Information Sciences Letters

Volume 12
Issue 1 *Jan. 2023*

Article 27

2023

An Enhanced Population Selection Algorithm for Timetabling System

B. Zaqaibeh
Department of Computer Science, Faculty of Science and Information Technology, Jadara University, Jordan, zaqaibeh@jadara.edu.jo

Information Sciences Letters
An International Journal

http://dx.doi.org/10.18576/isl/120127

An Enhanced Population Selection Algorithm for Timetabling System

B. Zaqaibeh

Department of Computer Science, Faculty of Science and Information Technology, Jadara University, Jordan

Received: 12 Jun. 2022, Revised: 12 Aug. 2022, Accepted: 4 Sep. 2022.
Published online: 1 Jan. 2023.

Abstract: The timetable preparation for systems that are based on credit hours is a challenge for heads of departments. Departments have to prepare their timetables, where the problem is increased when a department serves other departments with some courses. Also, university's management cannot assure that the offered courses are more than the needs or not. Many algorithms have been tested without an optimal solution. A new proposed algorithm which is called the Enhanced Population Selection (EPS) Algorithm has been implemented and tested with a suitable number of students, courses, lecturers, and venues that is based on the harmony search algorithm and genetic algorithm. The new proposed EPS algorithm has scheduled the timetables for two semesters with academic advisors satisfaction without conflicts. Furthermore, all specified constraints are tested and satisfied.

Keywords: Timetabling, Bees algorithm, harmony search algorithm, and genetic algorithm.

1 Introduction

The problem of getting an optimal timetable is common in universities. The heads of departments are suffering from timetable preparation for their students and lecturers in addition to exams invigilation. Preparing an optimal timetable means finding suitable timeslots and appropriate venues for students, lecturers, and invigilators that are fitted students, courses, venues, and lecturers under the satisfaction of a set of constraints.

Academic departments have varied and wide-range rules that constrain the timeslots distributions. In this paper, three types of constraints are considered which are hard constraints, soft constraints, and general constraints.

Researchers have introduced many algorithms to solve timetabling problems. The previous proposed algorithms show a high performance in many cases but they are not good enough to get an optimal solution. Furthermore, researchers have presented several optimization methods for some algorithms like bee's algorithm, ant colony algorithm, harmony search algorithm, genetic algorithm, and many others.

Algorithms are diverse and each one has input, process, and output as Figure 1 shows. An algorithm may perform better if it is reviewed and optimized according to the proposed constraints. In general, optimization is the process of making something better [1] but the better direction should be specified.

Optimization refers to find a better output for a specific input in such a way that the performance is achieved as expected under some constraints. Optimizing algorithms varies from one algorithm to another, but in mathematical definitions, it refers to the best or worst one by varying the input and processes. Generally, the aim of the optimization that is done on harmony search algorithm is to find an optimal solution for a determine problem.

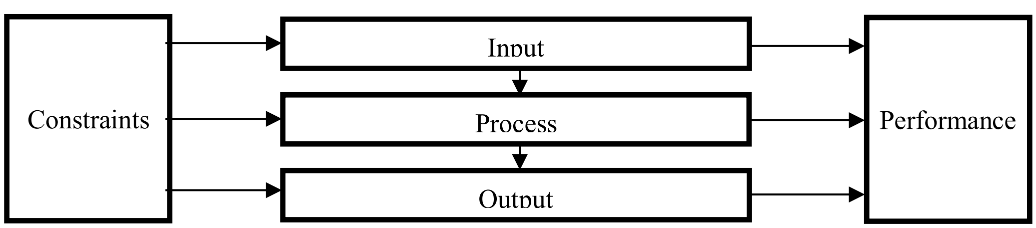

Fig. 1. Processes in Algorithms

This paper is organized as follows: section two shows the literature review and related work, section three represents the methodology of the proposed algorithm, section four shows the results, and the conclusion and future work is written in section five.

*Corresponding author e-mail: zaqaibeh@jadara.edu.jo

2 Preliminary

Universities run their systems in different ways; a university may use the year-based system, semester intake system, or credit hour system. The timetables in year-based system and semester intake system are almost fixed and can be controlled easily comparing with credit hours systems. In the credit hours system, students can choose courses with their timeslots and lecturers especially when a course has multi sessions. Therefore, a timetable should be prepared optimally to suite all students desires, in the meanwhile, the selected courses must be selected according to syllabi and to serve students ongoing study with considerations to time conflicts for both students and lecturers.

The timetabling problem is considered as Nondeterministic Polynomial (NP) problem [2][3], and there are many algorithms that are proposed to find an optimal solution. Researchers like Pham [4] and his colleagues have introduced the Bees Algorithm models and described it in its basic formulation that increased the accuracy of the search. The Bees Algorithm performed optimally in some cases. Nevertheless, in [5] researchers have descripted an experimental investigation into four nature-inspired population-based continuous optimization methods which are: The Bees Algorithm, Evolutionary Algorithms, Particle Swarm Optimization, and the Artificial Bee Colony algorithm. The aim of their study is to compare specific capabilities of each optimized algorithm. The experimental has highlighted the strengths and weaknesses of these algorithms, also configuration is tested.

In [6], the authors have proposed a bee colony optimization algorithm as part of swarm intelligence. A hybrid bee colony optimization algorithm is presented for examination timetabling problems. Also, they introduced three selection strategies which are tournament, rank, and disruptive selection strategies. The disruptive selection strategy outperforms tournament and rank selections. They also introduce a self-adaptive mechanism to select a neighborhood structure to enhance the neighborhood search.

Several versions of the Bees Algorithm have some drawbacks [7] as it requires a large number of parameters to be set, lack of methodology for parameter setting, and computational complexity. In [7], bees are grouped to search different sites with different neighborhood sizes rather than discovering two types of sites, and benchmark functions for 12 well-known algorithms are presented.

An experimental study on the Bees Algorithm is presented in [8] where its strengths and weaknesses are analyzed. The algorithm is tested on 18 functions. The results obtained on the benchmarks prove the effectiveness and robustness of the bees foraging metaphor. A solution to the problem is defined by a given number of parameters and cost via a fitness function.

Recently, metaheuristic algorithms came to solve timetabling problem like the harmony search algorithm [9]. The metaheuristic algorithms are divided into two types that are local based algorithm which starts with one solution and tries to satisfy the constraints iteratively based on a fitness function and the second type is a population based algorithm [9][10].

Harmony search algorithm is a relatively metaheuristic algorithm [11], it has been demonstrated in various real world problems like water distribution networks, energy-saving dispatch, bus routing problem, Sudoku puzzle, music composition - medieval organum, truss structure design, and timetabling. It is music-based and inspired by observing the aim of music and searching for an optimal state of harmony where the search process is compared to jazz music in [12].

The cellular harmony search is proposed in [13] on concern with individual perspective, the cellular automata provides a population of a particular structure formulated as a toroidal grid where the cellular automata concepts are embedded into the harmony search algorithm. In cellular harmony search, the population is arranged as a two-dimensional matrix, where each individual interacts with its neighbors. The individual who communicates with its closest neighboring individuals has the same number of neighbors.

Genetic algorithm is a search based algorithm that is based on the notions of natural selection. Genetic algorithm depends on the population to give possible solutions where population is a subset of proposed solutions for a specific problem. In genetic algorithm [1], fitness is calculated to indicate of optimal solution. A candidate solution is taken as an input and produces a solution as the output, the output replaces the existing individuals in the population if it can be fitted, this process is called fitness function or objective function [1][14]. The genetic algorithm uses three types of rules at each step to create the next generation from a given population which are:

- Selection rules which selects the individuals and called parents of the next generation.
- Crossover rules which combines two parents to form children for the next generation.
- Mutation rules which applies random changes to individual parents to form children.

The fitness value is required to be calculated iteratively and therefore it should be fast executed. The genetic algorithm has several operators like selection, crossover, and mutation. In crossover, more than one parent is selected and mutation is related to exploration search and is used to introduce population diversity [15]. The fitness function in [16] is calculated for each solution. This causes memory consuming while the population is increased, in addition to the system slow and less efficiency.

$$\begin{pmatrix} a_{0,0}^{1} & a_{0,1}^{1} & \cdots\cdots & a_{0,p}^{1} \\ a_{1,0}^{1} & a_{1,1}^{1} & \cdots\cdots & a_{1,p}^{1} \\ a_{2,0}^{1} & a_{2,1}^{1} & \cdots\cdots & a_{2,p}^{1} \\ \cdots & \cdots & \cdots\cdots & \cdots \\ a_{k,0}^{1} & a_{k,1}^{1} & \cdots\cdots & a_{k,p}^{1} \end{pmatrix} \begin{pmatrix} a_{0,0}^{2} & a_{0,1}^{2} & \cdots\cdots & a_{0,p}^{2} \\ a_{1,0}^{2} & a_{1,1}^{2} & \cdots\cdots & a_{1,p}^{2} \\ a_{2,0}^{2} & a_{2,1}^{2} & \cdots\cdots & a_{2,p}^{2} \\ \cdots & \cdots & \cdots\cdots & \cdots \\ a_{k,0}^{2} & a_{k,1}^{2} & \cdots\cdots & a_{k,p}^{2} \end{pmatrix} \cdots \begin{pmatrix} a_{0,0}^{HMS} & a_{0,1}^{HMS} & \cdots\cdots & a_{0,p}^{HMS} \\ a_{1,0}^{HMS} & a_{1,1}^{HMS} & \cdots\cdots & a_{1,p}^{HMS} \\ a_{2,0}^{HMS} & a_{2,1}^{HMS} & \cdots\cdots & a_{2,p}^{HMS} \\ \cdots & \cdots & \cdots\cdots & \cdots \\ a_{k,0}^{HMS} & a_{k,1}^{HMS} & \cdots\cdots & a_{k,p}^{HMS} \end{pmatrix}$$

Fig. 2. Harmony memory structures for timetabling representation

An algorithm has been proposed in [17] to find an optimal solution for a university course timetabling problem by using the harmony search algorithm and several matrixes as shown in Figure 2, this leads to use a huge space in the memory which effects negatively on the performance of the system and its efficiency. The proposed algorithm has used a matrix that was proposed in [18].

The university course timetabling is represented in a huge number of two-dimensional matrixes where they contains courses and rooms timeslot or -1 if it is empty.

The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection and the process that drives biological evolution. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation, they are utilized to build an automated course timetable system and the proposed timetabling system requires minimal effort from the administration staff to prepare the course timetable [14].

3 The Enhanced Population Selection Algorithm

The new proposed algorithm which is called the Enhanced Population Selection (EPS) algorithm is an upgrade of the OHSADS which was proposed in [19]. The EPS algorithm has succeeded to obtain an optimal solution for timetabling system by optimizing the harmony search algorithm and the genetic algorithm toward of distribution of courses with their timeslots, venues, and lecturers. Figure 3 shows the pseudocode of EPS algorithm.

To apply the EPS algorithm, settings must be specified first by creating Three Dimensional Matrix (3DM) and initialized with null values, determine the maximum number of courses (N) for the student to be enrolled in the semester, and determine the maximum number of lectures per day. Reading information is essential, the information is specified for Students (id, name, syllabi id), Syllabus (syllabi number, courses in each syllabi), Courses (course id, name, credit hours), Lecturers (ids, names, specialization, preferred courses, taught courses), and Venues (id, capacity, contents, location).

Step 1: Read the database views
Step 2: Analyze and optimizing data
Step 3: Store the required information
Step 4: Initialize the 3DM
Step 5: Store the pre-assigned blocked slots in the 3DM
Step 6: Store the pre-assigned courses in the 3DM
Step 7: Constraints checking
Step 8: Apply the algorithm
Step 9: Reporting

Fig. 3. The pseudocode of the EPS

Furthermore, the pre-definitions conditions have to be inserted into 3DM and will have the privileges: ability to specify some courses at a specific time, ability to block venues, allowing lecturers to select some courses, and allowing the possibility to prevent a specific lecturer from teaching at a certain time. In addition, courses have to be determined by reading and analyzing syllabi for each student, choosing the most needed courses by students, a way that takes into account the number of courses and the category to which they are belonging.

Syllabi has to be inserted into an intersection matrix, all elective courses are to be offered across the semesters, so that an elective course is offered at least once every four semesters, and all modified information have to be saved into the database, subsequently, data has to be analyzed and optimized using an intelligent optimization method then stored. The 3DM is used to classify courses and select required information depending on intended constraints.

Data is imported from the university database for all departments for the active semester where data could be students' information, courses with their sections, rooms' capacities, and lecturers' information. Day, Room, and Hour represent the 3DM dimensions as Figure 4 shows.

The populations are stored and represented in the 3DM with their distributions by the parameters D, R, and H which refer to Days, Rooms, and Hours respectively. D can have the values from 1 to the number of days per week, which is represented with $_d$, R represents the venues IDs which is represented with $_r$, and H can have the values from 1 to the number of hours per day which is represented with $_h$. To ease tracing of forward and backward selections conflicts among timeslots with constraints' satisfactions consideration are represented in a Conflict Matrix (CM).

The number of students who needed a specific course can be represented as an output of the CM, so data is optimized and adjusted to accumulate the minimum number of required courses for each student. Expected graduate students have a priority to be satisfied and their data is analyzed first by the algorithm. Therefore, the CM provides the real needed courses for each member where this happens during data preparation. Data is distributed and courses satisfied population with the consideration of satisfying an intended constraints.

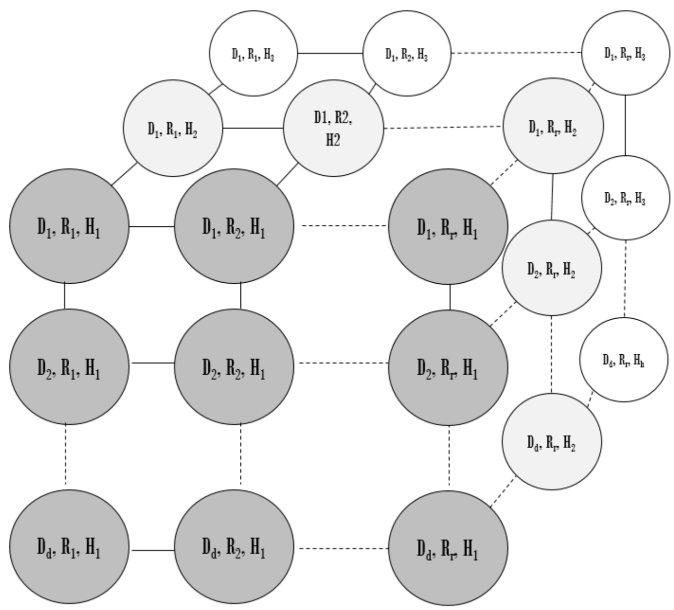

Fig. 4. Data Representation in the 3DM

In general, constraints control and direct timetable slots in each generation production, and this reflects positively to reduce data conflicts and relationships. The EPS allows pre-defined constraints to be performed once the data is available to the system, and to ensure the efficiency of the system, the EPS stores its fitness. Satisfaction of the pre-defined constraints is required, and they must be considered before the timeslot in the timetable is processed. The following pre-defined constraints are considered in the EPS.

- Specific timeslots can be defined to have no lectures on the university, faculty, or department level, this to have extracurricular activities.
- A course can be booked to be offered on a selected time or date.
- A venue can be booked for some lectures.
- A lecture can be pre-assigned to be offered in a time, date, or venue.
- A Lab can be pre-assigned to a time, date, or venue.
- A lecturer can select a suitable slot for his/ her lecture or/and avoid a specific slot.
- A time, date, or venue can be booked for some activities.

The fitness of the pre-defined constraints has values between 1 and 7, the fitness of the pre-defined constraints will be calculated as Figure 5 shows, where *pre* represents the number of pre-defined constraints, the fitness for each slot after checking all constraints will be stored in *Fit1*.

Efficient timetables should satisfy their major constraints where the major constraints are mandatory conditions that direct the courses distribution over the timetable. Some items are to be considered when generation courses timetable such as: courses, students, lecturers, venues, timeslots, and constraints. The following constraints are considered during the timetable generation:

- Courses may be split into multi-sections, if a course is needed by several students that their number exceeds a venue capacity.
- If a course is set as pre-scheduled course in a day, time, or venue then the course must be offered in the timetable as set.
- If a course is assigned to a lecturer as pre-scheduled course in a day, time, or venue then the course must be assigned to the lecturer as set.
- Each member must be offered at least five courses.
- Enough number courses are to be offered for expected graduate students.
- A student must not offered more than four courses per day.
- Number of members should suite venues capacities.
- Lecturers should take courses that suite their majors.
- A lecturer must not take more than a course in the same time and date.
- A lecturer should have number of courses that are correspond to their academic rank and duties.
- A venue must not have more than a lecture in the same time.
- Venues must suite type of teaching and learning methods.

Step 1: declare i, j, and *FitX*
Step 2: define c and *Cons*
Step 3: initialize i to 1
Step 4: initialize j to 0
Step 5: increase j
Step 6: get C_j
Step 7: call constraint checking procedure
Step 8: count satisfied constraints
Step 9: if j less than *Cons* then goto step 5
Step 10: increase i
Step 11: store the fitness value in *FitX*
Step 12: if i less than *c* then goto step 4
Step 13: return the *FitX*

Fig. 5. Calculating constraints' fitness

Constraints have to be checked against the generated timetable during its generation. The fitness of the major constraints will be calculated as shown in Figure 5 with respect to its required parameters, where *h* represents the number of major constraints, and *Fit2* will store the fitness for the major constraints.

To generate an optimal courses timetable, an additional enhanced constraints should be considered. The enhanced constraints are in the contrary of pre-defined and major constraints which can be unsatisfied if required such as:

- Courses with multi-sections have to be proposed in different time, this to offer more choices for students and prevent lecturers timeslots conflicts.
- A student should not have more than three courses per day.
- Timeslots should be distributed within close intervals with the consideration of students' desires.
- Lecturers may select a preferable time.
- A course leader is selected for multi-section courses.

- Courses should be distributed fairly among lecturers.

Constraint has to be checked after the timetable is created, the fitness of enhanced constraints is calculated as shown in Figure 5, where *e* represents the number of constraints, *Fit3* stores the fitness for each slot after checking all constraints.

Fitness

As mentioned earlier, constraints are used to calculate the fitness. If all courses are scheduled with the satisfaction of all constraints and have the maximum fitness, this means an optimal solution is obtained. Nevertheless, if a course is not scheduled at all, it will be recorded in the unscheduled table. Furthermore, the forward and backward selections method will be called and rescan the schedule for enhancing population selection by:

- Scan the timetable to look for gaps
- Balancing the courses timeslots by replacing courses that have alternating periods for a group of students.
- Reschedule the unscheduled courses by using select, test, and replace algorithm.
- Check the constraints satisfaction and recalculate their fitness

The fitness value of the timetable is the average of fitness of its constraints as formula (1) shows. The results are represented by floating point number between 0 and 1.

$fitness = \text{Average}(Fit1 + Fit2 + Fit3)$ ……… (1)

The fitness is evaluated after the population is created. The objective of the EPS is to reach 1 as a final fitness value for the final timetable.

Implementation

The EPS provides all possible distributions of publications and this leads to get a suitable timetable for students and lecturers. As shown in Figure 2, the likelihood slots are selected depending on the constraints' satisfaction and the cross matrixes that is shown in Table 1 among students for similar needed courses according to their curriculum and in Table 2 among selected courses.

Table 1: The cross matrix among students and their curriculum

Std/ Sub	=77	=107	101099	101111	301433	302102	302211	302342	505105	……	302411
201620389	1	1	1	1	1	1	1	0	0	…	1
201720085	0	1	1	1	0	0	1	0	1	…	1
201720275	1	0	0	1	1	1	1	0	1	…	1
201810322	0	0	0	1	1	0	1	1	1	…	0
202010332	1	1	1	1	1	1	0	0	1	…	0
202010344	1	0	0	1	0	1	1	0	1	…	0
202010452	1	1	0	1	1	1	0	0	0	…	0
…	…	…	…	…	…	…	…	…	…	…	…
202020132	1	1	1	0	0	0	0	0	1	…	0
Total	451	378	392	201	267	176	98	102	323	…	47

Courses with conflicts are represented with 1, if all values are stored in the matrix, this leads to use a large memory space, but EPS keeps only the values that are represented with 0's, subsequently, this leads to use less space in the memory and speeds up the data processing.

Table 2: The cross matrix among selected courses

Std/ Sub	=77	=107	101099	101111	301433	302102	302211	302342	505105	……	302411
=77	0	1	0	0	1	0	1	0	0	…	0
=107	0	0	1	0	1	0	0	0	1	…	0
101099	1	0	0	0	0	0	1	0	0	…	1
101111	0	0	0	0	1	0	1	1	0	…	0
301433	0	1	1	1	0	1	0	0	0	…	0
302102	0	0	0	1	1	0	1	0	1	…	0
302211	1	1	0	1	1	1	0	0	0	…	0
302342	0	1	0	0	1	0	0	0	1	…	1

505105	0	1	1	1	0	1	0	1	0	…	1
…	…	…	…	…	…	…	…	…	…	…	…
302411	0	0	0	0	1	0	1	0	1	…	0

Equivalent courses are considered and re-coded in a hatch table with a shared code among all courses that are from the same sets of equivalent courses, for more clarifications: The courses 301241, 301341, and 302343 are equivalent courses and coded to ‘=77’, the ‘=’ sign is added to a sequence number which is initialized to 1 and increased by 1 for each set of equivalent courses.

Courses are not expected to be fitted in the right position at the first try, therefore, the EPS checks courses against pre-assigned slots that are in the database.

Once the EPS generates the timetable schedule, the selection method checks the multiple timeslots against the constraints to select the optimal timeslots. The timeslots are sorted in descending order where the less number of possibilities are processed first, subsequently, the timetable is modified accordingly. Reports are produced to represent the timetable and a summary on its courses and their sections, lecturers, venues, timeslots, and relationships.

4 Results

The proposed EPS is implemented and tested on Jadara Database using PHP software and MySQL Database and applied at Jadara University for 4122 students that have registered for 723 courses and lectured by 128 lecturers in about 98 venues that are 11 labs, 1 lecture hall and 86 lecture rooms as Figure 6 shows. The system has worked perfectly and scheduled all required courses. The fitness value is calculated which resulted 1 as a total fitness, where it proves an optimal solution is obtained and suited the university needs.

Input summary,,,
# of students: 4122
# of courses: 723
# of sections: 903
# of lecturers: 128
# of venues: 98
Output summary,,,
# of scheduled courses: 723
# of scheduled labs: 73
# of scheduled sections: 903
# of unscheduled courses: 0
# of unscheduled labs: 0

Fig. 6. Summary of input and output data

Lecturers and students are allowed to access the system and see their timeslots. Figure 7, shows the available timetable for a student.

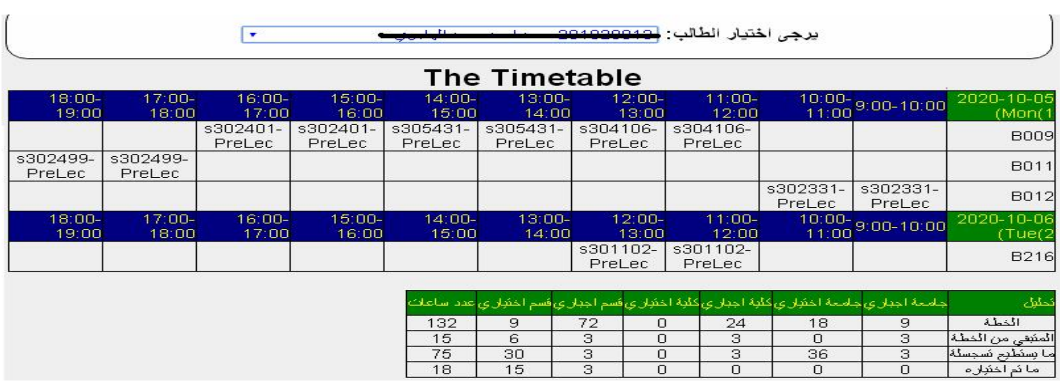

Fig.7. A student timetable

Courses are appeared according to the proposed timeslots, also students can see a statistic of their academic plan, so, students will know this is an optimal suggestion for their coming semester.

The whole timetable shows all lectures time and date, courses IDs, and venues. Table 3 shows the “=” sign beside the code means they are equivalent courses, and the empty cell means there are no lecture or lab in that venue on that date. Courses are represented by their IDs, and the comment under the course ID shows if the course is predefined of system generated.

Table 3: The timetable

Tue(2) …	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00
B009	s302211-PreLec	s302211-PreLec	s403444-PreLec	s403444-PreLec	s403214-PreLec	s403214-PreLec	s302461-PreLec	s302461-PreLec	s302421-PreLec	s302421-PreLec
B011	s302455-PreLec	s302455-PreLec	s305301-PreLec	s305301-PreLec	s302381-PreLec	s302381-PreLec	s302411-PreLec	s302411-PreLec	s403492-PreLec	s403492-PreLec
B012	s403481-PreLec	s403481-PreLec	=s301101 =s803113 =s831101-PreLec	=s301101 =s803113 =s831101-PreLec	s301433-PreLec	s301433-PreLec	s302102-PreLec	s302102-PreLec	=s832102 =s803111 =s302110-PreLec	=s832102 =s803111 =s302110-PreLec
B216	s403262-PreLec	s403262-PreLec			s403361-PreLec	s403361-PreLec	s302462-PreLec	s302462-PreLec	s302371-PreLec	s302371-PreLec
B217	=s302373 =s306373-PreLec	=s302373 =s306373-PreLec	s306491-PreLec	s306491-PreLec	s302372-PreLec	s302372-PreLec	s800203-PreLec	s800203-PreLec	s403343-PreLec	s403343-PreLec
B218	=s841322 =s403322-PreLec	=s841322 =s403322-PreLec	s308201-PreLec	s308201-PreLec	s308402-PreLec	s308402-PreLec	s306483-PreLec	s306483-PreLec	s308404-PreLec	s308404-PreLec
B219	=s853106 =s807106 =s505104-PreLec	=s853106 =s807106 =s505104-PreLec	s308301-PreLec	s308301-PreLec	s403393-PreLec	s403393-PreLec	s308499-PreLec	s308499-PreLec	s403382-PreLec	s403382-PreLec
B307	s306320-Lecture	s306320-Lecture	s302402-Lecture	s302402-Lecture	s306121-Lecture	s306121-Lecture	=s833102 =s803105 =s304105-Lecture	=s833102 =s803105 =s304105-Lecture	s403291-Lecture	s403291-Lecture

The deanship of admission and registration and heads of departments can get an excel file as shown in Table 4.

Table 4: An excel copy of the timetable

- 11:00 13:00	الاقتصاد الكلي	304106	المحاسبة	أحد، ثلاثاء، خميس
- 17:00 19:00	ادارة الاعمال الالكترونية + ادارة الاعمال الالكترونية	302423 + 306423	المحاسبة	أحد، ثلاثاء، خميس
- 11:00 13:00	مبادىء القانون التجاري	403141	المحاسبة	أحد، ثلاثاء، خميس
- 09:00 11:00	ادارة الموارد البشرية	302331	المحاسبة	أحد، ثلاثاء، خميس
- 15:00 17:00	السلوك التنظيمي	302261	المحاسبة	أحد، ثلاثاء، خميس
- 09:00 11:00	احصاء لطلبة الاعمال	505105	المحاسبة	أحد، ثلاثاء، خميس
- 15:00 17:00	التخطيط والرقابة الادارية	302481	المحاسبة	أحد، ثلاثاء، خميس
- 17:00 19:00	مبادىء الادارة المالية (1) + مبادئ الادارة المالية + مبادىء الادارة المالية	803104 + 833101 304101 +	المحاسبة	أحد، ثلاثاء، خميس
- 09:00 11:00	مبادىء الادارة (1) + مبادئ الادارة + مبادىء الادارة	803101 + 832101 302101 +	المحاسبة	أحد، ثلاثاء، خميس
- 11:00 13:00	مبادىء التسويق + مبادئ التسويق + مبادىء التسويق	803107 + 305101 835101 +	المحاسبة	أحد، ثلاثاء، خميس
- 09:00 11:00	الاساليب الكمية في الادارة	302211	المحاسبة	سبت،اثنين، اربعاء
- 11:00 13:00	ادارة التسويق	305301	المحاسبة	سبت،اثنين، اربعاء
- 11:00 13:00	مبادىء المحاسبة (1) + محاسبة (1) بالانجليزية(طلبة 2013 فما بعد) + مبادىء المحاسبة (1) بالانجليزية	803113 + 301101 831101 +	المحاسبة	سبت،اثنين، اربعاء
- 13:00 15:00	المحاسبة الادارية بالانجليزية	301433	المحاسبة	سبت،اثنين، اربعاء
- 17:00 19:00	اساليب البحث العلمي لطلبة الاعمال + اساليب البحث العلمي (طلبة الاقتصاد والاعمال) + اساليب البحث العلمي (طلبة الاقتصاد والاعمال)	803111 + 832102 302110 +	المحاسبة	سبت،اثنين، اربعاء

Surveys are conducted for admin staff in the deanship of admission and registration, head of departments, lecturers, and students to get their satisfaction with the system. The results are impressive.

5 Conclusion and future work

This paper has introduced the EPS algorithm and its result, the EPS has provided an optimal solution for solving the timetabling problem. Students can see their proposed courses without time conflict, with minimum number of lectures per day, and suitable distributed time. Basically, testing the algorithm on a huge data assures its performance and efficiency.

More work can be done on students' desired and preferable time and date. In addition, allowing students to exclude specific timeslots for reasonable excuse.

Conflict of interest

The authors declare that there is no conflict regarding the publication of this paper.

References

[1] J. Lee, *Genetic Algorithms in Java Basics*, Apress, 1st edition, (2015).

[2] R. Michael, Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York - United States, 338, (1990).

[3] A. Bettinelli, V. Cacchiani, R. Roberti and P. Toth, *An Overview of Curriculum-based Course Timetabling*, Springer Berlin Heidelberg, 23, 2, 313-349, (2015).

[4] D. T. Pham, M. Castellani, *The Bees Algorithm: Modelling Foraging Behaviour to Solve Continuous Optimization Problems*, Journal of Mechanical Engineering Science, 12, 2919-2938, (2009).

[5] D. T. Pham and M. Castellani, *Benchmarking and Comparison of Nature-inspired Population-Based Continuous Optimisation Algorithms*, Soft Computing, 18, 871–903, (2014).

[6] M. Alzaqebah, S. Abdullah, *Hybrid Bee Colony Optimization for Examination Timetabling Problems*, Computers and Operations Research, 54, 142-154, (2015).

[7] H. Nasrinpour, A. Bavani, and M. Teshnehlab, *Grouped Bees Algorithm: A Grouped Version of the Bees Algorithm*, Computers, 6, 1, (2017).

[8] D. Pham and M. Castellani, *A Comparative Study of the Bees Algorithm as a Tool for Function Optimization*, Cogent Engineering, 2. 1, 1–28, (2015).

[9] A. Alomoush, A. Alsewari, H. Alamri, K. Aloufi, and K. Zamli, *Hybrid Harmony Search Algorithm with Grey Wolf Optimizer and Modified Opposition-based Learning*, IEEE Access, 7, 68764-68784, (2019).

[10] C. Blum and A. Roli, *Metaheuristics in Combinatorial Optimization: Overviewand Conceptual Comparison*, ACM Computing Surveys, 35, 3, 268–308, (2003).

[11] X. Yang, *Harmony Search as a Metaheuristic Algorithm*, Studies in Computational Intelligence, 191, 1-14, (2009).

[12] Z. W. Geem, J. H. Kim, and G. V. Loganathan, *A new Heuristic Optimization Algorithm: Harmony Search*, Simulation, 76, 2, 60–68, (2001).

[13] M. Al-Betar, A. Khader, M. Awadallah, M. Alawan, and B. Zaqaibeh, *Cellular Harmony Search for Optimization Problems*, Journal of Applied Mathematics, 20, (2013).

[14] M. Al-Jarrah, A. Al-Sawalqah, and S. Al-Hamdan, *Developing a Course Timetable System for Academic Departments using Genetic Algorithm*, Jordanian Journal of Computers and Information Technology, 3, 1, (2017).

[15] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham, *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*, Chapman and Hall, 1st Edition, (2009).

[16] M. Chiarandini, M. Birattari, K. Socha, *An effective hybrid algorithm for university course timetabling*. Journal of Scheduling, 9, 5, 403-432, (2006).

[17] M. Al-Betar, A. Khader, and T. AbdulGani, *A Harmony Search Algorithm for University Course Timetabling*, Annals of Operations Research, 194, 1, 3–31, (2012).

[18] R. Lewis, *Metaheuristics for university course timetabling*, PhD thesis, School of Computing, Napier University, Edinburgh, (2006).

[19] K. Alomari, O. Almarashdi, A. Marashdh and B. Zaqaibeh, *A New Optimization on Harmony Search Algorithm for Exam Timetabling System*, Journal of Information and Knowledge Management, World Scientific Publishing Co., 19, 1, (2020).