their skills are not on par, or because teaching loads are based on constraints that can be both hard and soft. The hybrid genetic algorithm in [12] solves the aforementioned problem while also creating an efficient timetable and allocating additional staff as needed. It also aids in determining the prospective workloads of new potential teaching staff, allowing for the avoidance of hiring unmatched skills. If the hybrid genetic algorithm fails to produce a viable solution, a repair operator is used to correct each infeasible gene. Despite satisfying all hard constraints, some soft constraints remained unresolved.

Marina Yusoff and Nurhikmah Roslan[13] compared the genetic algorithm hill climbing with elitist and the genetic algorithm with elitist on a real-world dataset. The authors conducted a survey to determine the soft constraints that needed to be addressed. Penalty points were allocated to both the hard and soft constraints, which had to be lowered. For the performance comparison, several computational tests based on population size, crossover rate, mutation rate, number of generations, and so on were conducted on both techniques: genetic algorithm hill climbing with elitist and genetic algorithm with elitist. In some areas, the genetic algorithm hill climbing with elitist outperformed the genetic algorithm with elitist technique to obtain the near optimal solution. Large dataset implementation on this approach is still a work in progress.

The authors of [14] used a Parallel genetic algorithm and a local search on the BenPaechter competition datasets to solve the problem of course scheduling. The authors tested the standard genetic algorithm on several samples and found that the results were unsatisfactory. As a result, the authors offered some algorithm modifications, including rehabilitative processes for children, an intelligent operator, and the usage of parallel structure. The crossover and mutation operators were used to reproduce the chromosome. The local search assisted in switching the genes of unused kid chromosome genes one by one, and the local search's stopping criteria were based on time slots. The proposed approach reduces the DF's accuracy quickly (distance to feasibility).This is owing to the structure of the mutation operator and its method for dealing with chromosomes like this. The structure of the method causes pressure on the soft restrictions, producing a rapid fall in their value at the first stages in small instances such as small, where the amount of DF is initially zero. The use of advanced genetic operators and heuristics could have resulted in a more accurate model to tackle the problem.

G. Alnowaini et al.[15] developed a strategy to address the scheduling problem for several university departments using a genetic algorithm and the dynamic chromosome in 2021. The chromosome's dynamic size made it possible to fit every semester's courses for each department, making the model adaptable to other faculties. The roulette wheel is used for selection in this variant as well, as it is in [5]. Although the constructed timelines have a 93 percent optimality rate, the model failed to create timelines with challenging constraints and did not use all of the principles of genetic algorithms.

3. PROPOSED METHODOLOGY:

The proposed system is based on the following Fig. 1. Genetic Algorithm Diagram.

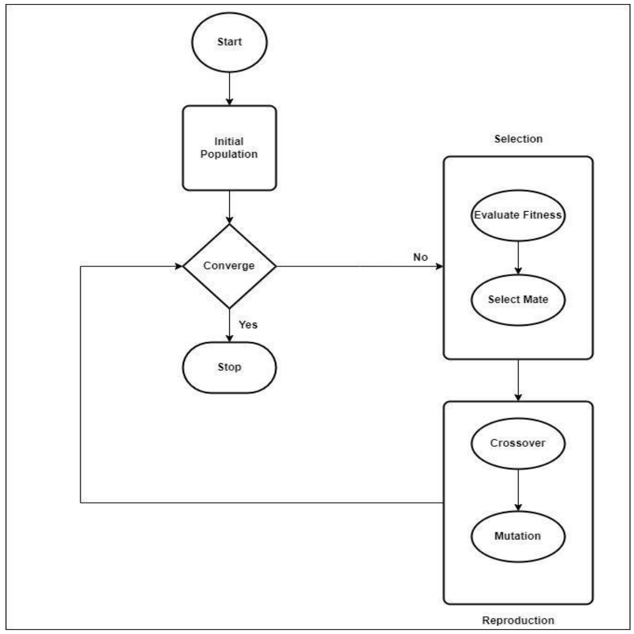


Fig. 1 Genetic Algorithm Diagram

The algorithm will go through each of the states listed above, but it is critical to understand the fundamentals of genetics before going ahead. Let's start with the fundamentals of genetics.

3.1 Fundamental Principles of Genetics:

Understanding the fundamental principles of genetics using the fig.2 below. A gene is a single piece of genetic information. Each gene has a distinct feature or parameter. The chromosome is made up of genes that are linked together as a string. Individuals, or chromosomes, can be viewed as a possible solution to the problem that the genetic algorithm is considering. The chromosome group is referred to as the population. We will now discuss the encoding scheme.

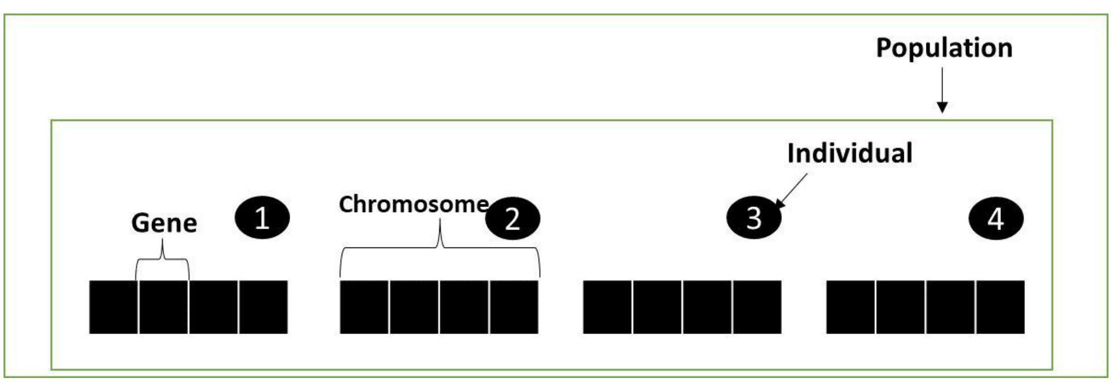


Fig. 2 Fundamentals of Genetics

3.2 Encoding:

Before a genetic algorithm can be utilized to solve a problem, it is necessary to have two crucial elements: i) Measuring the solution's quality (fitness function), ii) Representing the solution (string, number, or character). Binary Encoding is employed in the suggested system. The chromosome is represented as 0's and 1's in this case. Each chromosomal location encapsulates the problem's distinct characteristics.

3.3 Initial Population:

The initial population or generation is often selected randomly using random numbers. It is the most important portion of the algorithm because it might affect the solution's quality and convergence rate.

Following the creation of the initial generation, we go on to the evolution of that generation. With the help of Genetic Algorithm Operators, this is feasible. The genetic algorithm operators direct the algorithm to provide the best possible solution to the problem.

3.4 Genetic Algorithm Operators:

Selection, Crossover, and Mutation are the three operators. These operators come in a variety of sorts, and they're employed in different ways depending on the task at hand. To arrive at the best solution, our proposed system employs a distinct type of each operator.

3.4.1 Selection Operator:

The goal is to prioritise those with high fitness scores so that they can pass on their genes to future generations. Various selection procedures, such as Roulette Wheel selection, Tournament selection, Elitism selection, and so on, have been employed in prior studies. Rank-based selection is used in our suggested method.

Chromosome	Fitness value	Rank	Area acquired
A	5	5	33.33 %
B	2	2	26.67%
C	0.5	0.5	6.67%
D	1.5	1.5	20%
E	1	1	13.33%

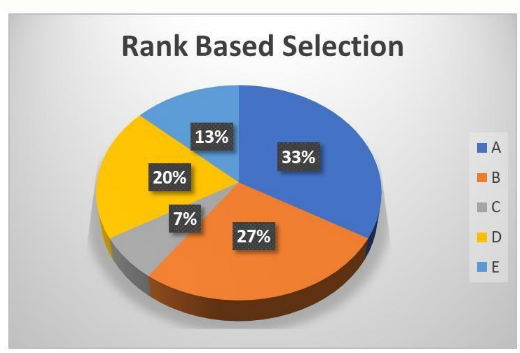


Fig. 3 Rank Based Selection

Figure 3 depicts how rank-based selection works. We have five chromosomes A, B, C, D, and E in this diagram, each with a fitness value of 5, 2, 0.5, 1.5, and 1 and an acquired area of 33.33 percent, 26.67 percent, 6.67 percent, 20 percent, and 13.33 percent, respectively. The chromosomes are then sorted according to their fitness value, with the least fittest chromosome receiving rank 1 and the fittest chromosome receiving rank N (N= Number of chromosomes), and so on. It also contains one fixed-position arrow, similar to a roulette wheel selection, that picks one chromosome at a time. Ranking creates a consistent scale across the population and is a simple and effective technique to manage selective pressure.The likelihood of any individual being chosen for reproduction is determined by its fitness normalized by the population's overall fitness. As a result, rank-based selection aids in the consideration of chromosomes with negative fitness values. Rank-based fitness assignment is more resilient than proportional fitness assignment. All chromosomes have a chance of being picked, restoring diversity.

3.4.2 Crossover Operator:

Following the selection of the two individuals, the crossover operator is used to improve the algorithm and produce an optimal result. The crossover operator aids in the variation of chromosomal characteristics from one generation to the next. To create superior progeny, two strings are randomly selected from the mating pool to crossover. Different types of crossover were used in prior studies, such as Single Point Crossover and Multi Point Crossover. Determination of which type of crossover must be chosen is done by keeping the encoding method in mind. The Uniform Crossover approach is used in our proposed system.

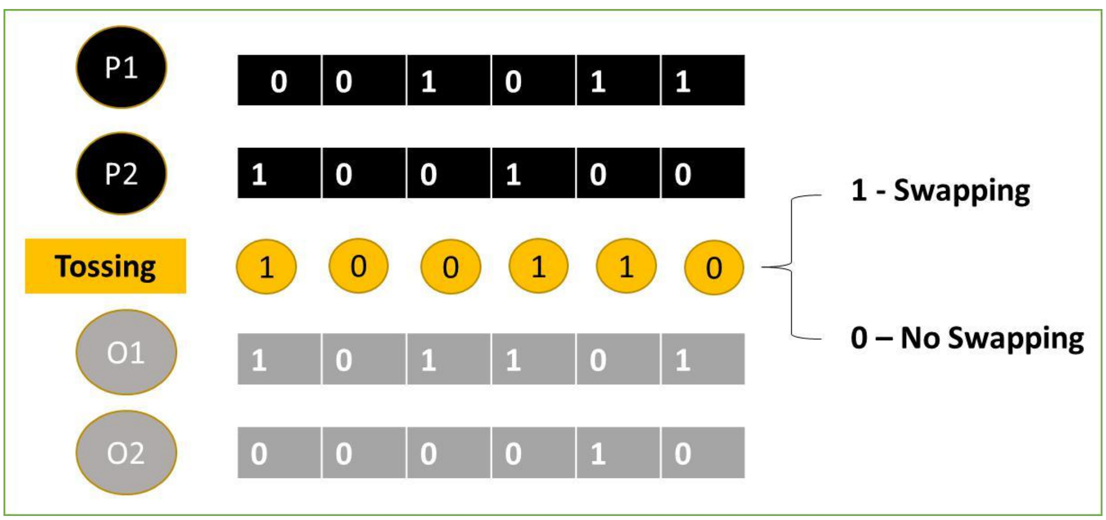


Fig. 4 Crossover Operator

The tossing method is used in Uniform Crossover to produce new children; if the tossing value assigned to a gene is 1, the gene will be swapped; if the tossing value allocated to a gene is 0, the gene will not be swapped. The example of uniform crossover shown in fig. 4 is shown above. Using the tossing method, two parents P1 and P2 generate offspring O1 and O2. If the parents are good, the probability of offspring being good is high. If the offspring isn't good (a bad solution), it will be deleted during the "Selection" phase of the next iteration.

3.4.3 Mutation Operator:

Mutation is defined as a change in the structure of a gene. Because of the binary coding, the Bit Flipping mutation is often utilised in genetic algorithms. This bit flipping mutation mechanism is also used in our proposed system.

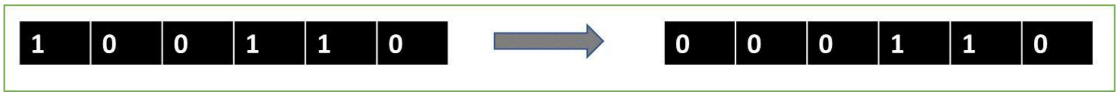


Fig. 5 Mutation Operator

Bit Flipping is demonstrated in Figure 5 above. This approach flips the bits of any one or more genes on a chromosome, converting 1s to 0s and vice versa. Mutation happens to maintain population variety and avoid premature convergence.

3.5 Convergence Test / Termination Condition:

The convergence test follows the reproduction phase. The algorithm's termination condition is determined here. Different approaches can be employed for this, such as manual checking, a fixed number of generations, and so on. Our proposed system employs the strategy of terminating the algorithm when we are unable to obtain a more suitable solution than the previous iteration or result. The genetic algorithm is thus considered to have generated a set of solutions to our problem.

There will be a list of hard and soft limits. This list can be created by sending out a survey form to professors and students, asking them to suggest constraints that their university should address. Then, depending on their university's needs, they can add or remove limits, making the system dynamic in nature.

Course durations should be adequately arranged with the mental health of students and instructors in mind. Mathematical courses, for example, should be taught early in the morning, while theory subjects like Software Engineering and Process Management should be taught after lunch. Students and teachers can suggest this type of course scheduling through a survey form. This will not cause pupils to become bored, sluggish, or sleep during lectures.

Thus,with the use of Rank based selection, Uniform crossover and Bit flip mutation the algorithm is enhanced to perform better and give optimal solution to the problem.

4. CONCLUSION:

During research to solve the university timetabling problem, the genetic algorithm was proven to be superior to the traditional method in every regard. As a result, the genetic algorithm was effectively used to tackle the problem of university course scheduling. The algorithm was improved by combining different types of genetic operators. The rank-based selection operator allowed for the inclusion of negative fitness values, which was not achievable with any other method, as well as for the gain of diversity and the avoidance of premature convergence. The system not only took care of improving the algorithm to reach a better result, but it also took care of the mental health of students and teachers by altering the course schedules, which was done using a survey. The ability to add and remove both soft and hard constraints was also included in the latter, which was extensively surveyed by students and professors. Although rank-based selection aided in premature convergence, it was unable to overcome the slow convergence rate because it was discriminatory to individuals with higher fitness values because the best chromosomes are not very different from the others.

5. REFERENCES:

1. Ahmad, Izah & Sufahani, Suliadi & Ali, Maselan & Mohd Razali, Siti. (2018). A Heuristics Approach for Classroom Scheduling Using Genetic Algorithm Technique. Journal of Physics: Conference Series. 995. 012050. 10.1088/1742-6596/995/1/012050.

2. Xu, Jing. (2021). Improved Genetic Algorithm to Solve the Scheduling Problem of College English Courses. Complexity. 2021. 1-11. 10.1155/2021/7252719.

3. Timilsina, Sandesh & Negi, Rohit & Khurana, Yashika & Seth, Jyotsna. (2015). Genetically Evolved Solution to Timetable Scheduling Problem. International Journal of Computer Applications. 114. 12-17. 10.5120/20077-2100.

4. Modibbo, Umar & Umar, Ibraheem & Mijinyawa, Mohammed & Hafisu, Rafiyatu. (2019). Genetic Algorithm for Solving University Timetabling Problem.

5. Shraddha Thakare , Tejal Nikam , Mamta Patil, 2020, Automated Timetable Generation using Genetic Algorithm, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 07 (July 2020)

6. Priyanshi Barod, Varsharani Hawanna,Vandana Jagtap , “Genetic Algorithm And Memetic Algorithm On Graph Coloring”,International Journal of Advance Engineering and Research Development, Volume 1,Issue 12, December -2014.

7. P. M. Chauhan, K. B. Parmar and M. B. Mendapara, "Solving time-table scheduling problem by novel chromosome representation using Genetic algorithm," 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], 2015, pp. 1-6, doi: 10.1109/ICCPCT.2015.7159319.

8. S. S. A. Alves, S. A. F. Oliveira and A. R. Rocha Neto, "A novel educational timetabling solution through recursive genetic algorithms," 2015 Latin America Congress on Computational Intelligence (LA-CCI), 2015, pp. 1-6, doi: 10.1109/LA-CCI.2015.7435955.

9. R. E. Febrita and W. F. Mahmudy, "Modified genetic algorithm for high school time-table scheduling with fuzzy time window," *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2017, pp. 88-92, doi: 10.1109/SIET.2017.8304115.

10. J. Soyemi, J. Akinode and S. Oloruntoba, "Electronic Lecture Time-Table Scheduler Using Genetic Algorithm," 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), 2017, pp. 710-715, doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.124.

11. Maram Assi, Bahia Halawi, Ramzi A. Haraty, Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem,Procedia Computer Science,Volume 126,2018,Pages 899-906,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2018.08.024.

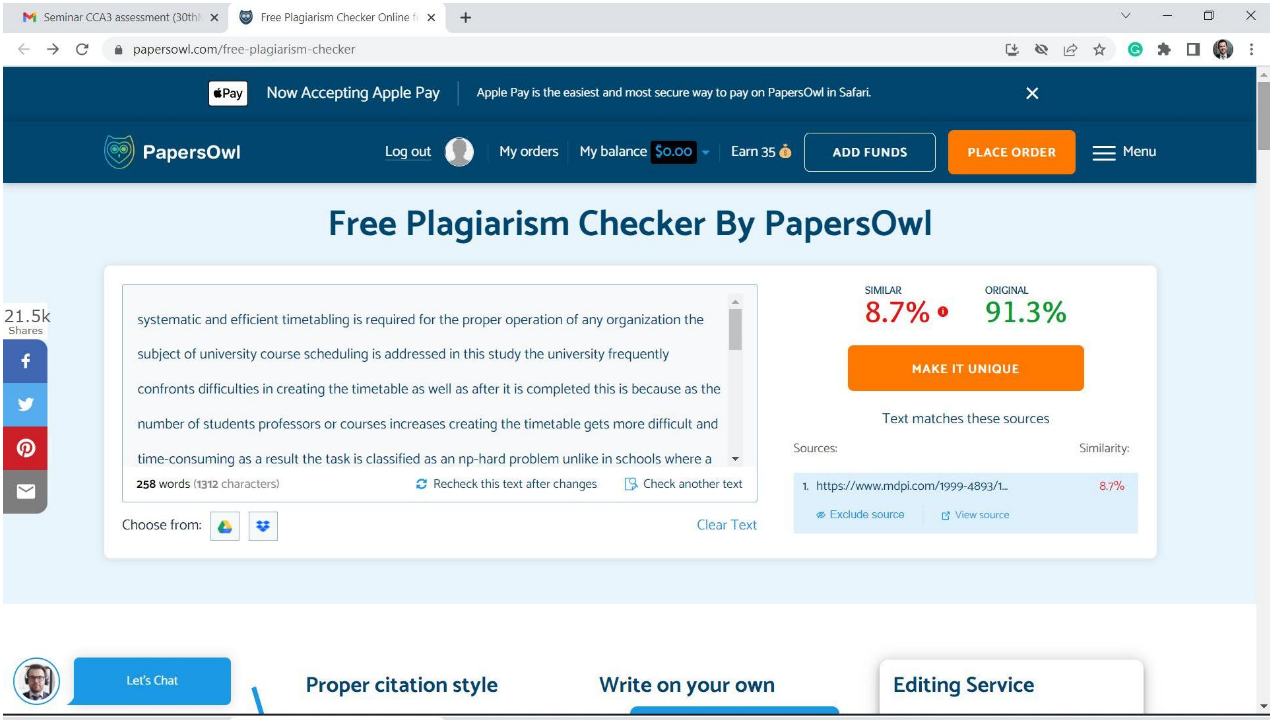
12. J. B. Matias, A. C. Fajardo and R. P. Medina, "A Hybrid Genetic Algorithm for Course Scheduling and Teaching Workload Management," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), 2018, pp. 1-6, doi: 10.1109/HNICEM.2018.8666332.

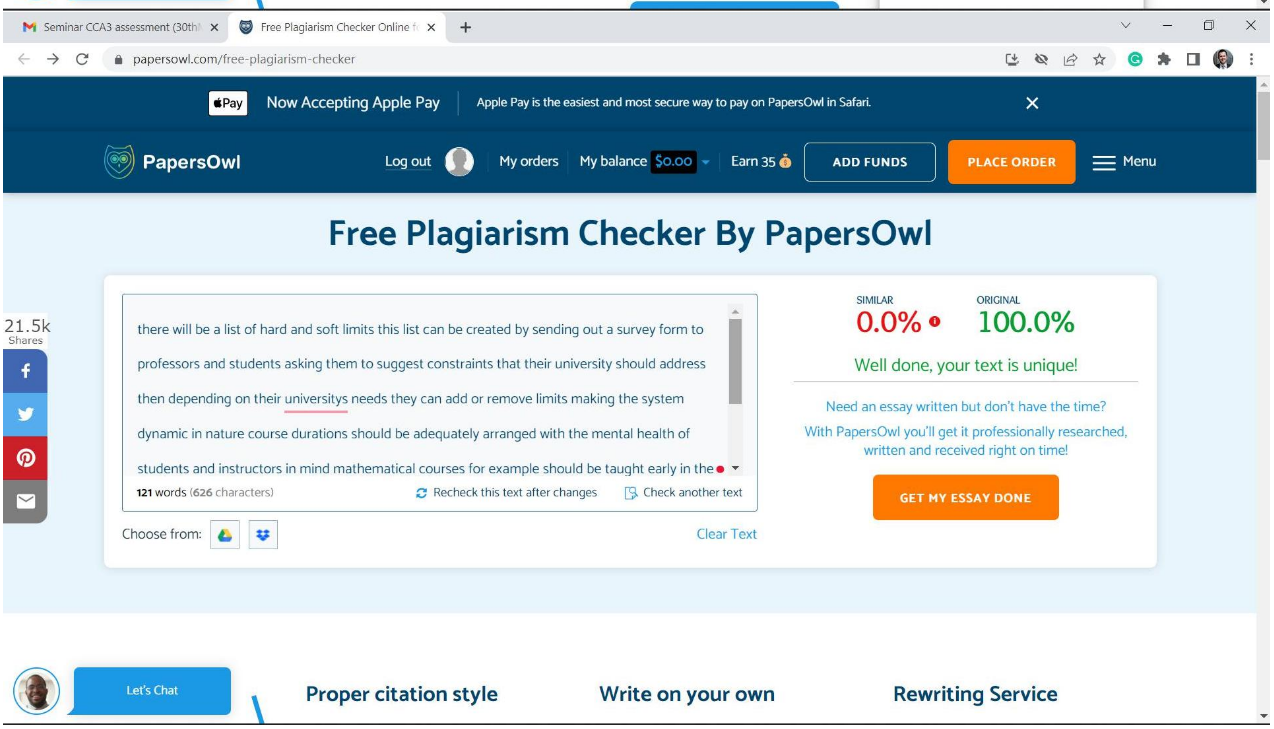
13. Marina Yusoff & Nurhikmah Roslan. (2019). Evaluation of Genetic Algorithm and Hybrid Genetic Algorithm-Hill Climbing with Elitist for Lecturer University Timetabling Problem. 10.1007/978-3-030-26369-0_34.

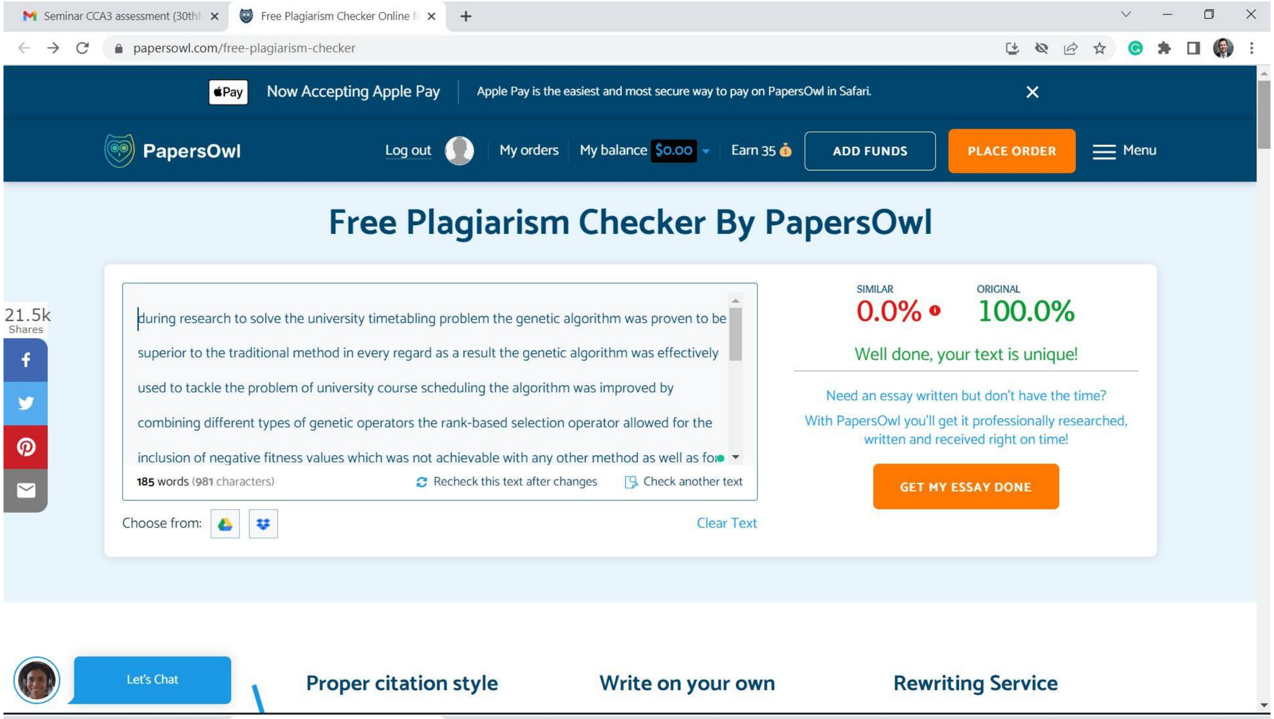
14. Rezaeipanah, Amin & Abshirini, Zahra & Zade, Milad. (2019). Solving University Course Timetabling Problem Using Parallel Genetic Algorithm. Journal of Scientific Research and Development. 7. 5-13.

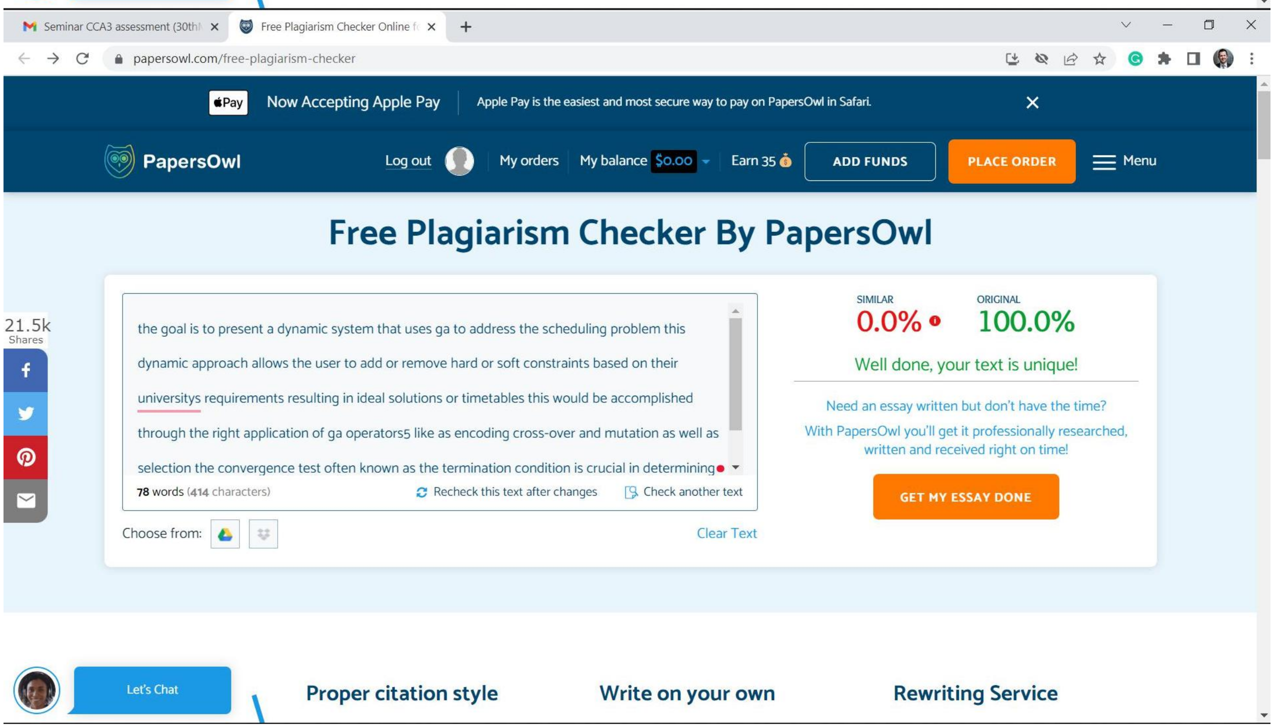
15. G. Alnowaini and A. A. Aljomai, "Genetic Algorithm For Solving University Course Timetabling Problem Using Dynamic Chromosomes," 2021 International Conference of Technology, Science and Administration (ICTSA), 2021, pp. 1-6, doi: 10.1109/ICTSA52017.2021.9406539

- **Plagiarism Check report:**






Seminar CCA3 assessment (30th
Free Plagiarism Checker Online
papersowl.com/free-plagiarism-checker
Now Accepting Apple Pay
Apple Pay is the easiest and most secure way to pay on PapersOwl in Safari.
PapersOwl
Log out
My orders
My balance $0.00
Earn 35
ADD FUNDS
PLACE ORDER
Menu
Free Plagiarism Checker By PapersOwl
21.5k
Shares
during research to solve the university timetabling problem the genetic algorithm was proven to be
superior to the traditional method in every regard as a result the genetic algorithm was effectively
used to tackle the problem of university course scheduling the algorithm was improved by
combining different types of genetic operators the rank-based selection operator allowed for the
inclusion of negative fitness values which was not achievable with any other method as well as fo
185 words (981 characters)
Recheck this text after changes
Check another text
Choose from:
Clear Text
SIMILAR
0.0%
ORIGINAL
100.0%
Well done, your text is unique!
Need an essay written but don't have the time?
With PapersOwl you'll get it professionally researched, written and received right on time!
GET MY ESSAY DONE
Let's Chat
Proper citation style
Write on your own
Rewriting Service


Seminar CCA3 assessment (30th
Free Plagiarism Checker Online
papersowl.com/free-plagiarism-checker
Now Accepting Apple Pay
Apple Pay is the easiest and most secure way to pay on PapersOwl in Safari.
PapersOwl
Log out
My orders
My balance $0.00
Earn 35
ADD FUNDS
PLACE ORDER
Menu
Free Plagiarism Checker By PapersOwl
21.5k
Shares
the goal is to present a dynamic system that uses ga to address the scheduling problem this
dynamic approach allows the user to add or remove hard or soft constraints based on their
universitys requirements resulting in ideal solutions or timetables this would be accomplished
through the right application of ga operators5 like as encoding cross-over and mutation as well as
selection the convergence test often known as the termination condition is crucial in determining
78 words (414 characters)
Recheck this text after changes
Check another text
Choose from:
Clear Text
SIMILAR
0.0%
ORIGINAL
100.0%
Well done, your text is unique!
Need an essay written but don't have the time?
With PapersOwl you'll get it professionally researched, written and received right on time!
GET MY ESSAY DONE
Let's Chat
Proper citation style
Write on your own
Rewriting Service