GENETIC ALGORITHM FOR EXAM TIMETABLING PROBLEM - A SPECIFIC CASE FOR JAPANESE UNIVERSITY FINAL PRESENTATION TIMETABLING

Jiawei LI and Tad Gonsalves

Department of Information & Communication Sciences
Faculty of Science and Technology, Sophia University, Tokyo, Japan

ABSTRACT

This paper presents a Genetic Algorithm approach to solve a specific examination timetabling problem which is common in Japanese Universities. The model is programmed in Excel VBA programming language, which can be run on the Microsoft Office Excel worksheets directly. The model uses direct chromosome representation. To satisfy hard and soft constraints, constraint-based initialization operation, constraint-based crossover operation and penalty points system are implemented. To further improve the result quality of the algorithm, this paper designed an improvement called initial population pre-training. The proposed model was tested by the real data from Sophia University, Tokyo, Japan. The model shows acceptable results, and the comparison of results proves that the initial population pre-training approach can improve the result quality.

KEYWORDS

Examination timetabling problem, Excel VBA, Direct chromosome representation, Genetic Algorithm Improvement.

1. INTRODUCTION

Examination Timetabling Problem (ETP) is a well-known NP-hard problem which tries to find the best examinations schedule for schools, colleges, and universities. As a discrete optimization algorithm, Genetic algorithm (GA) is naturally suitable to solve ETP. Moreover, compared to other search algorithms, GA is more robust in searching complex search spaces [1]. This paper focuses on a special case of Examination Timetabling Problem (ETP), which is common in Japanese universities or colleges. In Japanese universities or colleges, every final-year student must give a presentation to evaluate their academic research. Each student has three examiners, so the examination timetabling problem is related to allocating proper time and rooms to students and examiners. It is preferable to put all students from the same laboratory together in the same room for presentations in successive time slots which make a SESSION. Moreover, one session should be held on one day. These two constraints make the traditional common GA no longer effective because the traditional crossover operation can break the intact session with a high probability resulting in a large number of infeasible solutions. Therefore, one has to design a novel variant of GA to satisfy this special case.

This paper designed an automatic ETP algorithm by using a variant of GA where the constraint-based initialization and crossover operations are applied to satisfy the two constraints about the

Natarajan Meghanathan et al. (Eds): CSE, BMLI, EDU, AI & FL, CCNET - 2020
pp. 01-12, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101701

sessions. The penalty point system is also implemented to optimize other soft constraints. However, during the research, it is found that the constraint-based initialization and crossover operations usually cause premature convergence and then output an unacceptable result. To improve the result quality of the proposed model, an improvement which we call initial population pre-training is adopted.

The application is written in VBA language. VBA is embedded in the Microsoft office suite. Although VBA code execution is slower than C or Python, it has the advantage of using the familiar MS Excel worksheet interface with an unobtrusive macro embedded in it. Further, since all the timetabling data is in Excel, direct encoding in Excel VBA becomes very handy for the university staff who is responsible for creating and managing the timetable.

2. RELATED WORK

As Qu et al. [2] and Carter, & Laporte [3] summarized, the early research– on ETPs can be roughly divided into four types of approaches, which are Cluster, Sequential, Constraint Based and Generalized Search. Generalized Search has another well-known name, namely, meta-heuristic algorithms. In the early research on ETP, simulated annealing algorithm [4][5] and Tabu search [6][7] were the two main Generalized Search approaches.

In recent years, with the development of ETPs research, meta-heuristic algorithms have become some of the main approaches to solve ETPs, and many different kinds of meta-heuristic algorithms are implemented. In 2005, Azimi [8] applied Simulated Annealing (SA), Tabu Search (TS), Genetic Algorithm (GA) and Ant Colony System (ACS) to solve the ETP. They then proposed three novel hybrid combinations of those four algorithms, which are Sequential TS–ACS, Hybrid ACS/TS, and Sequential ACS–TS algorithms. By testing more than 10 different scenarios of the ETP, they demonstrated that all the three hybrid algorithms performed significantly better than the performance of four non-hybrid algorithms. In 2006, Eley [9] applied two ant colony approaches, Max-Min and ANTCOL approach for solving the ETP and compared these two approaches with other timetabling heuristics. In 2015, Mandal & Kahar [10] applied the great deluge algorithm to partial exam assignment. In their approach, the total exams are ordered in advance based on graph heuristic and then partial exams are improved by the great deluge algorithm one by one until all exams have been scheduled. Compared to the state-of-the-art approaches, this novel method shows a competitive performance. In 2018, Leite et al. [11] solved the ETP with the cellular memetic algorithm. The cellular memetic algorithm organizes the population in a cellular structure to provide a smooth actualization and improve the diversity of the population. The algorithm gives improvements on partial functions of incapacitated Toronto and capacitated ITC 2007 benchmark sets.

As one of the most popular optimization approaches, GA has also been widely used to solve ETPs. In 2017, Rozaimee, et al. [12] tried to use GA to construct the final exam timetable automatically for the UniSZA computer system, to save time for the university staff. In 2017, Shatnawi et al. [13] proposed a two-stage approach optimization algorithm by running Greedy Algorithm and GA in parallel, to help the Arab East College of High Education in Saudi Arabia solve the problem of scheduling exams. Their result shows that the required number of conflicts, exam days and available venues had been reduced successfully. In 2019, Dener [14] introduced a two-stage GA, where the first stage carries out the assignment of courses to sessions and the second stage assigns the students who participated in the test session to the examination room. The system was designed to allocate students and supervisors in a more efficient way to reduce the number of rooms and time consumption. In 2020, Gozali, et al. [15] attempted to solve the university course timetabling by using localized island GA with dual dynamic migration policy

(DM-LIMGA). The results show that the proposed algorithm can produce a feasible timetable in student sectioning problem with a better result than previous works.

3. OBJECTIVE PROBLEM

The proposed model is designed to solve the thesis presentation final exams of the Informatics Graduate School of Sophia University, Tokyo, Japan, which is different from the general Examination Timetabling Problems. At the end of the academic year, all final-year students in Science and Engineering School at Sophia University must give a presentation to explain and discuss their research. Each student's presentation is evaluated by three examiners, one of which is the student's supervisor, who is called the prime examiner, and the two other examiners are called deputy examiners, who are familiar with the student's research topic. The student's examiners are decided in advance which cannot be changed. The capacity of the room is limited. Each room can only have one presentation at a time. A room can hold a maximum of 10 presentations per day. The purpose of the proposed problem is to allocate all students to the proper timeslots and rooms, by considering the examiners' available time and some other constraints which will be explained below. The specific information about the proposed problem is shown in Table 1.1.

Table 1. Algorithm input, basic information of the proposed problem.

Presentation event dates	3 days
Presentation event start time of a day	9:00
Presentation event end time of a day	17:30
Length of presentation	40 minutes
Lunch period	12:00—13:00
Length of break	5 minutes
Room numbers of each day	2/3/3

For each timeslot, an examiner has three options, "O", "X" and "Δ". "O" means the examiner is available for that time period. An "X" means the examiner is not available for that time period. A "Δ" means the examiner is available during that time period, but that should be avoided if possible. "X" has higher priority than "Δ".

The proposed timetabling problem also has 3 hard constraints and 4 soft constraints as follows:

Hard constants

1. All students with the same supervisor should hold the presentation one after another, which is called a session.

2. A session should be held on a single day.

3. No student or examiner can be removed from the session.

Soft constraints

1. Examiners should be allocated to appropriate time depending on the examiner's own schedule.

2. Examiners should be assigned to contiguous time slots, if possible.

3. The timetable should be compact.

4. An examiner should not occur in two places at the same time. (Note that, although from the feasibility perspective this is a hard constraint, for fast convergence of our algorithm, it is classified as a soft constraint).

In the common ETP problem, each examination is independent, which means a single examination could be moved independently to wherever feasible. The common GA operations can be directly applied to solve the problem because the arbitrary chromosome cut and join does not break the feasibility of the solutions. However, in Japan, people prefer to make the students with the same supervisor together to conduct the presentation one after another, which is called a session. Moreover, a whole session should be held on the same day. These two constraints corresponds to hard constraint 1and 2. The two hard constraints make the common GA operation no longer work because the arbitrary chromosome cut and join does not break the session with a high probability, and it is hard to recombine the scattered session again by the arbitrary GA operation. To solve this problem, a constraint-based initialization and crossover operations are proposed, where these two operations will never break the three hard constraints and therefore, the three hard constraints can be satisfied automatically. Moreover, the four soft constraints are optimized by using a penalty points system. This is the reason why we classify the constraint "An examiner should not occur in two places at the same time" as a soft constraint, but not a hard constraint, although commonly people will regard this constraint as a hard constraint intuitively. In the proposed model, all hard constraints are satisfied automatically by the specific GA operations, and all soft constraints are satisfied by the penalty point systems.

4. PROPOSED VARIANT GA MODEL

4.1. Chromosome representation

The proposed model uses the direct chromosome as the encoding, where each chromosome corresponds to a specific arrangement to the generated blank position unit. The total chromosome length is equal to the total number of available timeslots. For example, there is a two-day presentation period, each day has 2 available rooms, each one can hold maximum 10 presentations each day, then the total chromosome length is equal to 2*2*10=40 timeslots. The students will be numbered from 1 to some maximum number. Each chromosome is a list of numbers which contain all numbers from 1 to the maximum number of students and numbers of 0, meaning a certain position unit does not have an arrangement for a student's presentation.

For example, assume we have a two-day presentation event, each day has two available rooms, Room A and Room B, then the chromosome is as follows:

[5,6,7,0,0,0,0,0,0,0, 0,0,0,0,0,0,11,10,9,8, 15,16,0,1,2,3,4,12,13,14, 0,0,0,0,0,0,0,0,0,0]

In this chromosome string, the student 5 is allocated into timeslot 1, which corresponds to the first presentation in the Room A of the first day. Similarly, the student 11 is allocated into timeslot 17, which corresponds to the 7th presentation in the Room B of the first day, the student 16 is allocated into timeslot 22, which corresponds to the 2nd presentation in the Room A of the second day, while there are no students allocated into the Room B of the second days which corresponds to the timeslots from 31 to 40.

4.2. Constraint-based initialization operation

To satisfy the hard constraints 1 and 2, a constraint-based initialization operation is introduced. The operation is divided into two steps:

First, the students are grouped with the same supervisor to form a session. Then, the order of the sessions and the order of students within one session are re-ordered.

Secondly, the first session of the new order is allocated into the first available place in the first room. The second session of the new order is allocated into the first available place in the second room and so on. If all rooms have been allocated with a session, the operation goes back to the top and allocates the next session to an available timeslot of a random room. If the selected random room does not allow all students from the same laboratory to hold the presentation in the same day and same room, a new random target room is looked for. If there is no way to allocate all laboratories to the proper position, an error is reported to ask the staff to re-arrange the rooms or add some new rooms. In this way, all the initial possible solutions can satisfy the hard constraints automatically, where the students from the same session can be allocated together and a session can hold all presentations on the same day. Meanwhile the constrained initialization can maintain a certain degree of randomness to maintain the diversity of the initial population. Below is an example of an initial chromosome. In this example, each room corresponds to 10 time slots.

[6,7,5,0,0,0,0,0,0,0, 10,8,9,11,0,0,0,0,0,0, 15,16,0,0,0,0,0,0,0,0, 13,12,14,0,1,2,3,4,0,0].

4.3. Fitness evaluation and penalty system

The fitness evaluation of the proposed model uses the penalty points system to optimize the soft constraints in the proposed problem. In penalty points system, if the solution breaks any soft constraints, the penalty will be given. The proposed timetabling problem then becomes an optimization problem to find the solution with minimum penalty. Four soft constraints in the proposed problem are decomposed into 6 types of penalties. The soft constraints 1, “Examiner should be allocated to an appropriate time depending on examiner’s own schedule” corresponds to penalty 1 and penalty 2. The soft constraint 2 corresponds to penalty 3 and 4. The soft constraints 3 and 4 correspond to penalty 5 and 6, respectively. Table 1.4 shows the specific distribution of the penalty points.

Table 2. Penalty category and penalty points.

Penalty	**Penalty points**
1. Examiner is allocated into time period with X.	242
2. Examiner is allocated into time period with △.	60
3. Examiners are placed in contiguous slots in the same session.	10
4. Examiners are placed in contiguous slots in different sessions.	9
5. An examiner occurs in two places at the same time.	390
6. Session did not start during 1st period in one room OR, two sessions are not contiguous.	1 per timeslot

During the calculation, the penalties which have larger penalty points are more likely to be avoided during the evolution of GA. Depending on the actual situation of the priority of the penalties, each penalty is allocated with different penalty points. Since it is physically impossible for “an examiner occurring in two places at the same time”, penalty 5 is allotted the highest penalty points. The penalty 1, “examiner is allocated into timeslots with X” is then another important penalty we want to avoid, therefore it is allotted the second-highest penalty points, followed by penalties 2, 3, 4 and 6. Moreover, according to the real situation, penalties 5 and 2 are two situations which we want to avoid as far as possible, the other penalties, however, have some degree of tolerance. Therefore, the value of penalties 5 and 2 should be far greater than the value of other penalties.

4.4. Selection

Tournament selection is implemented, the tournament size is set as 2.

4.5. Crossover

In this paper, a constraint-based crossover operation is designed to satisfy the hard constraints. During the constrained crossover operation, the chromosomes have been decomposed into two parts. The first problem is the optimization of the examiner’s schedule problem, which corresponds to penalty 1 and 2 in Table 1.4. The second problem is to make sure the same examiner can attend the presentations contiguous in the same session and between two sessions to save the examiner’s time, which is called examiner’s time continuity problem, which corresponds to penalty 3, 4 and 5 in Table 1.4. Therefore, the constraint-based crossover must achieve the exploration for both, examiner’s schedule problem search region and examiner’s time continuity problem search region. In the proposed model, a variant multi-point crossover has been applied to achieve the information exchange for both examiner’s schedule problem and examiner’s time continuity problem, and meanwhile, maintain the feasibility for each chromosome.

In the first step, two parent chromosomes will be selected. Then, for each parent chromosome, two random sessions, session A and session B will be selected. Session A and session B could be either real sessions or zero sessions. A zero session means no session is to be placed in these timeslots. Then session A on the first parent chromosome and session B on the second parent chromosome will be swapped to get two new chromosomes. Similarly, session B on the first parent chromosome and session A on the second parent chromosome will be swapped. Since the different sessions could have different number of students, once a session with fewer students is swapped with a session with more students, the algorithm will check if there is enough blank space beside the shorter session to fill the position of longer session. If not, the model will then check if it is possible to move the adjacent session up or down to vacate enough space. However, the moving of the session should maintain this session within one single room and one single day. Secondly, if enough space cannot be vacated by moving the adjacent session up or down, the adjacent session will be moved to another random place, where this random place should also satisfy the constraints to maintain the session within one single room and one single day. In this way, the exchange of gene segments could be achieved, and meanwhile, the integrity of each session can be maintained and ensured that each session is held on the same day.

For example, there is a problem with 5 sessions, [1,2,3,4],[5,6,7],[8,9,10,11],[12,13,14] and [15,16], and there are two parent chromosomes,

[15,16,0,0,0,0,0,0,0,0, 9,8,10,11,0,0,0,0,0,0, 13,12,14,0,0,0,0,0,0,0, 5,6,7,0,1,2,3,4,0,0] and

[0,0,0,0,0,0,0,0,0,0, 8,9,10,11,0,1,2,3,4,0, 0,0,00,0,0,0,0,0,0, 7,6,5,0,0,12,13,14,15,16]

Assuming that session [8-11] and [15, 16] are selected. The session [8-11] from parent chromosome 1 and session [15, 16] from parent chromosome 2 will be swapped. Similarly, the session [15, 16] from parent chromosome 1 and session [8-11] from parent chromosome 2 will be swapped as well. However, since the length of session [15, 16] is shorter than that of session [8-11] and in parent chromosome 2, there are not enough blank timeslots beside the session [15, 16]. Therefore, the session [12-14] in the parent chromosome 2 will be moved two timeslots forward to vacate enough position for session [8-11]. And the offspring chromosomes after the crossover are:

[8,9,10,11,0,0,0,0,0,0, 15,16,0,0,0,0,0,0,0,0, 13,12,14,0,0,0,0,0,0,0, 5,6,7,0,1,2,3,4,0,0]

[0,0,0,0,0,0,0,0,0,0, 15,16,0,0,0,1,2,3,4,0, 0,0,00,0,0,0,0,0,0, 7,6,5,12,13,14,9,8,10,11]

Moreover, if the session [12-14] in the parent chromosome 2 could not vacate enough timeslot for the session [8-11] then the session [12,13,14] in the parent chromosome 2 will be moved to another random feasible position. This operation is to guarantee every session would have the same opportunity to be exchanged.

However, compared to the traditional crossover, the amount of information exchanged by this constraint-based crossover is relatively limited, where only two sessions of information can be exchanged during each crossover operation. Moreover, as we mentioned in section 4.3, the constraints with high priority are given much higher penalty points. The limited information exchange and uneven proportion of penalty points make the algorithm usually be stuck into the local optima solution with a high penalty point. To improve the result quality, an improvement approach called initial population pre-training is applied to the proposed algorithm, which will be described in section 4.8.

4.6. Mutation

During the constrained mutation operation, one session in the chromosome is selected first and then two random students in this session are swapped.

4.7. Elitism

The research of [16] [17] shows that the elitism of GA can improve the convergence speed. However, the increased number of remained elitism individuals can increase the evolution pressure which may cause premature convergence [18]. To preserve the diversity of the population, in the proposed model, only the first best solution in the proposed model can be retained to the next iteration.

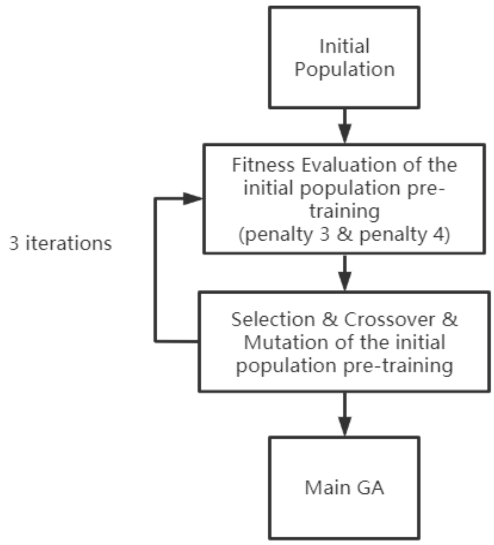
4.8. Initial population pre-training

As we mentioned in crossover part, the proposed problem can be decomposed into two optimization problems: examiner's schedule problem and examiner's time continuity problem. However, it is usually hard for the proposed algorithm to both the objective problems at the same time. If relatively larger penalty points are applied to the examiners' schedule problem, the examiners' time continuity problem will be stuck in a local optimum at a higher opportunity. Similarly, larger penalty points on the examiner's time continuity problem could cause a bad result on examiners' schedule problem. In the proposed model, depending on the priority of the penalties in the real situation, examiners' schedule problem is given relatively higher penalty points. Moreover, in the proposed algorithm, the constraint-based crossover makes a concession on search ability to satisfy the hard constraints. Therefore, this constraint-based crossover operation is not able to make enough information exchange and cannot keep the diversity of the population. Therefore, during the research of the variant GA algorithm, it is found that the algorithm is usually stuck into the local optimal solution with high penalty points on penalty 3 and penalty 4. For the sake of keeping balance between the examiners' schedule problem and the examiners' time continuity problem, and to further improve the diversity of the population, we propose an improvement approach, namely, initial population pre-training which has been proved to be effective in enhancing the result quality.

In 1993, Schoenauer and Xanthakis [19] introduced a genetic optimization based on the Behaviour Memory Paradigm. The method first only considers only one constraint; when sufficient number of feasible individuals satisfy this constraint, the algorithm will then consider next constraint and eventually, all constraints can be satisfied.

The initial population pre-training operation in the proposed model refer to this idea. The initial population pre-training operation is conducted on the population before the main iterations of GA. During pre-training operation, the populations are evaluated by only the examiner's time continuity problem for serval iterations. In this way, some good solutions on the examiners' time continuity problem can be generated in advance to reduce the search pressure on examiner's time continuity problem. However, the iteration number of pre-training cannot be so high to avoid the homogeneity of the initial population. In the proposed model, the pre-training operation runs only 3 iterations. The test result shows the initial population pre-training can improve the result quality. Figure 1 shows the flowchart of the initial population pre-training operation.

Figure 1. Flowchart of the initial population pre-training operation



4.9. Flowchart of the whole model

Figure 2 shows the flowchart of the whole proposed model.

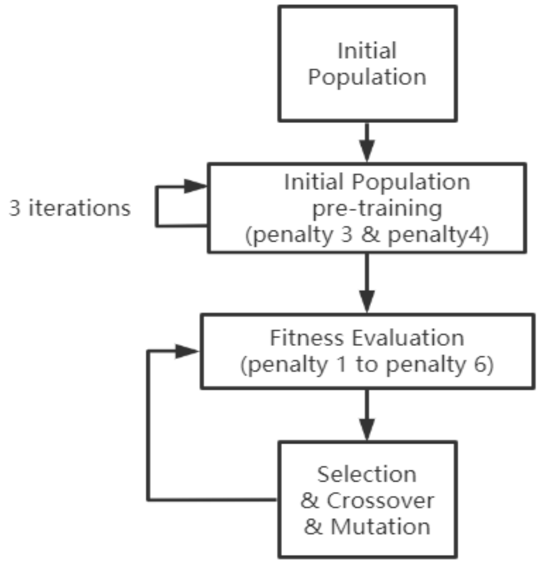


Figure 2. Flowchart of the whole proposed model

5. TESTING

The test uses the real data from Sophia University, 2020 Winter Presentation event for Informatics Graduate School, which contains 31 students and 25 examiners. Below, two different variants of the proposed model are tested, where model 1 is the proposed model and model 2 is the control group. Each model is tested 10 times to get an average value. The total population size for each model is equal to 120. The initial population pre-training runs 3 iterations. The toleration for the stop condition is 30 iterations. The crossover ratio is set as 0.7, and the mutation ratio is set as 0.1.

Model 1: Proposed GA with initial population pre-training

Model 2: Proposed GA only

Table 3 and Table 4 show the testing result of model 1 and model 2 respectively, where the penalty 1 to penalty 6 in the tables means the quantity of penalty imposed on the solution for constraint violations.

For model 1, the average penalty points are 116.8, which breaks down to an average 0.1 times for penalty 1, 5.0 times for penalty 3, 4.2 times for penalty 4 and 4.8 times for penalty 6. Examiners can be allocated into their available time for most of the time. The model shows an acceptable performance as verified by the university staff.

The comparison between model 1 and model 2 shows that, by adding the pre-training, the average penalty points of penalty 1 slightly decrease from 0.2 to 0.1, the average penalty points of penalty 2 decreases from 0.8 to 0.0, and the average penalty points of penalty 3 decreases from 7.1 to 5. On the contrary, the average penalty points of penalty 5 slightly increase from 4.0 to 4.2, the average penalty points of penalty 6 increase from 3.4 to 4.8. However, penalty 4 and penalty

6 have the lowest priority in the problem, therefore, we think the increase of the penalty points on penalty 4 and penalty 6 are acceptable. In total, model 1 obtained lower total average penalty points than that of model 2, which is 116.8 compared to 206.8. The result shows that the adding of initial population pre-training can improve the result quality. Moreover, the average calculation iteration of model 1 is 104.1 plus 3 iterations of initial population pre-training, compared to the 103.7 average iterations of model 2, the calculation speed of the improved model does not decrease so much.

Table 3. Test result for Model 1.

test	penalty points	iteration	penalty 1	penalty 2	penalty 3	penalty 4	penalty 5	penalty 6
1	135	59	0	0	6	8	0	3
2	352	91	1	0	7	4	0	4
3	92	127	0	0	5	4	0	6
4	124	117	0	0	7	5	0	9
5	91	82	0	0	4	5	0	6
6	70	90	0	0	3	4	0	4
7	61	54	0	0	5	1	0	2
8	97	182	0	0	4	6	0	3
9	72	195	0	0	4	3	0	5
10	74	44	0	0	5	2	0	6
Average	**116.8**	**104.1**	**0.1**	**0**	**5**	**4.2**	**0**	**4.8**

Table 4. Test result for Model 2.

test	penalty points	iteration	penalty 1	penalty 2	penalty 3	penalty 4	penalty 5	penalty 6
1	160	162	0	1	9	1	0	1
2	180	55	0	1	10	2	0	2
3	198	54	0	1	9	5	0	3
4	167	96	0	1	5	6	0	3
5	225	67	0	1	8	9	0	4
6	740	50	2	2	8	5	0	11
7	131	119	0	1	5	2	0	3
8	68	153	0	0	3	4	0	2
9	107	188	0	0	7	4	0	1
10	92	93	0	0	7	2	0	4
Average	**206.8**	**103.7**	**0.2**	**0.8**	**7.1**	**4**	**0**	**3.4**

6. CONCLUSION

This paper focuses on a specific examination timetabling problem which commonly occurs in Japanese universities and proposed a variant genetic algorithm approach to solve the exam timetabling problem. The proposed model is written in VBA programming language, which is easy for the university staff to use. Constraint-based initialization and crossover operations are used to satisfy the hard constraints, and a penalty system is implemented to optimize the soft constraints. To improve the result quality, the initial population pre-training, which optimizes partial objective problem in advance for several iterations, is applied. The proposed model is compared with the model without pre-training, by using the real exam data from Informatics Graduate school from Sophia University in 2020. The positive comparison results support the idea that the initial population pre-training is an effective approach to improve the result quality of the proposed model.

However, to compromise on the feasibility of the problem constraints, the constraint-based crossover operation cannot reach wide-enough search space. Therefore, the solution often got stuck into the local-optima solution. Even though the initial population pre-training can solve this problem in a way, the research on a better crossover approach for this specific examination timetabling problem is still needed. In the proposed case, the discrete search space of ETP is further scattered because of the hard constraints 1 and 2. The local search approaches have the advantage of exploitation, while the population-based approaches have the stronger ability of exploration. Therefore, applying other types of population-based algorithms is one direction of further research. Another research direction is improving the exploration ability and the result quality of the GA. A lot of researches show that parallel GA can improve the diversity of the population to give a better result compared to the normal GA [20] [21]. Therefore, as for the future direction of this research, we plan to use the GA island model for faster convergence and better results.

REFERENCES

[1] Song, Y., Wang, F., & Chen, X. (2019). An improved genetic algorithm for numerical function optimization. Applied Intelligence, 49(5), 1880-1902.

[2] Qu, R., Burke, E. K., McCollum, B., Merlot, L. T., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. Journal of scheduling, 12(1), 55-89.

[3] Carter, M. W., & Laporte, G. (1995, August). Recent developments in practical examination timetabling. In International Conference on the Practice and Theory of Automated Timetabling (pp. 1-21). Springer, Berlin, Heidelberg.

[4] Thompson, J. M., & Dowsland, K. A. (1996). Variants of simulated annealing for the examination timetabling problem. Annals of Operations research, 63(1), 105-128.

[5] Thompson, J. M., & Dowsland, K. A. (1998). A robust simulated annealing based examination timetabling system. Computers & Operations Research, 25(7-8), 637-648.

[6] Di Gaspero, L., & Schaerf, A. (2000, August). Tabu search techniques for examination timetabling. In International Conference on the Practice and Theory of Automated Timetabling (pp. 104-117). Springer, Berlin, Heidelberg.

[7] Abdullah, S., Ahmadi, S., Burke, E. K., Dror, M., & McCollum, B. (2007). A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem. Journal of the Operational Research Society, 58(11), 1494-1502.

[8] Azimi, Z. N. (2005). Hybrid heuristics for examination timetabling problem. Applied Mathematics and Computation, 163(2), 705-733.

[9] Eley, M. (2006, August). Ant algorithms for the exam timetabling problem. In International Conference on the Practice and Theory of Automated Timetabling (pp. 364-382). Springer, Berlin, Heidelberg.

[10] Mandal, A. K., & Kahar, M. N. M. (2015, April). Solving examination timetabling problem using partial exam assignment with great deluge algorithm. In 2015 International Conference on Computer, Communications, and Control Technology (I4CT) (pp. 530-534). IEEE.

[11] Leite, N., Fernandes, C. M., Melicio, F., & Rosa, A. C. (2018). A cellular memetic algorithm for the examination timetabling problem. Computers & Operations Research, 94, 118-138.

[12] 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.

[13] Shatnawi, A., Fraiwan, M., & Al-Qahtani, H. S. (2017, February). Exam scheduling: A case study. In 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI) (pp. 137-142). IEEE.
[14] Dener, M., & Calp, M. H. (2019). Solving the exam scheduling problems in central exams with genetic algorithms. arXiv preprint arXiv:1902.01360.
[15] Gozali, A. A., Kurniawan, B., Weng, W., & Fujimura, S. (2020). Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy. IEEJ Transactions on Electrical and Electronic Engineering, 15(3), 389-400.
[16] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," Evol. Comput., vol. 8, no. 2, pp. 173–195, Summer 2000.
[17] G. Rudolph, "Evolutionary search under partially ordered sets," Dept. Comput. Sci./LS11, Univ. Dortmund, Dortmund, Germany, Tech. Rep. CI-67/99, 1999.
[18] Ahn, C. W., & Ramakrishna, R. S. (2003). Elitism-based compact genetic algorithms. IEEE Transactions on Evolutionary Computation, 7(4), 367-385.
[19] M. Schoenauer and S. Xanthakis, "Constrained GA optimization," in Proc. Int. Conf. Genetic Algorithms, vol. 5, 1993, pp. 573–580.
[20] Gozali, A. A., & Fujimura, S. (2019). DM-LIMGA: dual migration localized island model genetic algorithm—a better diversity preserver island model. Evolutionary Intelligence, 12(4), 527-539.
[21] Sun, X., Chou, P., Wu, C. C., & Chen, L. R. (2019). Quality-oriented study on mapping island model genetic algorithm onto CUDA GPU. Symmetry, 11(3), 318.

AUTHOR

Jiawei Li is a PhD student in the Department of Information & Communication Sciences. Faculty of Science and Technology, Sophia University, Tokyo, Japan. His research interests are Evolutionary Computation and Neuro-Evolution.