J Inno Sci Eng,2022, 6(2):220-232
https://doi.org/10.38088/jise.1006070

Research Article

Performance Improvement of Genetic Algorithm Based Exam Seating Solution by Parameter Optimization

Fatih Agalday [1*], **Ali Nizam** [2]

[1*] *Department of Computer Engineering, Mardin Artuklu University, 47060, Mardin, Turkey*
[2] *Department of Computer Engineering, Fatih Sultan Mehmet Foundation University, 34445, Istanbul,*

Abstract

Exam seat allocation has become a complex problem, with an increasing number of students, subjects, exams, departments, and rooms in higher education institutions. The requirements and constraints of this problem demonstrate characteristics similar to extensively researched exam timetabling problems. They plan for a limited capacity effectively and efficiently. Additionally, exam seating requires a seating arrangement to reduce the number of cheating incidents. In the literature, several genetic algorithm-based methods have been recommended to prevent students, who are close friends, from sitting close during the exams while providing the best exam session arrangement. We improved the performance of the genetic algorithm using parameter optimization and a new elitism method to increase the saturation rate and accuracy. The algorithm was tested on a real-world dataset and demonstrated high potential for the realization of a high-quality seating arrangement compatible with the requirements of educational institutions.

Keywords: Data science applications in education, Genetic algorithm, Multi-parameter optimization.

Cite this paper as: Fatih A. and Ali N. (2022). Performance improvement of genetic algorithm based exam seating solution by parameter optimization, 6(2):220-232.

*Corresponding author: Ali Nizam

E-mail: ali.nizam@fsm.edu.tr

Received Date: 07/10/2021
Accepted Date: 26/07/2022

1. Introduction

Cheating during academic examinations is a form of plagiarism that decreases education quality, reduces student satisfaction, and subverts the course evaluation system [1]. However, recent research has revealed that cheating is prevalent and has increased dramatically over the last thirty years [2]. According to [3], 54 percent of US and 45 percent of Canadian faculties report observing cheating during the exam, and 22 percent of US and 19 percent of Canadian students self-report serious test/exam cheating. Research in the US and Britain has repeatedly shown that more than half of university students cheat during their undergraduate years [4]. Preventing cheating in the first place is the best way to ensure fairness among students by measuring their academic achievements fairly [5].

Friendship was listed among the top five reasons as “My friend wants me to help” in the research on cheating in exams at universities [6]. In other research, the rate of copying from the closest seated colleague was higher than 50 percent, and helping others to cheat was more than 70 percent [7]. The faculty reporting rate of using an electronic device for cheating (11 percent) was lower than copying from another student with their knowledge (33 percent) and helping someone else cheat on a test (29 percent) [3]. Communication during the exam is significantly higher among neighbors who are also friends outside the examination room and unfamiliar neighbors will be less confident in interacting, which partially inhibits unwanted communication [8]. Additionally, detecting and punishing student whispering is often difficult and subjective. The easiest method to decrease cheating is to spread students out, leaving an empty seat between them in each row. Unfortunately, classroom size may not be sufficient to spread students effectively [9].

This study aims to provide the best session arrangement to prevent placing familiar students in a close location during an exam to minimize the possibility of cheating prevailing in central exams via GA. We investigated the best methods for passing elite individuals to the next population in the exam seating problem domain. In addition, we tuned the GA parameters, such as the mutation probability, crossover probability, and population size to determine reasonable settings. It reduces time consumption and manual human dependency and optimizes the usage of classroom spaces. We employed multiple parameters such as graduation from the same department or school, living with the same relative, or the same birthplace to identify friendship.

The remainder of this paper is organized as follows. Section 2 presents the motivation for the study and problem definition and describes various research works that have been conducted on exam scheduling and seating problems using GA and other methods. Section 3 discusses the selection mechanism, crossover and mutation operators, and parameters of the existing and proposed methods. Section 4 presents results that demonstrate the effectiveness of the proposed method. Finally, concluding remarks and discussions are presented in Section 5..

2. Background and Related Works

The steps of the exam scheduling process include examination timetabling, room assignment, and student seat placement. Examination timetabling begins by listing exams that cannot be scheduled in the same time slot, generating available examination time slots; and identifying available exam rooms and courses that need to be scheduled [10]. A large number of events to be scheduled and constraints imposed on timetabling make the search space for the problem extremely large. Indeed, a manual solution can require considerable effort or is nearly impossible [11]. Many constraints should be satisfied, as a course taught at many places must have the same exam date and an invigilator can be assigned to any campus [12]. Timetabling problems have attracted the attention of the scientific community and interest in this field has increased [8]. However, the lack of room for information in currently used datasets means that the optimization function used to measure solutions does not incorporate all necessary issues [13].

The problem of assigning exams to classrooms has been investigated in research using seating capacity in the classroom as a hard constraint [14]. Some researchers have focused on the invigilation schedule which aims to assign rooms or invigilators after the exam timetable has been approved; however, it has received less attention from the research community because of the lack of available datasets [15]. There is relatively little research on the assignment of exams to classrooms by targeting the minimization of the total used capacity [16].

Generally, student placement on the seat is not considered when generating examination timetables. Some studies have focused on minimizing student movement between rooms by assigning them to the same room when they are scheduled to sit consecutive examinations on the same day [10]. Students should not undergo two examinations simultaneously over a designated period within a finite area of space. A viable solution must satisfy this ‘hard’ constraint [17].

When the focus shifts to the prevention of cheating, the successful distribution of seats in an exam is one of the primary concerns of a well-prepared exam hall, ensuring the maximum distance between students while making the best use of seats. The general aim is to replace friends separately during an exam and place non-befriended students next to each other as much as possible [8]. However, exam seats are usually allocated manually based on the basic rules [18]. Many institutes perform this task manually using Excel sheets causing excessive wastage of time and manpower [19].

Social network analysis has been used to detect student friendships and avoid cheating [8]. Friendships constructed by a friend’s friend create a complex network topology [20]. Because not every student has a public social media account, we determined the friendship level via the student’s department, program, and birthplace parameters using a real-world dataset.

3. Methods

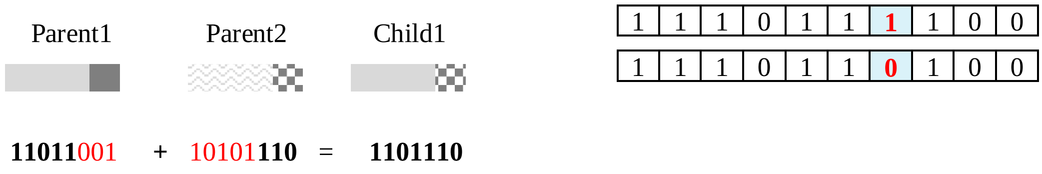
This section explains the proposed method, its advantages, and the differences between the proposed and previous methods.

3.1. Genetical Algorithm Basic Structure

The solution to exam seating is constraint programming, which solves a given set of variables, a finite set of possible values that can be assigned to each variable, a list of constraints, and values of the variables that satisfy every constraint as the timetabling problem [21]. GA, which is a suitable optimization technique for solving hard and highly constrained problems, is a popular meta-heuristic method that addresses the university timetabling problem [22]. GA was chosen to solve the exam seating problem because of its robustness in many studies [23–26]. In addition, in some studies, the selection strategies of GA were incorporated into other algorithms, such as bee colony optimization, to increase the effectiveness of the solution to the examination timetabling problem [27].

GA which was first introduced by John Holland [28] allows the building of a solution inspired by nature. It aims to find the best solution for the survival of living organisms that adapt to the natural environment and the elimination of living organisms that cannot adapt [29]. The population is composed of chromosomes, and the population size (PS) is the chromosome number. The initial population is generated from random values [30] and interchanges to produce new individuals (chromosomes). The population is assigned a suitability value by comparing each solution with other solutions using a fitness function for each generation number (GN) and the best individuals of the population are transferred to the next generation. This iteration continues to try solutions until the fitness value is zero or the maximum number of generations (GN_{max}) is reached.

The natural selection process can be clogged somewhere and circulates the same solution. Crossing and mutation are two important genetic operators used to create new solutions [31]. In the crossover operation, the bits of the first chromosome and the other chromosome shift according to a predefined rule to produce chromosomes with different structures and increase chromosome richness, as shown in Figure 1a.



(a) Crossover

(b) Mutation

Figure 1. Operators in GA.

The mutation operator changes the homogeneous structure in the vicious circle formed when searching for the best solution. If all chromosomes are identical or similar, the optimum mean fit value may not be reached. Randomly selected genes in the chromosome are changed as shown in Figure 1b.

3.2. Applying Genetical Algorithm to Exam Seating

The exam seating problem has a variety of constraints similar to those of the timetabling problem. When searching for an optimal solution, E events can be allocated to P places in P^E ways that create an NP-hard problem [32]. Heuristic algorithms are used to solve NP problems in polynomial time or approximate it. Therefore, GA was applied to exam seating in the literature [33]. The students were placed into classrooms with different capacities as a matrix layout as shown in Figure 2.

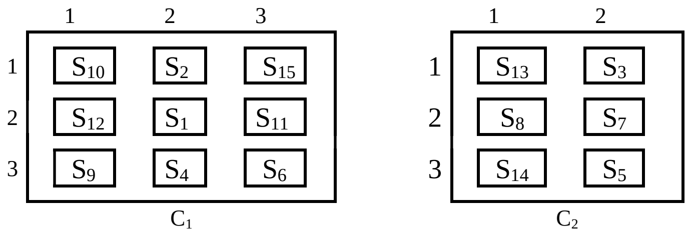


Figure 2. The session layout in the classrooms.

The initial population, chromosome encoding, and fitness function structures are arranged to solve the exam seating problem. The pseudo-code of the proposed GA method is shown in Figure 3.

```
begin
    Data: Create the initial population
    Result: $F$
while do
    Calculate F while $(GN_i < GN_{max})$: The modified fitness function is applied on each chromosome;
    Cumulative Probability Calculation: Roulette wheel transfer to the intermediate population;
    CrossOver: Selection of chromosome pairs with an integrated repair algorithm;
    Mutation: Permutation coding;
    Elitism: Replace the worst members of the existing population with the best members of the intermediate
    population;
end
```

Figure 3. The proposed GA for exam seating

3.3. Chromosome Encoding and The Initial Population

Each chromosome was constructed by using a random distribution of student information to create an initial population. The initialization procedure is an important issue in GA implementation because it supports the fact that individuals across the first generation spread across the entire search space with as much diversity as possible [34]. The index information can be given by:

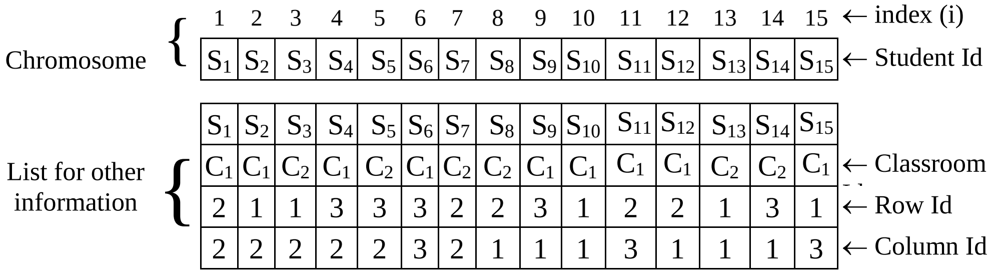
$$\mathrm{c} \in 1, \ldots, \mathrm{R} \ \text{ where R is the number of classrooms} \tag{1}$$

$$\mathrm{i} \in \{1, \ldots, \mathrm{L}\} \ \text{ where L is the number of students} \tag{2}$$

$$\mathrm{t} \in \{1, \ldots, \mathrm{T}\} \ \text{ where T is the total number of seats} \tag{3}$$

$$\mathrm{s_i} = (\mathrm{a_1}, \mathrm{a_2}, .., \mathrm{a_n}) \text{ where } \mathrm{s_i} \text{ is a tuple containing the attribute } (a) \text{ of } i\text{th student} \tag{4}$$

The form and definition of the chromosome structure are very important to improve the performance of the algorithm. One student was placed in each row or a chromosome $index = i$ represents $(ClassId, RowId, ColumnId)$ information [36]. The other information about the student such as undergraduate program, birthplace, and residence information has been stored in another list and referenced using student identity to reach a light and high-performance structure. Figure 4 shows an example distribution of a chromosome structure for 15 students and 2 classrooms.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	← index (i)
Chromosome	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	← Student Id

List for other information	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	
	C_1	C_1	C_2	C_1	C_2	C_1	C_2	C_2	C_1	C_1	C_1	C_1	C_2	C_2	C_1	← Classroom
	2	1	1	3	3	3	2	2	3	1	2	2	1	3	1	← Row Id
	2	2	2	2	2	3	2	1	1	1	3	1	1	1	3	← Column Id

Figure 4. Chromosome structure

3.4. The Fitness Function

The system creates a penalty when two students are likely to recognize each other sitting back-to-back or next to each other in a chromosome. The fitness function $\mathrm{F}(k)$, which represents the total penalty for kth generation, is given by Equation (5) as

$$\mathrm{F}(k) = \sum_{\mathrm{i=1}}^{\mathrm{L}} \sum_{\mathrm{j=i+1}}^{\mathrm{L}} proximity(\mathrm{s_i}, \mathrm{s_j}) \tag{5}$$

To detect proximity, the relationship immediately behind and to the right of each student's seat is examined. When proximity is detected, the chromosome receives a $\mathrm{proximity}(s_i, s_j)$ score representing the degree of proximity between the ith and jth students' locations as s_i and s_j. In Equation (6), the first row indicates that the jth student sits next to the seat of the ith student and they have the same proximity attributes. The second row indicates the jth student sitting behind the ith student. Only the seat behind the student is examined in the rightmost column of the classroom and only the seat next to the student is examined in the last row of the classroom. The proximity calculation for the fitness function is

$$\mathrm{proximity}(t_i, t_j) = \begin{cases} 1 \text{ if } i = j + 1 \ \wedge (s_1 = s_2) \\ 1 \text{ if } i = j + N \ \wedge (s_1 = s_2) \\ 0 \qquad \text{otherwise} \end{cases} \tag{6}$$

The classroom contains a total of M rows and N columns. We accepted all friendship parameters to be the same and the similarity coefficients as 1. The fitness function is subject to the following additional constraint: more than one student cannot be assigned to one seat:

$$\sum_{i=1}^{L} v_{it} \ \leq 1 \ \text{ for each i} \in 1, \ldots, \mathrm{L} \tag{7}$$

v_{it} 1 if student i is assigned to seat t and 0 otherwise.

The target solution must minimize the fitness value to zero or near zero.

3.5. The Crossover Operator

After determining the initial population, the chromosomes are crossed according to their fitness values, and new offspring individuals are obtained at the end of each iteration to determine which individuals are the correct solution, as shown in Figure 5. $\mathrm{s_1}, \mathrm{s_3}, \mathrm{s_6}$ have a friendship; thus, $Chromosome_2$ has one close seat (s_1, s_3) and $F = 1/3$. $Chromosome_3$ has two close seats $((s_1, s_3), (s_1, s_3))$ and $F = 2/3$.

Chromosome 2 (F=1/3)

S_1	S_2	**S_3**	S_4	S_5	**S_6**	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
C_1	C_1	**C_2**	C_1	C_2	**C_1**	C_2	C_2	C_1	C_1	C_1	C_1	C_2	C_2	C_1
1	2	**1**	3	3	**2**	2	2	3	2	3	1	1	3	1
2	2	**2**	3	2	**1**	1	2	2	3	1	1	1	1	3

Chromosome 3 (F=2/3)

S_1	S_2	**S_3**	S_4	S_5	**S_6**	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
C_1	C_1	**C_1**	C_1	C_2	**C_2**	C_1	C_2	C_1	C_1	C_2	C_2	C_1	C_2	C_1
3	3	**2**	3	3	**2**	1	3	2	2	2	1	1	1	1
3	1	**2**	2	1	**1**	3	2	1	3	2	1	2	2	1

Figure 5. The crossover of chromosomes

i.th chromosome roulette selection probability function is given in Equation (8)**:**

$$P(i) = \frac{\mathrm{F}(i)}{\sum_{j=0}^{CN} F(j)} \tag{8}$$

Chromosome number (CN) defines the crossing point. Candidate chromosomes for selection are determined by generating a random number using a crossing point. For example, when random point 5 is determined, the first five genes of $Chromosome_2$ and the last ten genes of $Chromosome_3$ combine to form the $NewChromosome$.

Chromosomes that will survive in the next population are determined by cumulative ratio values using a randomly generated number between 0 and 1 in the roulette wheel selection method. Crosspoint selection is implemented by generating random values for chromosomes that address the locations of the chromosome pairs to be crossed. For example, the first pair of chromosomes to be crossed is 7 and 2, when random numbers $(7-2-5-8-1-4-3-6)$ are generated in a population. The genes after the crossing point of the first and second chromosomes are combined to obtain the second offspring.

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
C_1	C_1	C_2	C_1	C_2	C_2	C_1	C_2	C_1	C_1	C_2	C_2	C_1	C_2	C_1
1	2	1	3	3	2	1	3	2	2	2	1	1	1	1
2	2	2	3	2	1	3	2	1	3	2	1	2	2	1

Figure 6. Crossover in SPCM

The direct implementation of GA crossover causes genes to repeat in the chromosome, as shown in Figure 6 [32]. When a crossover is performed without considering the repeating genes in the chromosome structure, over 20 percent of the chromosome structures are repeated causing the same student to be replaced in more than one place in the room or two different students to be replaced in the same seat [37]. In the example above, the students, s_5 and s_8, were in the third seat of the second classroom. Therefore, this settlement plan cannot be applied in the real world. During the exchange process, the generic partially mapped crossover operator performs no feasibility check by default, and a repair mechanism is required to make each child chromosome feasible [34]. An improved crossover model was designed to eliminate an additional repair function as satisfying the constraints of the problem [37]. The crossover of ith gene in chromosomes X and Y is given by equation (9):

$$PMX = \{X_{Y[i]} = X_i \ then \ X_{[i]} \leftrightarrow Y_{[i]}\} \tag{9}$$

The process begins with selecting random crossover points on the parents. The gene on the first chromosome is copied on the same chromosome to the location of the corresponding gene on the second chromosome when transferring the segment from the first parent to the first offspring. The same procedure is repeated for the second chromosome with the parents' roles reversed. This prevents the replication of the same seating position on the same chromosome, as shown in Figure 7.

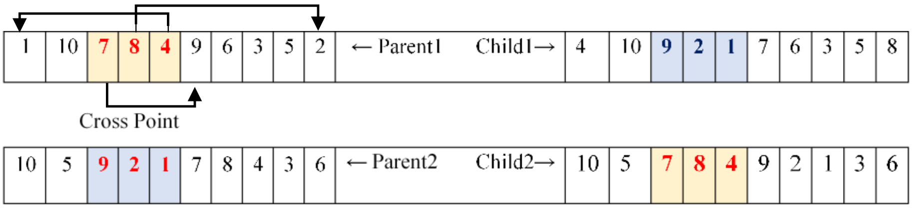


Figure 7. Partially matched crossover [37]

The genes (7,8,4) in Cromodsome_1 were crossed with the genes (9,2,1) in Cromodsome_2. Gene 9 in Cromodsome_2 was replaced 7 in Cromodsome_1. Gene 9 in Cromodsome_1 was placed at the location of gene 7 in Cromodsome_1. The same process was applied to others.

3.6. Elitist Selection

Elitist selection based on cloning the best single individual from one generation to the next is popular [36]. The average fitness of the population will increase by transferring the best chromosomes to the next generation. Thus, in a previous work [37], the best chromosome of the generation, which had the best fitness value, was selected as the elite and transferred to the next generation. Although an elitist procedure is sufficient for many GA applications, there may be some cases that benefit from selection schemes that provide different types of search capabilities [36]. Our observations showed that the method of preventing repeated genes by identifying transfer location using gene values [37] caused the distribution of genes to different points during the transition and slowed down the saturation in the crossover operator. To improve system performance, this paper explores several elitist procedures for the design of GA-based exam seating. We attempted to replace a number of the worst chromosomes of the previous generation with the best chromosomes from the new generation. We evaluated the effects of varying the number of replaced chromosomes against generation number.

The effects of the MR, CR, and PS values on the results of the previous algorithms were examined in detail and then applied to our elitist selection to compare the results. We determined the best values for the parameters step-by-step and used them as the initial values for the next step. We evaluated the saturation process by observing the penalty score of each algorithm over generations until the saturation point was reached. Each test was repeated 10 times, and the results of the analysis were averaged to reduce the error rate to less than 10 [38]. The proposed algorithm was implemented using Python.

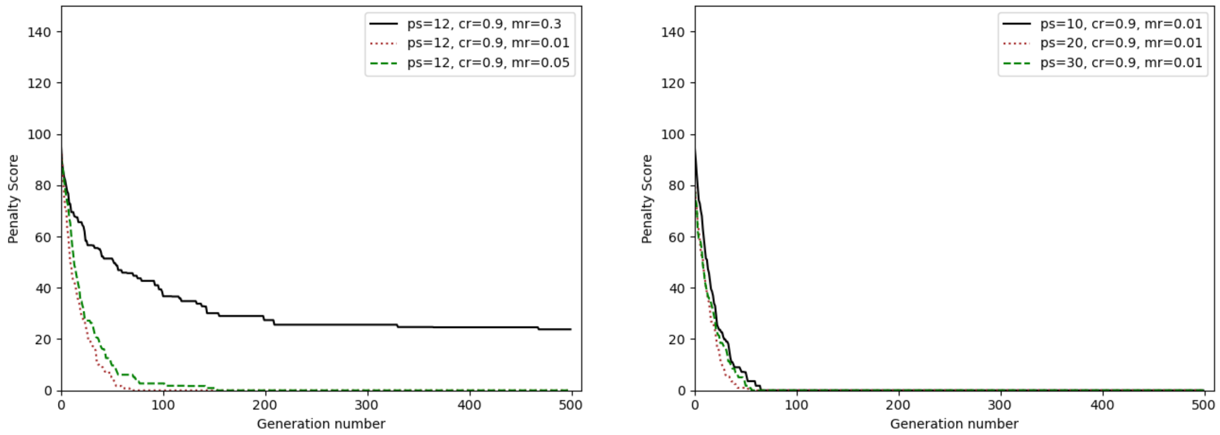
The data obtained from the distance education application and research center of a university were used in this study. The data contained 68 unique undergraduate programs, 157 birthplaces, and 138 residential addresses.

4. Results and Discussion

Although there is no common judgment on the most suitable parameter values, we used David Schaffer's experimental study values as $0.75 - 0.95$ for CR, $0.005 - 0.01$ for MR, and $20 - 30$ for PS [39]. In addition, it is recommended to take PS as an average of 10, in which PS affects the performance [40]. Thus, PS the value was chosen as 12, as described in [37].

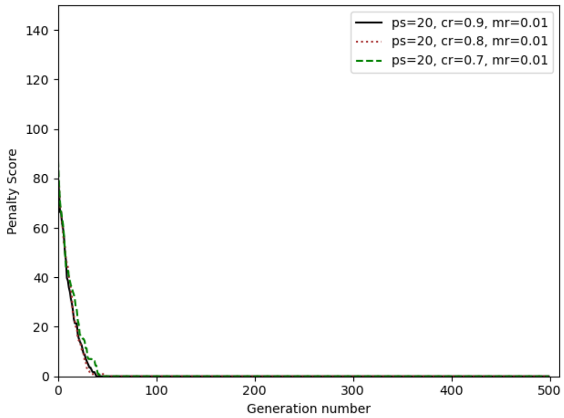
The effect of the parameter values on the saturation process of the previous algorithm is shown in Figure 8a-c. When the number of generations increased, the production of defective genes in the chromosome increased respectively until the saturation point was reached. The maximum number of generations that supported reaching the maturation point was detected as in the range of 50 to 500.

It was claimed that increasing MR has a positive contribution to the solution of the problem [41]. Thus, our investigation focused on MR parameter values having a range between 0.05 and 0.3 as in [37]. However, the results indicated that low MR values provided better saturation rates as shown in Figure 8a. We evaluated the effect of PS on the results using the best CR values determined as 0.01 as shown in Figure 8b. Thus, after testing different values, the parameter values providing the best session order were chosen inside the limits mentioned in the literature [39] as CR= 0.9, MR = 0.01, and PS=20.



(a) MR effect analysis

(b) PS effect analysis



(c) CR effect analysis

Figure 8. The effect analysis of the parameters

Figure 9 shows the parametric analysis of the proposed elitism procedure. We have added the results of the previous algorithm for the determined parameter values above to allow a comparison of the results of different elitisms to the figure.

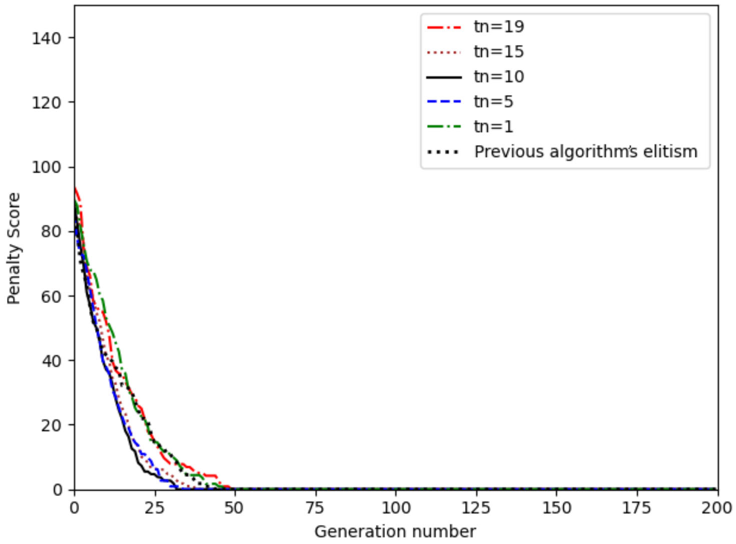


Figure 9. Demonstration of chromosomes in SPCM

The penalty scores decreased to zero for the 30th generation compared to the 50th generation in the previous algorithm, as shown in Figure 9. The TN values yielded better results when approximately 50 percent of the genes were altered.

Our comparative analysis results suggest that the proposed elitism method provides better saturation rates and lower penalty scores than the previous algorithms. The saturation rate decreases when the number of transferred chromones is low, and the elitism results correlate with those of the previous algorithm [37]. In addition, the model result did not contain any misplacement of students as appeared in the study [32].

Despite the important improvements, there are still some potential threats to the validity of our study. We evaluated the effects of parameter values step by step and transferred the best values to the next evaluation stage to decrease the size of the test attempts. This could have caused some parameter combinations to be missed. Trying combinations of all possible parameter values can provide slightly improved results. Additionally, the dataset on which the model was tested was small, considering the number of students in the centralized exams. This limits the evaluation of the algorithm performance against large examination datasets.

5. Conclusion

The principal contribution of this study is the parametric optimization of GA-based exam seating by analyzing the effect of parameter values on the system performance and saturation rate. The accuracy and performance of the algorithm were evaluated by using the student dataset from a university. The results indicated that the parametric optimization and improved elitism methods fulfilled two purposes simultaneously: increasing the robustness of the algorithm and preventing unrealistic seating plans.

The proposed model has important theoretical and practical implications. From a theoretical perspective, researchers can use the method and dataset as inputs for new exam seating, timetabling, and GA research. From a practical perspective, our process model can increase the speed of preparing reliable plans with optimal session orders for centralized exams where thousands of people are held simultaneously.

Some limitations of our study should be addressed to strengthen its applicability further. First, the effects of different parameters, such as school friendship and class friendship on exam seating accuracy and performance should be evaluated in detail. Second, the determination of the minimum and maximum classroom spaces should be addressed to determine the optimum space allocation for exams. Finally, the integration of exam timetabling and seating should be investigated to develop a comprehensive plan for seat placement.

6. Acknowledgements

We thank the Academic Management Office of Mardin Artuklu University for providing the examination data.

References

[1] Walker J. (1998). Student plagiarism in universities: What are we doing about it? High Educ Res Dev, 21(1), pp. 89–106.

[2] McCabe D.L., Klebe L.T., Butterfield K.D. (2010). Cheating in Academic Institutions: A Decade of. Ethics Behav., 8422, pp. 37–41.

[3] Mccabe D.L. (2005). Cheating among college and university students: A North American perspective. Int J Educ Integr, 1(1).

[4] Newstead S. (1996). Individual differences in student motivation. J Educ Psychol., 88(2), pp. 229–41.

[5] Yee K, MacKown P. (2009). Detecting and preventing cheating during exams. Pedagogy, not Policing.

[6] Wang J., Tong Y., Ling M., Zhang A., Hao L, Li X. (2015). Analysis on test cheating and its solutions based on extenics and information technology. Procedia Comput Sci., 55(2015):1009–14. DOI:10.1016/j.procs.2015.07.102

[7] David L.T. (2015). Academic cheating in college students: relations among personal values self-esteem and mastery. Procedia - Soc Behav Sci., 187, pp. 88–92, DOI:10.1016/j.sbspro.2015.03.017

[8] Topîrceanu A. (2017). Breaking up friendships in exams: A case study for minimizing student cheating in higher education using social network analysis, Comput Educ., 115, pp. 171–87, DOI: 10.1016/j.compedu.2017.08.008

[9] Danielsen R.D., Simon A.F., Pavlick R. (2006). The culture of cheating: From the classroom to the exam room. J Physician Assist Educ. , 17(1), pp. 23–9.

[10] Ayob M., Malik A. (2011). A new model for an examination-room assignment problem. J Comput Sci., 11(10):187–90.

[11] Güler M.G., Geçici E. (2020). A spreadsheet-based decision support system for examination timetabling. Turkish J Electr Eng Comput Sci., 28(3): pp. 1584–98.

[12] Elsaka T. (2017). Autonomous generation of conflict-free examination timetable using constraint satisfaction modelling. In: IntArtificial Intelligence and Data Processing Symposium (IDAP), p. 1–10.

[13] Vasupongayya S., Noodam W., Kongyong P. (2013). Developing examination management system: Senior Capstone Project, a case study. Int J Comput Inf Eng., 7(7), pp. 1046–52.

[14] Dammak A., Elloumi A., Kamoun H. (2006). Classroom assignment for exam timetabling. Adv Eng Softw., 37(10), pp. 659–66.

[15] Mohmad Kahar M.N., Kendall G. (2014). Universiti Malaysia Pahang examination timetabling problem: Scheduling invigilators, J Oper Res Soc., 65(2), pp-214–26.

[16] Elloumi A., Kamoun H., Jarboui B., Dammak A. (2014). The classroom assignment problem: Complexity, size reduction and heuristics, Appl Soft Comput J., 14 (PART C), pp-677–86. DOI: 10.1016/j.asoc.2013.09.003

[17] Burke E.K., Mccollum B., Mcmullan P., Qu R. (2006). Examination timetabling: A new formulation, In Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling, Brno, pp. 373-375.

[18] Chaki P.K. (2016). Algorithm for efficient seating plan for centralized exam system, Int Conf Comput Tech Inf Commun Technol., pp. 320–5.

[19] Inamdar A., Gangar A., Gupta A., Shrivastava V. (2018). Automatic Exam Seating & Teacher Duty Allocation System, Second Int Conf Inven Commun Comput Technol. (Icicct), pp. 1302–6.

[20] Wang X.F., Chen G. (2003). Complex networks: Small-world, scale-free and beyond, IEEE Circuits and Systems Magazine, 2003, 3(1), pp. 6–20.

[21] Brailsford S.C., Potts C.N., Smith B.M. (1999). Constraint satisfaction problems: Algorithms and applications, European Journal of Operational Research, Vol. 119, pp. 557–81.

[22] Hosny M., Fatima S. (2011). A Survey of Genetic Algorithms for the University Timetabling Problem A Survey of Genetic Algorithms for the University Timetabling Problem Manar Hosny and Shameem Fatima, Int Conf Future Inf Technol IPCSIT 13., pp. 34–9.

[23] Abdelhalim E.A., Khayat G.A. El. (2016). A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA), Alexandria Eng J, 55(2), pp.1395–409. DOI: 10.1016/j.aej.2016.02.017

[24] Karova M. (2004). Solving Timetabling Problems Using Genetic Algorithms, 2Th Int’l Spring Seminar on Electronics Technology, pp. 20–2.

[25] Rozaimee A., Shafee A.N., Anissa N., Hadi A., Mohamed M.A. (2017). A Framework for University’s Final Exam Timetable Allocation Using Genetic Algorithm, World Appl Sci J., 35(7), pp. 1210–5.

[26] Pillay N., Banzhaf W. (2010). An informed genetic algorithm for the examination timetabling problem, Appl Soft Comput J., 10(2), pp. 457–67.

[27] Abdullah S., Alzaqebah M. (2013). A hybrid self-adaptive bees algorithm for examination timetabling problems, Appl Soft Comput J., 13(8), pp. 3608–20. DOI: 10.1016/j.asoc.2013.04.010

[28] Holland J.H. (1992). Adaption in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press

[29] Şen Z. (2004). Genetik Algoritmalar ve En İyileme Yöntemleri, İstanbul, Su Vakfı Yayınları

[30] Karaboğa D. (2004). Yapay Zekâ Optimizasyon Algoritmaları, İstanbul, Atlas Yayınevi, pp. 75–112.

[31] Negnevitsky M. (2005). Artificial intelligence: A guide to intelligent systems, Systems (2nd Edition), Pearson Education.

[32] Tindell K.W., Burns A, Wellings A.J. (1992). Allocating hard real-time tasks: An NP-Hard problem made easy, J Real-Time Syst., (4), p. 145.

[33] Bulut F., Subaşı Ş. (2015). Best Seating Plan For Centeral Exams Using Genetic Algorihtms. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Derg., 17(51), pp. 122–37.

[34] Beligiannis GN, Moschopoulos C, Likothanassis SD. (2009). A genetic algorithm approach to school timetabling, J Oper Res Soc., 60(1), pp. 23–42.

[35] Akkan C., Gülcü A. (2018). Computers and Operations Research A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem, Comput Oper Res., 90:pp. 22–32. DOI:10.1016/j.cor.2017.09.007

[36] Soremekun, G., Gürdal, Z., Haftka, R. T., & Watson, L.T. (2001). Composite laminate design optimization by genetic algorithm with generalized elitist selection. Computers & structures, 79(2), 131-143.

[37] Ağalday, M. F. (2018). Genetik algoritma ile merkezi sınavlarda tek ve çok boyutlu yakınlığa göre en iyi oturum planının oluşturulması (Master's thesis, Fatih Sultan Mehmet Vakıf Üniversitesi, Mühendislik ve Fen Bilimleri Enstitüsü).

[38] Nadeau C., Bengio Y. (2003). Inference for the Generalization Error. Mach Learn Norwell Kluwer Acad Publ, 52(3), pp. 307–13.

[39] Schaffer J.D. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In: 3rd International Conference on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, p. 51–60.

[40] Aksu Ö. (2008). Yeni Bir Paralel Genetik Algoritma Modeli Ve Analog Devre Tasarımına Uygulanması Yüksek Lisans Tezi, Kayseri Erciyes Üniversitesi.

[41] Grefenstette J.J. (1992). Genetic algorithms for changing environments. Parallel Probl Solving from Nat., 2, pp. 137–44.