



Available online to www.journal.unipdu.ac.id

Unipdu

S2-Accredited – SK No. 34/E/KPT/2018

Journal page is available to www.journal.unipdu.ac.id:8080/index.php/register



Multi-parent order crossover mechanism of genetic algorithm for minimizing violation of soft constraint on course timetabling problem

Ahmad Miftah Fajrin ^a, Chastine Fatichah ^b

^{a,b} Department of Informatic Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

email: ^a miftah.fr@gmail.com, ^b chastine@if.its.ac.id

ARTICLE INFO

Article history:

Received 26 July 2019
Revised 5 August 2019
Accepted 6 August 2019
Published 3 April 2020

Keywords:

course timetabling problem
Genetic Algorithm
multi-parent crossover
order crossover
soft constraint

IEEE style in citing this article:

A. M. Fajrin and C. Fatichah,
"Multi-parent order
crossover mechanism of
genetic algorithm for
minimizing violation of soft
constraint on course
timetabling problem,"
*Register: Jurnal Ilmiah
Teknologi Sistem Informasi*,
vol. 6, no. 1, pp. 43-51, 2020

ABSTRACT

A crossover operator is one of the critical procedures in genetic algorithms. It creates a new chromosome from the mating result to an extensive search space. In the course timetabling problem, the quality of the solution is evaluated based on the hard and soft constraints. The hard constraints need to be satisfied without violation while the soft constraints allow violation. In this research, a multi-parent crossover mechanism is used to modify the classical crossover and minimize the violation of soft constraints, in order to produce the right solution. Multi-parent order crossover mechanism tends to produce better chromosome and also prevent the genetic algorithm from being trapped in a local optimum. The experiment with 21 datasets shows that the multi-parent order crossover mechanism provides a better performance and fitness value than the classical with a zero fitness value or no violation occurred. It is noteworthy that the proposed method is effective to produce available course timetabling.

© 2020 Register: *Jurnal Ilmiah Teknologi Sistem Informasi* (Scientific Journal of Information System Technology) with CC BY license.

1. Introduction

Genetic algorithm is an optimized algorithm that adopts the process of natural selection to obtain the best collection of individuals in a population. It is a metaheuristic algorithm often used in optimizing various studies such as in creating timetables for courses in colleges. There are many kinds such as ant colony and swarm, however, the genetic is the best as it contains least risk of violations to the constraint [1].

Adaptive Genetic Algorithm is used to improve the effectiveness in solving problems associated with automated examination timetable in universities [2]. It was analysed using graph colouring, which proved its efficiency in solving problems associated with course timetabling [3]. However, the possibility of premature convergence and stagnation tended to occur while determining solutions [4]. Therefore, a study to develop the right genetic algorithms is still on-going, especially regarding timetabling courses.

Course timetabling does not only involve courses but also instructors in a particular space and time scheduling to fulfil several requirements or constraints, which is categorized into hard and soft. The hard constraints are not allowed to violate any rule, while the soft are not strictly prohibited, but

preferably fulfilled, therefore, the less violated it is, the better the schedule generated. Course timetabling problems are NP-hard since it is not easy to obtain optimal solutions for large algorithms such as scheduling because they have to meet the constraints [4].

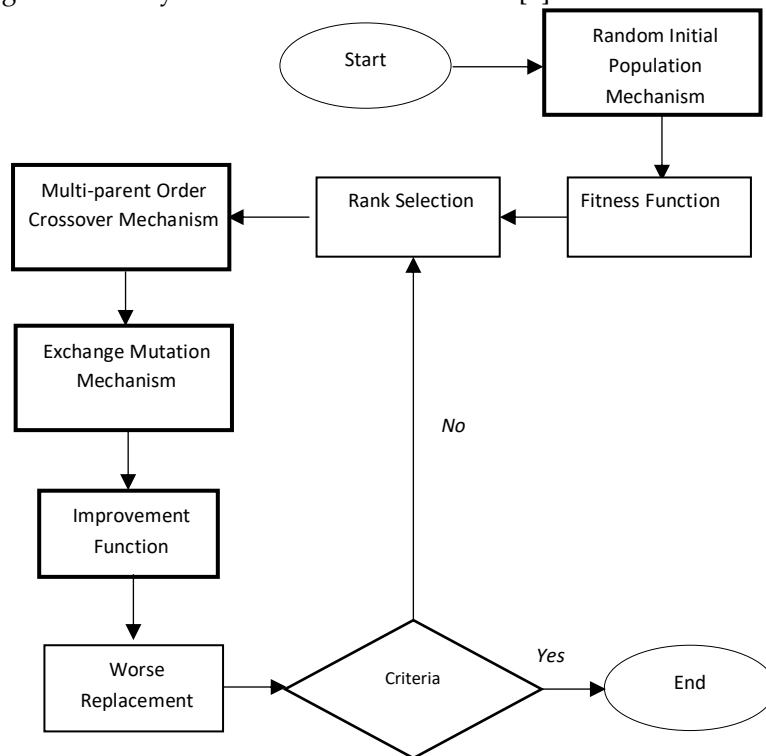


Figure 1. Optimization using genetic algorithm modification

Genetic algorithm (GA) consists of several operators that are important in obtaining good solutions, namely selection, crossover and mutation. From the technical point of view, the three operators are the basic development in order to produce the desired performance [5]. The crossover operator plays an important role in GA and is dubbed as its backbone, therefore, lots of study is still needed for its development [6]. The GA modification in the form of Directed-Based Crossover (DBX) is conducted to obtain a better population in each generation [7]. Study on its comparison such as Order Crossover, Partially Mapped Crossover and Cycle Crossover has been carried out, resulting order crossover to be the most effective algorithm in producing feasible scheduling solutions [8]. Modifications of OX in recursive GA have also been conducted to produce decently scheduled timetabling courses [9]. PMX is modified using a feasibility check during gene exchange based on the period to obtain a good offspring [10]. The feasibility check carried out during the exchange process generates a much faster and better performance than conducting it after the exchange process. Meanwhile, to increase diversity in GA, several studies on Multi-parent Crossover have been conducted producing solutions which prohibit similarity among them [11, 12]. The BLX- α multi-parent crossover was also developed to take advantage and enhance the exploration process in GA into a larger search space [13].

The modification of GA is conducted to avoid the possibility of premature convergence by increasing exploration in the solution search space. The use of multi-parent crossover makes chromosomes diverse and different from one another with an exploration increase. In the case of timetabling courses with soft constraints a solution with a small level of violation is very crucial. Therefore, this study aims to develop GA specifically for crossover operators using a multi-parent order mechanism. Its development is expected to produce high diversity that can explore solution space potency and reduce the level of violation on soft constraints.

2. Research Method

Courses Timetabling is a routine activity which occurs in each semester in universities. The problem associated with it includes NP-Hard class which is very difficult to solve using a classical algorithm [14]. A number of Events (courses, lecturers) needs to be allocated to a limited space and time to fulfill various

interests [15]. A set number of constraints need to be ensured in solving this timetabling problem [16]. In this study, the ITC-2007 dataset with curriculum-based course timetabling (CB-CTT) type is used to describe the number of constraints.

The hard constraint in the dataset consists of:

- H.1. **Lectures:** All lectures need to be scheduled in the available timeslot (day, period).
- H.2. **Room occupancy:** Two lectures may not be placed in the same room and period.
- H.3. **Conflicts:** Courses in one curriculum group or taught by the same lecturer need to be scheduled at different periods.
- H.4. **Availability:** Courses should have a specific period that cannot be placed in another.

While soft constraints consist of.

- S.1. **Room Capacity:** The number of students taking the course need to be less or equal to the room capacities. Every student without a seat is penalized.
- S.2. **Room stability:** All lectures in one course should be placed in the same room. Those with different rooms should be penalized.
- S.3. **Minimum Working Days:** Lectures from each course need to be distributed to the assigned minimum number of days. Every shortfall is subjected to a penalty of five points.
- S.4. **Curriculum Compactness:** Lectures in one curriculum should corporate. Those that fail, are subject to a penalty of two points.

3. Research Methodology

In this research, GA is used to solve problems associated with course timetabling. Genetic Algorithm is an excellent search algorithm for solving problems based on the principles of Darwinian evolutionary biology, and is currently used in various studies, including scheduling [17, 18]. Some operators are part of the main contributors to its addition to the multi-parent order crossover mechanism which comprises of random initial population mechanism, exchange mutation mechanism, and improvement function. The system design is shown in Figure 1.

3.1. Dataset

In this research, the dataset was obtained from the International Timetabling Competition (ITC-2007), and it consists of several problems. This study uses CB-CTT data type which comprises of 21 dataset files, with the aim of reducing violations of soft constraints to the minimum and preventing the hard constraints. CB-CTT data has several attributes, including:

1. Courses: Each course consists of the number of lectures, and students set to be scheduled in a certain period.
2. Rooms: Each room has a chair capacity.
3. Days: Number of days available.
4. Period Per Day: Number of periods in one day.
5. Curricula: A group of subjects.
6. Min-Working-Days: The minimum number of days in a course that need to be available.

3.2. Chromosome representation

<i>Course</i>	<i>Day</i>	<i>Period</i>	<i>Room</i>
Co1	D1	P1	R1
Co2	D1	P1	R2
...
Co9	D4	P4	R4

Figure 2. Chromosome representation

Chromosome encoding is conducted to determine the success of the process of GA before an initial population is carried out. These effects are not only based on the efficiency, but also on the speed and quality of getting the results of a solution to the problem.

In this research, chromosome represents course, day, period, and room. Its length depends on the number of courses, because the greater it is, the bigger the length. For further explanation, see Figure 2.

3.3. Fitness function

The fitness function is used as a parameter to candidates irrespective of the problem [19]. It also depends on the specifications of the problem that needs to be solved. In terms of scheduling, the quality of a chromosome is measured by the number of constraints violated. Each chromosome is checked for validity by analysing the hard and soft constraints. In hard constraints, all cases involving no violations, while in soft constraint, it may occur, but should be as small as possible. Every violated soft constraint is given a penalty value.

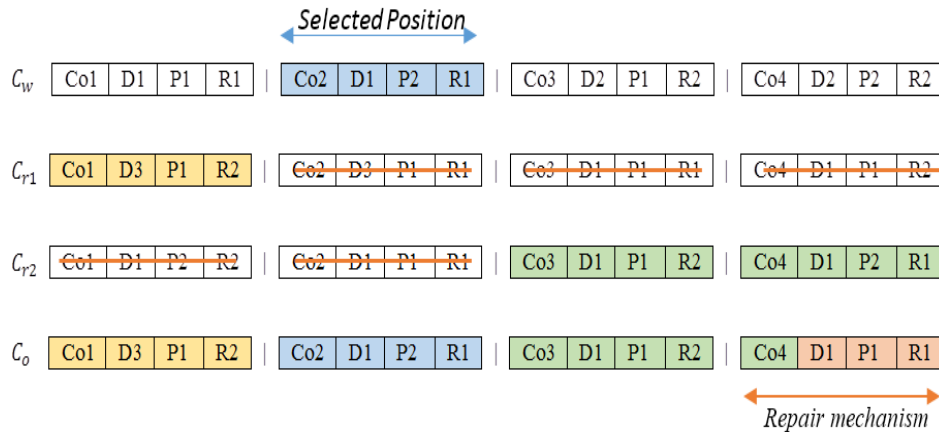
3.4. Rank selection

Rank Selection (RS) is a technique that uses explorative concepts to prevent convergence. It sorts the population based on fitness value and gives it a rank, with the highest first and the smallest last. In this research, three parents would be chosen and used in the crossover process. Two would be selected from the RS while another based on the chromosome with the greatest fitness. This is conducted to maintain the diversity of the population and explore a wider solution space. The wider the search space, the easier it is for GA to obtain the optimum global solution.

3.5. Multi-parent order crossover mechanism

The existence of additional parent in the crossover process reduces the pressure in the selection process and increases the diversity of the population. The higher the level of diversity the smaller the occurrence of incest (mating between similar genes). This multi-parent order crossover mechanism consists of three parent resulted from the previous selection process. Multi-parent OX mechanism works by choosing the gene sequence on the best chromosome in a random position to be filled in a new offspring.

An additional chromosome is then inserted into one parent since it enhances the exploration process into a wider search space [13]. Next, the empty block of offspring on the left side is sequentially filled with the first chromosome genes RS chose. To fill in the empty blocks on the right, genes from the second choice chromosome and from the RS is utilized. Assuming the exchange process is restricted by a violation on the hard constraint, the repair mechanism is carried out. An illustration of the Multi-parent OX mechanism operation is seen in Figure 3.



Note:

- C_w : Chromosome with lower fitness value
- C_{r1} : First chromosome from rank selection
- C_{r2} : **Second chromosome from rank selection**
- C_o : **Offspring**

Figure 3. Multi-parent order crossover mechanism process

3.6. Exchange mutation mechanism

The existence of mutation helps to improve the quality of the offspring previously formed. The role of mutation is to make small changes to the chromosomes to meet the predetermined constraints. There are several types of mutations namely exchange (EX), insertion (INS), and swap (SWP). The results of studies on the mutations showed that EX was the best mutation operator compared to INS and SWP [20]. The exchange (EX) mutation operation was used due to its effectiveness which allows good results. EX works by selecting and exchanging the position of two pairs of genes during which a violation on the hard constraint, leads to execution of the repair mechanism. An illustration of the operation (EX) mechanism is seen in Figure 4.

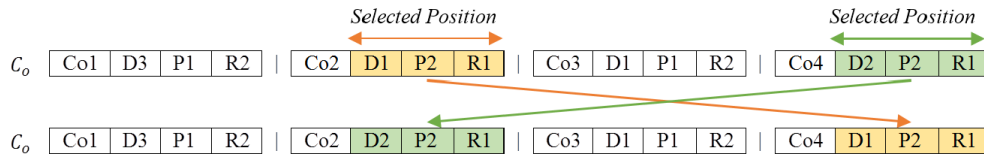


Figure 4. Exchange mutation mechanism process

Table 1: Improvement function

No.	Description
IF_1	When the chosen course and number of student is greater than the room capacity, move it randomly.
IF_2	When the chosen course is a specific lecture in different rooms move to the same room with day and room available
IF_3	When the chosen course are not distinct in the same day move it to the same curriculum with others.

3.7. Improvement function

Offspring may not be optimal, therefore at this stage, the improvement function process is inspired by the neighborhood function carried out by previous study [21] Repair function is performed on an infeasible chromosome to hasten the resulting solution thereby, making it feasible, by minimizing the constraint. The improvement function in this study consists of three stages, as follows IF_1, IF_2, IF_3 . More details of improvement function is seen in Table 1.

3.8. Worse replacement

After passing through a mutation process, the offspring formed is included in the population. One of the replacement operations is the Worst Replacement (WR) operation and it has been applied in several studies, especially in scheduling problems [14]. It flows when the offspring is formed, by replacing the individuals in the population with high violation or fitness value. This operation produces the best individuals collection in the population.

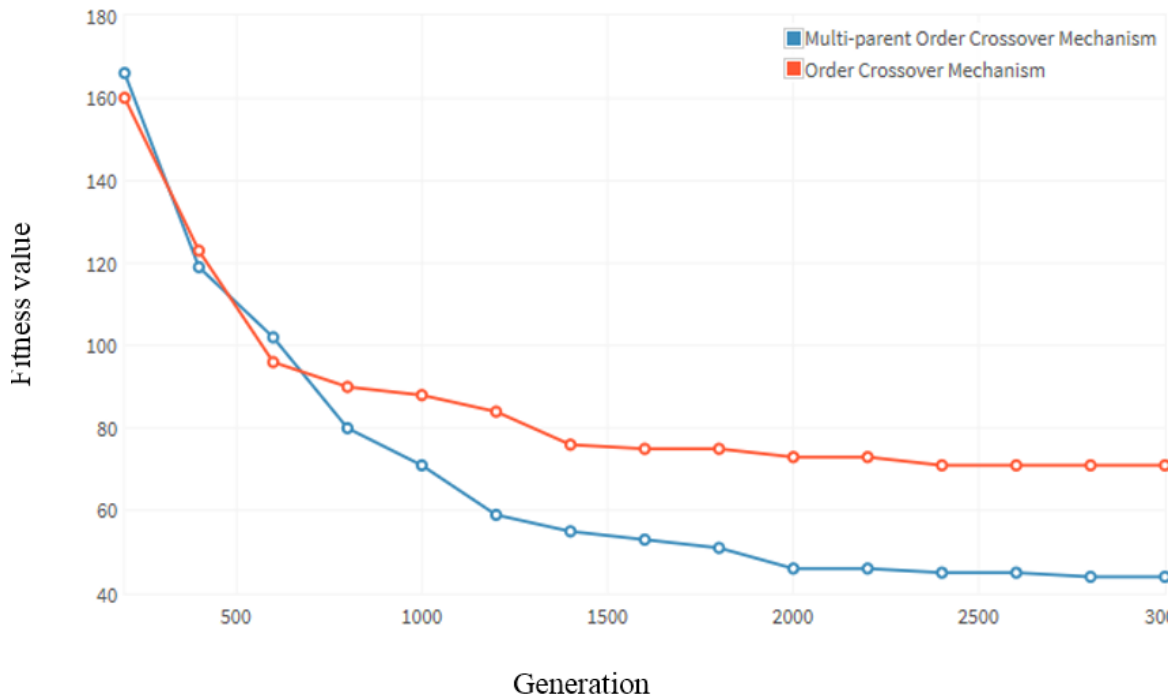


Figure 5. Changes in fitness values (2nd dataset)

4. Result and Discussion

Tests in this study are divided into two scenarios. The first comprises of classical GA with the proposed scenario from other literatures.

4.1. Comparison of fitness value between classical GA and proposed GA

At this stage, a fitness value test was performed between classical GA and the proposed method. The higher the soft constraint violations, the greater the fitness value was generated, with the ITC-2007 dataset, comprising of 21 data utilized.

Table 2. The results of the best fitness values between the classical GA and the proposed GA method based on the type of soft constraint

Dataset	Classical GA					Proposed GA				
	S.1	S.2	S.3	S.4	Total	S.1	S.2	S.3	S.4	Total
1	4	0	0	0	4	4	0	0	0	4
2	0	4	35	32	71	0	8	10	26	44
3	0	4	15	64	83	0	2	0	64	66
4	0	3	5	62	70	0	5	15	48	68
5	0	0	85	174	261	0	19	75	152	246
6	0	7	25	50	82	0	8	5	30	43
7	0	12	25	50	87	0	11	15	38	64
8	0	5	10	64	79	0	1	30	42	73
9	0	6	20	98	124	0	2	5	100	107
10	0	8	15	26	49	0	7	5	36	48
11	0	0	0	0	0	0	0	0	0	0
12	0	3	60	228	291	0	6	80	168	254
13	0	2	15	76	93	0	5	15	68	88
14	0	4	0	64	68	0	4	22	25	51
15	0	0	10	70	80	0	2	5	58	65
16	0	5	10	44	59	0	5	15	38	58
17	0	9	10	72	91	0	5	10	58	73
18	0	8	15	60	83	0	6	15	44	65
19	0	9	0	70	79	0	8	5	48	61
20	0	9	20	54	83	0	8	0	38	46
21	0	3	10	84	97	0	7	10	70	87

Table 3. Comparison of the average fitness values between proposed classical GA and GA (30 trials).

Dataset	Classical GA		Proposed GA	
	Best Fitness	Average Fitness	Best Fitness	Average Fitness
1	4	4.77	4	4.37
2	71	98.41	44	63.47
3	83	120	66	81.03
4	70	89.6	68	89.6
5	261	343.96	246	304.1
6	82	94.73	43	74.47
7	87	107.8	64	87.47
8	79	100.7	73	93.36
9	124	132.9	107	117.43
10	49	78.33	48	71.1
11	0	0	0	0
12	291	317.73	254	297.8
13	93	113.83	88	106.93
14	68	103.37	51	89.53
15	80	101.9	65	82.3
16	59	87.57	58	74.97
17	91	111.23	73	91.37
18	83	109	65	84.03
19	79	99.52	61	80.8
20	83	112.2	46	82.16
21	97	132.7	87	108.77

Figure 5 shows the flow process of changing the fitness value from each generation using classical GA, using sample from the 2nd dataset. Two algorithms tend to experience a change in fitness value because they pass through the rank selection process which select a new parent, using a multi-parent order crossover mechanism and mutated through exchange. The result of the mutation was improved

using the improvement function, while the offspring replaces the worst chromosomes in the population. The process stopped when the fitness value reached 0, also known as its optimal global or generates 3000.

As seen in the performance graph in Figure 5, the fitness value in the proposed GA has decreased more than the classical. In the generation between 300 and 3000 for classical GA the value of fitness only decreased by 150 while in the proposed method the value decreased by 80. In the classical generation between 2500 and 3000, they were stagnant or trapped at the local optimum, however, in the proposed method it was still going down. This indicates that the use of multi-parent order crossover mechanism increased exploration potency in the search space, with a higher possibility of obtaining optimal solutions.

Table 2 presents the number of violations of soft constraints in classical GA testing and the proposed method. Each algorithm obtains a fitness value based on the type of soft constraint violated. In the 11th dataset, each algorithm which was able to reach the global optimum successfully satisfied soft constraints with no violations or, in other words, had a fitness value of 0. In the 12th dataset, both algorithms consisted of the highest fitness value due to high violations which occurred in the S.4 constraint with a fitness value of 291 for the classical GA and 254 for the proposed. In all datasets tested, the proposed GA had better performance than the class due to its smaller fitness value.

Table 4. Comparison of fitness values with other GA literature (30 trials).

Dataset	Akkan & Gülcü	Proposed GA
1	5.07	4.37
2	81.43	63.47
3	98.77	81.03
4	46.5	89.6
5	367.8	304.1
6	75.33	74.47
7	47.47	87.47
8	58.1	93.36
9	118.83	117.43
10	39.57	71.1
11	0	0
12	367.2	297.8
13	87.6	106.93
14	68.93	89.53
15	96.37	82.3
16	61.23	74.97
17	100.77	91.37
18	90.8	84.03
19	82.63	80.8
20	62.8	82.16
21	132.33	108.77

Table 3 presents the results of testing the classical and proposed GA method based on the average fitness value carried out 30 times on each dataset. The table also presents the best and average values for each algorithm which is tested 30 times. It is seen in Table 3 that the best value in the proposed method produces a smaller fitness value than classical GA in all datasets. This indicated that the probability of getting an optimal solution on the use of the proposed method is higher. It was also concluded that the use of multi-parent in crossover increases exploration potency in the search space and provided effective results to reduce violations of soft constraints.

4.2. Comparison of GA method from other literature with proposed GA.

At this stage, the fitness value between the proposed GA method and other literature is performed [10]. The literature presents the results of the algorithm using a hybrid GA to solve course scheduling problems. The method proposed with other literature used the same dataset, ITC-2007 with CB-CTT track type consisting of 21 dataset which was tested for 30 times.

Table 4 shows the performance of each algorithm based on violations of soft constraints. In some datasets the proposed GA method had a smaller fitness value than other literature. In the 11th dataset each algorithm obtained an average fitness value of 0, which meant that the solution had acquired the global optimum without any violations. In the 12th dataset, GA from Akkan & Gülcü was 367.2 while

the proposed method was 297.8 with a significant difference of 69.4. This showed that the results of the proposed method had been able to achieve good results compared to the same GA in the dataset based on the average fitness value test that has been conducted.

5. Conclusions

Based on the test results and discussion, it can be concluded that the use of multi-parent on crossover orders increases the exploration potency in the search space in order to reduce the level of violation of soft constraints rather than using traditional crossovers. It can be seen from the test result using 21 ITC-2007 dataset with CB-CTT tracks type values in 21 smaller datasets smaller. In one dataset, the proposed method obtains a fitness value of 0 or no violations, which proves its effectiveness in producing a reasonable course schedule. The number of constraints also affects the performance of genetic algorithms. The higher the constraints, the more difficult for the algorithm to determine optimal results. Conversely, the smaller the violations that occur in soft constraints, the better course timetabling can be produced.

6. References

- [1] J. S. Gill and G. k. Kaur, "Optimizing the Energy Efficiency of the Modern Data Centers using Ant Colony Optimization," *Indian Journal of Science and Technology*, vol. 9, no. S1, 2016.
- [2] S. Innet, "A novel approach of genetic algorithm for solving examination timetabling problems: A case study of Thai Universities," in *13th International Symposium on Communications and Information Technologies (ISCIT)*, Surat Thani, Thailand, 2013.
- [3] M. Assi, B. Halawi and R. A. Haraty, "Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem," *Procedia Computer Science*, vol. 126, no. 2018, pp. 899-906, 2018.
- [4] H. Babaei, J. Karimpour and A. Hadidi, "A survey of approaches for university course timetabling problem," *Computers & Industrial Engineering*, vol. 86, no. August, pp. 43-59, 2015.
- [5] N. Metawa and M. K. H. M. Elhoseny, "Genetic algorithm based model for optimizing bank lending decisions," *Expert Systems with Applications*, vol. 80, no. September, pp. 75-82, 2017.
- [6] R. Kumar, G. Gopal and R. Kumar, "Novel Crossover Operator for Genetic Algorithm for Permutation Problems," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, no. 2, pp. 252-258, 2013.
- [7] Y. Song, F. Wang and X. Chen, "An improved genetic algorithm for numerical function optimization," *Applied Intelligence*, vol. 49, no. 5, pp. 1880-1902, 2018.
- [8] W. Chinnasri, S. Krootjohn and N. Sureerattanan, "Performance comparison of Genetic Algorithm's crossover operators on University Course Timetabling Problem," in *8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, Seoul, South Korea, 2012.
- [9] S. S. A. Alves, S. A. F. Oliveira and A. R. R. Neto, "A Recursive Genetic Algorithm-Based Approach for Educational Timetabling Problems," in *Designing with Computational Intelligence*, Cham, pp. 161-175, 2017.
- [10] C. Akkan and A. Gülcü, "A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem," *Computers & Operations Research*, vol. 90, no. February, pp. 22-32, 2018.
- [11] M. Z. Ali, N. H. Awad, P. N. Suganthan, A. M. Shatnawi and R. G. Reynolds, "An improved class of real-coded Genetic Algorithms for numerical optimization," *Neurocomputing*, vol. 275, no. January, pp. 155-166, 2018.
- [12] X. Zhao, J. Chen, R. Li, D. Gong and X. Li, "An Orthogonal Genetic Algorithm with Multi-parent Multi-point Crossover for Knapsack Problem," in *International Conference on Bio-Inspired Computing: Theories and Applications BIC-TA 2018: Bio-inspired Computing: Theories and Applications*, Singapore, 2018.

- [13] D. Kristianto, C. Faticah, B. Amaliah and K. Sambodho, "Prediction of Wave-induced Liquefaction using Artificial Neural Network and Wide Genetic Algorithm," *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, vol. 8, no. 1, pp. 1-10, 2017.
- [14] M. S. Kohshori, D. Zeynolabedini, M. S. Liri and L. Jadidi, "Multi Population Hybrid Genetic Algorithms for University Course Timetabling Problem," *I.J. Information Technology and Computer Science*, vol. 4, no. 6, pp. 1-11, 2012.
- [15] J. B. Matias, A. C. Fajardo and R. M. Medina, "A fair course timetabling using genetic algorithm with guided search technique," in *5th International Conference on Business and Industrial Research (ICBIR)*, Bangkok, Thailand, 2018.
- [16] T. Song, S. Liu, X. Tang, X. Peng and M. Chen, "An iterated local search algorithm for the University Course Timetabling Problem," *Applied Soft Computing*, vol. 68, no. July, pp. 597-608, 2018.
- [17] S. Jaengchuea and D. Lohpetch, "A hybrid genetic algorithm with local search and tabu search approaches for solving the post enrolment based course timetabling problem: Outperforming guided search genetic algorithm," in *7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Chiang Mai, Thailand, 2015.
- [18] H. Sani and M. Yabo, "Solving timetabling problems using genetic algorithm technique," *Int. J. Comput. Appl.*, vol. 134, no. 15, pp. 33-38, 2016.
- [19] A. Hideg, "Comparing genetic operators for the timetabling problem," in *13th International Symposium on Applied Machine Intelligence and Informatics (SAMi)*, Herl'any, Slovakia, 2015.
- [20] S. Karakatič and ViliPodgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem," *Applied Soft Computing*, vol. 27, no. February, pp. 519-532, 2015.
- [21] S. Abdullah, H. Turabieh, B. McCollum and P. McMullan, "A hybrid metaheuristic approach to the university course timetabling problem," *Journal of Heuristics*, vol. 18, p. 1-23, 2012.