Journal of Engineering and Applied Sciences 13 (23): 9976-9984, 2018
ISSN: 1816-949X

Construction a MATLAB Program to Solving the Timetable Scheduling Problem

Wasan Saad Ahmed Saad Qasim Abbas
Diyala University, Baqubah, Iraq

Abstract: This study presents algorithm for generating the university timetabling problem. The inputs are the classes, teachers and teacher availability as well as other parameters which can be input into the program. This will give schools more flexibility to design the optimum table lessons. The run time of the program is improved by applying conditions that reduce the time needed to give the same result. It is shown that the time complexity of algorithm to generates the desired D×H table with classes and L levels is $0 = (DH)^L \prod_L S!S^L$. The program was tested using the input data from the Troy University (USA) Mathematics Department which included a large number of courses and teachers and resulted in more than one solution.

Key words: Time table scheduling problem, schedule classes, MATLAB, input data, optimum table, flexibility

INTRODUCTION

The idea of this study is to solve the teacher resource allocation problem (which is known as the time tabling problem) for many universities. There are a significant number of teachers who donot have all the time to teachbecause they have other activities like meetings, advising graduate students, doing research, etc. or performing different school relate dactivities or functions. In addition, the student's ability to understand decreases over time and fewer lessons in theory are needed later in the day (Wile and Shouppe, 2011). These scheduling constraints provided the stimulus to design a program giving the user the optimum weekly class schedule which can provide more than one choice to school administrators to provide the best schedule for both teachers and students. In addition, the programs offer the possibility for the school to reduce the number of classrooms needed by estimating the minimum number of required classrooms.

The programs run in MATLAB (R 2017a) and the run time varies from a few seconds to a couple of minutes (unless one wants more than 10900 solutions that can reach up to 7.5 h) to give the result. The program comes with default inputs but the user can adjust them to meet their needs. The output of this program is a table or multiple tables of weekly class schedules. This output is saved into a text file and arranged into tables. The specific table number is dependent on the value of input n, the solution number. The user can then choose which table from the set of solutions is the best fit.

The time table problem is one of the NP-complete problems (Even *et al*., 1976) meaning that there is no general solution found, so far that can solve the time table problem in a reasonable time when the volume of data becomes very large.

The timetable problem has been investigated during the last four decades. A good survey can be found by De Werra (1985). Schmidt and Strohlein (1980) describe contributionsof this problem prior to 1980. Many approaches have been proposed to solve the timetable problem. Welsh and Powell (1967) point out the connection between the coloring of the vertices graph and time table problem while De Werra (1997) gives various formulations in terms of the coloring problem in graph theory.

By Lawire (1969) used an integer programming approach to find the solution of the time table problem; this research is considered to be the earliest work in this field. Additionally, Akkoyunlu (1973) formulated the problem in terms of cost. By Bakir and Aksop (2008) solved the problem as 0-1 integer programming. Hultberg and Cardoso (1997) and Badri *et al*. (1998) modeled the time table problem as amixed integer program.

Heuristics techniques invesigated by Burke *et al*. (2003) and Alvarez-Valdes *et al*. (1997) Tabu search. Genetic algorithem research has been done by Carrasco and Pato (2000). Chu and Fang (1999) provide comparisons between the genetic algorithm and tabu search approaches and shows the latter to produce the better result.

Abramson (1991) used Simulated Annealing (SA) to find the solution using the Monte Carlo scheme as an optimization technique. Additionally, Abramson (1999) compared the performance of six different SA cooling schedules.

A constraint-based reasoning technique was used by Fen *et al*. (2009). Deris *et al*. (1997) applied this techniqe to university timetable planning. Also, Deris *et al*. (2000) implemented an object-oriented approach. A hybird algorithem combines two or more than one previous

techniqes as shown by Gunawan *et al.* (2007) which combines an integer programming approach, a greedy heuristic and a modified simulated annealing algorithm collaboratively to solve the problem. Fen and Ho (2009) developeda hybrid algorithm consisting of a particle swarm optimization and constraint-based reasoning in generating a feasible and near-optimal solution.

The details of the program: To understand how the program works, suppose a given school hasthe following parameters: n teachers, subjects {Teach 1, Teach 2, ..., Teach n}, m subjects {Subj 1, Subj 2, ..., Subj m} and the user wants to develop a schedule for them. As there are different starting days for work, dependingon the country or the policy of the institution, the user can select one day among the following days: Sunday, Monday or Tuesday as the first business day of the week fdow = {Sat, Sun, Mon}) enter the number of business days per week (bd) and the Number of lecturesor Classes per Day (NCD).

Also, the user can set the Duration of Class (DC) and the Duration Between Classes (DBC); The user should also set the start time of the first lecture (STC) or leave it to the default setting (8:00).If the school has more than one class at the same time of day, then the Number of Grades (NOG) is set to 2, 3 or more.

Before the program runs, the user can set the Number of Solutions (NOS) to 1or more but the user should remember that as the number of solutions increases, the required time for acquiring the solutions will be longer. However, the running time will still be reasonable if the (NOS) is below 100 which are morethan enough solutions from which the user may choose the best fit.

The Number of Halls (classrooms) (NOH) should be chosen to cover all the given lectures or classes. One advantage of using this program is that the user can start from the smallest estimated number of halls (which covers all the given lectures) to see if they can acquire the solution; If not, then the user increases the number until the solution is achieved. This step allows using the minimum number of halls to cover all given lectures and this will enable them to keep some halls to be used in another school's activities without affecting the school's daily work. To better understand how the program works, see the following example:

Suppose the school has 10 teachers, 5 classrooms and 4 groups of classes tobe given in the same time (i.e., 4 grades) with 5 subjects in every grade and the administration of the school wants to arrange them into a table of weekly lessons. Every teacher comes in on a specific day and on each day there are specified hours in which the teacher can teach as shown in Table 1.

The information in Table 1 should be input to the program in the same way it is written in this example. The upcoming days indicatethe days in which the teacher is available or is on campus doing another activity. The business days are referred to as Mon, Tue, ..., Fri. as an abbreviation to Monday and Tuesday, etc. The default first business day in the program is Monday and as mentioned, the user could change it to Saturday or Sunday. Also, the last business day is Friday and so, the number of business days (bd) should be 5. However, the user could change it to 6 or more if needed to extend the work week and in this case the last business day would be Saturday. The 0 in the table means that the teacher is not available to teach on that given day.

The "teaching hours" in Table 2 indicate the hours in which the teacher is available to teach. For example, the numbers in the table (1234007) referto the coming hours of the first teacher (Teach1) on Monday.This teacher can teach from the first working hour until the end of the day (for example, starting at 8:00 and ending at 3:00 pm) except

Table 1: Teacher's names with the upcoming days of each teacher and the teaching hours of each coming day

Teachers names	The available daysof the teacher				The teaching hours of each day			
Teach 1	Mon.	Tue.	Wed.	The.	1234007	0034560	1234567	1234560
Teach 2	Mon.	Tue.	The.	Fri.	1234000	1200560	1200560	0034567
Teach 3	Mon.	Tue.	Wed.	The.	1234567	1234567	0034567	1234007
Teach 4	Tue.	The.	Fri.	0	0034007	1234567	1200560	0
Teach 5	Mon.	Wed.	The.	Fri.	0034567	1234560	1234560	1234567
Teach 6	Mon.	Tue.	0	0	1350000	1200567	0034007	0
Teach 7	Mon.	Tue.	The.	0	1234560	1000060	1234567	0
Teach 8	Mon.	Tue.	0	0	0034007	1234560	0	0
Teach 9	Tue.	Wed.	0	0	0034007	1234567	1200567	0
Teach 10	Mon.	Tue.	Wed.	The.	0034507	1234567	0234567	1204060

Table 2: The names of the hallswith the available days for each hall and the available hours each day

Hall names	The available days of the hall				The available hours each day			
H_1	Mon.	Tue.	The.	Fri.	1234560	1234560	0000000	1234560
H_2	Tue.	Wed.	The.	Fri.	1230000	1234000	1200560	0034567
H_3	Mon.	Tue.	Wed.	The.	1234000	1234567	0034567	1234007
H_4	Mon.	Wed.	The.	0	0034567	1234567	1234000	0
H_5	Mon.	Wed.	The.	Fri.	0234567	1234560	1234560	1234567

Table 3: The subjects with teachers and duration of each subject for all schoolgrades

Subjects	Teachers	No. of hours
Grade one		
Subj 1 I	Teach 1	2
Subj 2 I	Teach 2	3
Subj 3 I	Teach 4	2
Subj 4 I	Teach 10	3
Subj 5 I	Teach 5	2
Grade two		
Subj 1 II	Teach 7	1
Subj 2 II	Teach 6	1
Subj 3 II	Teach 3	3
Subj 4 II	Teach 7	2
Subj 5 II	Teach 8	1
Grade three		
Subj 1 III	Teach 3	1
Subj 2 III	Teach 5	1
Subj 3 III	Teach 4	3
Subj 4 III	Teach 6	2
Subj 5 III	Teach 3	1
Grade four		
Subj 1 IV	Teach 7	1
Subj 2 IV	Teach 9	1
Subj 3 IV	Teach 4	3
Subj 4 IV	Teach 1	2

the 5 and 6th h (which are the zeros in the number 1234007). The 0 in the table indicates that the teacher will not teachat this hour because they are not available.

The information of the halls (classrooms) should also, be input to the program with the days and hours in which these halls will be available to be used for teaching in the week as Table 2.

In Table 2, the first row represents the first hall (classroom) with all necessary information whereas the second till the fifth column represents the days in which the classes will be taught which are Monday-Friday. The last four columns represent the hours of these days in which the halls or classrooms are available. Since, there are four grades all the subjects should be input with the teachers who teach them and the duration of each subject in all of these grades as shown in Table 3.

After running the program, the user will have the tables of weekly lessons. The number of these tables depends on the value of the Numbers of Solutions (NOS) that the userinputs and the number of similar tables. For example, if the Numbers of Solutions (NOS) is 5 and two of these tables are identical, the output of the program will only be 4, if 2 of the 5 tables are identical, then the output will be 3.

In our example, we set NOS = 2, fdow = 'Mon', wh = 7, bd = 5, LC = 0:40, LDBC = 0:10, STC = 0:30, NOL = 4, NOH = 5, NOS = 20 and the output for the program will be two tables. The output is shown in Table 4.

In Table 4 notice that there is an arrangement of 10 teachers and in the non-empty cells of the table for example (Subj 4-Teach 1 (H_3)), the name of the subject is shown (Subj 4) beside the name of the teacher (Teach 1). The name of the hall (H_3) in the table shows that the start time of the work is 8:30 am.

The school can select the best and most appropriate schedule. This depends on many factors such as the availability of the teachers and the classrooms in the school building. Also, the nature of the subject taught may be considered. For example, subjects of a theoretical nature should be instructed during the time of day in which the brain is in the best degree of understanding which is usually in the morning (Wile and Shouppe, 2011). And sometimes planning should consider the nature of classes to help the studentstay focused andavoid distraction.

Because of the difficulty in MATLAB in dealing with characters (letters), all the characters and names in the programs are converted to numbers, treated inside the program and then returned to characterforms after making the necessary operations to achieve the solution. The algorithm of the program is listed below.

Algorithm 1:

1. Input the values of the following: The first work day (fdow), number of business days (bd), Number of Classes in the Day (NCD), Duration of lectures (Classes (DC), Duration Between two Classes (DBC) Start Time of first Class (STC), Number of Grades (NOG), Number of Solutions (NOS), the number of Halls, the names of teachers with the coming days (or available days) and hours of availability in each day, the names of halls with the available days and hours in each day as well as the subjects and teachers with the duration of each subject at all grade levels
2. Convert the Teachers, Halls and Lectures from names to number
3. Set the value grade to 1
4. While the gradeis less than or equal to thenumber of grades, do steps 5-14
5. Set the value of subject to 1
6. While the subject is less than or equal to the number of classes in the grade, do steps 7-14
7. Set the value ofthe table's locations i to 1
8. While the table's locationi is less than or equal to the last location (do steps 9-14
9. If the table's location i is empty and the teacher who teaches the subject isavailable, place the subject and name of teacher in location i
10. Set the value of hall to 1
11. While the value of the hall is less than or equal to the number of halls, do steps 12-13
12. If the hall is empty, place its name in location iof the table
13. Else (hall is not empty), increase the value of the hall by 1
14. Else (the table location i is not empty or the teacher who teaches the subject isnot available), increase the value of the table's location by 1
15. Stop

Note that, the program needs to check all the table's locations (bd×NCD locations) even if the teachers are not available on that day. For example in Fig. 1, suppose the program wantsto evaluate Subject1. The program will start

Table 4: The number of possible schedles is 2

Days	08:30-09:10	09:20-10:00	10:10-10:50	11:00-11:40	11:50-12:30	12:40-13:20	13:30-14:10
Mon.	Subj 1 I-Teach 1 (H_1)	Subj 1 I-Teach 1 (H_1)					
Mon.	Subj 1 II-Teach 7 (H_3)	Subj 2 II-Teach 6 (H_3)	Subj 3 II-Teach 3 (H_1)	Subj 3 II-Teach 3 (H_1)	Subj 3 II-Teach 3 (H_1)		Subj 5 II-Teach 8 (H_4)
Mon.		Subj 1 IV-Teach 7 (H_5)					
Mon.						Subj 1 III-Teach 3 (H_1)	Subj 2 III-Teach 5 (H_5)
Tue.			Subj 2 IV-Teach 9 (H_2)	Subj 4 IV-Teach 1 (H_3)	Subj 4 IV-Teach 1 (H_3)		
Tue.	Subj 4 III-Teach 6 (H_1)	Subj 4 III-Teach 6 (H_1)					
Tue.			Subj 3 I-Teach 4 (H_1)	Subj 3 I-Teach 4 (H_1)	Subj 4 I-Teach1 0 (H_1)	Subj 4 I-Teach 10 (H_1)	Subj 4 I-Teach 10 (H_3)
Tue.							
Wed.							
Wed.							
Wed.							
Wed.							
The.	Subj 3 III-Teach 4 (H_3)	Subj 3 III-Teach 4 (H_3)	Subj 3 III-Teach 4 (H_3)	Subj 5 III-Teach 3 (H_3)			
The.	Subj 4 II-Teach 7 (H_2)	Subj 4 II-Teach 7 (H_2)					
The.				Subj 3 IV-Teach 4 (H_4)	Subj 3 IV-Teach 4 (H_2)	Subj 3 IV-Teach 4 (H_2)	
The.							
Fri.			Subj 2 I-Teach 2 (H_1)	Subj 2 I-Teach 2 (H_1)	Subj 2 I-Teach 2 (H_1)	Subj 5 I-Teach 5 (H_1)	Subj 5 I-Teach 5 (H_2)
Fri.							
Fri.							
Fri.							
Mon.	Subj 1 I-Teach 1 (H_1)	Subj 1 I-Teach 1 (H_1)					
Mon.	Subj 1 II-Teach 7 (H_3)	Subj 2 II-Teach 6 (H_3)	Subj 3 II-Teach 3 (H_1)	Subj 3 II-Teach 3 (H_1)	Subj 3 II-Teach 3 (H_1)		Subj 5 II-Teach 8 (H_4)
Mon.		Subj 1 III-Teach 3 (H_5)	Subj 2 III-Teach 5 (H_3)				
Mon.			Subj 1 IV-Teach 7 (H_4)				
Tue.	Subj 3 I-Teach 4 (H_1)	Subj 3 I-Teach 4 (H_1)	Subj 4 I-Teach 10 (H_1)	Subj 4 I-Teach 10 (H_1)	Subj 4 I-Teach 10 (H_3)		
Tue.							
Tue.	Subj 4 III-Teach 6 (H_1)	Subj 4 III-Teach 6 (H_1)		Subj 4 III-Teach 6 (H_1)	Subj 4 III-Teach 6 (H_1)		Subj 4 III-Teach 6 (H_1)
Tue.			Subj 2 IV-Teach 9 (H_2)	Subj 4 IV-Teach 1 (H_3)	Subj 4 IV-Teach 1 (H_3)		
Wed.							
Wed.							
Wed.							
Wed.							
The.							
The.	Subj 4 II-Teach 7 (H_2)	Subj 4 II-Teach 7 (H_2)					
The.	Subj 3 III-Teach 4 (H_3)	Subj 3 III-Teach 4 (H_3)	Subj 3 III-Teach 4 (H_3)	Subj 5 III-Teach 3 (H_3)			
The.				Subj 3 IV-Teach 4 (H_4)	Subj 3 IV-Teach 4 (H_2)	Subj 3 IV-Teach 4 (H_2)	
Fri.	Subj 2 I-Teach 2 (H_1)	Subj 2 I-Teach 2 (H_1)	Subj 2 I-Teach 2 (H_1)	Subj 5 I-Teach 5 (H_1)	Subj 5 I-Teach 5 (H_2)		
Fri.							
Fri.							
Fri.							

with the first location l_{11} and determine if the teacher is available during the 1st h of the 1st day. If the teacher is available, then the program will enterthe information (the name of the subject and teacher) in this location. If

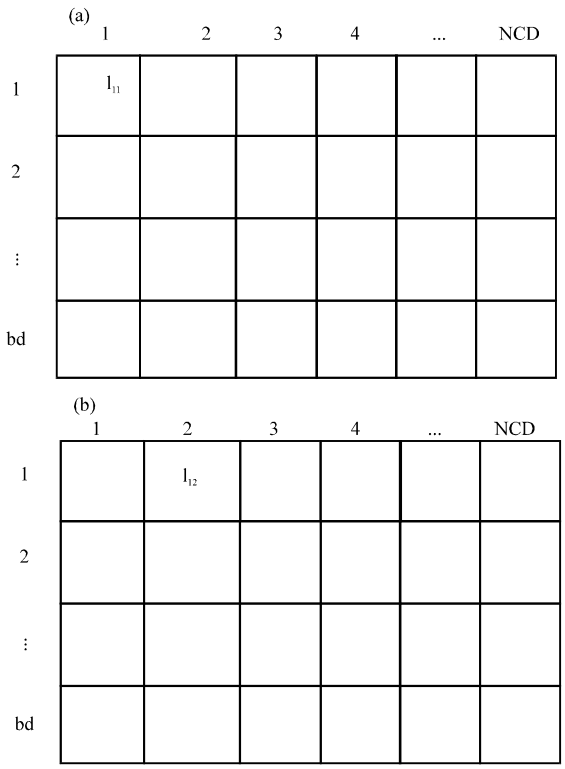

Fig. 1: The program will start with the location l_{11} and l_{12}

the teacher is not available, then the program will move to the next location l_{12} and check if the teacher is available as shown in Fig. 1.

If the teacher is available, the program will add the teacher's name and subject to the location. If the teacher is not available, then the program moves to the next location and checks the availability of the teacher in the next hour of the first day.

When the program moves to the next location it will begin to check the free time of the teacher in the second hour of the first day, so, if they are available, then the subject and the name will be added to this location. If not, then the program will move to the next location, l_{13}, re-evaluate and so on until the end of the first row is reached as shown in Fig. 2.

Each of the seiterations requirestime for the program to work. The next section describes advanced techniques used to reduce the time required to achieve the same result with less time.

The advanced technique: The idea of reducing the run time required of the program is important in the field of Applied Mathematics. There are two ways to reduce the runtime. The first is by setting a path for the program to follow when checking the teacher's availability. The

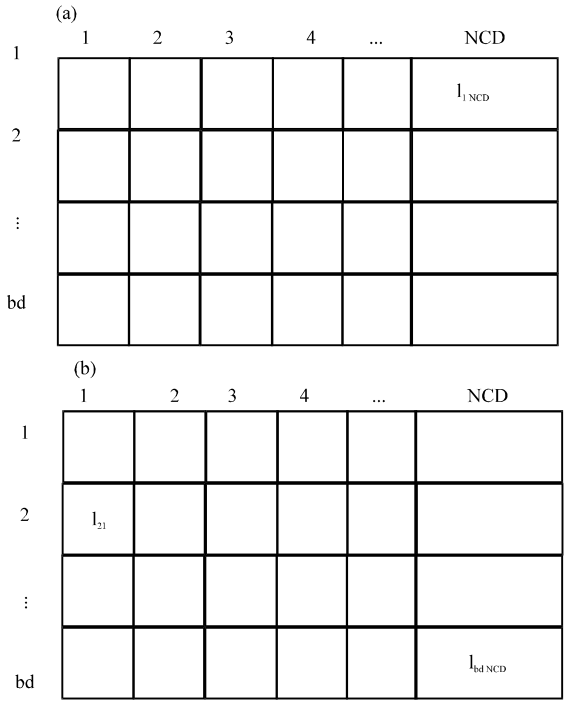

Fig. 2 a, b): If the teacher is not available, the program will move to the first location of the second column l_{21} and re-evaluate until it reaches the last location $l_{bd\ NCD}$

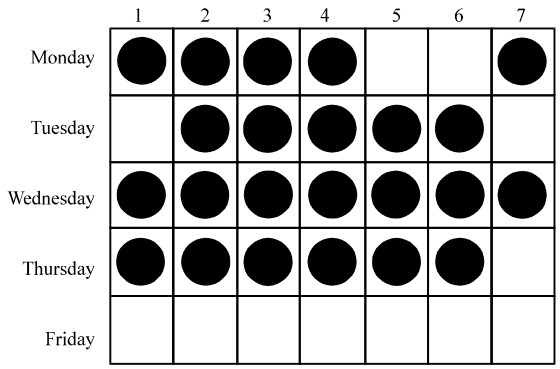

Fig. 3: The days and hours that the teacher can teach

second is by adding conditions to the program to prevent it from doing unnecessary work. These two ways will make the program complete the same task that it did in study 2 but in less time.Theorem 3.1 shows the expected time for steps 7-16 of algorithm 2.

The paths generated inside the program present the days and hours that the teacher can teach. For example, suppose the teacher (Teach 1) in Table 1 is available on Monday-Thursday, so, the table of days available for this teacher will be as shown in Fig. 3.

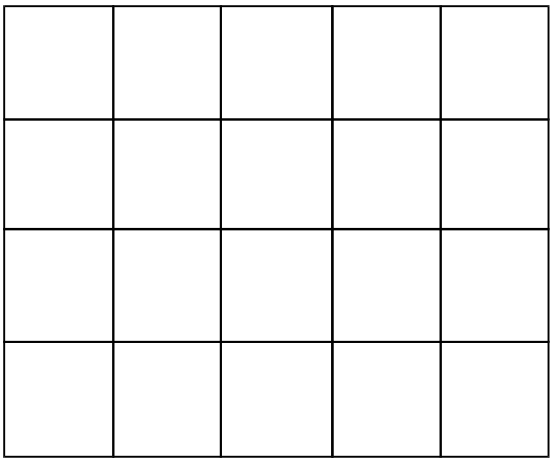

Fig. 4: 20 locations to arrange all 9 elements of the sets m_1-m_4

The black dots represent the hours in which this teacher can teach on the given day. These available days are represented inside the program as a sequence of numbers that signify the locations that should be checked. The path of this teacher will be p = {1, 2, 4, 7, 10, 11, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27} and the program will check these positions while ignoring the locations outside this sequence {5, 6, 8, 9, 14, 28, 29, 30, 31, 32, 33, 34, 35}. This method will decrease the program run time significantly.

The second way of reducing the time is by adding two conditions to the program to stopit when these conditions are not satisfied. The first one depends on the following conjunction. Suppose there are sets and each setcontains m_n elements and we want to arrange them in s locations. Then there are only $s-\sum_{i=1}^{j}$ locations to arrange the particular set m_j, $1 \le j \le n$. To illustrate this conjunction consider the following example: suppose there are n = 4 sets with m_1 = {1,1}, m_2 = {2, 2, 2}, m_3 = {3, 3}, m_4 = {4, 4, 4, 4} and we want to put the elements of m_4 in the following 4×5 table in Fig. 4-7.

There are 20 locations to arrange all 9 elements of the sets m_1-m_4 . If we begin the arrangement with the elements of then one of the arrangements will be as follows: when we arrange the elements of 20 locations to arrange all 9 elements of the sets m_2, then one of these arrangementswill be: to continue the arrangement of m_3, then one of these arrangements will be:

So, we have only 20-(2+3+2) = 20-7 =13 locations or empty locations, to arrange the elements of. According to this hypothesis, the program will ignorethe locations with numbers inside when making arrangements and this will reduce the runtime. Another program addition contributed to a reduction in the runtime. When the program does notfind any appropriate location for the teacher, the program will not continue to arrange the next teacher

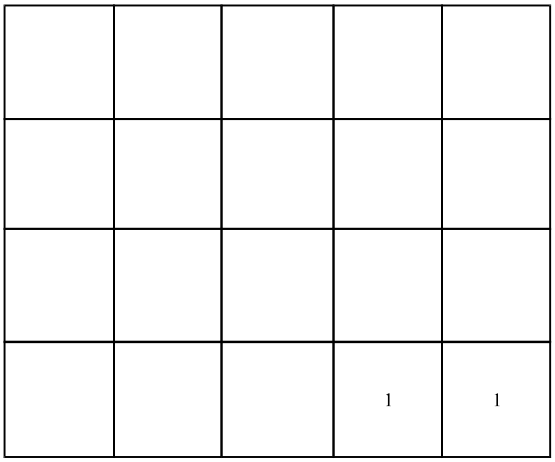

Fig. 5: Arrange elements of the sets m_2

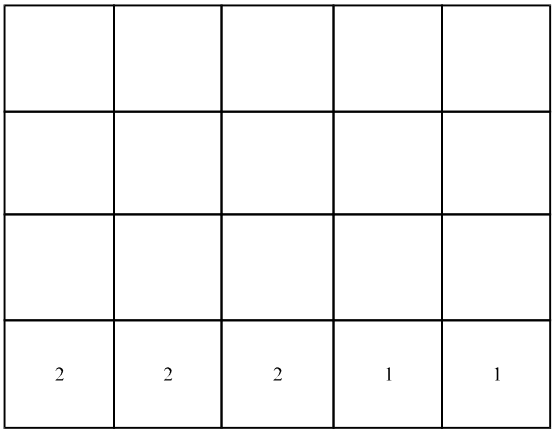

Fig. 6: Arrangement of m_3

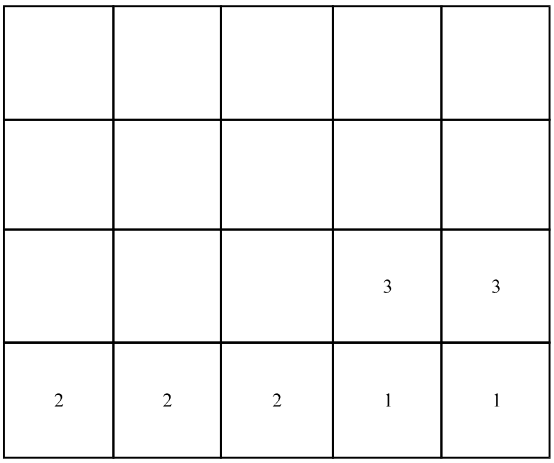

Fig. 7: Arrange elements of the sets m_4

because it will not be useful to continue to build the table with one missing teacher. For example, if the program tries to place the teacher (Teach 4) into a time slot but there is not a location to put the teacher inside the table, this will make the program end this loop and skip all the teachers {Teach 5, Teach 6, ..., Teach}. The program will then check the next location, then the next. The algorithm of this advanced techniqueis listed as:

Algorithm 2:

1. Input the values of the following: the first work day (fdow), number of business days (bd), Number of Classes in the Day (NCD), Duration of lectures Classes (DC), Duration Between two Classes (DBC), Start Time of first Class (STC), Number of Grades (NOG), Number of Solutions (NOS), the Number of Halls (NOH), the names of teachers with the coming days (or available days) and hours of availability in each day, the names of halls with the available days and hours in each day as well as the subjects and teachers with the duration of each subject at all grade
2. Convert the teachers, halls and lectures from names to number
3. Generate the teacher's path of the available days
4. Set the value grade to 1
5. While the gradeis less than or equal to thenumber of grades, do steps 5-16
6. Set the value of subject to 1
7. While the subject is less than or equal to the number of classes in the grade, do steps 7-16
8. Set the value of table locations i to the value of the first number of the teacher's path who teach the subject
9. While table's locations i is less than or equal to value of the last number of teacher's path who teach the subject, do steps 9-16
10. If the table's location i is empty and the teacher who teaches the subject is available, place the subject and name of the teacher in location i
11. If the tested teachers not have free time in the remaining location,go to step 9
12. Set the value of hall to 1
13. While the value of the hall is less than or equal to number of Halls, do steps 14-16
14. If the hall is empty, place its name in location i of the table
15. Else (hall is not empty), increase the value of the hall by 1
16. Else (the table location i is not empty or the teacher who teaches the subject is not available), increase the value of the table's location by 1
16. Stop

This algorithm is similar to algorithm 1 in section two, except steps 3, 8, 9 and 11 which summarizethe new technique we used. These two ways of the paths and conditions reduce the program runtimeas shown in Fig. 3. The run speed of the two techniques is measured using different parameters. It is clear from these figures that the run speedis significantly less using the advanced technique (Fig. 8-10). The following Theorem shows the expected time for steps 7-16 of algorithm 2.

Theorem 2.1: The time required to implement steps 7-16 of Algorithm 2 to generates:

$$O = (DH)^{L}\prod_{L} S!S^{L}$$

D×H table with S classes and L levels is.

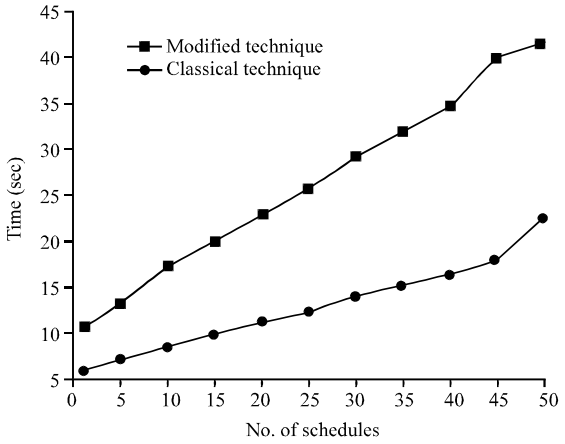

Fig. 8: The run speed of the advanced technique is less than the classic technique, the program runs with fdow = Mon, bd = 5, Wh = 7, LC = 00:40, STC = NOH = 5

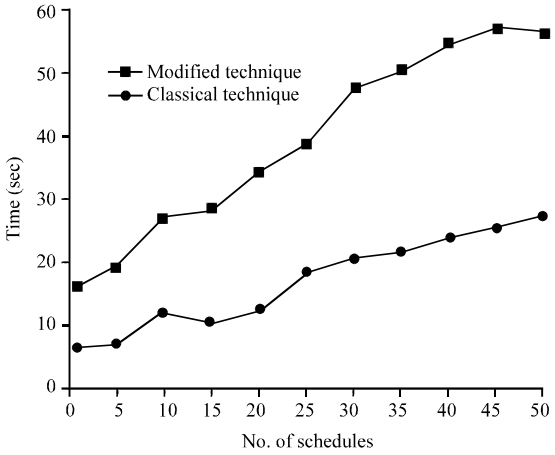

Fig. 9: The run speed of the advanced technique is less than the classical technique, the program runs with fdow = Sun, bd = 6, Wh = 10, LC = 01:00, STC = 0.8:00, NOH = 5

Proof: Let D be the number of business days per week, H be the number of classes per day, L be the number of levels and suppose the number of classes in each levels (S) then according to steps 7-16 of algorithm 2, the lop starts from the first class and end until the last class, since, there is S_L classes in each level L, the probability of searching through all the classes will be $S_L!$. In step 8 the algorithm will set the value of the first number of the teacher's path who teach the subject, since, there is S classes in each level then the number of screeching all classes will be S_L, the remaining steps will implicitly do the work of step8, exceptstep 13, the algorithm try to fill all the D×H cells in the table that will make the total number of searching.

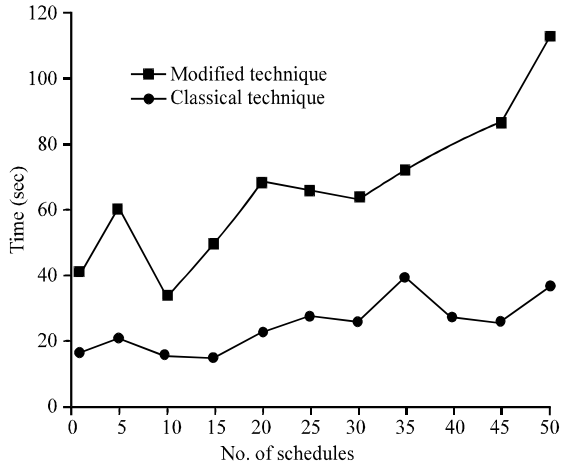

Fig. 10: The run speed of the advanced technique is less than the classic technique; the program runs with fdow = Sat, bd = 7, Wh = 11, LC = 00:45, LDBC = 0:05

$$DHS!S^{L} \quad (1)$$

this number will happened for each level, so, the final number of searching will be:

$$O = (DH)^{L}\prod_{1}^{L} S!S^{L} \quad (2)$$

Troy university example: Data from the faculty of Troy University's Mathematical Science Departmentare applied to this program. The number of lectures is 50 with 21 teachers and 7 classrooms. It consists of Monday through Friday and each day there are 7 h of lectures starting from 8:00 am to 3:00 pm. The schedule includes 35 time slots with 5 days of 9 h each. The program ran on a PC with an Intel i7 processor with a 6 M cache. The program took only minutes to find a solution. However, theprogram achieved a maximum of 10,969, solutions which took 7.5 h.

CONCLUSION

A computer program was implemented and testedto solve the timetable problem using the data from Troy University's Math Sciences Department. Many of its characteristics are common to other universities and the procedures could be adapted and used in other places. The main difference of this algorithm among the other methods which solve the time tabling problem is the possibility to have more than one solution in very reasonable computing times, depending on the size of the problem and the complexity.

SUGGESTIONS

One advantage of using this program is the possibility of using the minimum number of halls or classrooms, to cover all given lectures. This will allow some halls to be used in another classor activity.

After the program is run with real world data, there are still many enhancements that can be made towards the run time to have more solutions with more differences between them.

REFERENCES

Abramson, A., H. Dang and M. Krisnamoorthy, 1999. Simulated annealing cooling schedules for the school timetabling problem. Asia-Pacific J. Oper. Res., 16: 1-22.

Abramson, D., 1991. Constructing school timetables using simulated annealing: Sequential and parallel algorithms. Manage. Sci., 37: 98-113.

Akkoyunlu, E.A., 1973. A linear algorithm for computing the optimum university timetable. Comput. J., 16: 347-350.

Alvarez-Valdes, R., E. Crespo and J.M. Tamarit, 1997. A tabu search algorithm to schedule university examinations. Questiio, 21: 201-215.

Alvarez-Valdes, R., E. Crespo and J.M. Tamarit, 2002. Design and implementation of a course scheduling system using Tabu search. Eur. J. Oper. Res., 137: 512-523.

Badri, M.A., D.L. Davis, D.F. Davis and J. Hollingsworth, 1998. A multi-objective course scheduling model: Combining faculty preferences for courses and times. Comput. Oper. Res., 25: 303-316.

Bakir, M.A. and C. Aksop, 2008. A 0-1 integer programming approach to a university timetabling problem. Hacettepe J. Math. Stat., 37: 41-55.

Burke, E.K., G. Kendall and E. Soubeiga, 2003. A tabu-search hyperheuristic for timetabling and rostering. J. Heuristics, 9: 451-470.

Carrasco, M.P. and M.V. Pato, 2000. A multiobjective genetic algorithm for the class-teacher timetabling problem. Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling PATAT, August 16-18, 2000, Springer, Berlin, Germany, ISBN:978-3-540-42421-5, pp: 3-17.

Chu, S.C. and H.L. Fang, 1999. Genetic algorithms vs. Tabu search in timetable scheduling. Proceedings of the 1999 3rd International Conference on Knowledge-Based Intelligent Information Engineering Systems (Cat. No.99TH8410), August 31- September 1, 1999, IEEE, Adelaide, Australia, pp: 492-495.

De Werra, D., 1985. An introduction to timetabling. Eur. J. Oper. Res., 19: 151-162.

De Werra, D., 1997. The combinatorics of timetabling. Eur. J. Oper. Res., 96: 504-513.

Deris, S., S. Omatu and H. Ohta, 2000. Timetable planning using the constraint-based reasoning. Comput. Oper. Res., 27: 819-840.

Deris, S.B., S. Omatu, H. Ohta and P.A.B.D. Samat, 1997. University timetabling by constraint-based reasoning: A case study. J. Oper. Res. Soc., 48: 1178-1190.

Even, S., A. Itai and A. Shamir, 1976. On the complexity of timetable and multicommodity flow problems. SIAM. J. Comput., 5: 691-703.

Fen, H.S., I.S. Deris and S.Z. Hashim, 2009. Investigating constraint-based reasoning for university timetabling problem. Proceedings of the International Multi Conference on Engineers and Computer Scientists (IMECS 2009) Vol. I, March 18-20, 2009, IAENG, Hong Kong, China, ISBN:978-988-17012-2-0, pp: 1-5.

Fen, S. and I. Ho, 2009. Incorporating of constraint-based reasoning into particle swarm optimization for university timetabling problem. Comput. Sci. Lett., 1: 1-21.

Gunawan, A., K.M. Ng and K.L. Poh, 2007. Solving the teacher assignment-course scheduling problem by a hybrid algorithm. Intl. J. Comput. Inf. Syst. Sci. Eng., 1: 136-141.

Hultberg, T. and D.M. Cardoso, 1997. The teacher assignment problem: A special case of the fixed charge transportation problem. Eur. J. Oper. Res., 101: 463-473.

Lawrie, N.L., 1969. An integer linear programming model of a school timetabling problem. Comput. J., 12: 307-316.

Schmidt, G. and T. Strohlein, 1980. Timetable construction-an annotated bibliography. Comput. J., 23: 307-316.

Welsh, D.J.A. and M.B. Powell, 1967. The upper bound for the chromatic number of a graph and its application to timetabling problems. Comp. J., 11: 85-86.

Wile, A.J. and G.A. Shouppe, 2011. Does time-of-day of instruction impact class achievement?. Perspect. Learn. J. Coll. Educ. Health Professions, 12: 21-25.