An innovative proposed solution to solve the problem of conflicts in course schedule

Yousif Karkaz[1], Mohamad Alrfaay[2] , and Aref Kurdali [3]

[1] Department of Computer Engineering and Networks, Al Jouf University, Skaka, Kingdom Of Saudi Arabia

[2] Department of Computer Engineering and Networks, Al Jouf University, Skaka, Kingdom Of Saudi Arabia

[3] Department of Computer Engineering and Networks, Al Jouf University, Skaka, Kingdom Of Saudi Arabia

Abstract

Registration systems are vital part in all universities. Furthermore, registration is the base stage for the academic and tuition process. The registration System provides the ability for students to register courses offered by their colleges during the announced registration periods. Conflicts are the most confused problem in registration system in universities. This article introduce a new innovative proposed solution for producing a free-conflicts *course schedule* *. So, is allowing students to register the maximum credit hours allowed to them by the academic system. As a case study we applied our analyses and algorithm in the college of computer and information sciences in Al Jouf university. By the proposed solution we achieved the following benefits: elimination of conflicts in students timetables, increase the total number of student’s registered hours, performance and reliability enhancement, fast way to solve the conflict problems, achieving fairness between students, accuracy of results, ease of use.

Keywords: *registration system, student timetable, course schedule.*

1. Introduction

It is not a secret that the *course schedule** is a critical matter in any university's registration system. As we know, the registration process is restricted by a lot of conditions, such as: prerequisites completion, corequisites enrolling, student CGPA threshold (which determines the maximum number of allowed credits for him/her), academic level limitations.

Course schedule **is published** before academic advising and registration begin for each semester, and students must know it in order to register. within this important process, the conflicts in students timetables are the most frequent problem. Conflict is the situation when the student tries to enroll into two or more courses where they are in the same time period of *course schedule*. As a result, the student cannot register all available courses for him/her because conflicts existence. There are many reasons which cause these conflicts, some of which related to the registration system itself and others to the academic progress of students. At any rate, these reasons are out of our interest here. The registration process is controlled by *course schedule* constraints. The problem appears when there are situations where the student can't register in some legal courses because of the discrepancy with other courses, although academic conditions do not prevent him/her from registering these courses. The idea of our solution is to allow all students to register all available courses they have the academic validity on it, regardless of the conflicts in *course schedule* between these courses. Then, our proposed algorithm will reconstruct a new free-conflicts *course schedule*. As a case study, we applied our proposed solution to the college of computer and information sciences in Al Jouf university. Our proposed solution is accomplished by firstly monitoring and analyzing the cases of conflicts in the college, and then design an algorithm to solve the problem based on registry data that has been collected through the registration process. Depending on our proposed solution the following benefits are acquired: Increase the total number of student’s registered hours, performance and reliability enhancement, fast way to resolve the conflict problems, achieving fairness between students, accuracy of results, and ease of use. The rest of the article is organized as follows: section

* *course schedule* lists each class being offered, its time, location, instructor, and its unique number, and which students must know in order to register.

2 gives a fast literature review. Section 3 introduce the methodology which we followed in our research. Results and discussion are introduced in section 4. Section 5 conclude the paper.

2. Literature Review

Researches in this domain are proceeded in different ways, many researchers have concentrated on satisfying the faculty constraints as we did. In [1] Vangel V. Ajanovski proposed a solution given in the form of a model in which a single unifying process is envisioned that covers both earlier processes. The integrated process defines a continuous dialogue between: the student groups, the enrolment administration, and scheduling administration. However, there are a lot of changes that should be done in the old system, also this paper stated that conflicts in the class schedule are likely introduced due to the lack of resources.

Muller et al. [2] built a system which provides a method for modeling the structure of course offerings and other essential problem constraints. It uses these along with student demand to build and modify course timetables and create individual student class schedules that satisfy student needs and minimize conflicts. However, this system is complicated and aim just to minimize conflicts while our research aims to resolve all the conflicts, beside to allowing students to register the maximum credit hours allowed by the academic system. Also the proposed system in [2] work from the first step of registration and give advice for student to select free courses –no conflicts- ,but our proposed solution will just depend on the data of currently registering system –no changes need- and then solve conflicts and reconstruct the *course schedule*.

At any rate, most common approaches are based on graph-coloring approximations [3], or on an iterative improvement process [4] in which an initial solution is improved over time. Other approaches included Monte Carlo methods [5] .

The comparison between different methods is difficult because of the nature of these researches which tends to resolve a local problem of different registration system in a university or a school. So, most papers reflect the locality of the research problem, and tend to use only local data when experiments are conducted. This data is not made available to other researchers so it is impossible to replicate the experiments.

So, our research will introduce a new good idea and different technique for solve conflicts. we will depend basically on database and a lot of iterations through database tables to extract the optimal case of the *course schedule* without conflicts and in friendly way for user. We will apply all tests and experiments to affirm that the proposed solution is effective.

3. Methodology

We began with collecting all necessary information about registration process based on our communications with admission and registration deanship in Al Jouf university. So that, we determine the student registration steps, conditions, restrictions, and criteria. Although the registration process is computerized, registration system still prevents students from enrolling in some courses because of conflicts in *course schedule* as we mentioned previously. Usually These conflicts are resolved manually by student problems committee. The registration process is controlled according to just the typical academic student progress, without taking into account the enormous considerations of special cases for all students, and for each student. For example but not limited to, a student may be allowed to register in many courses which belong to different levels. And the students who expected to graduate, academically are allowed to enroll in any course with their prerequisite in the same time. At any rate, as we mentioned previously, these details are out of our interest in our research.

3.1 Algorithm Requirements

Initially, in our proposed system, students are allowed to register all their desired courses even if there is a conflict in the time (two or more courses are registered in the same time), but without breaking the academics criteria. This is to give the students the ability to register the maximum studying hours under the academic conditions. So, the proposed algorithm which will be used to resolve the conflicts depends on the following data: the available courses in the current semester, the registered courses which are the desired courses of students under the academics criteria.

3.2 Basic Definitions

Courses Table: a table includes all registered courses in the current semester.

Study Period: a period of time includes all available non-conflicting courses. In view of the fact that the study day is divided into periods, the period may be one, two, or three hours depending on the courses specifications.

Period Table: a table includes study periods, each record represents one period.

Candidate non-conflict courses table: a table includes all available non-conflicting courses against the courses in the current study period.

Remaining courses table: a table includes the remaining courses of the courses table after deleting the courses that are included in the period table.

Study schedule: a schedule represents the distribution of the period table along the study days in the week.

3.3 proposed solution steps

In our proposed solution we give freedom and dynamism to the user to construct robust *course schedule* depending on the solution algorithm. So that, the proposed solution is applied as follow:

1. Our proposed system constructs the courses table and fills it with registration data that got from the admission and registration deanship.
2. Our proposed system constructs remaining courses table which initially includes all data in the courses table ordered descending according to the number of students registered in each course.
3. The user selects a course from the remaining course table.
4. Our proposed system appends new record in the period table.
5. Our proposed system inserts the selected course into the period table, and deletes it from the remaining course table.
6. Our proposed system constructs the candidate non-conflict courses table ordered descending according to the number of registered students in each course.
7. The user selects a course from the candidate non-conflict courses table.
8. Our proposed system inserts the selected course in period table, and deletes it from remaining courses table.
9. Our proposed system reconstructs the candidate non-conflict courses table.
10. Repeat steps 7, 8, and 9 until the candidate non-conflict courses table becomes empty.
11. Go to step 3 until the remaining courses table is empty.
12. The user depends on the period table to build free-conflicts *course schedule*.

4. results and discussion

In this section we will make a comparison between the manual *course schedule* (which is made by the schedule committee) and the automated *course schedule* (which is produced by our proposed algorithm). This comparison is made based on the statistical functions applied for the two previous mentioned course schedules.

The important thing that should be noted, that the manual *course schedule* **never see the light and does not become usable until** the end of third or fourth week of study. This is because of the time taken to resolve the conflicts and problems that begin to appear in succession since the first day of study, and continue until the end of fourth week, while our automated *course schedule* is produced and be usable from the first day of study.

4.1 Results Analysis

After getting the manual and the automated *course schedules*, we generate the histogram of the accumulated teaching hours for both schedules. Depending on these two histograms, we calculated the statistical functions and used them to make the comparison between the two schedules.

4.2 The manual *course study* statics

Table 1 shows the frequency distribution table of teaching hours of manual *course schedule* along study day. For illustration, as it shown from table 1, line3, there are 34 teaching hours between 10 to 11 am.

Table 1: manual *course study* frequency table

Line number	Frequency	Bin
1	33	8 – 9
2	33	9 – 10
3	34	10 – 11
4	30	11 – 12
5	3	12 – 13
6	22	13 – 14
8	24	14 –15
9	10	15 – 16
10	10	16 –17
11	9	17 –18
12	10	18—19
13	6	19—20
14	3	20—21
15	227	Total

Table 2 shows the cumulative distribution table

Table 2: the cumulative distribution table of manual course schedule

Cumulative %	*Frequency*	*Bin*
14.54%	33	8
29.07%	33	9
44.05%	34	10
57.27%	30	11
58.59%	3	12
68.28%	22	13
78.85%	24	14
83.26%	10	15
87.67%	10	16
91.63%	9	17
96.04%	10	18
98.68%	6	19
100.00%	3	20
100.00%	0	More

From table 2 we generate the histogram which is shown in figure1.

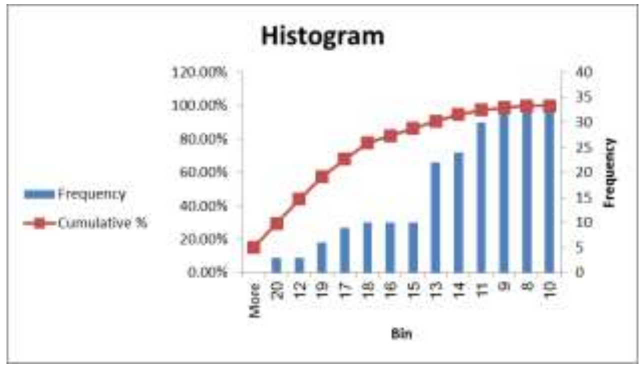

Fig. 1 The Manual *course study* Histogram

From Table 1 and Fig 1 we generate the descriptive function statics which are shown in table 3

Table 3: The manual *course schedule* Descriptive

11.920705	Mean
0.2175747	Standard Error
11	Median
10	Mode
3.2780935	Standard Deviation
10.745897	Sample Variance
-0.5875304	Kurtosis
0.6685935	Skewness
12	Range
8	Minimum
20	Maximum
2706	Sum
227	Count
20	Largest(1)
8	Smallest(1)
0.4287345	Confidence Level(95.0%)

4.3 The automated *course schedule* statics

Similar to the previous subsection, table 4 shows the automated *course schedule* frequency distribution of the same teaching hours which were in the manual *course schedule*.

Table 4: Automated *course study* frequency

Frequency	Bin
35	8 – 9
36	9 – 10
37	10 – 11
36	11 – 12
1	12 – 13
19	13 – 14
17	14 –15
12	15 – 16
12	16 –17
12	17 –18
6	18—19
4	19—20
0	20—21
227	Total

IJCSI International Journal of Computer Science Issues, Volume 12, Issue 2, March 2015
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

From Table 4 we generate the automated *course schedule* histogram which is shown in figure 2.

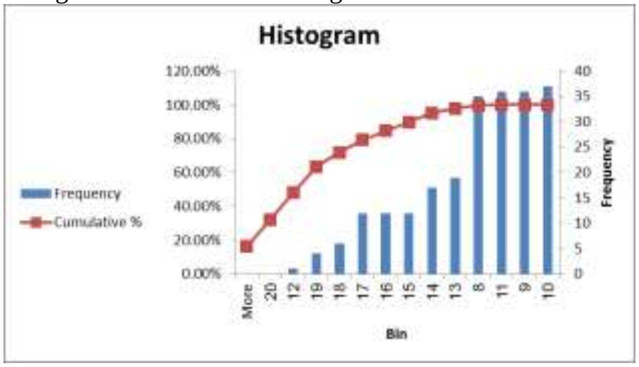

Fig. 2 The Automated *course schedule* Histogram

4.4 Comparison of descriptive statics between manual and automated *course schedules*

Table 5 and table 6, summarize the descriptive statics for manual and automated course schedules.

Table 5: A comparison of descriptive statics between manual and automated schedule

Automated course		Manual course schedule			
Frequency	*Cumulative %*	*Frequency*	*Cumulative %*	*Bin*	*Line number*
15.42%	35	14.54%	33	8	1
31.28%	36	29.07%	33	9	2
47.58%	37	44.05%	34	10	3
63.44%	36	57.27%	30	11	4
63.88%	1	58.59%	3	12	5
72.25%	19	68.28%	22	13	6
79.74%	17	78.85%	24	14	7
85.02%	12	83.26%	10	15	8
90.31%	12	87.67%	10	16	9
95.59%	12	91.63%	9	17	10
98.24%	6	96.04%	10	18	11
100.00%	4	98.68%	6	19	12
100.00%	0	100.00%	3	20	13
100.00%	0	100.00%	0	More	14

Table 6: A comparison of statics function between manual and automated *course schedules*

Automated	**Manual course**	**Statics functions**	**Line**
11.57268722	11.92070485	Mean	1
0.203548817	0.217574706	Standard Error	2
11	11	Median	3
10	10	Mode	4
3.06677215	3.278093477	Standard Deviation	5
9.405091419	10.74589685	Sample Variance	6
-0.612137001	-0.587530354	Kurtosis	7
0.714042968	0.668593469	Skewness	8
11	12	Range	9
8	8	Minimum	10
19	20	Maximum	11
2627	2706	Sum	12
227	227	Count	13
19	20	Largest(1)	14
8	8	Smallest(1)	*15*
0.40109625	0.428734492	Confidence	16

Figure 3 shows a graphical differentiation between manual and automated *course schedules.*

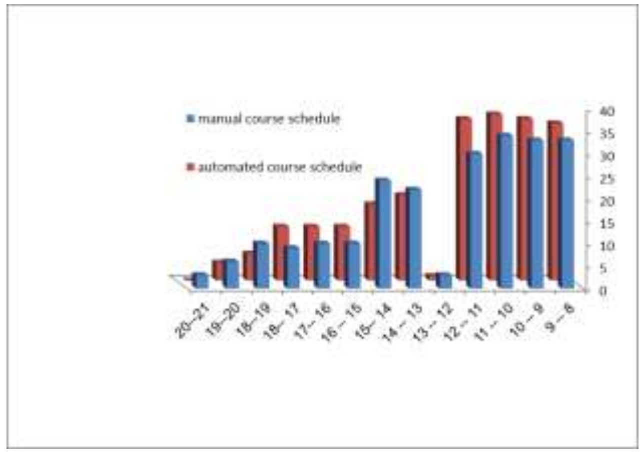

Fig. 3 A graphical differentiation between manual and automated *course schedules*

4.5 Discussion of Results

we can summarize our observations in the following points:

Firstly, from table 5, line6: (72.25%) of the teaching hours are allocated before one o'clock pm in the automated *course schedule*, while this percentage in manual *course schedule* is (68.28%).

Secondly, from table 5, lines12,13: the number of teaching hours after the 7 o'clock pm in manual *course schedule* is (9) hours; while it is only (4) hours in the automated *course schedule*.

Thirdly, from table 6, line1: the arithmetic mean of the automated *course schedule* is (11:57), while it equals to (11.92) in manual schedule.

Fourthly, from table6 line7: there is difference in the two kurtosis functions. Kurtosis expresses the rise of the peak in the histogram. If The kurtosis is more Negative, the peak is more broad. So, kurtosis in the automated *course schedule* is (-0.612), while it is (-0.587) in the manual one.

Fifthly, line8 in table6: there is difference in skewness between automated and manual *course schedule*s. If skewness is positive and closer to one, this means that the histogram peak is closer to the vertical axis. The skewness for automated *course schedule* equals to (0.714), whereas it is (0.668) for the manual *course schedule*.

From the previous comparisons it is clear that after applying our solution, the studying hours are closer to the first hours of the day . Also the teaching hours allocation along the study day hours is better. This will lead to fast stability of teaching process and more satisfying of students.

Acknowledgment

The research for this paper was financially supported by Al-Jouf University, grant no. 34/259. So we would like to express our deepest appreciation and thanks to Al-Jouf University; Vice President for Scientific for Graduate Studies and Scientific Research for supporting this work.

5. conclusion

University and students suffer from conflicts in the study timetables, the thing which affects the quality of teaching process. We proposed innovative solution to get rid of conflicts by allowing students to register all academically valid courses even if there are conflicts with other courses in the initially *course schedule*. After the end of registration period, we take the registration data and apply our novel algorithm to produce a free-conflicts *study schedule*. . By the proposed solution we achieved the following benefits: Getting a free-conflicts *study schedule* from the first study day, teaching hours become more concentrated in the first hours of the study day, better allocation of teaching hours along the hours of study day, qualifying the learning process and make it more effective, while in the old situation we note confusion and irregularity in the teaching process.

References

[1] Vangel V. Ajanovski. INTEGRATION OF A COURSE ENROLMENT AND CLASS TIMETABLE SCHEDULING IN A STUDENT INFORMATION SYSTEM., International Journal of Database Management Systems (IJDMS) Vol.5, No.1, February 2013.

[2] Tomas Muller, Keith Murray, Stephanie Schluttenhofer. University Course Timetabling & Student Sectioning System. 400 Centennial Mall Drive, West Lafayette, IN 47907-2016, USA ,2007.

[3] D. Abramson. Constructing school timetables using simulated annealing: sequential and parallel algorithms, Management Science, vol. 37, no. 1, January 1991, 98-113.

[4] T. Franklin, E. Jenkins, and K. Woodson. A case study in scheduling courses, UMAP Journal, 15, no. 2, 1995, 115-122.

[5] G. Lewandowski. Practical implementations and applications of graph coloring, Ph.D. Dissertation, University of Wisconsin-Madison, Computer Sciences Department, 1994.

Yusuf A Gargaz received his bachelor (1986), MSc (1988), Ph.D (2003) all in Computer Engineering (Networks), from University of Bucharest (UPB), Romania. Worked as Ass. Prof. in Al-Balqa university, head of computer science department in Al-Huson college (2007-2009). Currently He is Ass. Prof. in department of computer engineering and networks at Al-Jouf university/ Kingdom of Saudi Arabia(since 2009).

Mohamad A Alrfaay received his bachelor degree (2000) in Electronics Engineering from Faculty of Mechanical and Electrical Engineering in University of Damascus, Syria. He received his Master degree (2006) in information systems and computer networks engineering from Faculty of Engineering ,Department of Computer Engineering, in Cairo University, Egypt.
Since 2007 he is a lecturer in the Department of Computer Engineering and Networks in Al Jouf University, Saudi Arabia.

Aref H Kurdali received his bachelor (2000) and diploma (2001) degrees in Electronics Engineering, and Programming and Operating Systems, respectively, from Faculty of Mechanical and Electrical Engineering in University of Damascus, Syria. He received his Master degree (2005) in computer networks engineering from Faculty of Engineering, Department of Communication and Electrical Engineering in Cairo University, Egypt. He was a chief of Software Division (2005-2007). Since 2008 he is a lecturer in the Department of Computer Engineering and Networks in Al Jouf University, Saudi Arabia.