

A Variant Genetic Algorithm for a Specific Examination Timetabling Problem in a Japanese University



Jiawei Li^{1,*} and Tad Gonsalves¹

¹Department of Information & Communication Sciences, Faculty of Science and Technology, Sophia University, Tokyo, Japan

Abstract: Background: Examination Timetabling Problem that tries to find an optimal examination schedule for schools, colleges, and universities, is a well-known NP-hard problem. This paper presents a Genetic Algorithm variant approach to solve a specific examination timetabling problem common in Japanese colleges and universities.

Methods: The proposed algorithm uses a direct chromosome representation Genetic Algorithm and implements constraint-based initialization and constraint-based crossover operations to satisfy the hard and soft constraints. An island model with varying crossover and mutation probabilities and an improvement approach called pre-training are applied to the algorithm to further improve the result quality.

Results: The proposed model is tested on synthetic as well as real datasets obtained from Sophia University, Japan and shows acceptable results. The algorithm was fine-tuned with different penalty point combinations and improvement combinations.

Conclusion: The comparison results support the idea that the initial population pre-training and the island model are effective approaches to improve the result quality of the proposed model. Although the current island model used only four islands, incorporating a greater number of islands, and some other diversity maintenance approaches such as memetic structures are expected to further improve the diversity and the result quality of the proposed algorithm on large scale problems.

Keywords: Examination timetabling problem, genetic algorithm, direct chromosome representation, Island model, parallel genetic algorithm, evolutionary algorithms, meta-heuristic algorithms.

1. INTRODUCTION

Examination Timetabling Problem (ETP), which tries to find the best examination schedule for schools, colleges, and universities, is a well-known NP-hard problem. The early research on ETPs can be roughly divided into four types of approaches: Cluster, Sequential, Constraint-Based, and Generalized Search [1, 2]. Generalized Search has another well-known name, *i.e.*, meta-heuristic algorithms. In the early research on ETPs, simulated annealing algorithm [3, 4] and Tabu search [5, 6] were the two main Generalized Search approaches. This paper focuses on a special case of the Examination Timetabling Problem, which is common in Japanese colleges and universities. Very few related studies can be found in the review literature, and many universities in Japan solve this tabling problem manually. Therefore, this study is an exploratory experiment to find an automated solution for this specific timetabling problem. The proposed

algorithm has been used to solve the presentation timetabling problem for the Graduate School of Information Science, Sophia University, Japan, in 2019, 2020, and 2021 successfully. Moreover, in Japan, school and university staff work with Microsoft Excel. To reduce their workload, the proposed model is written in Excel VBA programming language, which can be easily executed as a macro.

In recent years, with the development of ETPs research, meta-heuristic algorithms have become the mainline approach to solving ETPs, and many kinds of meta-heuristic algorithms are implemented. In 2005, Azimi [7] applied Simulated Annealing (SA) [8], Tabu Search (TS) [9], Genetic Algorithm (GA) [10], and Ant Colony System (ACS) [11] to solve the ETP. The author further proposed three novel hybrid combinations of the four algorithms: Sequential TS-ACS, Hybrid ACS/TS, and Sequential ACS-TS algorithms and demonstrated that all three hybrid algorithms performed significantly better than the four non-hybrid algorithms when tested on more than 10 different scenarios of the ETP. In 2006, Eley [12] applied two ant colony approaches: Max-Min and ANTCOL, for solving the ETP and compared these two approaches with other timetabling heuristics. In 2015, Mandal & Kahar [13] applied the great del-

*Address correspondence to this author at the Department of Information & Communication Sciences, Faculty of Science and Technology, Sophia University, Tokyo, Japan; E-mail: jiaweitk@eagle.sophia.ac.jp

uge algorithm to the partial exam assignment. In their approach, the total exams are ordered in advance, based on the 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.

The Genetic Algorithm (GA) is one of the most popular optimization algorithms, widely used to solve theoretical and real-life problems. The discrete version of the GA is suitable to solve this type of optimization problem since, compared to other search algorithms, it is more robust in searching complex search spaces [14]. GA has developed into a mature technology for solving ETPs. In 2017, to save time for the university staff, Rozaimee *et al.* [15] constructed the final exam timetable on the UniSZA computer system by using GA, while Shatnawi *et al.* [16] proposed a two-stage approach optimization algorithm by running the 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 results show that the required number of conflicts, exam days, and available venues have been reduced successfully. In 2019, Dener and Calp [17] 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 more efficiently to reduce the number of rooms and time consumption. In 2020, Ezike *et al.* [18] hybridized the GA with the TS to develop the GATS algorithm for ETPs. In this hybrid algorithm, the TS is used to improve the child solutions created by the GA. An enhanced crossover operator is also introduced. The GATS is compared with the classic GA and TS on the problem of first Order Conflict Counts (OCC) and second OCC and it obtained higher quality solutions compared to the individual algorithms. Owing to the superior performance of GA in the ETP research, the discrete GA was chosen as the main algorithm at the beginning of this exploratory project, and it shows satisfactory results in terms of quality and computation time. In 2021, Ngo *et al.* [19] applied the multi-objective GA to an enrollment-based university course timetable problem with around 3,000 students. The proposed TPS aims to create student schedules automatically before each semester, based on a student's study preferences, the professors' schedule, number of lecturers, available rooms, and many other conditions. The proposed algorithm shows its effectiveness in both computational cost and solution quality.

However, most of the optimization algorithms share the premature problem drawback, not only in ETPs but in nearly all kinds of optimization problems. Developing algorithms and constructs to evade local optima traps is still a hotbed of research in optimization.

The island model is an effective approach for improving the population diversity and avoiding local optima. Many research studies show that the island model effectively improves the result quality of the algorithms. Further, the island model is naturally suitable for parallel computation, which allows multiple computation units to work together to increase the overall speed.

In 2016, Kurdi [20] applied the island GA to solve the shop scheduling problem. He proposed a new evolution model and migration selection mechanism to improve the island mode. Furthermore, in 2017, Palomo-Romero *et al.* [21] applied the island GA to solve the Unequal Area Facility Layout Problem. The results show that the island GA approach shows a better performance compared to that of the previous approaches. In 2020, García-Hernández *et al.* [22] applied the island model to the Coral Reefs Optimization Algorithm (IMCRO) to solve the Unequal Area Facility Layout Problem (UA-FLP). They made two different versions of the proposed approach, and both show superior performance. In 2020, Gozali *et al.* [23] proposed a localized GA island model with a dual dynamic migration policy (DM-LIMGA) to solve the University Course Timetabling Problem (UCTP). Their study shows that the DM-LIMGA could solve both Telkom UCTP 2011/2012 and 2016/2017 enrollment years problems with acceptable results. In 2021, Rezaeipannah *et al.* [24] solved the ETPs by introducing a hybrid method, IPGALS, based on the Improved Parallel Genetic Algorithm (IPGA) and Local Search (LS). In IPGA, there is an island GA with shared memory programming. In the local search, some unused genes in one chromosome are randomly swapped to generate new individuals. Their experiments show that the IPGALS algorithm gives higher quality solutions than the other similar methods. However, the performance of IPGALS on some large instances is still limited. It cannot always reach a feasible solution on large-scale problems.

Memetic algorithms that combine the population-based and local search algorithms are also found to improve the performance on ETPs. For instance, in 2018, Leite *et al.* [25] 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 improves partial functions of the incapacitated Toronto and capacitated ITC 2007 benchmark sets. One disadvantage is that the memetic algorithms are prone to call the fitness evaluation function many times. In our study, since more than half of the computation time is taken by the fitness evaluation, memetic algorithms are not considered applicable.

This paper introduces an exploratory project that tried to solve a special ETP common in Japanese universities. As one of the most popular and mature technology in ETPs, GA has the advantage of applicability and robustness, and it is easy to be applied to a variety of problems. Therefore, GA is chosen to solve the proposed problem.

In the proposed timetabling problem, every final-year student must give a presentation to get his/her academic research evaluated. Each student is evaluated by three examiners. The examination timetabling problem is related to allocating appropriate timeslots and rooms to students and examiners. It is preferable to put all the students from the same research group together in the same room for presentations in successive time slots. This is called a session. Any given session should be completed in a single day without being extended to the next day. These two constraints make the traditional GA no longer effective because the traditional

crossover operation can break the intact session with a high probability, resulting in numerous infeasible solutions.

Therefore, a novel variant of GA is designed to satisfy this special case, where the constraint-based initialization and crossover operations are applied to satisfy the two constraints imposed on the 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 leading to unacceptable results. To improve the result quality of the proposed model, an improvement called initial population pretraining is adopted to balance the weight between different soft constraints. A 4-island structure is applied to improve population diversity. Each island is assigned different crossover and mutation probabilities. This multiple-probability approach shows an improvement in the result quality.

The rest of the paper is organized as follows: Section 2 describes the objective problems in detail. Section 3 explains the methodology of the proposed algorithm. Section 4 contains the experiments and results. Section 5 contains the conclusion and directions for further research.

2. MATERIALS AND METHODS

2.1. Objective Problems

The proposed model is designed to solve the thesis presentation and final exams of the Graduate School of Information Science, Sophia University, Tokyo, Japan, which is different from the general ETPs. This ETP problem is common in Japanese universities. However, there are few research studies dealing with this specific problem. Moreover, since the scale of the problem is usually not very large, many universities still solve this problem manually.

At the end of the academic year, all final-year students in Japanese universities 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. The examiners are decided in advance and cannot be changed. The university requires putting the students with the same supervisor together in a session to conduct the presentations one after another. Moreover, the whole session should be held on the same day. These requirements break the conventional GA operations. In the common ETPs, each examination is independent, which means a single examination could be moved independently wherever feasible. The common GA operations can be directly applied to solve the problem. However, in the proposed ETP, the arbitrary chromosome cut and join would 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 operation is proposed to satisfy the requirements automatically.

The hard constraints and soft constraints are listed below. In the proposed algorithms, the hard constraints are satisfied by the chromosome operations. The soft constraints are satisfied by the penalty points systems. Therefore, the

objective of the genetic algorithm is to find the solution with the smallest penalty points.

2.2. Hard Constraints

1. The students with the same supervisor should hold the presentations consecutively, which is called a session.
2. Any given session should be held on a single day.
3. No student or examiner can be removed from the session.
4. The number of available rooms is limited. The capacity of the examination rooms is also limited. A room can hold a maximum of 10 presentations per day. Each room can only have one presentation at a time.

2.3. Soft Constraints

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

For each timeslot, the examiner has three states: "Available", "X", and " Δ ". "Available" means the examiner is available for that time period. An "X" means the examiner is not available for that period. A " Δ " means the examiner is available during that period, but that timeslot should be avoided if possible. "X" has a higher priority than " Δ ", so the "X" is allocated with a higher penalty (Refer to Appendices).

2. Examiners should be assigned to contiguous time slots, if possible.
3. The timetable should be compact.

For each day, the presentation should start from the very first timeslot and the time gap between any two sessions should be minimized.

4. An examiner should not be present in two places at the same time. (Note that, although from the feasibility perspective, this is a hard constraint, for the fast convergence of the proposed algorithm, it is classified as a soft constraint and is satisfied by the penalty point system).

3. PROPOSED VARIANT GENETIC ALGORITHM

Fig. (1) presents the flowchart of the whole algorithm. The main body of the proposed algorithm is the discrete GA with direct chromosome encoding. The constraint-based initialization operation and constraint-based crossover operation, selection, mutation, and elitism are implemented to build the main body of GA. The pre-training operation is applied between the constraint-based initialization operation and the main iterations of GA to improve the result quality. The entire search population of GA is divided into 4 parts. Each part is called a sub-island, does independent GA computing and shares its good solutions with the other islands *via* the migration operation.

3.1. Chromosome Representation

The proposed model uses the direct chromosome method for encoding, where each chromosome corresponds to a specific arrangement of the generated blank position unit.

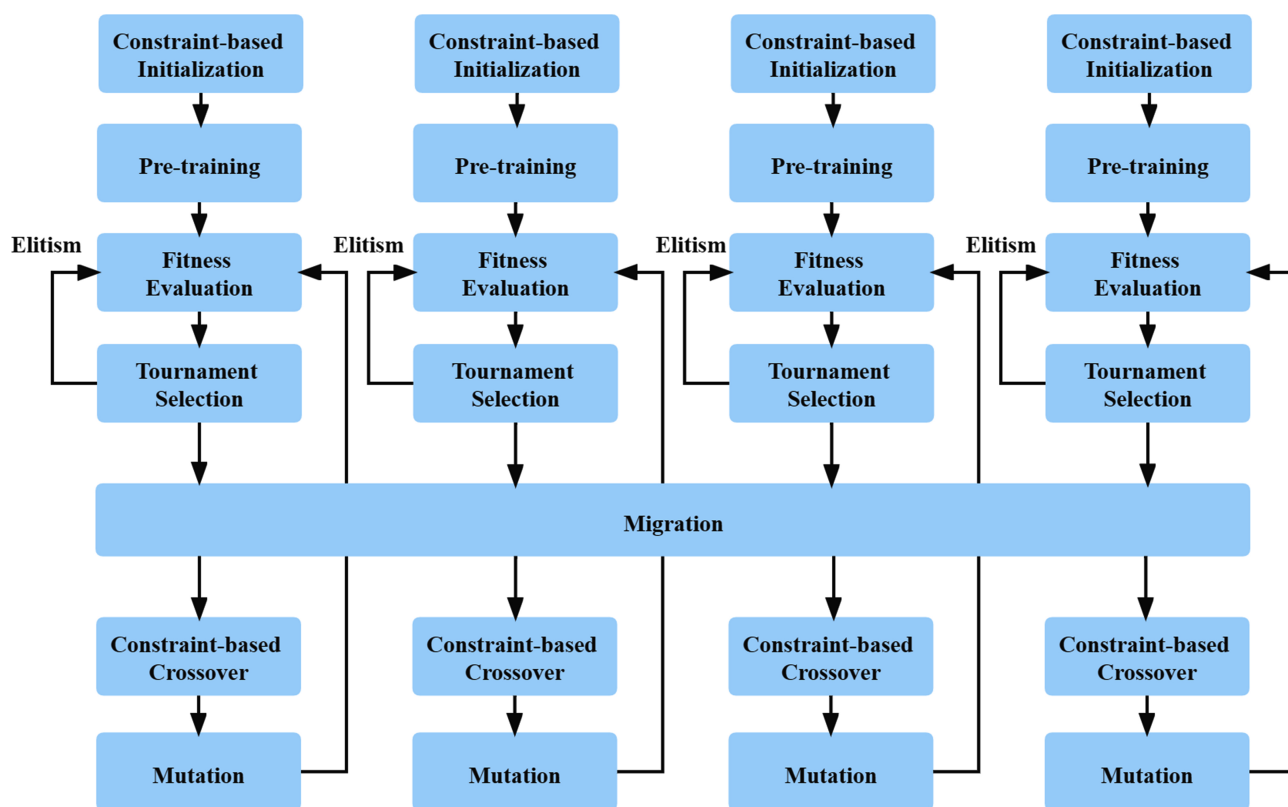


Fig. (1). Flowchart of the proposed algorithm. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

The total chromosome length is equal to the total number of available timeslots. For instance, assuming there is a two-day presentation period, each day has 2 available rooms, each one can hold a maximum of 10 presentations per day, then the total chromosome length is equal to $2 \times 2 \times 10 = 40$ timeslots. The students will be numbered from 1 to the maximum number. Every non-zero number in the chromosome indicates a specific student with his/her examiners in a specific timeslot. Any timeslot that does not have an arrangement for a student's presentation is represented by a 0. Accordingly, a candidate chromosome is shown in Fig. (2). In this chromosome string, student 5 is allocated to timeslot 1, corresponding to the first presentation in Room A of the first day. Similarly, student 11 is allocated to timeslot 17, which corresponds to the 7th presentation in Room B on the first day, student 16 is allocated to timeslot 22, which corresponds to the 2nd presentation in Room A on the second day, while there are no students allocated to Room B on the second day, which corresponds to the timeslots from 31 to 40.

3.2. Constraint-based Initialization Operation

To satisfy hard constraints 1 and 2, which are described in Section 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 reordered randomly.

Secondly, the first session is allocated to the first available place in the first room on the first day. The second ses-

sion is allocated to the first available place in the second room, first day and so on. When all rooms have been allocated to a session, the rest of the sessions will be randomly allocated to any feasible room. If there is no way to allocate all research groups to the proper position, an error is reported to ask the staff to rearrange the rooms or add some new rooms. In this way, all initial possible solutions can satisfy the hard constraints automatically (*i.e.*, the students from the same research group can be allocated together in a session, and all presentations can be held on the same day). The constrained initialization, while satisfying some of the initial conditions, can also maintain a certain degree of randomness to maintain the diversity of the initial population.

3.3. Fitness Evaluation and Penalty System

The fitness evaluation of the proposed model uses the penalty point system to optimize the soft constraints in the proposed problem. In the penalty points system, a penalty is imposed if a candidate solution breaks any of the soft constraints. The proposed timetabling problem then becomes an optimization problem to find the solution with a minimum penalty. During the GA computation, the four soft constraints mentioned above could be further decomposed into six soft GA constraints (SGACs), with the specific distribution of the penalty points, as shown in Table 1.

The SGACs with larger penalty points are more likely to be avoided during the GA evolution. From priority considerations, SGACs 1 and 5 are allotted the highest penalty points. The SGAC 2 "the examiner is allocated into timeslots with Δ " is another important penalty we want to

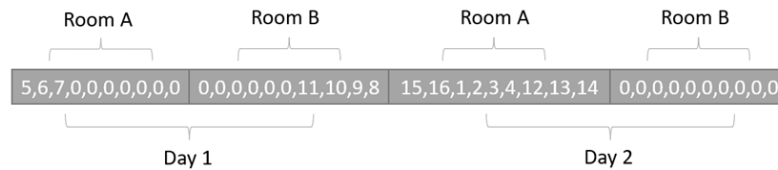


Fig. (2). Chromosome representation example. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

avoid. Therefore, it is allotted the second highest penalty points, followed by SGACs 3, 4, and 6. Moreover, in the real situation, SGACs 5 and 1 are two situations that must be avoided as far as possible; however, the other penalties have some degree of tolerance. Therefore, the penalty points of SGACs 5 and 1 should be far greater than the value of other penalties.

Table 1. Soft GA constraints’ penalty points.

S. No.	Soft GA Constraints	Penalty Points
1	Examiner is allocated to period with X.	242
2	Examiner is allocated to period with Δ.	60
3	Examiners are not placed in contiguous slots in the same session.	10
4	Examiners are not placed in contiguous slots in successive sessions.	9
5	An examiner occurs in two places at the same time.	390
6	Session did not start during the 1st period in one room or two sessions are not continuous.	1 per timeslot

The penalty points are also adjustable for different cases. If the examiner’s schedule is overcrowded, which means it is relatively hard to allocate all examiners in their available time, the penalty points of SGACs 3 and 4 are reduced to put more weight on avoiding the infeasible time of the examiner. On the other hand, if the schedule is not crowded, the penalty points of SGACs 3 and 4 can be increased to reach a better overall solution. The details are explained in the testing part.

3.4. Selection Operator

Tournament selection [26] is chosen in the proposed model to enhance the convergence speed of the model. Tournament selection provides a controllable selection pressure by changing the value of the tournament size [27]. In the proposed algorithm, the tournament size is set to 2 to obtain a low selection pressure and a high population diversity.

3.5. Constraint-based Initialization Crossover Operator

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 corre-

sponds to SGACs 1 and 2 in Table 1. The second problem is to make sure an examiner is placed contiguously in the same session as well as in the adjoining sessions. This situation corresponding to SGACs 3 and 4 in Table 1 is referred to as the examiner’s time continuity problem. Therefore, the constraint-based crossover must achieve the exploration for both the examiner’s schedule problem search region and the examiner’s time continuity problem search region. In the proposed model, a variant multipoint crossover has been applied to achieve the information exchange for both the examiner’s schedule problem and examiner’s time continuity problem, meanwhile satisfying the hard constraints 1 and 2 to maintain the feasibility of 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 B could be either real sessions or zero sessions. A zero session means no session is to be placed in these timeslots. In this algorithm, a real session has a 25% probability of crossover with a zero session.

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 different sessions could have different numbers 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 the longer sessions. 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. In this way, the exchange of gene segments can be achieved, maintaining the integrity of each session, and ensuring its completion in a single day.

For example, consider 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 shown in Fig. (3).

Assuming that sessions (8-11) and (15, 16) are selected, session (9, 8, 10, 11) from parent chromosome 1 and sessions (15, 16) from parent chromosome 2 will be swapped. Similarly, the session (15, 16) from parent chromosome 1 and sessions (8-11) from parent chromosome 2 will be swapped as well. However, since the length of the session (15, 16) is shorter than that of session (9, 8, 10, 11), and in parent chromosome 2 there are not enough blank timeslots beside the session (15, 16), session (12-14) in the parent chromosome 2 will be moved two timeslots forward to vacate enough positions for sessions (9, 8, 10, 11). The off-

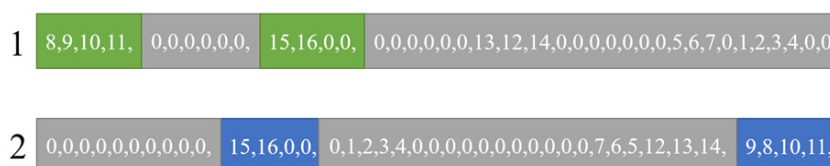


Fig. (3). Constraint-based Initialization crossover operator, parents' example. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

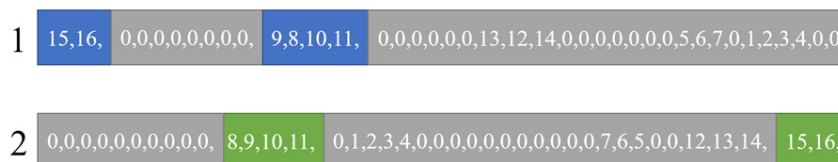


Fig. (4). Constraint-based initialization crossover operator. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

spring chromosomes after the crossover operation are shown in Fig. (4).

However, compared to the traditional crossover, the amount of information exchanged by this constraint-based crossover is relatively limited since only two sessions can be exchanged during each crossover operation. Moreover, as mentioned in Section 4.3, the constraints with higher priority are given much higher penalty points. The limited information exchange and uneven proportion of penalty points usually make the algorithm fall into a local optimal solution with a high penalty score. To improve the result quality, an improvement approach called initial population pretraining and island model are applied to the proposed algorithm as described in Sections 3.8 and 3.9.

3.6. Mutation Operator

Mutation operator can be divided into two steps. In the first step, one session in the chromosome is selected first and then two random students in this session are swapped. In the second step, a new session is selected randomly. This session is randomly moved up or down for several timeslots to an available position.

3.7. Elitism

Elitism strategy can decrease genetic drift by passing the most fitting individuals directly to the next iteration without genetic operations [28, 29]. However, the increased number of remaining elitism individuals can increase the evolution pressure, which may cause premature convergence [30]. To preserve the diversity of the population in the proposed model, only the first best solution can be retained for the next iteration.

3.8. Initial Population Pre-training Operator

As mentioned in the crossover part, the proposed problem can be decomposed into two optimization problems: the examiner's schedule problem and the examiner's time continuity problem. However, it is usually hard for the proposed algorithm to solve both the objective problems simultaneously. If larger penalty points are applied to the examiners' schedule problem, the examiners' time continuity problem

will be stuck in a local optimum with greater probability. Similarly, larger penalty points on the examiner's time continuity problem could cause a bad result in the examiners' schedule problem.

In the proposed model, depending on the priority of the SGACs in the real situation, the examiners' schedule problem is given higher penalty points. Moreover, in the proposed algorithm, the constraint-based crossover makes a concession on the search ability to satisfy the hard constraints. Therefore, this constraint-based crossover operation cannot make enough information exchange and cannot keep the diversity of the population. During several runs of the variant GA algorithm, it is found that the algorithm usually gets stuck into the local optimal solution with high penalty points on SGACs 3 and 4. For the sake of keeping the 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 pretraining, which has been proved to be effective in enhancing the result quality.

In 1993, Schoenauer and Xanthakis [31] introduced a genetic optimization based on the Behavior Memory Paradigm. The method first only considers only one constraint. When enough feasible individuals satisfy this constraint, the algorithm will then consider the next constraint and eventually, all constraints can be satisfied.

The initial population pretraining operations in the proposed model refer to this idea. The initial population pretraining operation is conducted on the population before the main iterations of GA. During pretraining operation, the populations are evaluated by the SGAC 3 only for several iterations. In this way, some good solutions to the examiners' time continuity problem can be generated in advance to reduce the search pressure on the examiner's time continuity problem. However, the iteration number of pretraining cannot be too high to avoid the homogeneity of the initial population. In the proposed model, the pretraining operation runs no more than 3 iterations. The test result shows that the initial population pretraining can improve the result quality. Fig. (5) shows the flowchart of the initial population pretraining operation.

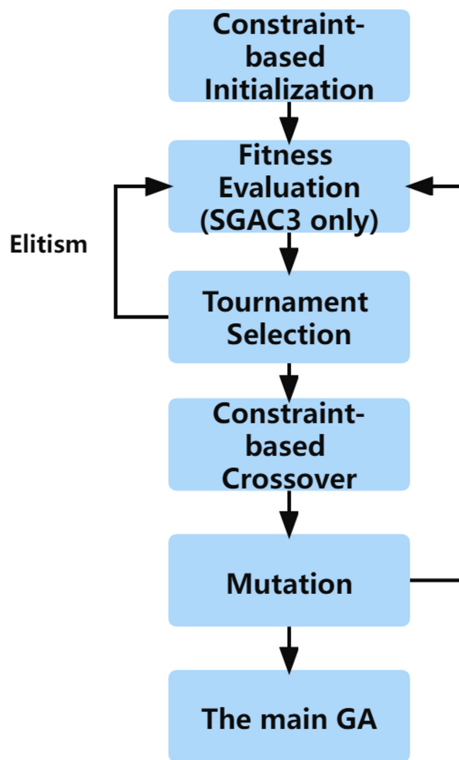


Fig. (5). Flowchart of the initial population pre-training operator.

3.9. Island Model

During the test of the single-series GA model, premature convergence occurred several times. This is because the proposed crossover operation sometimes cannot reach enough wide search region and therefore gets stuck in the local optimal solution.

A lot of research shows that the island GA delivers faster performance and higher solution quality than the conventional GA [32-34]. There are several sub-GAs in the island model and each of them does the iteration independently most of the time. For every certain number of iterations, the islands share their fitter individuals with each other [35]. This kind of cooperation between islands can increase the convergence speed and maintain some degree of independence of the sub-GA groups to increase the diversity of the population and reach a wider search region [36, 37]. Therefore, the island GA model is used to improve the diversity of the populations and avoid premature convergence.

Since the scale of the proposed problem is not too large, 4-island model is applied to the proposed algorithm. During the migration operation, 4 islands will be paired two by two randomly. For each island pair, the island with a better current optimal solution will give its current optimal solution to the other island and replace the position of a random individual in the target island. Furthermore, in the proposed model, the migration interval is equal to 1, which means the migration operation happens for every iteration. Moreover, each island has different crossover mutation probabilities (p_c, p_m) to further improve the population diversity of the

whole system and arrive at quality solutions. The p_c are 0.5, 0.6, 0.7 and 0.8 respectively, the p_m are 0.1, 0.2, 0.6 and 1.0 respectively. The sub-GAs with lower p_c and p_m are used to maintain the convergence efficiency, and the responsibility of the sub-GAs or islands with higher p_c and p_m are used to reach a wider search region. The test results show that the island model with multi-crossover and multi-mutation ratios could obtain a better solution than the solution of island GA with the same p_c and p_m .

4. RESULTS AND DISCUSSION

In Section 5, three different datasets have been used, where datasets 1 and 2 are synthetic and dataset 3 is the real data from Sophia University. Dataset 1 and dataset 2 have the same students and examiner list, which contains 50 students, 18 sessions, and 39 examiners. The difference is that dataset 1 has more crowded examiners' schedules, which means the examiners have more unavailable time. Dataset 3 contains 31 students, 11 sessions, and 25 examiners. During the testing, each variant of the algorithm is tested 10 times to get average performance.

The test configuration information is shown in Table 2.

Table 2. Test configuration information.

OS System	Windows 10
CPU	Intel Core (TM) i5-8300H CPU @ 2.30GHz
RAM	8GB
GPU	NVIDIA GeForce GTX 1060 Max-Q Design
Microsoft Office Edition	Microsoft Office 365 (2021)

As mentioned in the introduction, to reduce the staff workload, the proposed model was written in Excel VBA programming language. Although the computation speed for this programming language is much slower compared to the mainstream programming languages, such as C++, Java, or Python, it is acceptable for practical applications. Hence, this paper will focus on the iteration number for performance comparison instead of the practical computation time.

4.1. Test 1 Various Penalty Points Combinations

In this part, several different combinations of penalty points (CPP) are applied to datasets 1 and 2 to check how the performance of the algorithm varies with the change of the penalty points under different situations. The multi-crossover probability is 0.5, 0.6, 0.7, and 0.9, respectively, and the multi-mutation probability is 0.1, 0.2, 0.5, and 1, respectively, for each island. The total population size is 200 (50 for each island). Table 3 shows the results in detail.

In Table 3, C1 to C6 correspond to the constraints from 1 to 6. The term "Violations" refers to the number of constraints that have been violated in the solution. A smaller value corresponds to a better result. The term "Generations" means how many generations the algorithm uses to output the final answer. A smaller value corresponds to a faster computation speed.

Table 3. Result of test 1.

-	Dataset 1												Generations
	Penalty Points						Violations						
	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6	
CPP 1	20	20	20	20	20	1	2	1.3	0.8	0.4	1.2	10.2	178.6
CPP 2	40	40	20	20	40	1	0.7	0.6	2.6	1.3	1.1	12	176.1
CPP 3	100	100	20	20	100	1	0.2	0.6	2.5	1.5	0.3	11.1	234.1
CPP 4	20	20	40	40	20	1	2.1	3.2	0.1	0.2	2.4	9	182.3
CPP 5	242	60	10	9	242	1	0.2	1	1.8	2	0	11.3	236.1
CPP 6	242	60	10	9	390	1	0.6	1.2	2.4	1.8	0.1	15.1	240.9
CPP 7	500	60	10	9	242	1	0.1	1.8	4.3	2.4	0	13.5	183.5
-	Dataset 2												Generations
	Penalty Points						Violations						
	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6	
CPP 8	20	20	20	20	20	1	0.2	1.2	1	0.4	1	8.5	155.6
CPP 9	40	40	20	20	40	1	0.3	1	0.7	0.9	0.3	10	176.1
CPP 10	100	100	20	20	100	1	0.1	0.5	2.4	0.9	0	13	178.5
CPP 11	20	20	40	40	20	1	0.8	1.7	0.2	0.1	1.2	8.1	163.1
CPP 12	242	60	10	9	242	1	0	1.1	1.4	1.4	0	11.7	190.6

CPPs 1, 2, and 3 show that the increase in the penalty points for constraints 1, 2, and 5 (C1, C2, and C5) reduces the number of violations for the corresponding constraints. It will also increase the violations for C3 and C4. Similarly, compared to CPPs 1 and 4, the penalty points for C3 and C4 decrease violations for C3 and C4 but an increase for C1, C2, and C5. In addition, the increase of penalty points for C1, C2, and C5 leads to an increase in the number of computation generations, which corresponds to a lower computation speed, but better results are obtained.

Now let us consider the CPPs from 1 to 3 and CPPs from 8 to 10. For CPPs from 1 to 3, the C1 and C2’s violation numbers decrease with the increase of C1 and C2’s penalty points. However, this phenomenon does not happen on CPPs 8 to 10, which are tested under a less crowded examiner schedule. This means the variation of the penalty points on the algorithm’s performance depends on the different datasets. Therefore, the penalty points should be designed appropriately, based on the density of the examiner’s schedule and the attendance list.

Compared to CPPs 5, 6, and 7, it can be concluded that an extremely high penalty point on a constraint will increase violation numbers for all other constraints. Therefore, the extremely high penalty points should be avoided during the algorithm design.

Penalty point combination of CPP 5 is the relatively best combination for dataset 1, which we have obtained in our

experiments. Therefore, this combination is chosen finally in the proposed algorithm and will be applied to 2 and 3.

4.2. Test 2 Various Improvements

In test 2, three different variants of the proposed model are tested and compared, where model 1 is the proposed model, model 2 is the model without pretraining approach, model 3 is the proposed model with single p_c and p_m , and model 4 is the single-GA model without the islands.

Model 1: 4-island structure, pre-training and multiple crossover and mutation probabilities.

Model 2: 4-island structure and multiple crossover and mutation probabilities (without pretraining).

Model 3: 4-island structure, pre-training and single crossover and mutation probabilities.

Model 4: Single-GA structure and pretraining.

Two datasets, dataset 1 and dataset 3, are used. For dataset 1, the total population size for each model is the same, which is equal to 120. For the 4-island model, the population size is divided equally into 4 parts; therefore, each island contains 30 individuals. The pretraining run 3 generations for dataset 1. For dataset 3, the total population size for each model is the same, which is equal to 80, and therefore each island contains 20 individuals. The pretraining runs 2 generations for dataset 3. For model 1 and model 2, the multi-crossover probability is 0.5, 0.6, 0.7, and 0.9, respectively, and the multi-mutation probability is 0.1, 0.2, 0.5, and 1,

respectively, for each island. For models 3 and 4, the p_c and p_m are 0.8 and 0.5, respectively. The details and reasons for the parameter decision of p_c and p_m will be discussed in section 5.3, test 3. The toleration for the stop condition is 30 iterations. Tables 4 and 5 show the testing results for dataset 1 and dataset 3, respectively.

Comparing with the results of models 1 and 2 for dataset 1, model 1 obtains a lower violation number than model 2 on C1, C2, and C3, and keeps the same violation numbers on C4, C5 and C6. However, compared to the generations computation of model 2, that of model 1 increases from 231.2 to 245.3. For dataset 3, compared to model 2, model 1 obtains a lower violation number than that of model 2 on C2, C3, and C4, and a higher violation number on C1. However, the generations computation of model 1 decreases from 150.2 to 130.8, which corresponds to a faster computation speed.

By comparing with model 3 and model 4, the significant result quality improvement shows that under the same p_c and p_m , the island model can help avoid premature convergence and improve the result quality.

4.3. Test 3: Various Crossover and Mutation Probabilities

In this test, four different combinations of p_c and p_m are applied to the four islands of the proposed algorithm to see how the different combinations will affect the performance of the algorithm. Datasets 1 and 3 are used. Except for the p_c and p_m , all other parameters are the same as those in Test 2. Table 6 shows the specific combinations of p_c and p_m for each island. Tables 7 and 8 show the comparison results.

The results show that combinations 1 and 3 produce better result quality compared to that of combinations 2 and 4, which means the multi-crossover and mutation probabilities, or a p_m around 0.5 would achieve a better result quality. This is because the session movement operation of the mutation could help the GA population reach a wider search region. However, the performance of combination 4 shows that an overlarge p_m , on the contrary, would reduce the result quality.

Moreover, the result on dataset 1 also shows that the result quality of combination 1 is better than that of combi-

Table 4. Result of test 2 Dataset 1.

-	Violations						Generations
	C1	C2	C3	C4	C5	C6	
Model 1	0.1	1	1.7	2.6	0.1	13.4	245.3
Model 2	0.3	1.3	1.9	2.6	0.1	13.3	231.2
Model 3	0.2	0.9	3.2	3	0.2	16.1	245.1
Model 4	0.8	2.3	5.1	7.9	0	21.7	116.7

Table 5. Result of test 2 Dataset 3.

-	Violations						Generations
	C1	C2	C3	C4	C5	C6	
Model 1	0.2	6	2.5	0.7	0	5.7	130.8
Model 2	0	7	3.5	0.8	0	6.2	150.2
Model 3	0	7	2.4	0.5	0	5.6	142.6
Model 4	0.1	6.5	5.4	1.6	0	13.7	89.7

Table 6. Test 3 combinations of p_c and p_m .

-	p_c				p_m			
	Island				Island			
	1	2	3	4	1	2	3	4
Combination 1	0.5	0.6	0.7	0.9	0.1	0.2	0.5	1
Combination 2	0.7	0.7	0.7	0.7	0.1	0.1	0.1	0.1
Combination 3	0.8	0.8	0.8	0.8	0.5	0.5	0.5	0.5
Combination 4	0.8	0.8	0.8	0.8	1	1	1	1

Table 7. Test 3 comparison results Dataset 1.

-	Violations						Generations
	C1	C2	C3	C4	C5	C6	
Combination 1	0.1	1	1.7	2.6	0.1	13.4	245.3
Combination 2	0.1	2.2	2.1	2.3	0	14.7	251.9
Combination 3	0.2	0.9	3.2	3	0.2	16.1	245.1
Combination 4	0.6	1.7	2.9	3.2	0.1	16.4	234.8

Table 8. Test 3 comparison results Dataset 3.

-	Violations						Generations
	C1	C2	C3	C4	C5	C6	
Combination 1	0.2	6	2.5	0.7	0	5.7	130.8
Combination 2	0.4	4.9	4.1	0.4	0	5.2	149.8
Combination 3	0	7	2.4	0.5	0	5.6	142.6
Combination 4	0.1	6.5	2.8	1.1	0	7.3	126.2

nation 3, which means the multi-crossover and mutation probability have a better performance on the increased data size.

CONCLUSION

This paper focuses on a specific examination timetabling problem that commonly occurs in Japanese universities and proposes a variant genetic algorithm approach to solve the problem. 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 island model and the initial population pre-training, which optimize partial objective problem in advance for several iterations, are applied. The proposed model is compared to its three variants that do not include the above improvements. The positive comparison results support the ideas that the initial population pre-training and the island model are effective approaches to improve the result quality of the proposed model. Additionally, the algorithm performs better with multi-crossover and mutation probabilities or large p_m .

However, to compromise the feasibility of the problem constraints, the constraint-based crossover operation cannot reach wide-enough search space during the computation. Therefore, the solution often got stuck into local optima. Even though the pre-training and island model could solve this problem in a way, the research on a better crossover approach for this special ETP is still needed. Although the memetic structure is not applied in this project owing to the long computation time, it is expected to apply the memetic structure [25, 38] and some other diversity maintenance approaches [39, 40] to the proposed algorithm to further improve the result quality in other projects. Finally, instead

of GA, other algorithms such as Simulated Annealing, ant colony approaches, Particle swarm optimization [41], and Greedy-Late Acceptance-Hyperheuristic [42] are expected to be implemented to this proposed ETP as well.

LIST OF ABBREVIATIONS

- CPP = Combinations of Penalty Points
- ETP = Examination Timetabling Problem
- GA = Genetic Algorithm
- OCC = Order Conflict Counts
- SA = Simulated Annealing
- TS = Tabu Search

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

No animals/humans were used for studies that are the basis of this research.

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

Not applicable.

FUNDING

None.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

Declared none.

SUPPLEMENTARY MATERIAL

Supplementary material is available on the publisher's website along with the published article.

REFERENCES

- [1] Qu, R.; Burke, E.; McCollum, B.; Merlot, L.; Lee, S. A survey of search methodologies and automated system development for examination timetabling. *J. Sched.*, **2008**, *12*(1), 55-89. <http://dx.doi.org/10.1007/s10951-008-0077-5>
- [2] Carter, M.; Laporte, G. Recent developments in practical examination time-tabling. *International conference on the practice and theory of automated timetabling*, Springer: Berlin, **1995**, 1153, 1-21.
- [3] Thompson, J.; Dowsland, K. Variants of simulated annealing for the examination timetabling problem. *Ann. Oper. Res.*, **1996**, *63*(1), 105-128. <http://dx.doi.org/10.1007/BF02601641>
- [4] Thompson, J.; Dowsland, K. A robust simulated annealing based examination timetabling system. *Comput. Oper. Res.*, **1998**, *25*(7-8), 637-648. [http://dx.doi.org/10.1016/S0305-0548\(97\)00101-9](http://dx.doi.org/10.1016/S0305-0548(97)00101-9)
- [5] Di Gaspero, L.; Schaerf, A. Tabu search techniques for examination timetabling. *International Conference on the Practice and Theory of Automated Timetabling*, Springer: Berlin, **2020**, pp. 104-117.
- [6] Abdullah, S.; Ahmadi, S.; Burke, E.; Dror, M.; McCollum, B. A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem. *J. Oper. Res. Soc.*, **2007**, *58*(11), 1494-1502. <http://dx.doi.org/10.1057/palgrave.jors.2602258>
- [7] Naji Azimi, Z. Hybrid heuristics for examination timetabling problem. *Appl. Math. Comput.*, **2005**, *163*(2), 705-733. <http://dx.doi.org/10.1016/j.amc.2003.10.061>
- [8] van Laarhoven, P.; Aarts, E. Simulated annealing. *Simulated annealing: Theory and applications*; Springer: Dordrecht, **1987**, pp. 7-15. http://dx.doi.org/10.1007/978-94-015-7744-1_2
- [9] Tuson, A.; Glover, F.; Laguna, M. Tabu search. *J. Oper. Res. Soc.*, **1999**, *50*(1), 106. <http://dx.doi.org/10.2307/3010402>
- [10] Galletly, J.E. An overview of genetic algorithms. *Kybernetes*, **1992**, *21*, 26-30. <http://dx.doi.org/10.1108/eb005943>
- [11] Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.*, **1997**, *1*(1), 53-66. <http://dx.doi.org/10.1109/4235.585892>
- [12] Eley, M. Ant algorithms for the exam timetabling problem *International Conference on the Practice and Theory of Automated Timetabling*, Springer: Berlin, **2006**, 3867, pp. 364-382.
- [13] Mandal, A.K.; Kahar, M.N.M. Solving examination timetabling problem using partial exam assignment with great deluge algorithm. *International Conference on Computer, Communications, and Control Technology*, **2015**, (I4CT), pp. 530-534. <http://dx.doi.org/10.1109/I4CT.2015.7219635>
- [14] Song, Y.; Wan, F.; Chen, X. An improved genetic algorithm for numerical function optimization. *Appl. Intell.*, **2019**, *49*(5), 1880-1902. <http://dx.doi.org/10.1007/s10489-018-1370-4>
- [15] Rozaimae, A.; Shafee, A.; Hadi, N.; Mohamed, M. A framework for university's final exam timetable allocation using genetic algorithm. *World Appl. Sci. J.*, **2017**, *35*(7), 1210-1215.
- [16] Shatnawi, A.; Fraiwan, M.; Al-Qahtani, H.S. Exam scheduling: A case study. *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, **2017**, pp. 137-142. <http://dx.doi.org/10.1109/ICACI.2017.7974498>
- [17] Dener, M.; Calp, M.H. Solving the exam scheduling problems in central exams with genetic algorithms. *arXiv preprint*, **2019**, 1902-01360.
- [18] Ezike, J.O.J.; Oyeleye, C.A.; Olabiyisi, S.O.; Omidiora, E.O. Development of a hybrid tabu search and genetic algorithms for the examination timetabling problem. *J. Sci. Technol. Res.*, **2020**, *2*(3), 127. <http://dx.doi.org/10.37933/nipes/2.3.2020.14>
- [19] Ngo, S.; Jaafar, J.; Aziz, I.; Nguyen, G.; Bui, A. Genetic algorithm for solving multi-objective optimization in examination timetabling problem. *Int. J. Emerging Technol. Learning*, **2021**, *16*(11), 4. <http://dx.doi.org/10.3991/ijet.v16i11.21017>
- [20] Kurdi, M. An effective new island model genetic algorithm for job shop scheduling problem. *Comput. Oper. Res.*, **2016**, *67*, 132-142. <http://dx.doi.org/10.1016/j.cor.2015.10.005>
- [21] Palomo-Romero, J.M.; Salas-Morera, L.; García-Hernández, L. An island model genetic algorithm for unequal area facility layout problems. *Expert Syst. Appl.*, **2017**, *68*, 151-162. <http://dx.doi.org/10.1016/j.eswa.2016.10.004>
- [22] Garcia-Hernandez, L.; Salas-Morera, L.; Carmona-Muñoz, C.; Garcia-Hernandez, J.A.; Salcedo-Sanz, S. A novel island model based on coral reefs optimization algorithm for solving the unequal area facility layout problem. *Eng. Appl. Artif. Intell.*, **2020**, *89*, 103445. <http://dx.doi.org/10.1016/j.engappai.2019.103445>
- [23] Gozali, A.; Kurniawan, B.; Weng, W.; Fujimura, S. Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy. *IEEE Trans. Electr. Electron. Eng.*, **2019**, *15*(3), 389-400. <http://dx.doi.org/10.1002/tee.23067>
- [24] Rezaeipannah, A.; Matoori, S.S.; Ahmadi, G. A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search. *Appl. Intell.*, **2021**, *51*(1), 467-492. <http://dx.doi.org/10.1007/s10489-020-01833-x>
- [25] Leite, N.; Fernandes, C.M.; Melicio, F.; Rosa, A.C. A cellular memetic algorithm for the examination timetabling problem. *Comput. Oper. Res.*, **2018**, *94*, 118-138. <http://dx.doi.org/10.1016/j.cor.2018.02.009>
- [26] Miller, B. L.; Goldberg, D. E. Genetic algorithms, tournament selection, and the effects of noise. *Complex Syst.*, **1995**, *9*(3), 193-212.
- [27] Yadav, S.L.; Sohal, A. A. Comparative study of different selection techniques in genetic algorithm. *Int. J. Eng. Sci. Mathematics*, **2017**, *6*(3), 174-180.
- [28] Du, H.; Wang, Z.; Zhan, W.; Guo, J. Elitism and distance strategy for selection of evolutionary algorithms. *IEEE Access*, **2018**, *6*, 44531-44541. <http://dx.doi.org/10.1109/ACCESS.2018.2861760>
- [29] Reed, P.M.; Minsker, B.S.; Goldberg, D.E. The practitioner's role in competent search and optimization using genetic algorithms. In *Bridging the Gap: Meeting the World's Water and Environmental Resources Challenges*, **2001**, 1-9.
- [30] Ahn, C.W.; Ramakrishna, R.S. Elitism-based compact genetic algorithms. *IEEE Trans. Evol. Comput.*, **2003**, *7*(4), 367-385. <http://dx.doi.org/10.1109/TEVC.2003.814633>
- [31] Schoenauer, M.; Xanthakis, S. Constrained GA optimization. In Proc. 5th International Conference on Genetic Algorithms, Morgan Kaufmann, **1993**, pp. 573-580.
- [32] Meng, Q.; Wu, J.; Ellis, J.; Kennedy, P.J. Dynamic island model based on spectral clustering in genetic algorithm *2017 International Joint Conference on Neural Networks (IJCNN)*, **2017**, pp. 1724-1731. <http://dx.doi.org/10.1109/IJCNN.2017.7966059>
- [33] Li, C.C.; Lin, C.H.; Liu, J.C. Parallel genetic algorithms on the graphics processing units using island model and simulated annealing. *Adv. Mech. Eng.*, **2017**, *9*(7), 1-14. <http://dx.doi.org/10.1177/1687814017707413>
- [34] Cantu-Paz, E. *Efficient and accurate parallel genetic algorithms*; Springer Science & Business Media: Newyork, **2000**.

- [35] Whitley, D.; Rana, S.; Heckendorn, R.B. The island model genetic algorithm: On separability, population size and convergence. *CIT J. Comput. Inf. Technol.*, **1999**, 7(1), 33-47.
- [36] Gozali, A.A.; Fujimura, S. DM-LIMGA: Dual migration localized island model genetic algorithm—a better diversity preserver island model. *Evol. Intell.*, **2019**, 12(4), 527-539.
<http://dx.doi.org/10.1007/s12065-019-00253-2>
- [37] Sun, X.; Chou, P.; Wu, C.C.; Chen, L.R. Quality-oriented study on mapping island model genetic algorithm onto CUDA GPU. *Symmetry (Basel)*, **2019**, 11(3), 318.
<http://dx.doi.org/10.3390/sym11030318>
- [38] Gong, G.; Deng, Q.; Chiong, R.; Gong, X.; Huang, H. An effective memetic algorithm for multi-objective job-shop scheduling. *Knowl. Base. Syst.*, **2019**, 182, 104840.
<http://dx.doi.org/10.1016/j.knosys.2019.07.011>
- [39] Hornby, G.S. ALPS: The age-layered population structure for reducing the problem of premature convergence. *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, **2006**, pp. 815-822.
<http://dx.doi.org/10.1145/1143997.1144142>
- [40] Hutter, M.; Legg, S. Fitness uniform optimization. *IEEE Trans. Evol. Comput.*, **2006**, 10(5), 568-589.
<http://dx.doi.org/10.1109/TEVC.2005.863127>
- [41] Souad. “A survey of particle swarm optimization techniques for solving university examination timetabling problem. *Artif. Intell. Rev.*, **2015**, 44(4), 537-546.
<http://dx.doi.org/10.1007/s10462-015-9437-7>
- [42] Muklason, A.; Bwananesia, P.C.; S. H., Y.T.; Angresti, N.D.; Supoyo, V.A. Automated examination timetabling optimization using greedy-late acceptance-hyperheuristic algorithm *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, **2018**, pp. 201-206.
<http://dx.doi.org/10.1109/ICECOS.2018.8605194>