2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)
February 4-6, 2017, Doha, Qatar

Exam Scheduling: A Case Study

Ali Shatnawi and Mohammad Fraiwan
Department of Computer Engineering
Jordan University of Science and Tech
P.O. Box 3030 Irbid 22110, Jordan
Emails: ali@just.edu.jo; mafraiwan@just.edu.jo

Hadi S. Al-Qahtani
Computer Tech and Admin System
Comm. & Inf. Technology Commission
Kingdom of Saudi Arabia
Email: hqahtani@citc.gov.sa

***Abstract*—Academic institutions are moving toward automated management of the educational process. One aspect of this process is the exam scheduling. The large number of students, classes, professors, and venues renders the manual scheduling process tedious and useless. In this paper, we describe the efforts of the Arab East College for High Education in Saudi Arabia in scheduling exams in the least number of conflicts, among other constraints. We give the details for a two-stage solution approach; the first stage is a greedy algorithm and the second one is a genetic algorithm. The two algorithms work in tandem to generate the best exam timetable. Automation of this process has greatly reduced the number of conflicts, exam days, and the required venues.**

***Keywords*—exam scheduling; genetic algorithms; greedy algorithm**

I. Introduction

The evergreen exam scheduling problem has been studied extensively in the literature with a wide range of solutions. Yet there seems to be a lag in the adoption and implementation of these solutions in practice. This problem becomes of paramount importance as educational institutes are moving toward automation to cope with the large number of students, classes, and instructors.

The exam scheduling problem typically consists of the three major parts; 1. The time slots, during which the various exams can be scheduled. Typically, these slots are uniform in duration and limited in number. 2. Resources: These are the human and logistical elements required to complete the exam in a correct manner (e.g., halls, proctors, etc.). 3. Scheduling constraints: These are conditions that prevent the concurrent scheduling of exams. Sometimes it is not possible to schedule an exam in a given time slot due to students having multiple exams at the same time, a certain number of exams on the same day, or any other constraints ordained by the academic institution.

In this paper, we discuss and give details of the experience of the Arab East College for High Education, Saudi Arabia, in implementing an automated system for exam scheduling. The solution approach is comprised of two stages; In the first one, a greedy algorithm generates a set of timetabling solutions that are fed to the second stage, which is a genetic algorithm, that searches through and modifies the timetables to reach the best possible solution within a certain number of iterations.

The remainder of this paper is organized as follows: Section II gives an insight into the background and related work. The problem formulation is discussed in Section III. Section IV goes through the solution approach, the implementation details and results. In Section V, we conduct a comparison with one selected previous work. Finally, we conclude the paper in Section VI.

II. Background and Related Work

The exam timetabling problem is an evergreen problem in the literature. In the next few paragraphs, we survey some of the work conducted in this context along with some of the design criteria employed in the corresponding study.

Genetic algorithms were employed by [1], using a hierarchy of constraints and an adaptive method to solve the problem of searching in the solution space. They employed a dataset comprised of 8000 students having a total of 437 exams, to be scheduled over a period of 22 days in a total of 42 rooms.

The authors in [2] developed a very large neighborhood solution using Tabu search. They assumed that all assignments are without any conflict and the full capacity of the rooms must not be exceeded. The objective from this research was to enhance the solution quality by benefiting from the tabu search using the large neighborhood algorithm. However, it required a large amount of computational time.

Reference [3] provided a solution for a high school in Turkey using a memetic algorithm (MA) for the class and exam timetabling. Experiments were conducted to compare the incremental MA with the conventional one. Similarly, the researchers in [4] developed a genetic algorithm solution in a university setup.

A plethora of studies do exist in the literature, we summarize here some of these methods:

- Memetic algorithms [3].
- Large neighborhood search [5], [6].
- Genetic algorithms [1].
- Tabu search [2], [7].
- Cluster-based/decomposition [10].
- Ant algorithm [11].
- Simulated annealing [12].
- Great deluge algorithm [13].
- Variable neighborhood [14].
- Iterated local search [15].
- Exact methods [16].
- Multi-objective approaches [17].
- Hyper-heuristics [18].

978-1-5090-4726-0/17/$31.00©2017 IEEE

- Case-based reasoning [19].
- Fuzzy approach [20].
- Neural networks [21].
- Constructive approaches [10].

The aforementioned studies use a variety of hard (i.e., must be met) and soft (i.e., preferable) constraints. Some of these constraints are as follows:

- Hard constraints:
  1) Student may not be assigned more than one exam per day.
  2) The rooms need to be large enough to accommodate the students in one class.
  3) The number of scheduled exams in a specific time slot may not exceed the number of available rooms.
  4) Single exam per room per time slot.
- Soft constraints:
  1) For each student, the exams are spread as far as possible from each other.
  2) Exams with a large number of students should be scheduled first to allow for sufficient grading time.
  3) Load balancing of the exam resources (e.g., halls, proctors).

III. Problem Formulation

The exam scheduling problem is the assignment of exams to time slots during the exam period. The objective is to cover all exams, providing students with a conflict-free schedule (if possible), while respecting the given time periods and the rooms' capacities. Table I shows the various terms used in the formulation.

Assuming that all sections of the same class are scheduled in the same time slot, construct a conflict matrix N representing the time conflict for each exam pair. That is,

$$N_{ij} = \begin{cases} 1, & \text{exam of course } i \text{ conflicts with exam of course } j \\ 0, & \text{otherwise} \end{cases},$$

$\forall i, \forall j \in I$.

Consider the registration matrix R which represents the courses registered by each student as follows:

$$R_{ij} = \begin{cases} 1, & \text{If student } s_i \text{ is registered in course } j \\ 0, & \text{otherwise} \end{cases},$$

$\forall s_i \in S, \forall j \in C_{s_i}$.

A student is registered in one section of each class of his course list. Hence,

$$\sum_{j \in T_k} R_{ij} = 1, \forall s_i \in S, \forall k \in C_{s_i}. \quad (1)$$

Then we need to enforce there are no conflicts in the exam schedule of a student, as follows:

$$\sum_{k \in C_{s_i}} R_{ij} N_{jk} \leq 1, \forall s_i \in S, \forall j \in C_{s_i}. \quad (2)$$

TABLE I
The Notation Used in the Problem Formulation

Symbol	Meaning
S	The set of all students
I	The set of all courses
C_{s_i}	The set of all courses registered by student i
T_i	The set of all sections of course i
Z_i	The number of students enrolled in course i
W	The set of all available time slots

. Each student will introduce a number of constraints equal to the number of classes that she/he is registered in.

Problem Complexity: Now, given this initial set of constraints, the technique complexity can be established; The exam scheduling problem is an NP-hard problem. The other parameters, like the class enrollment, can be further added to take into account the capacity of the exam venue. This will make the problem even harder to solve.

The following discussion demonstrates that the problem is NP-hard:

- Consider the problem of scheduling M exams with three available time slots.
- Two exams are in conflict if they cannot be offered at the same time.
- Construct a graph G, such that the nodes represent exams.
- If two exam nodes are conflicting then they are connected with an edge.
- To schedule the exams with no conflicts, the nodes incident to the same edge cannot be scheduled during the same time slot.
- Let the three available time slots represent colors, then nodes incident to the same edge can not have the same color.
- The problem reduces to the 3-colorability graph, which is a known NP-hard problem [20].

IV. Proposed Solution and Implementation Results

A combination of two algorithms is used to solve the problem; A greedy algorithm, see Fig. 2, and a genetic algorithm, see Fig. 3. In the following, we describe in details some aspects of the algorithms.

Input: The input consists of a table of all offered sections with non-zero enrollment, along with the ID numbers of the registered students and the assigned instructors.

Output: The output is comprised of two parts; 1. A schedule of sections that includes the section name, day, time slot, and the exam venue(s). 2. Individual exam schedule for each student including the class name, day, time slot, and the exam venue.

The greedy algorithm is used to generate a set of configurable timetables that are later fed as an input to the genetic algorithm. The order of sections is set randomly beforehand. These tables represent a set of non-intersecting class sections.

The pseudo code for timetable generation is shown in Alg. 1. After that, the genetic algorithm is run for a predetermined number of generations (2000 in our case) to minimize the cost of each table. For example, consider the following set of classes:
$C = \{c_{23}, c_{25}, c_{65}, c_{53}, c_{19}, c_{123}, c_{76}, c_{55}\}$,
and let the application of the greedy algorithm result in two sets, based on conflicts, as follows:
$G1 = \{c_{23}, c_{53}, c_{19}, c_{123}\}$,
$G2 = \{c_{25}, c_{65}, c_{76}, c_{55}\}$.
These two groups form the input to the genetic algorithm, which is basically a search heuristic used to find the best solution given the number of generations.

The genetic algorithm starts by randomly generating a population (i.e., an exam schedule list) of X chromosomes using the greedy algorithm, then the fitness of each chromosome, $f(x)$, is evaluated. $f(x)$ is defined as:

$$f(x) = \sum_{i=1}^{n}(H_i \times K_{hard}) + \sum_{i=1}^{n}(S_j * K_{soft}),$$

where x is the exam schedule chromosome, n is the number of constraints, H_i is the total weight of the violated hard constraints, K_{hard} is the penalty for violating the hard constraints, K_{Soft} is the penalty for violating the soft constraints, and S_j the total weight of the violated soft constraints.

After that, a new population is generated by selecting two chromosomes as parents. Crossover (i.e., a new exam schedule from the parents) and mutation (i.e., modification of the child after crossover) are performed with a certain probability based on the fitness values till acceptance is reached and a full population is achieved.

```
GenerateTimeTabel(G: set of classes of sections [ ])
FirstTimeSlot=0
LastTimeSlot=Number of Days X Number of time
slots
for ∀Class C_i ∈ G do
    for ∀ Section S_j in C_i do
        for
        Time Slot T_S between FirstSlot and LastSlot
        do
            Find the best fit empty Room R for
            Section S_j
            Reserve room R and T_S for section S_j
            if Day(current T_S) > LastClassDay then
                LastClassDay= Day(current T_S)
            end
        end
    end
    FirstTimeSlot of the next day= Number of time
    slots X (LastClassDay+1)
end
```

Algorithm 1: The pseudo code for the greedy algorithm.

TABLE II
THE CASE STUDY PARAMETERS.

Parameter	Value
DEFAULT_EXAM_HOURS	3
DEFAULT_ROOM_CAPACITY	30
MAX_DAYS	20
RANDOM_SECTIONS_ORDER	1
TIME_SLOTS_PER_DAY	1
DAYS_BETWEEN_EXAMS	1
GENERATIONS	2000
POPULATION_SIZE	100
CROSSOVER_METHOD	1
MUTATION_PROBABILITY	0.05
MAX_CLASS_CAPACITY	250
ENVIRONMENT_CAPACITY	10

TABLE III
THE COST PARAMETERS EMPLOYED IN THE CASE STUDY.

Constraint	Cost
H_OVER_ROOM_CAPACITY_COST	10000
H_TIMESLOT_ROOM_CONFLICT_COST	10000
H_SAME_DAY_CONFLICT_COST	50000
S_SPREAD_STUDENT_COURSES_COST	50
S_PROF_COURSE_STUDENTS_EQUALITY_COST	10
S_UNUTILISED_ROOM_CAPACITY_COST	100

Case Study:

The system was designed to fulfill the needs of the Arab East College for Higher Education in the Kingdom of Saudi Arabia. Both the greedy and genetic algorithms were implemented in C++. Fig. 1 shows the various classes employed in the UML diagram of the system.

The required tables and data structures were either extracted from the registration database or constructed via preprocessing. Tables II and III show the algorithm parameters employed in the run. A sample of the records is shown in Table IV and the conflict matrix in Table V. Table VI shows the input and output obtained during one of the runs of the scheduling system. Table VII shows a sample of the output indicating the assigned time slot and room. Fig. 4 shows the improvement in the cost as more generations are used.

V. COMPARISON WITH PREVIOUS WORK

We have chosen to compare our work with the work proposed by Jha in [21], since it is the closest to our work in terms of the problem size and assumptions.

Jha used three constraints: each course is scheduled in exactly one period, no student is assigned two exams in the same period, and no room is allocated to more than one course

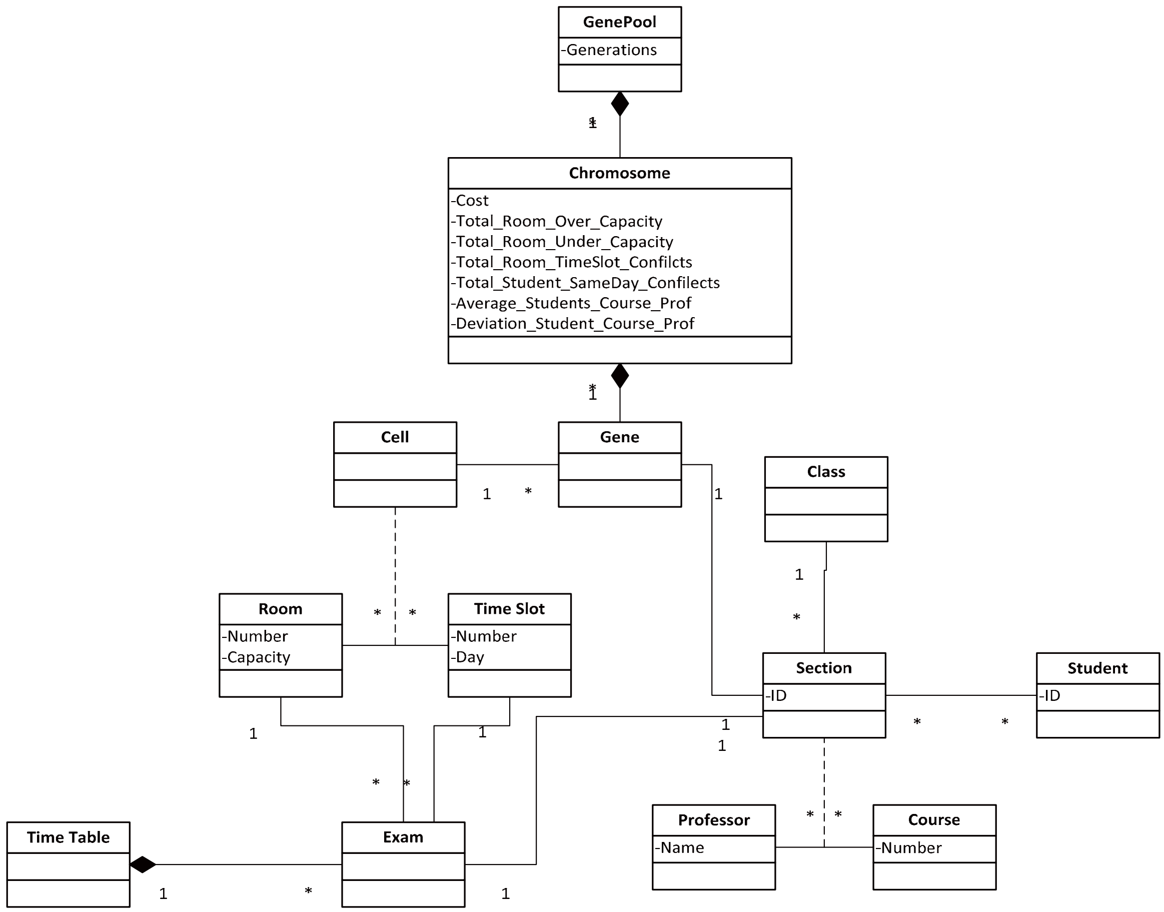

Fig. 1. The UML diagram showing the software design of the solution.

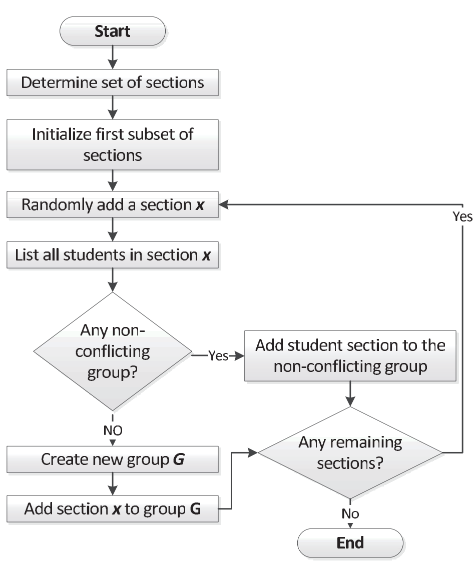

Fig. 2. The greedy algorithm

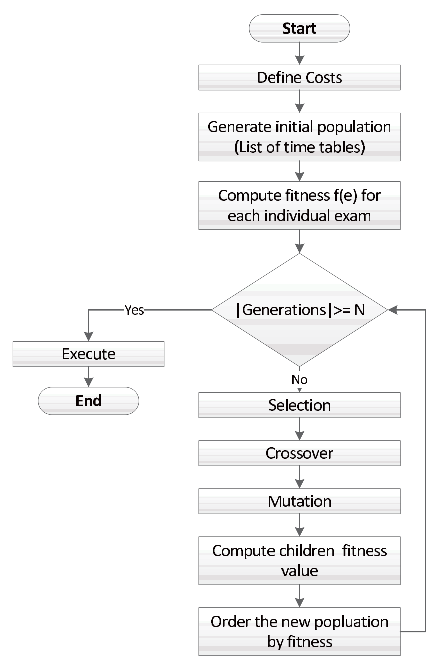

Fig. 3. The genetic algorithm

TABLE IV
SAMPLE INPUT DATA

Course	Section	Professor	Student No.
TAN_571	TAN_571_1	prof1	3000000
ADM_522	ADM_522_1	prof2	3000001
CAL_502	CAL_502_1	prof3	3000002
GNN_528	GNN_528_1	prof4	3000003
GNN_532	GNN_532_1	prof4	3000004
DHT_512	DHT_512_1	prof5	3000005
GNN_525	GNN_525_1	prof6	3000006
GNN_521	GNN_521_1	prof7	3000007
GNN_525	GNN_525_1	prof8	3000008

TABLE V
THE REGISTRATION CONFLICT MATRIX. DIAGONAL ENTRIES ARE THE SECTION ENROLLMENT FIGURES.

AS_513 prof1	AS_513 prof58	AS_513 prof32	AS_513 prof26	AS_513 prof1	
0	0	0	3	4	AS_513 prof1
0	0	0	17	3	AS_513 prof26
2	0	38	0	0	AS_513 prof32
0	11	0	0	0	AS_513 prof58
8	0	2	0	0	AS_513 prof1

in one period. It is also assumed that any room is large enough to accommodate the students of any course. He applied his pure genetic algorithm on a hypothetical scenario comprising thirty six courses and twelve rooms. The solution resulted in the need for twenty periods.

In our work, we handle more constraints as mentioned earlier. We have also started with a valid solution using the Greedy Algorithm and proceeded to use the Genetic Algorithm to tune the solution such that the hard constraints are met and the soft ones are maximally satisfied. On the other hand, we have tested our algorithm on a real case with ninety seven courses and twenty four rooms. The solution resulted in the need for twenty four periods.

VI. CONCLUSION

Even though the exam scheduling problem is a well-studied problem, practical solutions are still in need to be developed. In this paper, we have showcased the experience of the Arab East College for High Education, KSA, in scheduling exams. The greedy and genetic algorithm that we presented offer reasonable arrangements for the students and the faculty.

Yet, more constraints may also be required to increase the quality of the produced schedule. For example, minimizing the number of exam days, maximizing the separation between exams for the same student to allow for sufficient preparation time, prioritizing certain exams, proctors availability, or any other arrangement that may be required. Moreover, making these solutions available as free or commercial tools is of great benefit.

TABLE VI
THE INPUT (FIRST FIVE) AND OUTPUT OF THE SAMPLE RUN.

Input/Output	Value
Number of students	682
Number of halls	24
Total available time slots	16
Number of classes	94
Number of sections	118
Number of allocated time slots	40
Number of conflicts	0

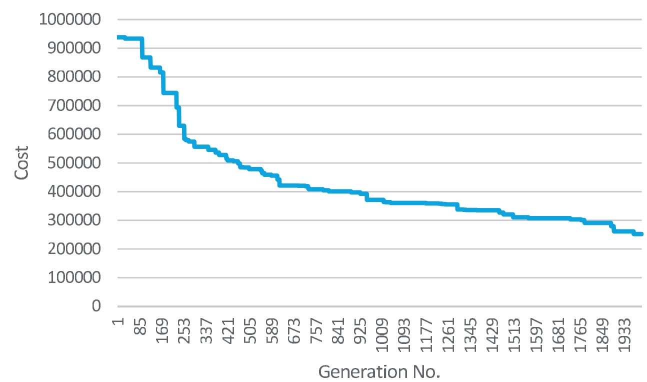

Fig. 4. The cost fitness versus the progress of the generations.

TABLE VII
SAMPLE OUTPUT.

# Students	Time slot	Day	Room	Section
4	0	0	23	AS_505 prof1
17	0	4	1	AS_509 prof26
38	0	4	2	AS_511 prof32
11	0	4	16	AS_511 prof58
8	0	2	16	AS_513 prof11
28	0	1	27	AS_513 prof32
10	0	3	16	AS_515 prof30
38	0	3	2	AS_517 prof32
11	0	2	13	AS_521 prof58

REFERENCES

[1] O. Arogundade, A. Akinwale, and O. Aweda, “Genetic Algorithm Approach for a Real-World University Examination Timetabling Problem,” *Int. J. Comput. Appl.,* vol. 12, no. 5, pp. 0975-8887, Dec., 2010.

[2] S. Abdullah, A. Ahmadi, E. Burke, M. Dror, and B. McCollum, “A tabu based large neighbourhood search methodology for the capacitated examination timetabling problem”, *J. Oper. Res. Soc.,* vol. 58, pp. 1494-1502, 2007.

[3] E. Ozcan and A. Alkan, “A Memetic Algorithm for Solving a Timetabling Problem:An Incremental Strategy,” In Multidisciplinary international scheduling: theory and applications (MISTA07), Paris, France, 2007.

[4] M. Basikhasteh and M. Movafaghpour, “Hybridizing Genetic Algorithm with Biased Chance Local Search,” *Proc. World Acad. Sci. Eng. Tech.,* vol. 80, pp. 354-359, 2011.

[5] S. Abdullah, S. Ahmadi, K. Burke, and M. Dror, “Investigating Ahuja-Orlins large neighbourhood search approach for examination timetabling,” *OR-Spektrum,* vol. 29, no.1, pp. 351372, 2007.

[6] A. Syariza, E. Burke, A. Bargiela, B. McCollum, and E. Ozcan, “A Constructive Approach to Examination Timetabling based on Adaptive Decomposition and Ordering,” *Ann. Oper. Res.,* vol. 218, no. 1, pp. 3-21, 2014.

[7] F. Glover, *tabu search fundamentals and uses*. University of Colorado, Boulder.
[8] R. Qu and R. Burke, "Adaptive decomposition and construction for examination timetabling problems," in *Proceedings of the 3rd Multi-disciplinary International Scheduling: Theory and Applications,* 2007.
[9] M. Eley, *Ant algorithms for the exam timetabling problem*, International Conference on the Practice and Theory of Automated Timetabling. Springer Berlin Heidelberg, 2006.
[10] J. Thompson and K. Dowsland, "Variants of simulated annealing for the examination timetabling problem," *Ann. Oper. Res.,* vol. 63, pp. 105128, 1996.
[11] K. Burke, Y. Bykov, P. Newall, and S. Petrovic, "A time-predefined local search approach to exam timetabling problem," *IIE Transactions,* vol. 36, no. 6, pp. 509528, 2004.
[12] K. Burke, J. Eckersley, B. McCollum, S. Petrovic, and R. Qu, "Hybrid variable neighbourhood approaches to university exam timetabling," *Eur. J. Oper. Res.,* vol. 206, no. 1, pp. 4653, 2010.
[13] M. Caramia, P. DellOlmo, and F. Italiano, "Novel local search based approaches to university examination timetabling," *Inform. J. Comput.,* vol. 20, no. 1, pp. 86-99, 2008.
[14] P. Boizumault, Y. Delon, and L. Peridy, "Constraint logic programming for examination timetabling," *J. Logic Program.,* vol. 26, no. 2, pp. 217-233, 1996.
[15] L. Mumford, "A multiobjective framework for heavily constrained examination timetabling problems," *Ann. Oper. Res.,* vol. 180, no. 1, pp. 3-31, 2010.
[16] E. Özcan, Y. Bykov, M. Birben, and E. Burke, "Examination timetabling using late acceptance hyper-heuristics," in *Proceedings of the 2009 IEEE congress on evolutionary computation (CEC 2009)*, Trondheim, Norway.
[17] E. Burke, S. Petrovic, and R. Qu, "Case based heuristic selection for timetabling problems," *Journal of Scheduling,* vol. 9, no. 2, pp. 115-132, 2006.
[18] H. Asmuni, E. Burke, J. Garibaldi, B. McCollum, and A. Parkes, "An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables," *Comput. Oper. Res.,* vol. 36, no. 4, pp. 981-1001, 2009.
[19] P. Corr, B. McCollum, M. McGreevy, and P. McMullan, "A new neural network based construction heuristic for the examination timetabling problem," *In Lecture notes in computer science: Vol. 4193. Parallel problem solving from naturePPSN IX,* pp. 392-401, 2006.
[20] D. Johanson and M. Gary, *computer and intractability a guide to the theory of np-completeness*, Publisher: W. H. Freeman and Company, 1979.
[21] S. K. Jha, "Exam timetabling problem using Genetic algorithm," *International Journal of Research in Engineering and Technology,* vol. 3, pp. 649-654, May, 2014.