Two-Stage Multi-Objective University Courses Timetabling Using Genetic Algorithms

M. S. Al-Ashhab*1,2 and Abdulrahman Alghamdi2

1 Design & Production Engineering Dept. Faculty of Engineering, Ain-Shams University– Egypt
2 Dept. of Mechanical Engineering, College of Engineering and Islamic architecture, Umm Al-Qura University, Makkah, Saudi Arabia.

*Corresponding author email: mohammedsem@yahoo.com

Abstract - In this paper, a University Courses Timetabling (UCTT) optimization model has been developed in MS Excel environment and solved using Evolver solver based on Genetic Algorithms. The proposed model has two stages, which may be used together or individually in accordance with the philosophy of the university. The proposed model was implemented to schedule the courses in the Mechanical Engineering Department of Umm Al-Qura University, and it provided an optimal schedule.

Keyword- UCTT, Optimization, GA, Evolver

1. Introduction

Timetables are organizational structures that can be found in various areas of human activity including sports [1], entertainment [2], transport [3], industry [4], and education [5]. In the context of higher education institutions, a timetable can be thought of as an assignment of events (such as lectures, tutorials, or exams) to a finite number of rooms and timeslots in accordance with a set of constraints; some of which will be mandatory, while others may be optional [6]. According to [7], the problem of constructing such timetables can be divided into two categories: exam timetabling problems and course timetabling problems. It is also suggested that course timetabling problems can be further divided into two sub-categories: “post enrolment-based course timetabling”, where the constraints of the problem are specified by student enrolment data, and “curriculum-based course timetabling”, where constraints are based on curricula specified by the university. T. Müller and H. Rudová [8] have also presented that these sub-categories are closely related, demonstrating how instances of the latter can be transformed into those of the former in many cases.

A Novel Genetic Algorithm Technique for Solving University Course Timetabling Problems [9], Trans Genetic Coloring Approach for Timetabling Problem [10], On improvement of Effectiveness in Automatic University Timetabling Arrangement with Applied Genetic Algorithm [11] are presented.

The CB-CTT problem consists of scheduling lectures of a set of courses into a weekly timetable, where each lecture of a course must be assigned a period and a room in accordance with a given set of constraints are proposed [12]. University Course Timetabling Problem is presented in Fig. 1 [13].

Figure 1. Diagram of University Course Timetabling Problem
Evolver can use any of the following six solving methods, depending on the type of optimization model. Three of the solving methods (Recipe, Order, and Grouping) make use of entirely different algorithms. The other three (Budget, Project, and Schedule) are descendants of these three, adding additional constraints[14].

The Schedule solving method has been used in this research to solve the UCTT problem. Each course is assumed to take the same amount of time.

In the Schedule solving method, there are eight kinds of constraints between courses:
1. (With) two classes must occur in the same time block.
2. (Not with) two classes must not occur in the same time block.
3. (Before) the first class must occur before the second class.
4. (At) the class must occur in the time block in a certain time block.
5. (Not after) the first class must occur at the same time or before the second task.
6. (Not before) the first class must occur at the same time after the second class.
7. (Not at) the class must not occur in a certain time block.
8. (After) the first class must occur after the second class.

2. Problem description

The problems of UCTT may involve two classes of hard and soft constraints. Hard constraints must be satisfied in the problem completely so that the generated solution would be possible without any conflict; no violation is allowed in these constraints. Soft constraints are related to objective function; the objective function is to maximize the number of satisfied soft constraints. Unlike hard constraints, soft constraints are not necessarily required to be satisfied, but as the number of these satisfied constraints increases, the quality of solutions of objective function increases. In the following, a list of hard and soft constraints is presented. These constraints are taken from literature [15-19].

a. Hard constraints
1. No student is required to attend more than one course at the same time.
2. No lecturer is required to attend more than one course at the same time.
3. Number of course occurred at the same time slot do not exceed the available number of teaching rooms.
4. Each course also requires a room capacity.
5. The double booking of rooms is prohibited.
6. Some events cannot be taught in certain timeslots.
7. Precedence constraints – some courses need to be scheduled before or after others.

b. Soft constraints
1. The teacher can have the choice to suggest priority to certain timeslots for her/his courses.
2. A teacher may request a special classroom for a given course.
3. The courses should not be scheduled for evening timeslots, as it is possible.
4. The teaching hours for teachers in a classroom are 2 hours.

3. Model formulation and implementation

The proposed model has been implemented to schedule the courses in the Mechanical Engineering Department of Umm Al-Qura University. The problem involves assigning a set of courses to 30 timeslots (5 days, with 6-timeslots per day) as shown in Table 1, which also includes the priority penalty (PP) of each slot as specified by the faculty policy. The lower the priority penalty, the number is the most prior slot.

The model has been solved using Evolver solver and run on an Intel® Core™ i3-2310M CPU @ 2.10 GHz (3 GB of RAM). The GA parameters include: population size N = 50, number of generations G = 40,000, probability of crossover Pp = 0.5, and probability of mutation Pm = 0.1.

The slots PPs may be chosen according to the requirements of each university. In this research, the slots PPs are assumed as shown in Table 1. Most priorities are given to the morning slots and priorities are reduced for the later slots. Thursday slots have the lowest priorities.
### Table 1. Time slot and priority penalty table

| Day | Period | Slots | Priority Penalty | Day | Period | Slots | Priority Penalty | Day | Period | Slots | Priority Penalty | Day | Period | Slots | Priority Penalty |
|-----|--------|-------|-----------------|-----|--------|-------|-----------------|-----|--------|-------|-----------------|-----|--------|-------|-----------------|
| Sunday | 1 | 1 | 0 | Monday | 1 | 1 | 7 | 0 | Tuesday | 1 | 13 | 0 | Wednesday | 1 | 19 | 0 | Thursday | 1 | 25 | 50 |
|       | 2 | 2 | 1 |               | 2 | 2 | 8 | 1 |               | 2 | 2 | 14 | 1 |               | 2 | 2 | 20 | 1 |               | 2 | 26 | 60 |
|       | 3 | 3 | 2 |               | 3 | 3 | 9 | 2 |               | 3 | 3 | 15 | 2 |               | 3 | 3 | 21 | 2 |               | 3 | 27 | 70 |
|       | 4 | 4 | 3 |               | 4 | 4 | 10 | 3 |               | 4 | 4 | 16 | 3 |               | 4 | 4 | 22 | 3 |               | 4 | 28 | 80 |
|       | 5 | 5 | 4 |               | 5 | 5 | 11 | 4 |               | 5 | 5 | 17 | 4 |               | 5 | 5 | 23 | 4 |               | 5 | 29 | 90 |
|       | 6 | 6 | 5 |               | 6 | 6 | 12 | 5 |               | 6 | 6 | 18 | 5 |               | 6 | 6 | 24 | 5 |               | 6 | 30 | 10 |

The list of the courses and the number of students registered in each course as tabulated in Table 2 are some of the model inputs.

### Table 2. Courses ID and no. of students

| ID | Course code | # St. | ID | Course code | # St. | ID | Course code | # St. |
|----|-------------|------|----|-------------|------|----|-------------|------|
| 1  | 804151-G1   | 20   | 2  | 804151-G2   | 29   | 3  | 804344-G1   | 42   | 4  | 804344-G2   | 26   | 5  | 804344-G3   | 31   | 6  | 804344-G4   | 28   |
| 7  | 804306-G1   | 46   | 8  | 804301-G1   | 45   | 9  | 804301-G2   | 37   | 10 | 804301-G3  | 27   | 11 | 804461-G1   | 21   | 12 | 804461-G2   | 34   |
| 14 | 804466-G1   | 33   | 15 | 804306-G2   | 46   | 16 | 804306-G3   | 41   | 17 | 804306-G4   | 24   | 18 | 804306-G5   | 34   | 19 | 804308-G1   | 34   |
| 20 | 804308-G2   | 25   | 21 | 804333-G1   | 37   | 22 | 804333-G2   | 27   | 23 | 804469-G1   | 36   | 24 | 804319-G1   | 37   | 25 | 804319-G2   | 42   |
| 26 | 804469-G1   | 48   | 27 | 804302-G1   | 32   | 28 | 804302-G2   | 33   |

DOI: 10.21817/ijet/2018/v10i4/180404030  Vol 10 No 4 Aug-Sep 2018  1104
The students’ list of each course is the second input of the model. Table 3 is a sample of students’ lists of some courses.

| Engineering Graphics | Engineering Economics |
|----------------------|-----------------------|
| Group (1)            | Group (2)            |
| 4300000              | 4300020              |
| 4300001              | 4300021              |
| 4300002              | 4300022              |
| 4300003              | 4300023              |
| 4300004              | 4300024              |
| 4300005              | 4300025              |
| 4300006              | 4300026              |
| 4300007              | 4300027              |
| 4300008              | 4300028              |
| 4300009              | 4300029              |
| 4300010              | 4300030              |
| 4300011              | 4300031              |
| 4300012              | 4300032              |
| 4300013              | 4300033              |
| 4300014              | 4300034              |
| 4300015              | 4300035              |

The optimization process is completed in two stages. The first stage is formulated to optimize the time slot for each course, while the second stage is formulated to optimize the location of each course. So, the first stage is called the time stage, and the second stage is called the location stage.

a. Time stage

Time stage consists of three steps as follows:
1. Student overlap
2. Lecturer overlap
3. Model Not-With (NW) constraints

1. Student overlap matrix

To ensure that no student has more than one lecture at the same time, the model is designed to compare the courses group’s lists automatically to determine the student overlap matrix. Table 4 shows a part of 81*81 student overlap matrix.

| ID | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|----|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|
| 1  |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |
| 2  |   |   |   |   |   |   |   |   | NW | NW |    |    |    |    |    |    |    |    |
| 3  |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |   NW|
| 4  |   |   |   |   |   |   |   |   |    | NW |    |    |    |    |    |    |    |    |
|    |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |
| 6  |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |
| 7  |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |
| 8  |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |
2. Lecturer overlap matrix

To ensure that no lecturer has more than one lecture at the same time, after entering lecturers responsible for teaching each group as shown in Table 5, the model will analyze it automatically to determine the lecturer overlap matrix. Table 6 shows a part of 81*81 lecturer overlap matrix.

| Courses (ID) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|--------------|---|---|---|---|---|---|---|---|---|----|----|----|
| Lecturer Code | L01 | L01 | L01 | L01 | L01 | L01 | L02 | L02 | L03 | L02 | L02 | L02 |
| Courses (ID) | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| Lecturer Code | L03 | L03 | L04 | L04 | L04 | L04 | L05 | L06 | L05 | L05 | L27 | L26 |
| Courses (ID) | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| Lecturer Code | L26 | L25 | L25 | L25 | L25 | L10 | L24 | L24 | L16 | L16 | L16 | L23 |
| Courses (ID) | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| Lecturer Code | L23 | L22 | L22 | L22 | L22 | L21 | L21 | L21 | L20 | L20 | L19 | |
| Courses (ID) | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| Lecturer Code | L19 | L18 | L17 | L17 | L14 | L14 | L16 | L16 | L16 | L16 | L15 | L15 |
| Courses (ID) | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 |
| Lecturer Code | L14 | L14 | L13 | L13 | L12 | L12 | L12 | L11 | L10 | L09 | L09 | L08 |
| Courses (ID) | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | | | |
| Lecturer Code | L07 | L07 | L07 | L07 | L07 | L06 | L06 | L07 | L07 | | | |

Table 6. Lecturer overlap matrix (sample)
3. Model “not-with” constraints determination

The Not-With constraints shown in Table 7 are resulted automatically in the model for the courses that have a student or lecturer overlap.

Table 7. Model NW Constraints

| ID 1 | Rel. | ID 2 | ID 1 | Rel. | ID 2 | ID 1 | Rel. | ID 2 | ID 1 | Rel. | ID 2 |
|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | NW   | 2    | 6    | NW   | 53   | 21   | NW   | 34   | 29   | NW   | 72   |
| 1    | NW   | 25   | 8    | NW   | 61   | 21   | NW   | 74   | 30   | NW   | 32   |
| 1    | NW   | 26   | 10   | NW   | 11   | 23   | NW   | 58   | 30   | NW   | 39   |
| 1    | NW   | 29   | 10   | NW   | 51   | 24   | NW   | 81   | 30   | NW   | 44   |
| 1    | NW   | 32   | 10   | NW   | 54   | 25   | NW   | 28   | 30   | NW   | 50   |
| 1    | NW   | 33   | 10   | NW   | 79   | 25   | NW   | 30   | 30   | NW   | 52   |
| 1    | NW   | 34   | 11   | NW   | 79   | 25   | NW   | 39   | 30   | NW   | 77   |
| 1    | NW   | 35   | 12   | NW   | 24   | 25   | NW   | 50   | 30   | NW   | 78   |
| 2    | NW   | 10   | 13   | NW   | 55   | 25   | NW   | 52   | 31   | NW   | 39   |
| 2    | NW   | 11   | 14   | NW   | 65   | 25   | NW   | 77   | 31   | NW   | 70   |
| 2    | NW   | 79   | 15   | NW   | 34   | 26   | NW   | 43   | 32   | NW   | 35   |
| 3    | NW   | 19   | 15   | NW   | 37   | 27   | NW   | 38   | 32   | NW   | 63   |
| 3    | NW   | 45   | 16   | NW   | 68   | 27   | NW   | 46   | 32   | NW   | 80   |
| 3    | NW   | 48   | 16   | NW   | 69   | 27   | NW   | 78   | 33   | NW   | 36   |
| 3    | NW   | 57   | 19   | NW   | 43   | 28   | NW   | 30   | 33   | NW   | 72   |
| 3    | NW   | 64   | 19   | NW   | 45   | 28   | NW   | 39   | 34   | NW   | 37   |
| 4    | NW   | 8    | 19   | NW   | 48   | 28   | NW   | 50   | 36   | NW   | 49   |
| 4    | NW   | 42   | 19   | NW   | 57   | 28   | NW   | 52   | 37   | NW   | 66   |
| 4    | NW   | 55   | 20   | NW   | 31   | 28   | NW   | 77   | 38   | NW   | 72   |
| 4    | NW   | 62   | 20   | NW   | 39   | 29   | NW   | 31   | 38   | NW   | 78   |

The output of the first stage is the optimal slot number for each course, which satisfies two objectives of minimizing both the penalty of priority and the excess number of classes as shown in Table 8. Limiting the number of classes at each slot by an available number of rooms, which is equal to four in this case, by reducing the excess classes’ number which is calculated by subtracting the number of the available rooms from the number of scheduled classes. Table 9 shows the resulted optimal solution, in which there are no required excess rooms in any time slot.

Table 8. Slot number and PP for the first six course (First Objective)

| ID | 1          | 2          | 3          | 4          | 5          | 6          |
|----|------------|------------|------------|------------|------------|------------|
|    | 804151-G1  | 804151-G2  | 804344-G1  | 804344-G2  | 804344-G3  | 804344-G4  |
| Slot No. | 11 | 2   | 7   | 14 | 23 | 9   |
| PP  | 4   | 1   | 0   | 1  | 4  | 2   |
### Table 9. The excess number of classes (Second Objective)

| Slots | Sched. classes | Excess classes | Slots | Sched. classes | Excess classes | Slots | Sched. classes | Excess classes |
|--------|----------------|----------------|--------|----------------|----------------|--------|----------------|----------------|
| 1      | 4              | 0              | 11     | 2              | 0              | 21     | 4              | 0              |
| 2      | 4              | 0              | 12     | 3              | 0              | 22     | 4              | 0              |
| 3      | 4              | 0              | 13     | 4              | 0              | 23     | 4              | 0              |
| 4      | 4              | 0              | 14     | 4              | 0              | 24     | 1              | 0              |
| 5      | 2              | 0              | 15     | 4              | 0              | 25     | 0              | 0              |
| 6      | 3              | 0              | 16     | 2              | 0              | 26     | 0              | 0              |
| 7      | 4              | 0              | 17     | 2              | 0              | 27     | 0              | 0              |
| 8      | 4              | 0              | 18     | 2              | 0              | 28     | 0              | 0              |
| 9      | 4              | 0              | 19     | 4              | 0              | 29     | 0              | 0              |
| 10     | 4              | 0              | 20     | 4              | 0              | 30     | 0              | 0              |
|        |                |                |        |                |                |        |                |                |
| Sum    | 81             | 0              |        |                |                |        |                |                |

b. Location stage

The location stage consists of two steps as follows:

1. Room NW matrix
2. Course location and schedule.

1. Room NW matrix

The objective of this stage is to reduce the number of excess students in the teaching rooms. It must ensure that there is no more than one lecture in the same room at the same time. Table 10 shows the NW relationships between courses at the same slot.

### Table 10. Room NW matrix (sample)

| ID/ID | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   |
|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 2     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 3     |      |      |      |      |      |      |      |      |      |      |      |      |      |      | NW   |
| 4     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 5     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 6     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 7     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 8     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |

2. Course location and schedule.

The resulted room number assigned for each course table is shown in Table 11.

### Table 11. Course location (sample)

| ID | Course code   | 1  | 2  | 3  | 4  | 5  | 6  |
|----|---------------|----|----|----|----|----|----|
|    | Course code   | 804151-G1 | 804151-G2 | 804344-G1 | 804344-G2 | 804344-G3 | 804344-G4 |
| #St |                | 20  | 22  | 42  | 26  | 31  | 28  |
| Slot No. |            | 11  | 2   | 7   | 14  | 23  | 9   |
| Room No. |              | 4   | 4   | 4   | 4   | 3   | 3   |
| Room Capacity |        | 70  | 70  | 70  | 70  | 35  | 35  |
| Excess students |    | 0   | 0   | 0   | 0   | 0   | 0   |
The final schedule of the program is shown in table 12, wherein it is noticed that there are no lectures on Thursday because of its higher PP.

| Day      | Room 1 | Room 2 | Room 3 | Room 4 |
|----------|--------|--------|--------|--------|
| Sunday   | Room 1 | 804466-G1 | 804465-G1 | 804465-G2 | 804465-G3 |
|          | Room 2 | 804466-G2 | 804465-G3 | 804465-G4 | 804465-G5 |
| Monday   | Room 1 | 804466-G1 | 804465-G1 | 804465-G2 | 804465-G3 |
|          | Room 2 | 804466-G2 | 804465-G3 | 804465-G4 | 804465-G5 |
| Tuesday  | Room 1 | 804466-G1 | 804465-G1 | 804465-G2 | 804465-G3 |
|          | Room 2 | 804466-G2 | 804465-G3 | 804465-G4 | 804465-G5 |
| Wednesday| Room 1 | 804466-G1 | 804465-G1 | 804465-G2 | 804465-G3 |
|          | Room 2 | 804466-G2 | 804465-G3 | 804465-G4 | 804465-G5 |
| Thursday | Room 1 | 804466-G1 | 804465-G1 | 804465-G2 | 804465-G3 |
|          | Room 2 | 804466-G2 | 804465-G3 | 804465-G4 | 804465-G5 |

The quality of the solution may be improved by repeating the optimization process without restoring the adjustable cells. The relation between achieved objective, processing time (PT) and the number of iterations for different number of trials for each run is represented in Table 13. Fig. 2 shows that increasing the number of trial for each run increases the quality of the solution by getting better values of the objective and decreases the number of iterations required to get the best solution. The solution does not change after six or seven iterations at most.

| No. of Trials | 10000 | 20000 | 30000 | 40000 | 50000 |
|---------------|-------|-------|-------|-------|-------|
| Obj. PT (Sec.) | 398   | 214   | 530   | 468   | 300   | 453   | 331   | 708   | 326   | 799   |
| 2             | 314   | 274   | 250   | 460   | 193   | 470   | 190   | 667   | 192   | 856   |
| 3             | 250   | 402   | 197   | 460   | 185   | 469   | 180   | 670   | 180   | 850   |
| 4             | 245   | 298   | 192   | 446   | 185   | 509   | 180   | 679   | 180   | 780   |
| 5             | 235   | 219   | 189   | 458   | 181   | 599   | 179   | 633   | 180   | 810   |
| 6             | 235   | 260   | 189   | 465   | 181   | 514   | 175   | 658   | 180   | 800   |
| 7             | 235   | 230   | 189   | 460   | 181   | 537   | 175   | 661   | 180   | 841   |
4. Conclusion

The University Courses Timetabling (UCTT) optimization proposed model has been built in MS Excel environment and solved using Evolver solver based on Genetic Algorithms. The model has been implemented in Umm Al-Qura University.

The proposed model has two stages, which may be used together or individually in accordance with the philosophy of the university. Some universities are assigning the time slot for all courses and ask their students to register and the registration system prevents occurring of any conflict of time for both lecturers and students. In this case, there is no need of the first stage.

The quality of the solution may be improved by repeating the optimization process without restoring the adjustable cells.

The proposed model has been implemented to schedule the courses in the Mechanical Engineering Department of Umm Al-Qura University, and it provided an optimal schedule.

References

[1] G. Kendall, S. Krust, C. C. Ribeiro, and S. Urrutia, "Scheduling in sports: An annotated bibliography," Computers & Operations Research, vol. 37, pp. 1-19, 2010.
[2] J. Eliashberg, Q. Hegie, J. Ho, D. Huisman, S. J. Miller, S. Swami, et al., "Demand-driven scheduling of movies in a multiplex," International Journal of Research in Marketing, vol. 26, pp. 75-88, 2009/06/01/ 2009.
[3] A. Capra, M. Fischetti, and P. Toth, "Modeling and solving the train timetabling problem," Operations research, vol. 50, pp. 851-861, 2002.
[4] I. Bööchler, "Modeling staff scheduling problems. A tutorial," European Journal of Operational Research, vol. 158, pp. 533-542, 2004.
[5] R. Lewis, "A survey of metaheuristic-based techniques for university timetabling problems," OR spectrum, vol. 30, pp. 167-190, 2008.
[6] D. Corne, P. Ross, and H. Fang, "Evolving timetables," The practical handbook of genetic algorithms, vol. 1, pp. 219-276, 1995.
[7] B. McCollum, A. Schaerf, B. Paechter, P. McMullan, R. Lewis, A. J. Parkes, et al., "Setting the research agenda in automated timetabling: The second international timetabling competition," INFORMS Journal on Computing, vol. 22, pp. 120-130, 2010.
[8] T. Müller and H. Rudová, "Real-life curriculum-based timetabling," Practice and Theory of Automated Timetabling (PATAT 2012), pp. 57-72, 2012.
[9] O. M. K. Alsmadi, S. Za'er, D. I. Abu-Al-Nadi, and A. Algsoon, "A novel genetic algorithm technique for solving university course timetabling problems," in Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on, 2011, pp. 195-198.
[10] G. Asham, M. M. Soliman, and A. Ramadan, "Trans genetic coloring approach for timetabling problem," Artificial Intelligence Techniques Novel Approaches & Practical Applications, ICA, pp. 17-25, 2011.
[11] P. Khonggamnerd and S. Innet, "On improvement of effectiveness in automatic university timetabling arrangement with applied genetic algorithm," in Computer Sciences and Convergence Information Technology, 2009. ICCIT'09. Fourth International Conference on, 2009, pp. 1266-1270.
[12] L. Di Gaspero, B. McCollum, and A. Schaerf, "The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3)," Citeseer2007.
[13] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," Computers & Industrial Engineering, vol. 86, pp. 43-59, 2015.
[14] Palisade. (2010). Guide to Using Evolver. Available: www.crystalballservices.com
[15] H. Asmuni, "Fuzzy methodologies for automated university timetabling solution construction and evaluation," University of Nottingham, 2008.
[16] M.-R. Feizi-Derakhshi, H. Babaei, and J. Heidarzadeh, "A survey of approaches for University course timetabling problem," in Proceedings of 8th International Symposium on Intelligent and Manufacturing Systems, Sakarya University Department of Industrial Engineering, Adrasan, Antalya, Turkey, 2012, pp. 307-321.
[17] C. Gotlieb, "The construction of class-teacher timetables," in Proc. IFIP Congress, 1963, pp. 73-77.
[18] J. Henry Obid, "Developing novel meta-heuristic, hyper-heuristic and cooperative search for course timetabling problems," University of Nottingham, 2010.
[19] T. A. Redl, "A study of university timetabling that blends graph coloring with the satisfaction of various essential and preferential conditions," Rice University, 2004.

AUTHOR PROFILE

Mohammad Sayed Al-Ashhab born in Cairo, Egypt. Is a full-time associate professor at UQU. Received a degree in Mechanical Engineering from Ain Shams University, located in Cairo, Egypt in 1996, Master in 2003, and PhD in 2008.

Abdulrahman Ali Alghamdi graduated from Umm Al-Qura University with a BSc in mechanical engineering in 2007. He subsequently awarded MSc and PhD in 2010 and 2014.