A Hybrid Genetic Algorithm for Course Scheduling and Teaching Workload Management

Junrie B. Matias
Technological Institute of the Philippines, Graduate Programs, Quezon City, Philippines
jbmatias@carsu.edu.ph

Arnel C. Fajardo
Technological Institute of the Philippines, Graduate Programs, Quezon City, Philippines
acfajardo2011@gmail.com

Ruji P. Medina
Technological Institute of the Philippines, Graduate Programs, Quezon City, Philippines
ruji.medina@tip.edu.ph

***Abstract*— Course scheduling is a common problem of all higher educational institutions in several developing countries. Most of these institutions experience a shortage of resources such as teachers and infrastructures. Having small time to recruit, institutions likely to resort in employing beginners to avoid overloading and to fill vacancies immediately. However, directly reassigning classes from senior teacher to the new teacher may lead to unmatched skills required to teach a particular course and more constraints violations. This study presents a hybrid genetic algorithm for course scheduling and teaching workload management. The genetic algorithm is employed with four neighboring operators identified using self-adaptive mechanism during the search process. A data structure of unused resources is used to guide the operators to unused periods. Repair operator is applied to increase the optimality of the solutions. Results show that the proposed algorithm generates feasible and optimized workloads and timetables. The automated system can relieve the decision-makers from the burden of tedious and time-consuming scheduling task in every semester. Hiring additional teaching staff and managing workloads are now much more convenient.**

Keywords*— *genetic algorithm, course timetabling, guided search, self-adaptive mechanism, workload management.

I. INTRODUCTION

The combined effect of decreasing number of enrollees and teacher attrition, staffing and retaining high-quality teaching personnel is very important for every higher educational institution [1]. To manage the shortage of teachers together with the limited budget, many educational institutions in several developing countries have employed an inexperienced and a non-tenured teaching staff often known as 'contract teachers' [2]. Inexperienced teachers are given smaller salaries and benefits compared to tenured personnel [3]. Contract teachers are mostly beginners that are also more likely to be given the most difficult classes to teach [1]. These circumstances will be most likely to happen when enrolment is near to the end and during the distribution of workloads. Most of the time, the tenured and well-experienced teachers are expected to teach the most difficult and complicated classes. However, senior teachers are also given an administrative task or non-teaching related workloads such as meetings, paper works, and many other responsibilities. Thus, teachers having administrative duties are given lesser teaching loads. Also, many teachers are leaving for some reasons such as compensation, teacher preparation and support, and teaching conditions [4]. With a sudden shortage of teachers, and have limited budgets, and with a short time to recruit, institutions will likely to resort in employing beginners to avoid overloading and to fill vacant teaching positions immediately. However, directly reassigning classes which are previously assigned from senior teacher to a new contract teacher likely will result in unmatched skills that are required for a particular course, conflicts, and other preferences are not satisfied. At present, many higher institutions are still constructing course schedules and assigning teaching loads manually [5]. The manually created schedules or timetables are stored in a database system for viewing, analysis and management purposes. Nevertheless, reconfiguring or making new class timetables and managing workloads is very difficult and will take a significant amount of time due to the variations in constraints and objectives across the different institutions [6].

Literature presents a frequently used methods and new algorithms to solve the educational timetabling problems. These includes but not limited to simulated annealing [7], evolutionary algorithms [8]–[10], tabu search [11], integer programming and constraint programming [12]. However, it is still challenging to solve or to find a perfect solution using conventional techniques [13].

More recently, there has been much research into hybridized methods which draw on two or more metaheuristics. Meta-heuristic methods are categorized as population-based approaches that are nature-inspired algorithms and single-solution approaches or a trajectory method that encompassed single point search-based meta-heuristics. Hybridizations have attracted more attention and represent improved quality and performance in solving NP-complete problems [14]. To enhance the algorithms fundamental decision-making skills useful characteristics of various methods are combined and applied simultaneously to solve problems [14]. Hence, these methods and techniques have its weaknesses, combining these methods can eliminate the gap, effectively exploit the strength of each algorithm and leads to a superior hybrid approach. Accordingly, this approach presents a promising direction for the development of descent methods and strategies that deal with large-scale optimization problems [15].

On the other hand, genetic algorithms are inspired by a natural evolutionary mechanism which is based on the following three steps: (1) selection, (2) regeneration and (3) replacement. GAs are known to be adaptive and flexible and has been demonstrated capabilities in solving various complex real-world problems [6]. However, genetic algorithms take large runtimes to produce a solution [16]. Moreover, several hybridization and modification are based on GA fundamental operations, aimed at improving the performance of the algorithms like in the works of [17]–[21] and [22].

In this study, we presented a hybrid genetic algorithm GA for course scheduling and teaching workload management for Higher Education Intuitions in the

978-1-5386-7767-4/18/$31.00 ©2018 IEEE

Philippines. The timetable is automatically generated by the proposed method which reduces the number of personnel involved in creating schedules and workloads. The proposed method can relieve the decision-makers from the burden of tedious and time-consuming scheduling task to be done in every semester.

After the timetable is created using the hybrid GA, a recommender system is developed to check the quality of the teaching loads base on the hard and soft constraints violations and if needed will recommend additional teaching staff. The hiring of additional teaching staff is efficient since the recommender system already identifies the possible workloads of the new potential teaching staff and hiring of unmatched skills can be avoided.

Lastly, the proposed algorithm uses a data structure similar to [23] containing the vacant pair of room and periods. The data structure will act as a memory that will guide the searching process to a feasible room and timeslot. Further, neighborhood structures are then performed to improve the solution, and these neighborhood structures are self-adaptively selected [24] during the searching process.

II. Problem Description

Timetables are configured to have 30 minutes to one hour every slot, and if a course needs three hours of lectures, then it will have six 30 minutes timeslots and distributed in a week. In this problem, students are blocked according to their degree or program has taken. Each class is arranged to 35 students or less per class, and the maximum should be 50 since 7m x 9m classroom is the standard size in the Philippines.

The following entities are considered in this work:

- Set of a period that is a pair of a day and a timeslot
- Set of group students of the same program or degree.
- Set of teacher that teaches a particular course.
- Set of a course which has a minimum number of days to assign in a week with a number of lectures.
- Set of lecture rooms and laboratory rooms.

A. Constraints

When thinking of a real timetable, there is a need to consider several types of conditions to satisfy, though these conditions are not of the same magnitude. It has distinguished these conditions as hard and soft constraints. Hard constraints cannot be violated physically so that solutions become feasible [13]. The hard constraints are based in [25] which are defined for this are the following:

h1. All the lectures of a course must be scheduled and they must be assigned to distinct time periods.

h2. Each room is allowed to accommodate at most one lecture per period.

h3. Lectures of courses in the same curriculum, or taught by the same teacher, must be scheduled in different periods.

Soft constraints are helpful but may be violated if there is no other feasible solution available. In this study, the soft constraints considered are the following:

s1. Teaching personnel priority periods are considered.

s2. Teaching personnel possibly have unavailable periods.

s3. Credit units or loads that are more than the maximum load are considered overload.

s4. Teaching personnel has a maximum number of preparations which is four.

s5. Teaching personnel has a maximum teaching hour in a classroom which is 4.

s6. Students should be vacant for at least one period in a day.

s7. Rooms might have unavailable periods.

s8. Every teacher that doesn't meet the maximum load or must have a full-time workload is considered as under-load.

s9. A penalty is assigned to each student who cannot take a seat in the assigned room for the lecture.

B. Objective Function

The objective is to find a feasible timetable while minimizing soft constraints penalties. The costs of every solution in the population are calculated on how it satisfies the constraints. The penalty function used to calculate the constraints violation might be formulated as:

$$f(x) = \sum_{h \in SH} f(x,h) + \sum_{s \in SC} f(x,s) \quad (1)$$

where $f(x)$ represents the objective function, $x \in X$ is the set of all candidate solutions, SH is the set of hard constraints, and $f(x,h)$ returns the total penalty of hard constraint h and SC is the set of soft constraints, and $f(x,s)$ returns the total penalty of soft constraint s [26]. Moreover, all constraints are given one penalty per violation.

III. PROPOSED METHOD

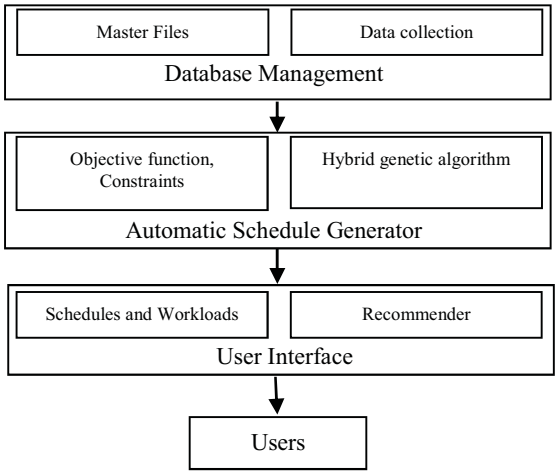

Fig. 1. The configuration of a course timetabling system

Adapted from [27], Fig. 1 describes the configuration of the proposed study, which contains three modules, namely: the database module, schedule generator module, and the user interface module. The database module includes objects such as subjects, teachers, and classrooms and laboratories. The scheduler modules include an objective function, constraints, and the proposed hybrid genetic algorithms. User interface modules are essential, especially in the implementation process. User interfaces modules will enable a user to configure constraints, manage the timetable generated by the system, and performed what if analyses. The recommendations are displayed in the user interface that includes hiring additional staffs, skills required, and subjects that are possibly be assigned to the new teaching personnel.

A. The Hybrid Genetic Algorithm

The flow of the proposed hybrid genetic algorithm is shown in Fig. 2. The proposed algorithm uses a data structure similar to [23] containing the vacant pair of room and periods. The data structure will act as a memory that will guide the searching process to a feasible room and timeslot. Further, neighborhood structures are then performed to improve the solution, and these neighborhood structures are self-adaptively selected [24] during the searching process. If a solution is not feasible, a repair operator is executed to fix every infeasible gene.

1. Randomly initialize a population of solution
2. Evaluate each individual in the population
3. While the termination condition is not reached do
4. Select parents through tournament selection
5. Perform crossover and mutation operators
6. Collect feasible rooms and periods
7. Apply self-adaptively selected neighborhood structure
8. Apply repair to infeasible individuals
9. Evaluate each individual in the population
10. End while

Fig. 2. The proposed hybrid Genetic Algorithm

1) Chromosome Representation: We use an array of integer values to represent a class in a chromosome. A shown in Fig. 3, each class consists of five elements which include a course and a teacher, student section, room, and periods.

Class 1					Class 2					…	Class n				
1	1	0	1	1	2	1	2	3	4	…	0	1	1	4	1

Fig. 3. Chromosome representation

2) Fitness Function: The hard constraints and soft constraints are given penalties whenever these types of constraints are violated. It can be represented as:

$$\text{fitness} = \frac{1}{1+f(x)} \quad (2)$$

where $f(x)$ is the total cost for violating hard and soft constraints from equation (1). The objective is to minimize the number of soft constraints violation in a feasible solution that generated by hard constraints.

3) Initialization: The initial population is usually generated randomly across the search space [28]. However, in this work, we had selected teachers based on their prepared subjects and allowed teaching workloads. The teachers having the largest available or unsigned workloads are always scheduled for a class first.

4) The Data Structure of Unused Rooms and Periods: Adapted from [23], every individual in the population has its own list or rooms and periods or data structure. The periods are a pair of days and timeslots. After an individual is evolved using the genetic operators, the unused pair of rooms and period are collected and stored it in the data structure. The data structure will act as a memory to guide the searching process to unused pair of rooms and periods.

5) Genetic Operators: Adopted from [23], we used a multi-point crossover of two parent individuals P1 and P2. P1 is selected through a tournament section and P2 is selected from the population with probability pc. After the two parents are selected, the following steps are performed:

Step 1. Create an empty offspring;
Step 2. For each gene of the offspring;
Step 3. If the penalty value of gene of P1 is less than the penalty value of gene of P2;
Step 4. Then assign the gene allocation to the offspring of P1;
Step 5. Otherwise, assign the gene allocation to the offspring of P2;
Step 6. Go to step 1 until all genes of P1 and P2 are all visited once.

The crossover operator will produce one offspring, however, it will contain the good genes from either of the two parents. Moreover, a two-gene swap from two parent individuals P1 and P2 is implemented with a low probability pm. The parents were selected through tournament selection and will produce two offspring.

When new offspring individuals are created using genetic operators, they are added to the parent population and a self-adaptively selected neighborhood structure is employed to all individuals in the population. The neighborhood structure are performed in the following steps:

Step 1. Select an $Individual$ in the population;
Step 2. Select $Neighborhood\ Structure$ from $Neighborhood\ List$;
Step 3. For all infeasible gene in the $Individual$ apply the selected $Neighborhood\ Structure$;
Step 4. If the neighborhood operator improve the quality of the $Individual$, keep the changes;
Step 5. Otherwise, discard the changes made by the neighborhood operator in the $Individual$.

On the other hand, if an individual in the new population is not feasible, a repair mechanism is employed. The infeasible genes and allele are replaced using the resources from the data structure. The data structure will serve as a memory of unused resources; however, if the required

resources such as rooms and periods are not found in the data structure, these are generated randomly.

Lastly, the population is sorted from the fittest to the weakest and to maintain the required number of individual in the population. The weakest individuals are deleted.

6) Self-Adaptive Mechanism: The self-adaptive mechanism is based on four neighborhood structures, defined as:

- An operator that select two classes randomly and swap periods.
- An operator that select a class randomly and moves it to a new feasible period from the data structure.
- An operator that selects four classes. The periods of first two classes are replaced by the other two classes and then that two class will be move to available periods from the data structure.
- An operator that selects two class randomly and moves them to available periods from the data structure.

Each individual has its list of neighborhood structures that is randomly generated which is adapted from the works [24], [29]. During the search process, a neighbor operator is taken out from the list and applied it to the current solution. The operators that are successfully improved the solution are stored in the winning list. The list of neighborhood structures is regenerated once it becomes empty, with 75 percent of the operator are coming from the winning list, and the remaining operators are generated randomly.

B. Recommender System

After the proposed hybrid GA automatically generates the timetables and teaching loads, all teachers with teaching overloads are collected. After that, all the subjects' that causes' overloads are summarized and displayed it to the user interface. Using the user interface, the decision maker or the administration can now do what-if analysis whether reassigning the teaching loads or to hire a new teaching staff and regenerate the timetable. Moreover, the algorithm will recommend additional staff if needed and the possible courses or subjects to be handled by the new contract teacher.

C. Datasets

The proposed method was tested using a typical dataset similar to [30] which from a higher educational institution in the Philippines. The dataset comprises of 118 classes with 308 hours workload per week that are needed to be assigned to a timetable with 45 timeslots, five laboratories, and five lecture rooms.

TABLE I. SUMMARY OF THE TEACHING STAFF

ID	Maximum Teaching Load	Administrative Load	Nature of Employment
1	3	18	Tenured
2	3	18	Tenured
3	15	6	Tenured
4	15	6	Tenured
5	0	21	Tenured
6	15	6	Tenured
7	21	0	Tenured
8	21	0	Contractual
9	18	3	Tenured
10	18	3	Tenured
11	12	0	Contractual
12	21	0	Contractual
13	21	0	Contractual
14	12	9	Tenured
15	12	9	Tenured
16	6	0	Contractual
17	21	0	Contractual
18	21	0	Contractual
19	21	0	Contractual

Table I shows 11 tenured teaching personnel are required to have a full-time workload of at least 21 units including non-teaching or administrative loads. On the other hand, the eight contract teachers are paid on an hourly basis meaning they can have a load lesser than the fulltime load. Each teacher has prepared subjects to teach that match their skills or field of discipline. Usually, these subjects or workloads are determined by the department chairs or an academic head. Moreover, all teachers are also allowed to have overloads as long as it is within the allowed number of units.

IV. RESULT AND DISCUSSION

Java programming language is used to implement the proposed approach, and testing is executed on Intel Core i3 2.4 GHz with 4 GB of RAM and Windows 10 operating system. We also used PostgreSQL, an open source object-relational database system to store master files, timetables and workloads.

Moreover, the timetable is generated under 60 seconds with pm=0.1 and a pc=0.7 to a population of 50 individuals within 250 generations. Shown in Fig. 4 is a sample teaching workload for one teacher with a teaching load of 18 units, 9 for administrative workload, and a total overload of 9. Table II shows the summary of violations concerning soft contains, hard constraints were not shown because it was all satisfied. The results are the optimal timetable for the given dataset with a total cost of 60 soft constraints penalties and no violations to all hard constraints.

Teacher ID : 14
Number of Preparation : 3
Teaching Load : 18.0 unit/s
Administrative Load : 9.0 unit/s
Over Load :6.0 unit/s

Course	Room	Day/s	Time
CSC 122	F1	M-Th	16:30 - 17:30
CSC 122.1	CL1	Sat	11:30 - 14:30
CSC 155	A2	M-Th	10:30 - 11:30
CSC 155.1	Cl3	T-F	11:00 - 12:30
CSC 104	A1	T-F	15:30 - 17:00
CSC 155	A2	Wed	08:30 - 10:30
CSC 155.1	NL	T-F	08:00 - 09:30

Fig. 4. Sample teaching load produced by the hybrid GA

As observed in Table II, the soft constraint S3 that pertains to the penalty per work overload has a total of 35 teaching units which also equivalent to 35 working hour. The other constraints are S1 that refers to a penalty per teacher's priority periods that are not satisfied, S2 refers to a penalty per teacher unavailable periods are being scheduled, S4 is the penalty per preparations above the allowed number, S5 is the penalty if the number of hours in a class per room is above the allowed allocation, S6 is a penalty that requires students to have at least one vacant period in a day, S7 is a penalty for every unavailable period of the classroom are allocated, S8 refers to under-load teachers violations and S9 refers to the room capacity constraints. With a total of 35 units of teaching overload, the administration can now decide whether to hire or not additional teaching staff.

The overloads can be distributed or transferred to other teaching staff so that they can have at least with acceptable overloads. However, teachers have a preferred subject to teach, and not all of them are capable of teaching a particular course or subject especially the beginners.

TABLE II. COST OF THE SOLUTION GENERATED BY THE HYBRID GENETIC ALGORITHM

Constraints	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**
Penalties	9	16	35	0	0	0	0	0	0

The courses shown in Table III are those courses that have a higher chance that it can be transferred to other staff or possibly to be assigned to the new contract teachers or transferred to other teaching staff.

TABLE III. RECOMMENDED SUBJECTS/LOADS FOR THE NEW CONTRACT TEACHER

Course ID	**Course Code**	**Possible Credit Units**
13	CSC 110	4
14	CSC 110.1	6
25	CSC 150	4
26	CSC 150.1	6
29	IT 116	4
30	IT 116.1	6
27	CSC 155	4
28	CSC 155.1	6

In this simulation, we suppose that the administration decided to hire an additional teaching staff using the recommended courses or teaching loads. In Table III, subjects or courses that are automatically identified and suggested by the recommender system as a possible teaching load for the new contract teacher and to avoid hiring of personnel that is not capable of teaching the remaining or unassigned classes.

Additionally, the administration can now focus on identifying hiring an additional teaching staff with skills that match the available teaching workloads. After regenerating the timetable, shown in Table IV is the summary of teaching work-loads of all teaching staff including the new contact teacher. The time needed to regenerate the timetable is also less than 60 seconds.

In this case, the allowed overload is three units, and as can be seen in the initial solution, the teacher ID 3, 4, 14 and 15 have overloads that are above of what is permitted. Moreover, in the second solution generated by the hybrid GA, we consider an additional teacher with preparation courses is those listed in Table III. It can be observed that the second solution considering the recommender system has balanced and equitably distributed workloads. The classes assigned to the new contract teacher that also recognized as a beginner are very easy to handle. The new teacher has, or they can ask help with other peers since the courses recommended by the system are familiar to most teachers, or many teachers teach the same classes. Moreover, the total penalty of the timetable has improved from 60 to 34, 8 for S1, 12 for S2, S3 has 14 and no violations to all hard constraints.

As observed, the solution with the help of the recommender system had eliminated the teachings loads that exceed the allowed overloads. Since all hard constraints are satisfied and most of the soft constraints are fulfilled, the proposed approach can create more effective room utilization patterns and improved students and teacher satisfaction levels, eliminated physical conflicts, and as much as possible it could satisfy all faculty desired periods and preparations.

TABLE IV. THE INITIAL SOLUTION COMPARED TO THE SOLUTION GENERATED USING THE RECOMMENDER SYSTEM

Solution	**Initial Solution**	**The solution with Recommender System**
No. of teachers	19	20
Total teaching loads	308 Units	308 Units
Total administrative loads	99 Units	99 Units
Total overloads (s3)	35 Units	14 Units
No. of teachers that exceeds the allowed overloads	4	0
No. of teachers with allowable overloads	6	8
No. of tenured teachers	11	11
No. of contractual teachers	8	9
Teaching personnel priority periods violations (s1)	9	9
Teaching personnel unavailable periods violations (s2)	16	16
Maximum number of preparations violations (s4)	0	0
Maximum teaching hour violations (s5)	0	0
Students one vacant period in a day violations (s6)	0	0
Rooms unavailable periods violations (s7)	0	0
No. of teacher that are underloaded (s8)	0	0
Room capacity violations (s9)	0	0
The time required in generating the timetables	< 60s	< 60s

Timetabling and teaching load management are a common problem of all institutions in higher education in the Philippines. Some of this task is done manually, and with the

help of the simple administrative system, which could end up with conflicting schedules and underutilized resources, it involves numerous personnel to plot schedules and could consume a large amount of time. Underutilized resources may result in a shortage and loss of profit, especially for private institutions. Through the proposed method, schedules are generated automatically, it only involved the scheduler or the department chairs, underutilization will be eliminated, and resources will be maximized.

Further, a sudden increase in students or an experienced teacher may leave which could result in a lack of teachers. With the help of the recommender system, managing the teaching workloads are trouble-free and hiring immediate teacher replacement is efficient since the recommender system already identifies the possible subjects or courses. Thus, hiring unmatched skills could be avoided.

V. CONCLUSION

The study presented in this paper implements a course scheduling and teaching load management with staff recommender based on the hybrid genetic algorithm. In this work, GA is infused with four neighboring structures that are identified using the self-adaptive mechanism. A repair operator also is applied to increase the optimality of the solutions. Both neighbouring structures and repair operator utilizes a data structure that act as a novel way to guide or memory of all unused pair of rooms and timeslots. Test results show that the proposed hybrid GA could generate feasible and optimized solutions which resulted in more optimal workloads and timetables. With the automated system, a considerable reduction in clerical efforts is attained. Compared to a timetable that is created manually and with the help of simple administrative system which involves numerous personnel and could take a large amount of time.

This can relieve the decision-makers from the burden of tedious and time-consuming scheduling task to be done in every semester. Also, with the recommender system, hiring additional teaching staff and managing workloads are much easier.

REFERENCES

[1] S. Bates and L. Feldstein, "Supporting Novice Faculty: Induction and Retention," J. Cross-Disciplinary Perspect. Educ., vol. 7, no. 1, pp. 26–32, 2014.

[2] A. Chudgar, M. Chandra, and A. Razzaque, "Alternative forms of teacher hiring in developing countries and its implications: A review of literature," Teach. Teach. Educ., vol. 37, no. 2014, pp. 150–161, 2014.

[3] A. Kezar and S. Gehrke, "Why Are We Hiring so Many Non-Tenure-Track Faculty?," Lib. Educ., vol. 100, no. 1, 2014.

[4] D. Carver-Thomas and L. Darling-Hammond, "Teacher turnover: Why it matters and what we can do about it BRIEF," no. August, pp. 1–50, 2017.

[5] J. Soria-Alcaraz, E. Özcan, J. Swan, and G. Kendall, "Iterated local search using an add and delete hyper-heuristic for university course timetabling," Appl. Soft Comput., vol. 40, pp. 581–593, 2016.

[6] E. A. Abdelhalim and G. A. El Khayat, "A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA)," Alexandria Eng. J., vol. 55, no. 2, pp. 1395–1409, 2016.

[7] H. Anderson, "School Timetabling in Theory and Practice A comparative study of Simulated Annealing and Tabu Search," 2015.

[8] P. Myszkowski and M. Norberciak, "Evolutionary algorithms for timetable problems," Ann. Univ. Mariae Curie-Sklodowska, Sect. AI – Inform., vol. 1, no. 1, pp. 1–9, Jan. 2015.

[9] K. Y. Junn, J. H. Obit, and R. Alfred, "The Study of Genetic Algorithm Approach to Solving University Course Timetabling Problem," Springer, Singapore, 2018, pp. 454–463.

[10] B. Y. Qu, Y. S. Zhu, Y. C. Jiao, M. Y. Wu, P. N. Suganthan, and J. J. Liang, "A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems," Swarm Evol. Comput., vol. 38, pp. 1–11, Feb. 2018.

[11] P. Amaral and T. C. Pais, "Compromise ratio with weighting functions in a Tabu Search multi-criteria approach to examination timetabling," Comput. Oper. Res., vol. 72, pp. 160–174, Aug. 2016.

[12] E. Demirović and P. J. Stuckey, "Constraint Programming for High School Timetabling: A Scheduling-Based Model with Hot Starts," Springer, Cham, 2018, pp. 135–152.

[13] H. Kanoh and S. Chen, "Particle swarm optimization with transition probability for timetabling problems," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7824 LNCS, pp. 256–265, 2013.

[14] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," Comput. Ind. Eng., 2015.

[15] S. A. Mirhassani and F. Habibi, "Solution approaches to the course timetabling problem," Artif. Intell. Rev., vol. 39, no. 2, 2013.

[16] N. Pillay, "A survey of school timetabling research," Ann. Oper. Res., vol. 218, no. 1, pp. 261–293, Jul. 2014.

[17] S. Abdullah, H. Turabieh, and B. McCollum, "An investigation of a genetic algorithm and sequential local search approach for curriculum-based course timetabling problems," Multidiscip. Int. Conf. Sched. Appl., 2009.

[18] C. Akkan and A. Gülcü, "A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem," Comput. Oper. Res., vol. 90, pp. 22–32, Feb. 2018.

[19] Y. Alshamaila, S. Papagiannidis, and F. Li, "Cloud computing adoption by SMEs in the north east of England," J. Enterp. Inf. Manag., vol. 26, no. 3, pp. 250–275, 2013.

[20] M. Rohaninejad, A. Kheirkhah, P. Fattahi, and B. Vahedi-Nouri, "A hybrid multi-objective genetic algorithm based on the ELECTRE method for a capacitated flexible job shop scheduling problem," Int. J. Adv. Manuf. Technol., vol. 77, no. 1–4, pp. 51–66, Mar. 2015.

[21] Q. Lin, Q. Zhu, P. Huang, J. Chen, Z. Ming, and J. Yu, "A novel hybrid multi-objective immune algorithm with adaptive differential evolution," Comput. Oper. Res., vol. 62, pp. 95–111, Oct. 2015.

[22] M. El-Sherbiny and R. Zeineldin, "Genetic Algorithm for Solving Course Timetable Problems," Int. J. Comput. Appl., vol. 124, no. 10, 2015.

[23] S. Yang and S. N. Jat, "Genetic algorithms with guided and local search strategies for university course timetabling," IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 41, no. 1, pp. 93–106, Jan. 2011.

[24] M. Alzaqebah and S. Abdullah, "Hybrid bee colony optimization for examination timetabling problems," Comput. Oper. Res., vol. 54, pp. 142–154, 2015.

[25] A. Bettinelli, V. Cacchiani, R. Roberti, and P. Toth, An overview of curriculum-based course timetabling, vol. 23, no. 2. 2015.

[26] J. Obit and D. Landa-Silva, "Computational study of non-linear great deluge for university course timetabling," Intell. Syst. From Theory to Pract., pp. 309–328, 2010.

[27] M. L. Pinedo, Scheduling. 2008.

[28] R. Klein, "Genetic Algorithms," Supply Chain Manag. Adv. Plan., pp. 529–536, 2008.

[29] M. Fatih Tasgetiren, P. N. Suganthan, and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," Inf. Sci. (Ny)., vol. 181, no. 12, pp. 2455–2468, 2011.

[30] J. B. Matias, A. C. Fajardo, and R. M. Medina, "A fair course timetabling using genetic algorithm with guided search technique," Proc. 2018 5th Int. Conf. Bus. Ind. Res. Smart Technol. Next Gener. Information, Eng. Bus. Soc. Sci. ICBIR 2018, pp. 77–82, 2018.