3rd International Engineering Conference (IEC 2019)
Federal University of Technology, Minna, Nigeria

Development of a Model for Generation of Examination Timetable Using Genetic Algorithm

Ahmed A[1], *Umar B. U[1], Abdullahi I. M[2], Maliki. D[3], Anda. I[4]. and Kamaldeen J. A[5].
[1,2,3,5]Department of Computer Engineering, Federal University of Technology, Minna, Niger-state, Nigeria
*Corresponding author email: buhariumar@futminna.edu.ng, +2348058001775

ABSTRACT
Examination time table scheduling problem is one of the complexes, NP-complete and typical combinatorial optimization problem faced by the university community across the globe. Many researchers have studied the problem due to its NP-complete nature and highly- multi-constrained problem which seeks to find possible scheduling for courses. Creating an examination timetable for university is a very difficult, time-consuming and the wider complex problem of scheduling, especially when the number of students and courses are high. Several factors are responsible for the problem: increases number of students, the aggregation of schools, changes in educational paradigms, among others. In most universities, the examination time table schedule is usually ended up with various courses clashing with one another. In order to solve this problem of time table scheduling for University examination and effective utilization of resources, this research proposed a model for examination time table generation using Genetic Algorithm (GA) probabilistic operators. GA has been successful in solving many optimization problems, including University time table. This is based on the fact that GA is accurate, precise, free from human error and robust for complex space problem. GA theory was also covered with emphasis on the use of fitness function and time to evaluate the result. The effects of altered mutation rate and population size are tested. By using Genetic algorithm, we are able to reduce the time required to generate a timetable which is more accurate, precise and free of human errors. The implication of this research is a solution, minimizing the time taken in timetable allocation and the clashing that usually characterize time table schedule.

Keywords: *Genetic algorithm, timetable, constraints, chromosomes, fitness function.*

1 INTRODUCTION

The Timetable Problem is one of the complex problems faced in any university in the world. It is a highly-constrained combinatorial problem that seeks to find possible scheduling for the university course offerings. There are many algorithms and approaches adopted to solve this problem, but one of the effective approaches to solve it is the use of metaheuristics. Genetic algorithms were successfully useful to solve many optimization problems including the university Timetable Problem. In this paper, we present a near-optimal solution for problem of time table scheduling using GA. The GA method was implemented in MATLAB (Assi, Halawi, & Haraty, 2018). The University timetabling problem is a typical scheduling problem that appears to be a tedious job in every academic institute once or twice a year (Datta, Deb, & Fonseca, 2006). In earlier days, time table scheduling was done manually with a single person or some group involved in task of scheduling it manually, which takes a lot of effort and time. Planning timetables are one of the most complex and error-prone applications.

Timetabling is the task of creating a timetable while satisfying some constraints. There are basically two types of constraints, soft constraints and hard constraints. Soft constraints are those if we violate them in scheduling, the output is still valid, but hard constraints are those which, if we violate them; the timetable is no longer valid (Doulaty, Derakhshi, & Abdi, 2013). The search space of a timetabling problem is too vast, many solutions exist in the search space and few of them are not feasible. Feasible solutions here mean those which do not violate hard constraints and as well try to satisfy soft constraints. We need to choose the most appropriate one from feasible solutions. Most appropriate ones here mean those which do not violate soft constraints to a greater extent (Doulaty et al., 2013). The goal is to have the near-optimal solution for the time table scheduling problem using GA. GAs are used in optimized searching processes and scheduling process. This GA is generally applied to the problem of where optimization is the key goal. A timetable problem is basically related to finding the exact time allocation within limited time period of number of events (courses-lectures) and assigning to them number of resources (teachers, students and Lecture Halls) while satisfying Hard Constraints and Soft constraints (Deeba Kannan, 2019).

Also, this is based on the fact that GA is accurate, precise, free from human error and robust for complex space problem. GA theory was also covered with emphasis on the use of fitness function and time to evaluate the result. The effects of altered mutation rate and population size are tested. By using Genetic algorithm, we are able to reduce the time required to generate a timetable which is more accurate, precise and free of human errors. The implication of this research is a solution, minimizing the time taken in timetable allocation and the clashing that usually characterize time table schedule.
.
The remaining part of this paper is organized as follows: section II deals with time tables issues. section III discussed GA; related works are discussed IV; V presents the proposed methodology and the implementation, while VI presents the results and discussion. Finally, section VII presents the conclusion.

2. TIMETABLING ISSUES

2.1 Examination Timetabling Problem

The examination timetabling problem is a well-known non-deterministic polynomial-time (NP) and is a hard optimization problem (solving problems without violation of the university's terms) faced by all educational institutions. However, the approach that the institutions adopt in preparing examination timetables depends on the variations in the institution's requirements and constraints (Burke, Eckersley, McCollum, Petrovic, & Qu, 2003; Burke et al., 2003).

The Federal University of Technology, Minna (FUTMINNA) like any other institutions of higher learning, face a similar problem and is used as the case study in this paper. Preparing the examination timetable takes much of the administrator's time and when it involves all the different programs, the task is overwhelming (Ross, Corne, & Fang, 1994; Birbas, Daskalaki, & Housos, 2009). Timetabling is complicated due to many factors. Below are some of the problems that university administrations usually face when preparing the final examination schedule (Birbas et al., 2009):

a. Limited time to prepare.
b. A limited number of rooms.
c. Limited number of staffs to prepare the schedule.
d. Last-minute timetable changes. e. Inadequate staff to invigilate.
f. Late submission of examination grades by Lecturers.
g. Absence of invigilators or students (sick or vacation leaves) at the appointed time.

3 GENETIC ALGORITHM

Genetic Algorithm (GA) is an optimization method that imitates the behaviour of biological evolution to solve particular problems programmatically. In aligning with a specific problem at hand, a quantitative evaluation is performed by a fitness function to each candidate solution by implementing GA that makes the solution better, i.e. improves the fitness level (Birbas et al., 2009). The algorithm evaluates each candidate at random according to the fitness function (Birbas et al., 2009). However, by chance, a few may succeed but with weak or imperfect results. These positive solution candidates are stimulated, reserved and permitted to reproduce (Obaid, Ahmad, Mostafa, & Mohammed, 2012). GA has been widely used in many areas that are difficult for humans to handle. The solutions are more credible, efficient, faster, and higher complexity than humans can solve. In many cases, genetic algorithms produce solutions that are far better than the programmer who wrote the algorithm. The Basic outline of GA in the step next below.

```
Create a population of creatures.
Evaluate the fitness of each creature.
While the population is not fit enough:
{
Kill all relatively unfit creatures.
While population size< max;
{
Select two population members.
Combine their genetic material to create a new creature.
Cause a few random mutations on the new creature.
Evaluate the new creature and place it in the population.
}}.
```

3.1 GA Operators

Chromosome representation: Chromosome is a set of parameters which define a proposed solution to the problem that the genetic algorithm is trying to solve. The chromosome is often represented as a simple string. The fitness of a chromosome depends upon how well that chromosome solves the problem at hand.

Initial population: The first step in the functioning of a GA is the generation of an initial population. Each member of this population encodes a possible solution to a problem. After creating the initial population, each individual is evaluated and assigned a fitness value according to the fitness function. It has been recognized that if the initial population to the GA is good, then the algorithm has a better possibility of finding a good solution and that, if the initial supply of building blocks is not large enough or good enough, then it would be difficult for the algorithm to find a good solution.

Selection: This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce (Mitchell, 1998).

***Crossover**:* In genetic algorithms, a crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. A crossover is a process of taking more than one parent solutions and producing a child's solution from them. There are methods for selection of the chromosomes. This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring.

***Mutation**:* Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. The mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to a better solution by using mutation. This operator randomly flips some of the bits in a chromosome (Mitchell, 1998).

***Fitness Function**:* The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. In particular, in the fields of genetic Programming and genetic algorithms, each design solution, are commonly represented as a string of numbers referred to as a chromosome. After each round of testing, or simulation, the idea is to delete the 'n' worst design solutions and to breed 'n' new ones from the best design solutions.

4 RELATED WORKS

Different approaches have been proposed to solve the university timetable problem by a different researcher. These research are: (Elsaka, 2017), (Matias, Fajardo, & Medina, 2018), (Dener & Calp, 2019), (Dener & Calp, 2019), (Soyemi, Akinode, & Oloruntoba, 2017), (Mohammed, 2017), (Rozaimee, Shafee, Hadi, & Mohamed, 2017), (Al-Jarrah, Al-Sawalqah, & Al-Hamdan, 2017), (Sani & Yabo, 2016), (Mittal, Doshi, Sunasra, & Nagoure, 2015), (Amaral & Pais, 2016) and (Abdelfattah & Shawish, 2016). Numerous works of some of these are presented in this section below.

Research by (Bhaduri, 2009) solved the problem with an evolutionary technique such as the Mimetic Hybrid Algorithm, Genetic Artificial Immune Network (GAIN) and compared the result with that obtained from genetic algorithm. His result showed that GAIN was able to reach the optimal feasible solution faster than that of GA but not as effective as GA.

Mei Rui 2010, in his paper, performed analysis and the summarization of the existing problems. A mathematical model for the course timetable system was proposed. At the same time, through the use of the pattern recognition technology in artificial intelligence, aiming at this mathematical model a new university course timetable system design program is proposed and realized. This program not only can well solve the shortages of the existing course timetable system but also is simple and easy to operate, has strong versatility.

A genetic algorithm (GA) based model to generate examination schedules such that they focus on students' success in addition to satisfying the hard constraints required for feasibility. The model is based on the idea that student success is positively related to adequate preparation and resting time between exams. Therefore, the main objective of the system is to maximize the time length among exams (i.e., paper spread) considering the difficulties of exams. Two different genetic algorithm models were developed to optimize the paper spread. In the first genetic algorithm model, a high penalty approach was used to eliminate infeasible solutions throughout generations. The second genetic algorithm model controls whether or not each chromosome joining the population satisfies the hard constraints. The model presented can be implemented in software to make it use practical. On the other hand, the models presented can be extended to include the fuzziness that may arise when understanding the difficulties of exams. Only constraints set by the institutions and lecturers have been taken into account (Kalayci & GUNGOR, 2012).

The author in (Parkavi, 2012), developed an e-College Time table Retrieval System Based on Service Oriented Architecture which provides support for staffs, HODs, Principal, Directors and other societies interested in e-college time table retrieval systems, as well as for students. This system provides the facility for coordinator to enter the time table or change the time table and the staffs to access the time table using their PDAs or by using their smartphones, alerts can be transferred about their duties over the mobiles and exchanging of duties can also be done using the mobile devices with the help of web services. The system is expensive due to the involvement of many entities and takes time to develop. Network failure can lead to inadequate message dissemination. Extra alerts sent over the media or internet require extra spending.

Research in Mugdha K. P., Rakhe S. S., Prachi A. P. & Naveena N. T. (2014), authors focused on the development of a web application for automatic time table generation. A mechanized system was designed with a computer-aided timetable generator. The system allows interaction between the staff and students and at the same time enable them to upload their queries, notes, presentations and e-books. The application makes the procedure of time table generation easier consistently which may otherwise need to be done using spreadsheet manually which might lead to constraints problem that is strenuous to establish when time table is generated physically. The purpose of the algorithm is to generate a timetable schedule mechanically. Keeping in mind the generality of the algorithm operation, it can further be modified to more particular scenarios, e.g. University, examination schedules, etc. A number of hours which are spent on creating a fruitful timetable can be reduced ultimately through the mechanization of the timetable issue. The cost of developing a site to operate only time table schedule can be minimized by developing a simple standalone application and also, the security aspect i.e. how secure the site is may allow hackers to easily penetrate and easy pasting of irrelevant posts is achieved by any students which are a disadvantage to the system too.

Also, research by (Mittal, Doshi, Sunasra, & Nagoure, 2015) developed an automatic timetable scheduler by using Genetic algorithm. The time requires to generate time table was able to reduce drastically and generate a timetable which is more accurate, precise and free of human errors. The first phase contains all the common compulsory classes of the institute, which are scheduled by a central team. The second phase contains the individual departmental classes. Presently, this timetable is prepared manually, by manipulating those of earlier years, with the only aim of producing a feasible timetable.

5 PROPOSED METHODOLOGY AND IMPLEMENTATION

In order to deal with timetabling issues, we are proposing a system which would automatically generate a timetable for the institute. Course and lectures will be scheduled in accordance with all possible constraints and given inputs and thus a timetable will be generated. The structure of time table generator consists of input data, relation between the input data, system constraints and application of a genetic algorithm. Figure 1, present flowchart for Timetable Generator.

5.1 The Structure for Time Table Generation

A. Input Data

The input data contains:

1. *Subject:* Data describes the name of courses in the current term.
2. *Room:* Data describes the room number and their capacity.
3. *Time intervals:* It indicates starting time along with the duration of a lecture.

B. Hard and Soft Constraints

A timetable must serve and overcome a number of constraints. Constraints are universally used to deal with timetabling problems. In GA, the types of constraints are soft and hard. Hard constraints in constructing an examination timetable should have no violation, an example is, a class cannot be at different locations at the same time. While Soft constraints are constraints that can be violated but violations must be minimized. Below are the lists of hard and soft constraints used in this research.

Hard constraints

a) No students should write two exams on the same day.
b) No two venues should be assigned for the same exam.
c) No two exams should be allocated to the same venue.
d) A number of days should not be violated.

Soft constraints

a) The Capacity of the venues should be considered.
b) Carryover students should be considered.

C. The Parameters for the implementation

Population Size = 60
Number of Generation = 100
Partially mapped Crossover
Single point Mutation
Rank Selection
Crossover probability = 0.8
Mutation rate = 0.03

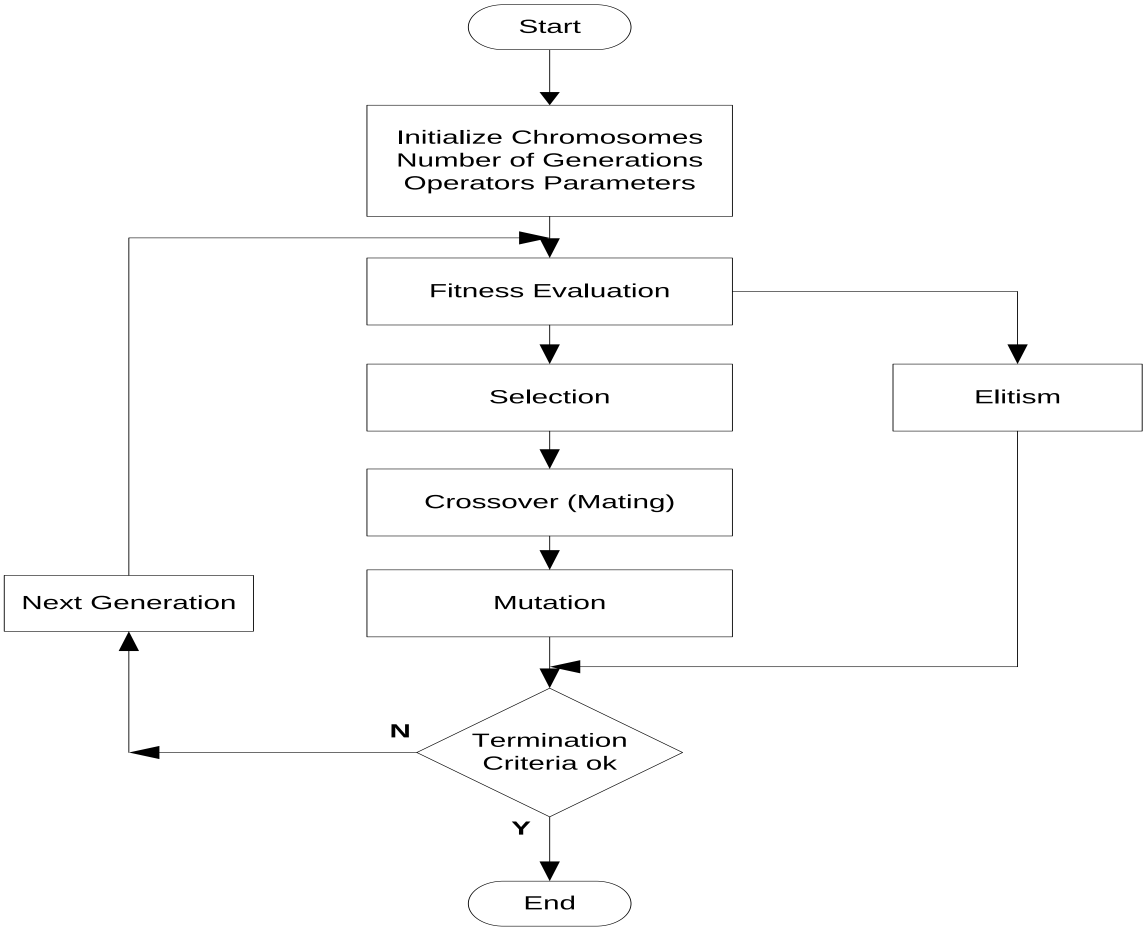

Figure 1: Flowchart of the Timetable Generator

6 RESULT AND DISCUSSION

The experiments aim to quantitatively determine the best results. The experiments focus on the best fitness that could be obtained with a fixed population size and generation iterations. Thus, the best value is assumed to have the lowest fitness value out of all values generated for each population size used.

In this paper, the time at which each result is generated and the fitness value of the generated result are used as the evaluation metrics. The results are presented in table 1, figure 2 and 3 respectively.

Figure 2 shows the minimum and maximum fitness value of the generated timetable against the population size, while figure 3 shows the maximum time of the generated timetable against different population size.

From the result in figure 2, it can be deduced that the fitness value is inversely proportional to the population size and directly proportional to the time spent to give results for each timetable i.e., the higher the population size, the better and smaller the fitness value and at the same time the longer it takes to give results for higher population size and vice versa.

The implication of this result is a solution, minimizing the time taken in timetable allocation and the clashing that usually characterize time table schedule.

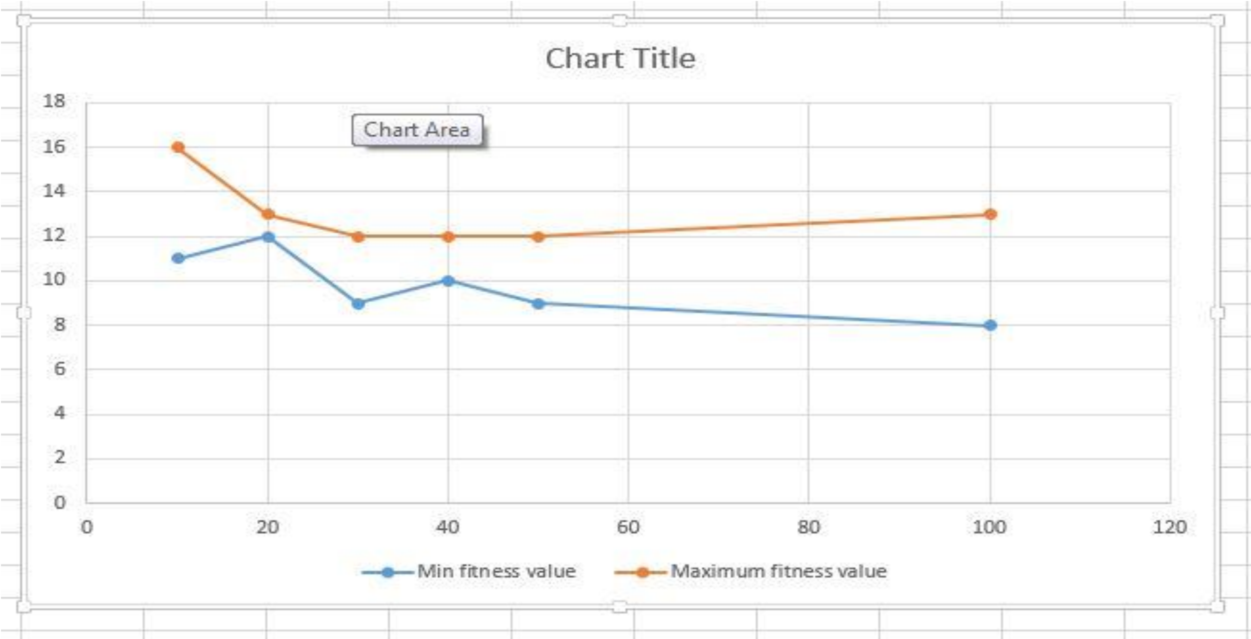

Figure 2: Population Size against Fitness Value

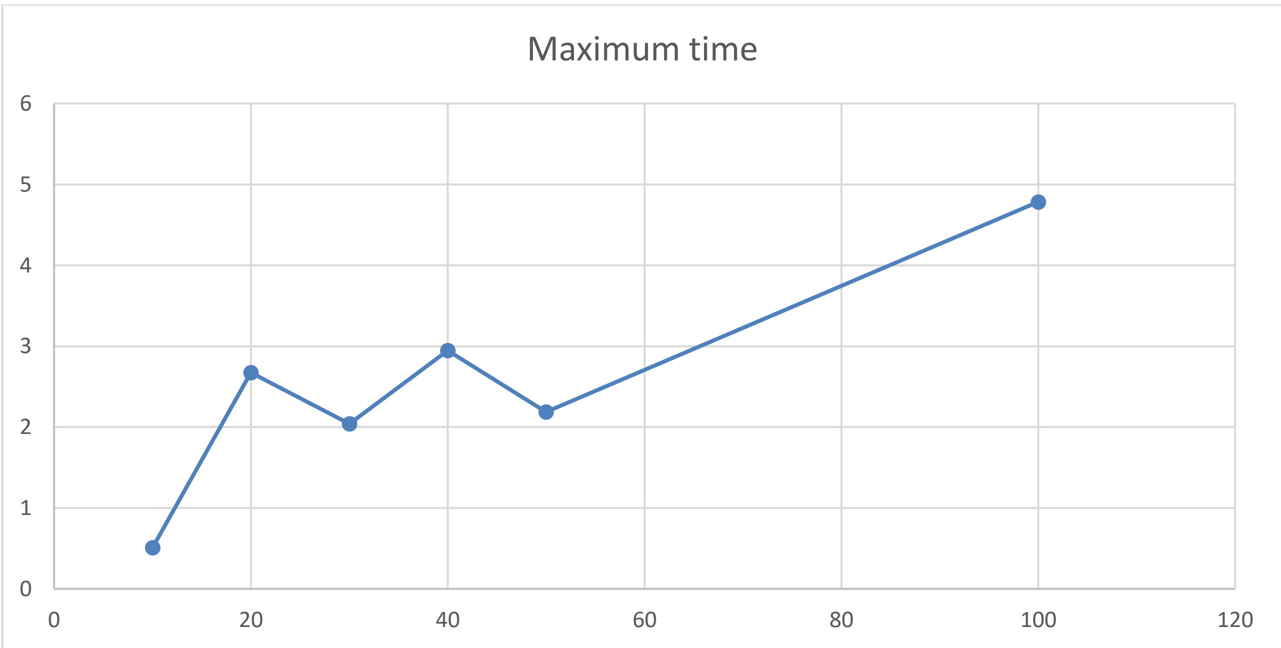

Figure 3: Maximum Time against Population Size

Table 1: Result Obtain for Proposed Timetable Generation using GA

S/n	Population size	Min fitness value	Meantime	Mean fitness value	Maximum fitness value	Maximum time
1	10	11	0.2744	11.8900	16	0.5084
2	20	12	1.5085	12.0300	13	2.6747
3	30	9	1.1091	9.6200	12	2.0425
4	40	10	1.6266	10.8300	12	2.9448
5	50	9	1.1002	9.7800	12	2.1885
6	100	8	8.2872	10.3300	13	4.7846

7 CONCLUSION

University time table scheduling is an NP problem. Many techniques exist for solving University time table scheduling problem. This research addresses the problem of time complexity and the clashes that usually associated with most university time table scheduling using GA. The undertaking is created in a manner that; no exam conflicts happen given elements to tailor the timetable as of a wish. This research also talks about the execution of a Genetic Algorithm (GA) to locate the best answer for examination timetabling issue. Furthermore, the selected operators' viability and effectiveness are concentrated on. An assortment of GA operators has been acquainted all together with meet the difficulties and the requests of examination timetabling issue. The proposed algorithm is tested with genuine information and the results present optimal solution for time table scheduling problem. The algorithm's conduct is investigated taking into account the varieties of the selected parameters. The best results are obtained with the moderate populace of 60 chromosomes, partially mapped crossover single point mutation, and rank selection. Consequently, the algorithm can initialize the population of examination timetable solutions that are feasible (satisfies hard constraints) at all time. The implication of this research is a solution, minimizing the time taken in timetable allocation and the clashing that usually characterize time table schedule.

REFERENCES

Abdelfattah, M., & Shawish, A. (2016). *Automated academic schedule builder for University's faculties.* Paper presented at the Proceedings of the World Congress on Engineering.

Al-Jarrah, M. A., Al-Sawalqah, A. A., & Al-Hamdan, S. F. (2017). DEVELOPING ACourse TIMETABLE SYSTEM FOR ACADEMIC DEPARTMENTS USING GENETIC ALGORITHM. *Jordanian Journal of Computers and Information Technology (JJCIT), 3*(1).

Amaral, P., & Pais, T. C. (2016). Compromise ratio with weighting functions in a Tabu Search multi-criteria approach to examination timetabling. *Computers & Operations Research, 72*, 160-174.

Assi, M., Halawi, B., & Haraty, R. A. (2018). Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem. *Procedia Computer Science, 126*, 899-906.

Bhaduri, A. (2009). *University time table scheduling using genetic artificial immune network.* Paper presented at the Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on.

Birbas, T., Daskalaki, S., & Housos, E. (2009). School timetabling for quality student and teacher schedules. *Journal of Scheduling, 12*(2), 177-197.

Burke, E., Eckersley, A., McCollum, B., Petrovic, S., & Qu, R. (2003). Similarity measures for exam timetabling problems.

Datta, D., Deb, K., & Fonseca, C. M. (2006). Solving class timetabling problem of IIT Kanpur using multi-objective evolutionary algorithm. *KanGAL, Report, 2006006*, 1-10.

Deeba Kannan, K. B. a. S. R. (2019). Solving Timetable Scheduling Problems Using Genetic Algorithm. *International Journal of Recent Technology and Engineering (IJRTE), 7* (5C), 168 - 170.

Dener, M., & Calp, M. H. (2019). Solving the exam scheduling problems in central exams with genetic algorithms. *arXiv preprint arXiv:1902.01360*.

Doulaty, M., Derakhshi, M. F., & Abdi, M. (2013). Timetabling: A state-of-the-art evolutionary approach. *International Journal of Machine Learning and Computing, 3*(3), 255.

Elsaka, T. (2017). *Autonomous generation of conflict-free examination timetable using constraint satisfaction modelling.* Paper presented at the 2017 International Artificial Intelligence and Data Processing Symposium (IDAP).

Kalayci, C. B., & GUNGOR, A. (2012). A Genetic Algorithm Based Examination Timetabling Model Focusing on Student Success for the Case of the College of Engineering at Pamukkale University, Turkey. *Gazi University Journal of Science, 25*(1), 137-153.

Matias, J. B., Fajardo, A. C., & Medina, R. M. (2018). *A fair course timetabling using genetic algorithm with guided search technique.* Paper presented at the 2018 5th International Conference on Business and Industrial Research (ICBIR).

Mei Rue (2010). Computer and Automation Engineering. *The 2nd International Conference (Volume: 4)*

Mittal, D., Doshi, H., Sunasra, M., & Nagoure, R. (2015). Automatic timetable generation using genetic algorithm. *International Journal of Advanced Research in Computer and Communication Engineering, 4*(2), 245-248.

Mohammed, M. A. (2017). A Review of Genetic Algorithm Application in Examination Timetabling Problem*" Mazin Abed Mohammed," Mohd Khanapi Abd Ghani," Omar Ibrahim Obaid," Salama A. Mostafa,"Mohd Sharifuddin Ahmad," Dheyaa Ahmed Ibrahim and" MA Burhanuddin" Biomedical Computing and Engineering Technologies (BIOCORE) Applied Research Group. *Journal of Engineering and Applied Sciences, 12*(20), 5166-5181.

Mitchell, M. (1998). *An introduction to genetic algorithms*: MIT press.

Mugdha K. P., Rakhe S. S., Prachi A. P. & Naveena N. T. (2014). Web Application for Automatic Timetable Generation. *International Journal of Current Engineering and Technology 2277 – 4106.*

Obaid, O. I., Ahmad, M., Mostafa, S. A., & Mohammed, M. A. (2012). Comparing performance of genetic algorithm with varying crossover in solving examination timetabling problem. *J. Emerg. Trends Comput. Inf. Sci, 3*(10), 1427-1434.

Parkavi, A. (2012). An e-College Time table Retrieval System Based on Service Oriented Architecture. *International Journal of Advances in Computing and Information Technology*.

Ross, P., Corne, D., & Fang, H.-L. (1994). Improving evolutionary timetabling with delta evaluation and directed mutation. *Parallel Problem Solving from Nature—PPSN III*, 556-565.

Rozaimee, A., Shafee, A. N., Hadi, N. A. A., & Mohamed, M. A. (2017). A framework for university's final exam timetable allocation using genetic algorithm. *World Applied Sciences Journal, 35*(7), 1210-1215.

Sani, H., & Yabo, M. (2016). Solving timetabling problems using genetic algorithm technique. *International Journal of Computer Applications, 134*(15).

Soyemi, J., Akinode, J., & Oloruntoba, S. (2017). *Electronic Lecture Time-Table Scheduler Using Genetic Algorithm.* Paper presented at the 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).