AJHAS

https://albalqajournal.ammanu.edu.jo/SC.aspx
Vol.(1) No.1 Jan 2022

Recived: 21 Oct 2021
Accepted: 30 Nov 2021
Published online: 30 Nov 2021

Al-Balqa Journal
for Health & Applied Sciences

A Hybrid Algorithm to Solve COVID-19 Physical Distancing for Examination Scheduling: Al-Ahliyya Amman University Case Study

Hassan Tarawneh[1], Khalid Kaabneh[2,*]

[1] Department of Data Science & Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman – Jordan, h.tarawneh@ammanu.edu.jo ORCID: 0000-0001-8952-1977
[2*] Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman – Jordan, Zip-Code (Postal Address): (19328), k.alkaabneh@ammanu.edu.jo ORCID: 0000-0002-2772-5664

Abstract

COVID-19 is rapidly spreading throughout the world's communities. The terms "physical distance" and "social distancing" are used to emphasize the importance of always remaining at least 6 feet away from others. As universities reopen, it's more important than ever to enforce the guideline when creating examination schedules with fixed variables like rooms and time slots. Many algorithms have been implemented to successfully schedule University Examination Time Tabling (UETT). This research offers a novel method for upgrading the genetic algorithm with tabu list memory. The proposed approach achieved results that reduce student crowding compared to the standard approach, with less than 50% of the actual university capacity over time slots. The quantitative results proved that the use of this Approach will reduce the risk of crowding during university exams, at university campus.

Keywords

Al-Ahliyya Amman University, COVID-19, Examination Timetable, Genetic Algorithm, Social Distancing, Tabu list.

Author Biographies

Hassan Tarawneh: received his bachelor's degree in Information Systems in 2005, and a MC Degree in Intelligent Systems in 2008. In 2013, He received his Ph.D. in Computer Science/Artificial intelligence from the National University of Malaysia. He worked for five years at Isra University and Middle East University as Assistant Professor. Currently, he is an Assistant professor at Al-Ahliyya Amman University at the Department of data sciences. His research interests focus on Artificial Intelligence fields include Optimization algorithms, Robotics, Expert Systems, Image processing, NP-Complete Problem. He published many journals and conferences papers in a good and high-referred journal index.

Khalid Kaabneh: received his bachelor's degree in Electrical Engineering and Computer Engineering in 1989. In 1991, he earned an MS Degree in Management Information Systems. A Ph.D. was bestowed to him in 2001 for his research in the area of Multimedia audio watermarking. All degrees were awarded from The George Washington University in Washington DC, USA. Prof. Alkaabneh was an assistant professor at Mutah University and an associate professor at the department of multimedia, Isra University. Faculty dean of Computer Science and Informatics at Amman Arab University. Currently, he is the vice president and dean of Scientific Research at Al-Ahliyya Amman University. His research interests include areas like Digital Communications, Wireless Networks, Digital Multimedia, Copyright Protection, and Data and Systems Security.

1. Introduction

The university examination timetabling problems is considered one of the most difficult tasks facing universities, as the construction of the examination schedule manually required a lot of effort (Abayomi-Alli, Olusola, et al ,2019), where the problem begins when universities need to define a set of exams for a fixed number of time slots.

The university examination timetabling problems (UETP) is task of allocating Exams to the rooms and timeslots, in such a way as to satisfy and minimize two types of constrains (hard constraints and soft constraints).

Hard constraints are the constraints that must not be violated in any case, such as:

- There are no two exams in the same time slot and room.
- The number of students taking exam should not exceed the number of seats in the exam room.

Soft constraints are the restrictions that should be minimized if possible, such as:

- Having an exam at the end of the day (last timeslot).
- A group of students have more than one exam per day.
- A group of students have more than three consecutive exams per day.

Scheduling problem such as university examination timetabling (UETT) is considered as a non-deterministic polynomial (NP) complete problem (Pillay N,2008). Many algorithms were implemented to accomplish scheduling tasks effectively. Abdullah, S., Burke, E. K., & Mccollum, B. (2005) defined a scheduling algorithm as "the allocation, subject to constraints, of given resources to objects being placed in space-time, in such a way as to satisfy, as nearly as possible, a set of desirable objectives." Thus, for any scheduling algorithm, a set of objectives needs to be achieved.

According to Tarawneh, H., Ayoub, M, & Ahmad, Z. (2013), there are three main timetable problems in the field of education: university timetabling, school examination timetabling, and courses timetabling. Several solutions were suggested to solve the problem of timetable scheduling. Aladag, C. H., Hocaoglu, G., & Basaran, M. A. (2009) implemented those solutions applying high-level strategies included meta-heuristics techniques, while Avanthay, C., Hertz, A., & Zufferey, N. (2003) used to solve similar problems such as the genetic algorithm (GA), Tabu Search (TS) algorithm, Simulated Annealing (SA) algorithm, and variable neighborhood search (VNS) algorithm.

Chen, M., et. (2020) said that: "We now have the elements of a theory of heuristic (as contrasted with algorithmic) problem solving; and we can use this theory both to understand human heuristic processes and to simulate such processes with digital computers", which means that heuristic search does not guarantee to find the optimal solution, but it has procedure to work. Chen, R.-M., & Shih, H.-F. (2013) argued that combining the basic heuristic search methods in high level frameworks can explore the searching space more effectively than basic heuristic, which is called metaheuristic. Duarte, A., Sánchez, Á., Fernández, F., & Cabido, R. (2005) defined the metaheuristic as: "an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions". In Haddadi, S. & Cheraitia, M. (2018): "Meta-heuristics are typically high-level strategies which guide an underlying, more problem specific heuristic, to increase their performance".

Metaheuristic starts with one solution or more initial solutions and employ search mechanisms/strategies to search for a better-quality solution. There are different ways to categorize the meta-heuristics algorithms. Hansen, P., & Mladenović, N. (2001) categorized the metaheuristic algorithms as follows: Trajectory methods vs discontinuous methods; one-neighborhood vs various neighborhood structures; population-based vs single point search; nature inspired vs non-nature inspiration; dynamic vs static objective function and memory usage vs memoryless methods.

All earlier works related to timetabling had focused on factors not related to social distance as a critical issue. However, new circumstances developed raised due to the perspective of a disease—COVID-19. COVID-19 is a pandemic that led to the worldwide practice of social distancing to minimize the spread of this virus. Universities and schools are two of the most crowded places, and mass gatherings at those places may cause the virus if social distancing is not maintained. This work supplies a new approach to help universities schedule their exams in the manner of keeping students safe as much as possible. For this purpose, a new real-world dataset were collected from the faculty of information technology at Al-Ahliyya Amman University (AAU). The Faculty of Information Technology has four departments and 16 curricula of students as shown in Table.1.

This chapter is organized as follows: Section 2 presents the overview background and related works . While section 3 presents the research methodology. Section 4 reports the experimental results and the comparable results with other works reported in the literature. Finally, section 5 presents the summary and conclusion of this paper.

2. BACKGROUND AND RELATED WORKS

Generally, the examination timetable quality is measured by the satisfaction degree of the soft constraint's penalties, meanwhile for the hard constraints, it is measured by the complete fulfilment degree. That is measured by penalizing each violation of soft constraints (penalties cost). Minimum number of penalty values indicate good quality solution. This evaluation function is also called the objective function or penalties cost function. "The problem of university examination timetable is consists of a set exams E = {e1, e2,,en} are to schedule in a set of p periods, T= { t1, t2, ..., tp }and a set of m rooms R= {r1,r2, ... ,rm}. Each courses group that share common students called curriculum Crk, CR= {Cr1, Cr2, ... , Crs}"(Lü and Hao 2010).

By the mid of 1990s, it was already suggested that incorporating some amount of local search technique within evolutionary algorithms which it might to enhance the quality of the final solutions , In general there are a growing number of solutions to different types of timetabling problems having different types of Constraints Werra(1985); (Colorni et al, 1992); (Burke et al., 1994); Burke et al, 1996); (Burke et al, 1997); Abramson et al., 1999); (Alkan & ozcan 2003); (Causmaecker et al., 2002); (Burke et al., 2003).

Jat and Yang (2011) presented a hybrid GA and TS approach to solve the post enrolment course timetabling problem. Two phases technique were presented in this work. The first phase is a guided search genetic algorithm with two local search methods and six neighborhood structures. The guided search strategy stores some useful information that was extracted from previous good generations to guide the offspring generations into the population. Whilst the local search methods applied to improve the individual's quality. In the second phase, the best solution from the first phase is further improved by the TS algorithm. The experimental results showed that the proposed work can produce good results.

Azizi et al. (2009) proposed a hybrid SA with evolution-based diversification approach called SAMED to solve job shop scheduling problem. SAMED includes SA, three types of memories and GA based crossover component. The first two types of memories are short term memories (tabu list and seed memory list), while the third type is long term memory. SAMED used short term memory (tabu list) to temporarily save the accepted solutions to avoid recycling. Whilst the second short term memory (seed memory) is to keep track of the best solutions with lowest objectives functions for further improvement.

Gao et al. (2006) presented a hybrid meta-heuristic algorithm by combined the characteristic of SA, GA and Chaos strategy to solve TSP problem. The experimental results showed that the hybrid meta-heuristic algorithm (CASAGA) is quite flexible with satisfactory results. Hint: Chaos strategy is a well-known phenomenon that exist in nonlinear system, it has better capacity of climbing hill by using intrinsic stochastic property to find the optimum solution.

Zolfaghari and Liang (2002) presented a comparative study of SA, GA, and TS to test the performance for each algorithm. They implemented their algorithms to solve the varying types of binary machine grouping problems, which involved machine/part types, sizes, machine capacities and processing times. The comparison results were made in terms of solution quality, pre-search effort and the behavior of the search convergence; indicated that SA outperformed both GA and TS, mostly for each large problem.

The early techniques used in solving timetabling problems were based on a simulation of the human approach in resolving the problem. These techniques were successive augmentations that were called direct heuristics. This based on the idea of creating a partial timetable by scheduling the most constrained lecture first and then extended this partial solution lecture by lecture until all lectures were scheduled (Schaerf, 1999).

Rahoual and Saad (2006) explained an idea of hybridizing of GA and TS systems from their desire to reap the benefits of both methods: the simplicity of the use of GAs on the one hand, and such ability to jump out from local optima through a more diversified and balanced search as provided by TS, on the other hand. One of these is the so-called Violation Driven Mutation, which provides an intelligent operator capable of detecting and solving sources of conflicts. The idle time that might be generated in the process is eliminated via TS.

Zhou and Shi-Xin (2006) presented a Self-Adaptive Genetic Algorithm (SAGA) approach to multiprocessor scheduling problems. SAGA adjusts the parameters according to the evolution process to obtain better solutions. The obtained results showed that the SAGA can perform better than the others GAs.

Tarawneh et al. (2022) presented an automated scheduling approach that can help universities and schools comply with the social distancing regulations by providing assistance in avoiding huge gatherings of people, the paper proposed a novel course timetable-scheduling scheme based on four main constraints. First, two meters must be maintained between each student inside the classroom. Second, no classroom should contain more than 20% of their regular capacity. Third, there would be no back-to-back classes. Lastly, no lectures should be held simultaneously in adjacent classrooms. The proposed approach was implemented using a variable neighborhood search (VNS) approach with an adaptive neighborhood structure (AD-NS) to resolve the problem of scheduling course timetables.

Based on literature, we found that each algorithm has strengths and weaknesses in terms of the search ability and the solution quality. The TS is performed effectively when the neighborhood structure is small and the landscape is fat, by escaping form the local optimum. The variable neighborhood descent improves the solution quality very fast. Local search can improve the solution very fast. SA performs effectively when the computational time is quite enough (large), the genetic algorithm may improve the search process to reach different solutions that qualify for better results, knowing that it may lead to ways far from the optimal solution, because it accepts worse solutions continuously. Therefore, the use of Tabu list memory has become an ideal solution to improve the genetic algorithm search to avoid falling into duplicate solutions and searching for better solutions during the search process. Furthermore, using long term memory in the TS will guide the diversification part in order to explore the search space around those solutions.

Based on the recommendations from the literature, there are two research questions that have emerged needs to be answered, as follows: -

Q1) How to enhance GA search process that can help effectively search for good quality solution and minimize the random swap and move?

Q2) How to build an automated examination timetable that helps to solve the problem of social distancing during pandemic, such as COVID-19?

3. RESEARCH METHOD

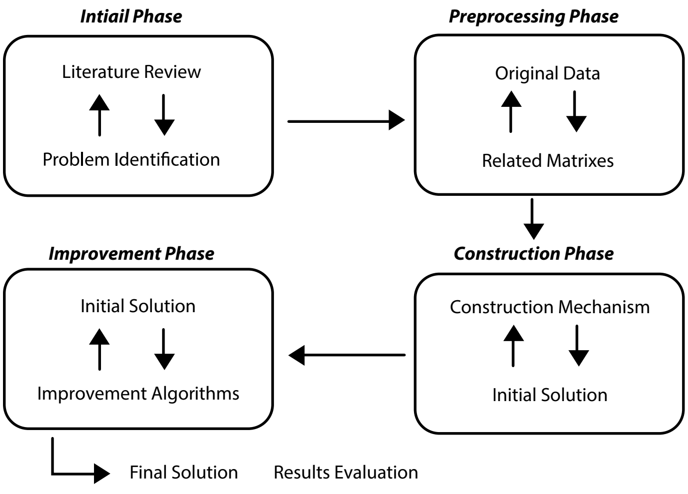


Figure 1: The research method for the University Examination timetabling Problems.

This research focusses on solving university examination timetable problem (UETT), by hybridizing GA and Tabu list memory, in order to improve the solution quality and reduces the computational time. The research method is divided into three phases: pre-processing phase, construction phase and improvement phases.

Initial Phase:

This phase focuses on the literature review to understand the problem of university examination timetabling. Moreover, to understand how to develop an effective way to avoid students overcrowding inside the university campus.

Pre-processing Phase:

This phase concentrates on reading and transferring the data into matrixes, by converting the original data fields to related symbols and numbers for AAU-faculty of information technology dataset.

Problem formulation:

The Faculty of Information Technology has four departments with thirty-two groups of students. Table 1 shows Faculty of Information Technology data information.

Table 1: Faculty of Information Technology Information.

	Value
No. Days	5
No. Courses	148
Total lectures	320
No. Lecturers	60
Timeslots(length)	2 (hours)
No. periods per day	5
No. Rooms	15
No. Groups	16

The construction of the university examination schedule differs from the schedule of lectures in many respects. An exam timetable consists of exams in timeslots such as courses. It depends on the number of students in each exam, and one or more classrooms can be used for each exam. Also, examiners must be assigned to each exam classroom. An exam timetable has two types of constraints. Hard constraints and soft constraints, meanwhile, social distancing to reduce infection with the Corona virus, has caused the emergence of new restrictions in building university schedules such as examination timetable, where the presence of students on the university campus needs a set of rules, to ensure that they are not infected with the virus.

Tables 2 and 3 shows the common constraints from the literature and the constraints suggested in this research (hard and soft constraints):

ISSN (Online): 2789-5289 AJHAS

Table 2: Examination Timetable Hard Constraints.

	Hard Constraints	Number of violations	Proposed by
1	There are no two exams in the same time slot and room.	20	De Smet, G. (2008).
2	A lecturer cannot have more than one exam in the same timeslot unless there is a combined exam.	20	De Smet, G. (2008).
3	The number of students taking an exam should not exceed the number of seats in the exam room.	20	De Smet, G. (2008).

Table 3: Examination Timetable Soft Constraints.

	Soft Constraints	Number of violations	Proposed by
1	A balanced distribution of exams of classes along exam days.	3	De Smet, G. (2008).
2	Avoiding assigning hard exams on the same day for any class.	3	De Smet, G. (2008).
4	Having an exam at the end of the day (last timeslot).	3	De Smet, G. (2008).
6	A group of students have more than one exam per day.	3	De Smet, G. (2008).
7	A group of students have more than three consecutive exams per day.	3	De Smet, G. (2008).
8	For each timeslot, avoid assigning exams in adjacent rooms. 10 violations are counted for every two exams scheduled in two adjacent rooms at the same timeslot.	10	Proposed in this study
9	For each time slot, the occupation of the building should not exceed 50% of the building's capacity.	10	Proposed in this study
10	Avoid consecutive exams in any room.	10	Proposed in this study

Concerning the violations that were calculated and determined in Table 3, the objective function here is to minimize the soft constraints (S1-S10) violations, i.e., Minimize $S_1+S_2+S_3+S_4+S_5+S_6+S_7+S_8+S_9+S_{10}$.

Construction Phase:

This phase focuses on building or initialising the initial solution by satisfying the hard constraints violations, the feasible initial solution is constructed by satisfying all hard constraints, using sequential greedy heuristic as in Lü and Hao (2010). Generating the possible solution using sequential greedy heuristic starts by inserting one lecture to the timetable at each time. By selecting one does not assign lecture of a course, then determines a period room pair to the selected lecture. In fact, the priority is to select the lectures that belong to a course with small number of available periods, and enormous number of unassigned lectures. Furthermore, "this heuristic and greedy colouring heuristic is similar" (Brélaz, 1979).

However, once the exam is selected, the procedure selects an available period to this exam. Therefore, when attempt to insert the feasible move; the procedure calculates the total number of the unfinished exams that become unavailable for the current period. Then, the exam with small value of this number is highly favoured to be inserted. However, there is no guaranty to find a solution using this greedy heuristic; Therefore, steepest decent scans will be applied until we obtain a feasible solution.

Improvement phase:

This phase concentrates on improving the initial solution produced by the construction phase, by minimizing the soft constraints violations. To achieve this target, this study applies the GA with tabu list memories.

The short-term memory is used to avoid cycling by forbidding revisiting recently solutions (which might lead to get trapped in poor local optima). Long term memory performs as a learning process that functions as the intensification and diversification strategies. That is the long-term memory is used to collect the information during the search process that allows the identification of common properties of elite solutions that are already visited and attempts to visit solutions with varying properties from those already visited.

GA was proposed by (Holland, 1975). "A genetic algorithm is a population-based method that is based on the principles of natural evolution" (Goldberg, 1989; Man et al., 1999; Michalewicz, 1999). In GA, A solution is a chromosome in the population. Each chromosome has associated objective function value called fitness value. A good chromosome is the one that has high/low fitness value depending upon the problem (maximization/minimization). A set of chromosomes is called the population. This population at a given stage of GA is referred to as a generation. Figure 1 shows a pseudo code for GA (Mitchell, 1996) with proposed tabu list memory.

Therefore, the use of Tabu list memory has become an ideal solution to improve the genetic algorithm search to avoid falling into duplicate solutions and searching for better solutions during the search process. Furthermore, using long term memory in the TS will guide the diversification part to explore the search space around those solutions.

The Hybrid Algorithm Starts using sequential greedy heuristic to generate n chromosomes, then evaluate the fitness f(x) of each chromosome x in population, then create a new population by repeating the following steps until the new population is complete; I. select two parent chromosomes from the population according to their fitness (the better fitness, the bigger chance to be selected); ii. Check the selected parents using Tabu list memory to avoid cycling, if it was visited before then select another parent (using ranking procedure); iii. with a crossover probability cross over the parents to form a new offspring. If no crossover was performed, offspring is the exact copy of parents; v. with a mutation probability mutate new offspring at each locus (position in chromosome); iv. place the new offspring in the new population. After the new population is created, the new generated population will be used for a further run of algorithm, see figure 2.

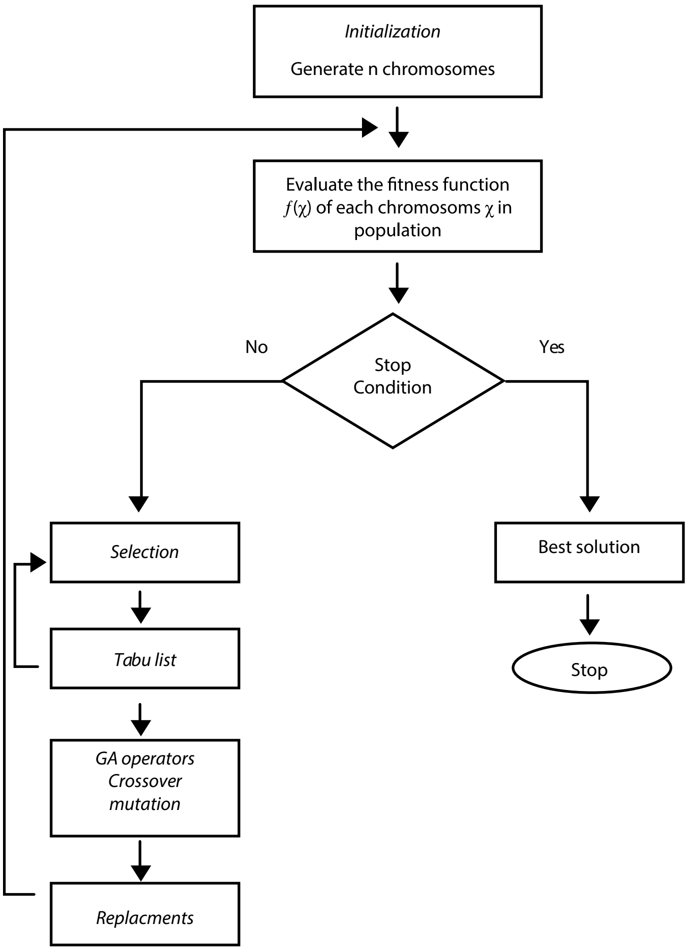


Figure 2: GA with Tabu List Algorithm.

The idea of hybridizing GA and TL memory is to reap the benefits of both methods: the simplicity of using GAs on the one hand, and such ability to jump out from local optima through a more diversified and balanced search as provided by TS, on the other hand. One of these is the so-called Violation Driven Mutation, which provides an intelligent operator capable of detecting and solving sources of conflicts. As shown in figure 2, the tabu list memory is used to reduce the problem of re-choosing the same parents each time of the selection stages, in order to avoid repetition.

4. EXPERIMENTS AND RESULTS

In this work, we performed several experiments for each part of our research questions as follow:

To answer the first research question "How to enhance GA search process that can help effectively search for good quality solution and minimize the random swap and move?"; we compare the standard GA with three types of tabu list memory; long term memory, short term memory and adaptive memory. Table 4 shows the parameter setting of the proposed approach.

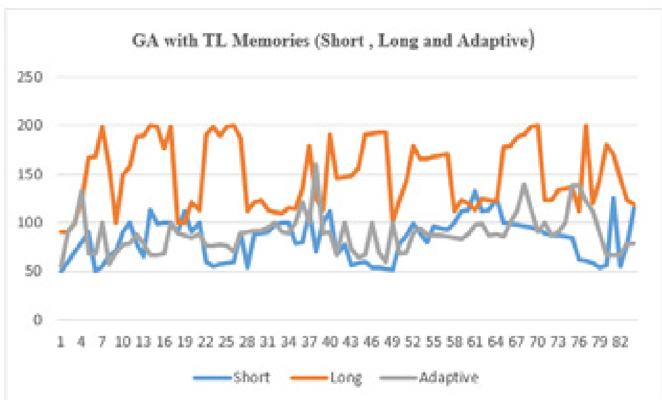
Table 4: The GA parameters setting with long- and short-term TL memories.

	Parameters
Population number	100
Crossover	Single point
Mutation percentage	2%
Long term TL	50
Sort Term TL	7

The main purpose of this experiment is to compare the performance of GA with different types of TL memories. For this comparison, short term memory size is seven as literature , while long term memory suggested to be fifty , and for adaptive TL is determined during the algorithm's search process, based on calculating the number of iteration that the solution is not improved, where if the solution does not improve within ten iteration, the memory size is increased by one until fifty move , where the preliminary experiments showed that when the TL memory more the fifty move, the solution could stick in local optimum very fast and could not escape from it.

Figure 3 and Table 5 shows the comparison results of GA with different Tabu list size using FIT- AAU real world datasets, each approach test over hundred runs, where the results show the improvement of the solution between the short, long, and adaptive tabu list memories with standard GA. However, using the long-term memory does not improve the results as well as other memories. Meanwhile, the short-term memory achieves best solution (50) and better average (83.11) compared with the adaptive TL memory. But as it is clear from the results, the use of adaptive TL achieves a better standard deviation than short memory, this gives an indicator that the results using adaptive TL are less scattered (see figure 1).

Figure 3: Comparison between Short, Long and Adaptive TL with standard GA.



ISSN (Online): 2789-5289 AJHAS

Table 5: Compression results between the proposed approach and standard VNS Using AAU dataset.

Instances	Short term TL				Long term TL				Adaptive TL			
	Best	Worst	Avg	STD	Best	Worst	Avg	STD	Best	Worst	Avg	STD
FIT-1-2019	50	132	83.11	21.03	90	200	149	34.69	55	160	88.5	19.57

Note: Best results in italic bold.

From the above experiments, using a genetic algorithm with tabu list memory achieves better results and can improve the solution by avoiding repetition in accepting the worst solution. Also, the use of short-term or Adaptive memory achieves better results than long-term memory. As for the examination timetable construction, the proposed approach can build it in the shortest time and the least effort compared to manual timetable.

Table 6 shows the comparison between the results of using standard GA and the proposed approach in this study, where the experimental results prove that using tabu list with GA achieves better results, improves the solutions, and minimizes the consuming time in building automated exam timetable.

Table 6: Compression results between the proposed approach and standard GA Using AAU dataset.

Instances	GA with Short term TL				Standard GA			
	Best	Worst	Avg	STD	Best	Worst	Avg	STD
FIT-1-2019	50	132	83.11	21.03	70	200	123.75	39.44

Note: Best results in italic bold.

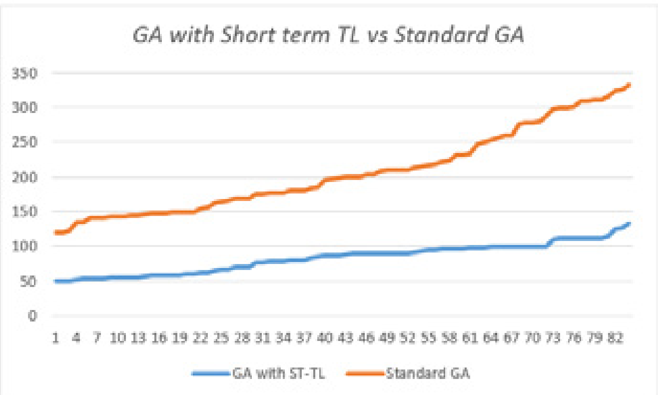


Figure 4: Comparison Results between the Proposed Approach vs the Standard Approach.

Figure 4 showed A comparison plot between the proposed approach with the standard approach, through eighty two runs, it shows the large distribution using standard genetic algorithm compared with the good distribution of the proposed approach, note that the results were arranged from smallest to largest in both tables.

Table 7 shows the comparison results between the proposed approach and the best known result in ITC2007. The stopping condition is set at 650 seconds (decided by benchmark programs provided at the ITC 2007 web site). The best results in the literature are shown in bold. The results show that GCCHH is competitive with the ITC 2007 competition winners' approaches.

Table 7: Compressing Results between the Proposed Approach and the best results of ITC 2007 winners.

	Our Approach	Pillay, 2007	Atsuta et al., 2007	Muller, 2009	Gogos et al., 2008	De Smet, 2008
Dataset1	5400	12035	8006	**4370**	5905	6670
Dataset 2	500	3074	3470	**400**	1008	623
Dataset 3	11000	15917	18622	**10049**	13862	-
Dataset 4	18200	23582	22559	**18141**	18674	-
Dataset 5	3000	6860	4714	**2988**	4139	3847
Dataset 6	27100	32250	29155	**26950**	27640	27815
Dataset 7	4300	17666	10473	**4213**	6683	5420
Dataset 8	8000	16184	14317	**7861**	10521	-

Note: Best results in italic bold.

Furthermore, to compare our approach of solving the problem of social distance during the Covid-19 pandemic, the achieved result of the proposed approach is compared with standard GA and common soft constraints presented by literature, by given the calculation of social distancing between exams rooms and students, and overcrowded buildings during exams, as the following proposed measurement (for both results, the hard constraints are satisfied) see equations 1, 2:

S= {χ | χ is a student's numbers attend exams at level i in slot t}

R = {r | **r** is a room capacity at level i}

$$SD_m = \sum_i^t x_{i,t} \quad i \qquad \text{eq. (1)}$$

$$\beta = \frac{SD_m * 100}{R} \qquad \text{eq. (2)}$$

ß is the percentage of student's number at each day SD_m using our approach in comparison with the actual number of students or actual room capacity **R**.

After 100 runs, the proposed approach shows the result of ß equal to 40%, whereas the standard approach achieved 70% in the case of social distancing for the exam schedule during the university day.

5. CONCLUSION

Social distancing is considered the main solution to prevent the spread of the Corona virus. University exams increase the chance of getting the disease and increase transmission. This paper offers a mechanism for distributing exams in such a way that student gatherings are minimized. The algorithm offers a novel soft constraint that reduces the presence of students in each time slot to the percentages indicated by health care authorities by upgrading the genetic algorithm with and tabu list memory.

The proposed approach improves the percentage of student gatherings in a way that reduces the number of students every day on campus. Thus, the proposed approach achieved results that reduce student crowding compared to the standard approach, with less than 50% of the actual university capacity over time slots. The proposed work can be carried out in all colleges and universities in Jordan or anywhere else where the same conditions apply to Al-Ahliyya Amman University.

Conflict of interests:

The author declares that there is no conflict of interests.

Funding statement

Research reported in this publication was supported by Al-Ahliyya Amman University.

References

Abayomi-Alli, O., Abayomi-Alli, A., Misra, S., Damasevicius, R., & Maskeliunas, R. (2019). Automatic examination timetable scheduling using particle swarm optimization and local search algorithm. In Data, engineering, and applications (pp. 119-130): Springer.

Abdullah, S., Burke, E. K., & Mccollum, B. (2005). An investigation of variable neighborhood search for university course timetabling. Paper presented at the 2nd multidisciplinary international conference on scheduling: theory and applications (MISTA).

Ahmad, Z., Tarawneh, H., & Ayob, M. (2013). A hybrid simulated annealing with solutions memory for curriculum-based course timetabling problem. Journal of Applied Sciences, 13(2), 262-269.

Al-Tarawneh, H., Al-Kaabneh, K., Alhroob, A., Migdady, H., & Alhadid, I. A Hybrid Heuristic Algorithm for Solving COVID-19's Social Distancing at Universities Campus. Computer Systems Science and Engineering, 41(3), 933-944.

Aladag, C. H., Hocaoglu, G., & Basaran, M. A. (2009). The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem. Expert systems with applications, 36(10), 12349-12356.

Alhroob, A., Tarawneh, H., & Ayob, M. (2014). A hybrid simulated annealing with tabu list and allowable solutions memory to solve university course timetabling problem. European Journal of Scientific Research, 126(1), 84-94.

Atsuta M, Nonobe K, Ibaraki T (2007) ITC2007 Track 1: an approach using general CSP solver. www.cs.qub.ac.uk/itc2007.

Avanthay, C., Hertz, A., & Zufferey, N. (2003). A variable neighborhood search for graph coloring. European Journal of Operational Research, 151(2), 379-388.

Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R. (2010). Hybrid variable neighbourhood approaches to university exam timetabling. European Journal of Operational Research, 206(1), 46-53.

Chen, M., Tang, X., Song, T., Wu, C., Liu, S., & Peng, X. (2020). A Tabu search algorithm with controlled randomization for constructing feasible university course timetables. Computers & Operations Research, 123, 105007.

Chen, R.-M., & Shih, H.-F. (2013). Solving university course timetabling problems using constriction particle swarm optimization with local search. Algorithms, 6(2), 227-244.

De Smet, G. (2008). Examination track, practice and theory of automated timetabling. Examination track, practice and theory of automated timetabling (PATAT'08), Montreal, 19-22.

Duarte, A., Sánchez, Á., Fernández, F., & Cabido, R. (2005). A low-level hybridization between memetic algorithm and VNS for the max-cut problem. Paper presented at the Proceedings of the 7th annual conference on Genetic and evolutionary computation, Boston, MA , USA.

Gogos, C., Alefragis, P., & Housos, E. (2008). A multi-staged algorithmic process for the solution of the examination timetabling problem. Practice and theory of automated timetabling (PATAT 2008), Montreal, 19-22.

Haddadi, S., & Cheraitia, M. (2018). Iterated local and very-large-scale neighborhood search for a novel uncapacitated exam scheduling model. International Journal of Management Science and Engineering Management, 13(4), 286-294.

Hansen, P., & Mladenović, N. (2001). Variable neighborhood search: Principles and applications. European Journal of Operational Research, 130(3), 449-467.

Kralev, V., Kraleva, R., & Kumar, S. (2019). A modified event grouping based algorithm for the university course timetabling problem. International Journal

on Advanced Science, Engineering and Information Technology, 9(1), 229-235.
Lewis, R., & Paechter, B. (2004). New crossover operators for timetabling with evolutionary algorithms. Paper presented at the 5th International Conference on Recent Advances in Soft Computing, Nottingham, UK.
Lü, Z., & Hao, J.-K. (2010). Adaptive tabu search for course timetabling. European Journal of Operational Research, 200(1), 235-244.
McCollum, B., & Ireland, N. (2006). University timetabling: Bridging the gap between research and practice. E Burke, HR, ed.: PATAT, 15-35.
Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. Computers & Operations Research, 24(11), 1097-1100.
Müller, T. (2009). ITC2007 solver description: a hybrid approach. Annals of Operations Research, 172(1), 429-446.
Pérez, J. A. M., Moreno-Vega, J. M., & Martın, I. R. (2003). Variable neighborhood tabu search and its application to the median cycle problem. European Journal of Operational Research, 151(2), 365-378.
Pillay, N., & Banzhaf, W. (2008). A developmental approach to the uncapacitated examination timetabling problem. Paper presented at the International Conference on Parallel Problem Solving from Nature, Dortmund, Germany.
Rahoual, M., & Saad, R. (2006, August). Solving Timetabling problems by hybridizing genetic algorithms and taboo search. In 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2006) (pp. 467-472), UK.
Schaerf, A. (1999). A survey of automated timetabling. Artificial intelligence review, 13(2), 87-127.
Viana, A., Sousa, J. P., & Matos, M. A. (2005). Constraint Oriented Neighbourhoods—A New Search Strategy in Metaheuristics. In Metaheuristics: progress as real problem solvers (pp. 389-414): Springer.
Wang, Y. Z. (2002). An application of genetic algorithm methods for teacher assignment problems. Expert systems with applications, 22(4), 295-302.
Wen-jing, W. (2018). Improved Adaptive Genetic Algorithm for Course Scheduling in Colleges and Universities. International Journal of Emerging Technologies in Learning, 13(6),29-42.
Wu, G., Wang, H., Pedrycz, W., Li, H., & Wang, L. (2017). Satellite observation scheduling with a novel adaptive simulated annealing algorithm and a dynamic task clustering strategy. Computers & Industrial Engineering, 113, 576-588.
Zhang, D., Liu, Y., M'Hallah, R., & Leung, S. C. (2010). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. European Journal of Operational Research, 203(3), 550-558.

Contribution of authors:

Ht: Conceptualization and designing the analysis, system programming, performed the analysis, applying the experimentation, writing the original draft, reading, and approving the final draft

Kk: Conceptualization and designing the analysis, data collection, performing the analysis, applying the experimentation, writing the original draft, reading, and approving the final draft

Citing the article:

Tarawneh H. and Kaabneh K. (2022). Hybrid Algorithm to Solve COVID-19 Physical Distancing for Examination Scheduling: Al-Ahliyya Amman University Case Study. . Al-Balqa Journal for Health and Applied Sciences. 1 (1), 32- 40.