



Development of Mechanism for Handling Conflicts and Constraints in University Timetable Management System

Constance Kalu

Department of
Electrical/Electronic and
Computer Engineering University
of Uyo
Akwa Ibom State Nigeria

Simeon Ozuomba

Department of
Electrical/Electronic and
Computer Engineering University
of Uyo
Akwa Ibom State Nigeria

Sylvester Isreal Umana

Department of
Electrical/Electronic and
Computer Engineering University
of Uyo
Akwa Ibom State Nigeria

ABSTRACT

In this paper development of mechanism for handling conflicts and constraints in university timetable management system is presented. The mechanism consists of a set of algorithms and analytical expressions for resolving conflicts and managing constraints in the university timetable scheduling system. In order to handle conflicts associated with courses, the courses are ranked based on their total ranking scores (TRS). In order to handle conflicts associated with course lecturers, the lecturers are ranked based on their ranking weight (W). Provision is also made to accommodate students' activities constraints. The venues are first matched with the courses based on their carrying capacity which is enough to accommodate the class size. Then the procedure for time allocation to courses on the timetable considers the course ranking first before the lecturers ranking. In all, the design constraints and decisions were based on information on the timetable issues at University of Uyo, Akwa Ibom state, Nigeria. The mechanism was used in a proof of concept web-based timetable management system for University of Uyo and with the sample data used to test the system there was no observed conflict in the timetable. Though the present mechanism for automating timetable conflict resolution is tailored towards University of Uyo as the case study, the idea presented can be adapted or further developed and generalized to apply to other institutions.

Keywords

Timetable, Venue Constraint, Map Mashup, Spatial Information System

1. INTRODUCTION

In academic institutions, lecture and examination timetable is indispensable. Most especially, in the developing countries where some tertiary institutions run several courses in multiple campuses with numerous shared facilities scheduling and managing such timetable system is very tedious [1,2,3,4,5,6,7,8,9]. The problem is more pronounced where the timetable is scheduled at the institutional level rather than at the departmental level [10,11,12, 13]. Again, such is the case in many institutions in the developing countries where there are inadequate resources as such, both personal and facilities are shared among many different departments and academic units [14,15,16,17,18,19,20,21].

In any case, the advent and advancements in information and communication technology (ICT) has made it possible to develop and automate timetable management system [22,23,24,25,26]. In addition, some institutions go further to use Google maps or other online map technology to visualize

the venues listed on the timetable [27,28,29]. Again, with such a system, members of the institution can remotely access the timetable over the Internet. In this case also, individual student and staff can extract their own personal timetable directly from the institutional online timetable.

Irrespective of the convenience and flexibility afforded by ICT, development of timetable management system is a tedious work. The major challenges in implementing a campus-wide timetable management system are the problems of conflicts and constraints which are numerous [30,31,32,33,34,35]. Considering the various constraints and conflicts that need to be tackled in developing robust campus-wide timetable management system a systematic approach is required. Consequently, in this paper, analytical expressions and algorithmic procedures are developed to handle the constraints and avoid conflicts that are identified in a campus-wide timetable. The study is based on University of Uyo as a case study. The idea presented in this paper is part of the larger work on web application and partial information system for campus-wide timetable management system. This paper focuses on the aspect of identifying and managing potential constraints and conflicts so as to avoid conflicts in the timetable and at the same time satisfy all identified constraints in the timetable.

2. METHODOLOGY

The major challenge in implementing a campus-wide timetable management system is the problem of conflicts. These include issues such as clashes in venue, students' lecture time, lecturers' lecture time, etc. In order to achieve an optimal solution to the above-mentioned problems, a systematic approach is required. In this paper, the proposed approach consists of the following four (4) major sections:

- i. Ranking of courses according to their total ranking score (TRS)
- ii. Ranking of lecturers based on lecturers ranking weight (W)
- iii. Provision for students' activities constraints
- iv. Assignment of venues and time to courses subject to the ranking and constraints obtained in sections I, II and III.

2.1 RANKING OF COURSES BASED ON THE COURSE TOTAL RANKING SCORE (TRS)

The courses offered in the university are listed and ranked based on the number of factors associated with the courses. In this paper, the three factors used to determine how courses are ranked are:

- i. Course prioritization by level or year of study.
- ii. Course prioritization by scope;
- iii. Course prioritization by estimated class size.

Course Prioritization By Level

In the case study university, UNIUYO, for undergraduates the levels /year have numerical values such as level 100 for year 1, level 200 for year 2, level 300 for year 3, and level 600 for year 6. In some departments, their academic programs stop at year 4 or 400 level. For the timetable conflict resolution mechanism, the courses are ranked based on the level of study of the students that are supposed to take the course. Courses offered at year one is ranked higher, followed by year two and so on. The rationale for this choice is because students at a higher level due to carryover may register for a course offered at a lower level but not vice-versa. A course priority score due to level, L can take the numerical values 6,5,4,3,2 and 1 for

year one, year two, year three, year four, year five and year 6 courses respectively.

The courses offered at different levels are assigned different level weights. For example, weights 6, 5,4,3,2 and 1 are assigned to year 1,2,3,4,5 and 6 courses respectively. The flowchart for automating the course prioritization by level on the proposed system is shown in Figure 1.

Course Prioritization By Scope

Course prioritization by scope considers the number of faculties as well as programs of study offering the course. This is denoted as S where S is given as:

$$S = (\text{number of faculties offering the course}) \times (\text{number of programs offering the course})$$

The higher the value of S, the higher the priority given to the course. In essence, a course offered by every program in the university has high priority over a course offered by a limited number of programs. The flowchart for course prioritisation by scope is given in Figure 2.

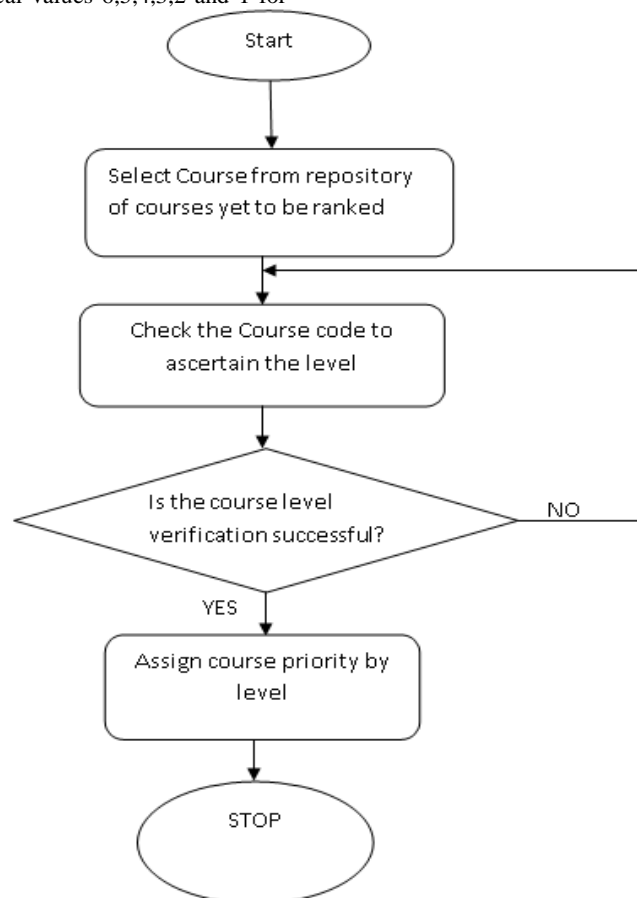


Figure 1 Flowchart for course prioritisation by level

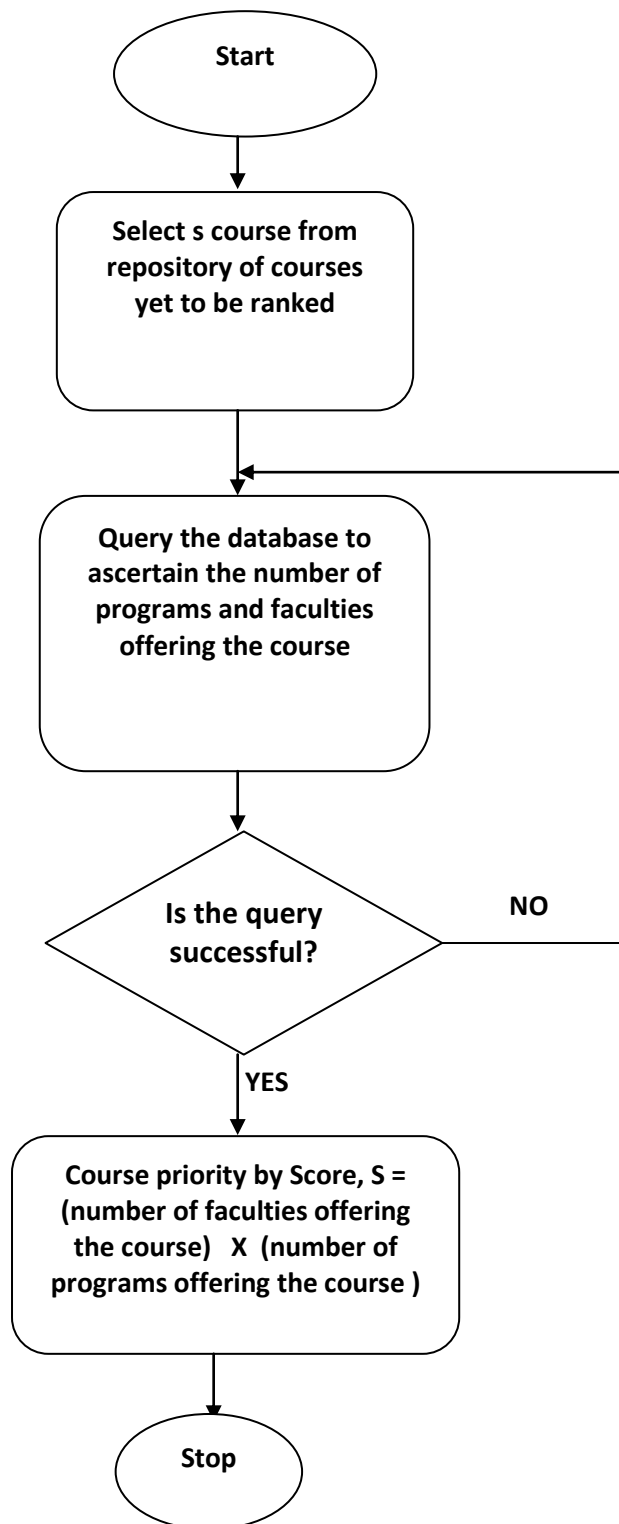


Figure 2 Flowchart for course prioritisation by scope

Course prioritization by estimated class size:

The final level of prioritization is achieved by considering the estimated class size. The flowchart for course prioritisation by estimated class size is given in Figure 3.

The estimated class size (E) for each course is computed as follows:



$$E = C_o + S_c + D_e$$

$$\left. \begin{aligned} D_e &= 0.1 (C_o + S_c) \text{ for } L = 200 \\ D_e &= 0.05 (C_o + S_c) \text{ for } L = 300 \\ D_e &= 0 \text{ for } L < 200 \text{ and } L > 300 \end{aligned} \right\}$$

Where,

C_o = number of carryover students in the course obtained from school result management system

S_c = number of students in the class or classes offering the course

D_e = number of direct entry students. D_e is applicable only for year two and three courses

The Course Ranking Process

Based on the three different course prioritization parameters, the total course ranking score (denoted as TRS) of each course can be computed as follows:

$$TRS = L + S + E$$

The course ranking process is illustrated in the flowchart of Figure 4. The courses are retrieved from the University course

registration portal. The retrieved courses and courses' information is stored in a repository for processing. The courses priorities by level are determined and assigned to the courses for further computation of the TRS of each course. The courses are stored and sorted by their TRS in a database for later use. The annotated diagrams for course ranking algorithm are shown in Figure 5.

The relevant course information in the repository includes:

- Course code;
- Course title;
- Credit hour;
- Course TRS;
- Estimated class size;
- Department in charge of course;
- Course lecturer or lecturers;
- List of departments offering the course
- Venue For the course

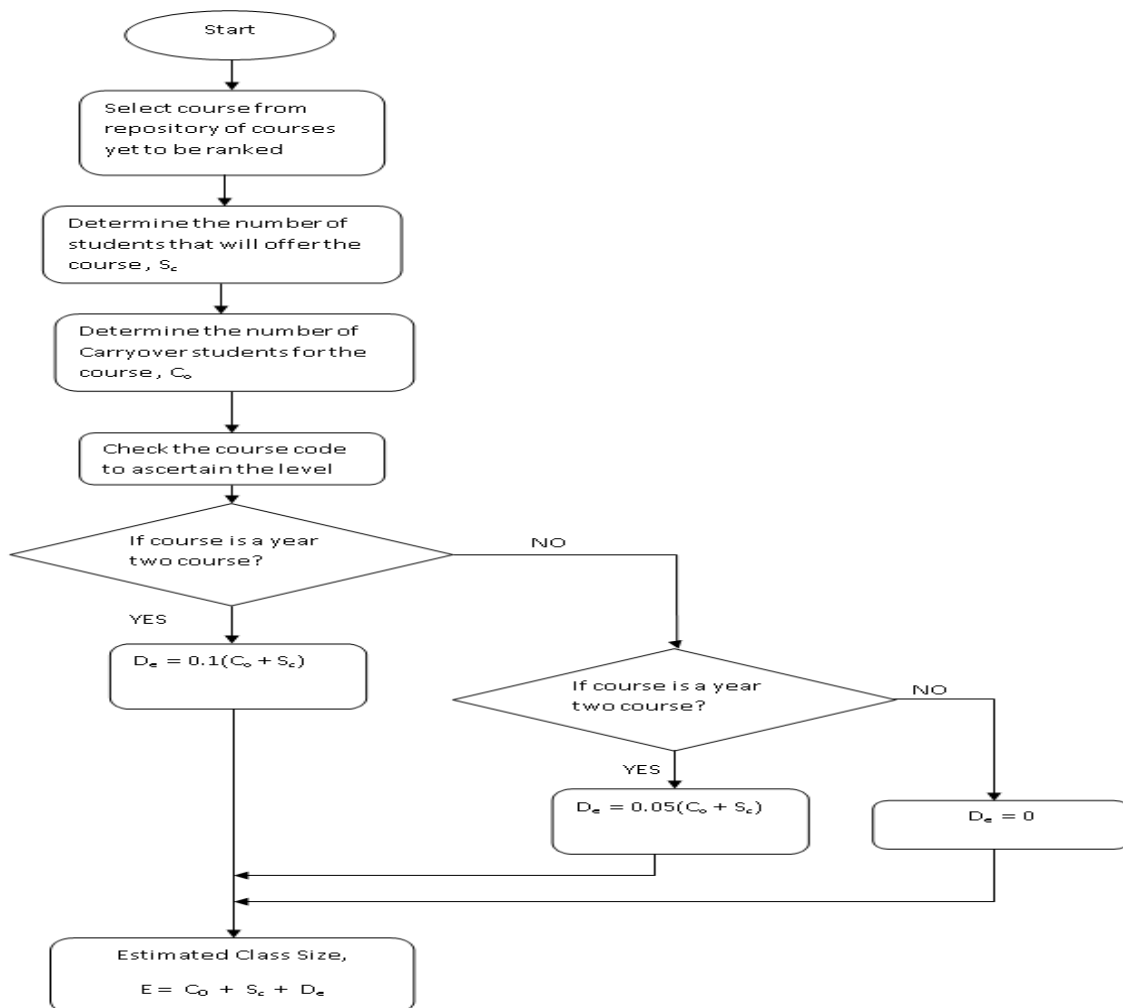


Figure 3 Flowchart for course prioritisation by Estimated Class Size

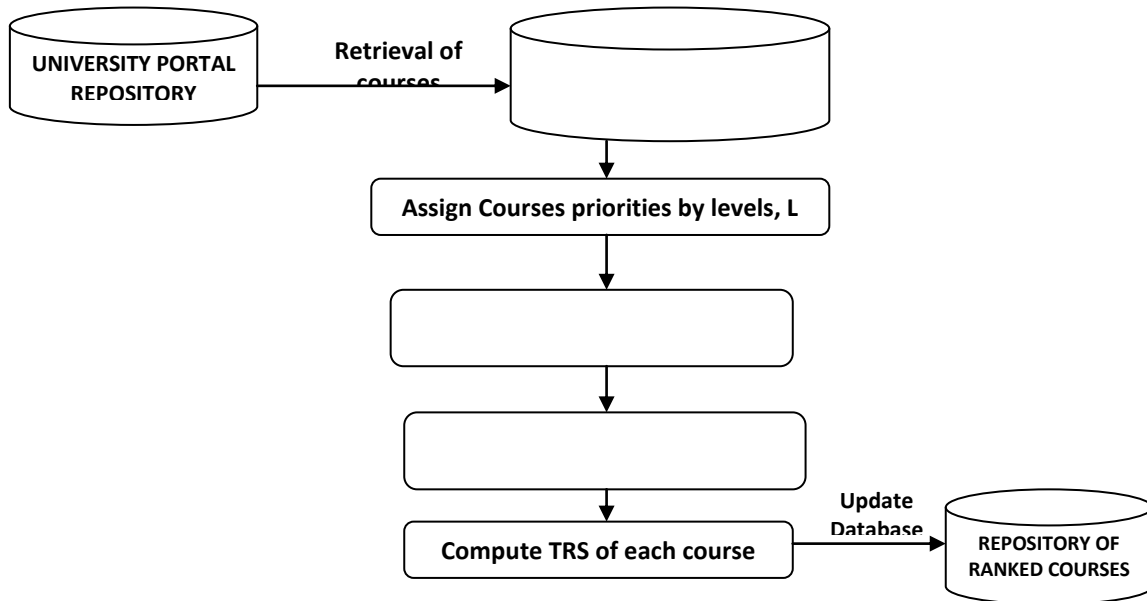


Figure 4: Course Ranking Process

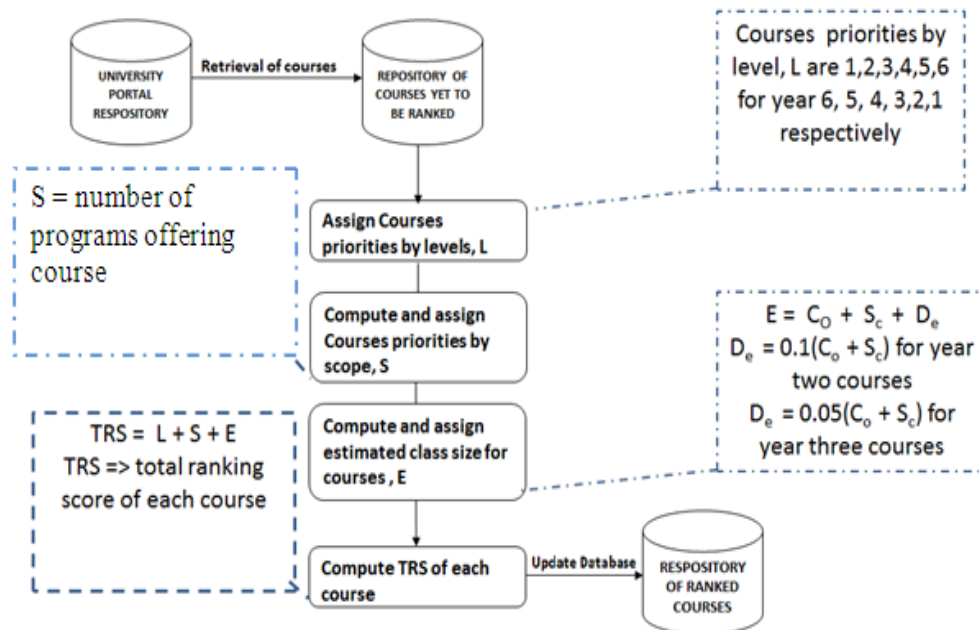


Figure 5 Annotated Diagrams For Course Ranking Algorithm

2.2 RANKING OF LECTURERS BASED ON LECTURERS RANKING WEIGHT (W)

Each departmental lecturer's course allocation list and each course lecturer's information are accessible by the system administrator through the ADMIN module of the system. Since a lecturer can take more than one courses in the university, care is taken to ensure that the courses handled by the lecturer never occur at the same time on the timetable. Moreover, some lecturers have fixed days or times for other official activities in the University. As such, the system administrator requires each lecturer's official engagement times or days for efficient temporal allotment to the lecturer. Specifically, a lecturer's course must be assigned to days and

times corresponding to the lecturer's free times and days. Figure 6 presents a lecturer's timetable schedule management procedure.

Another major constraint on the proposed system is managing the time allocation when two or more lecturers need to be scheduled within a limited time period. Particularly, some lecturers due to their ranks and other engagements in the university may have few available periods for lectures, As such, cases may arise where two or more lecturers have similar free days or times and there are limited available free periods to schedule their lectures. These problems are addressed in this work by using lecturers' engagement weight. In this work, a lecturer's weight computation algorithm illustrated in Figure 7 is used.

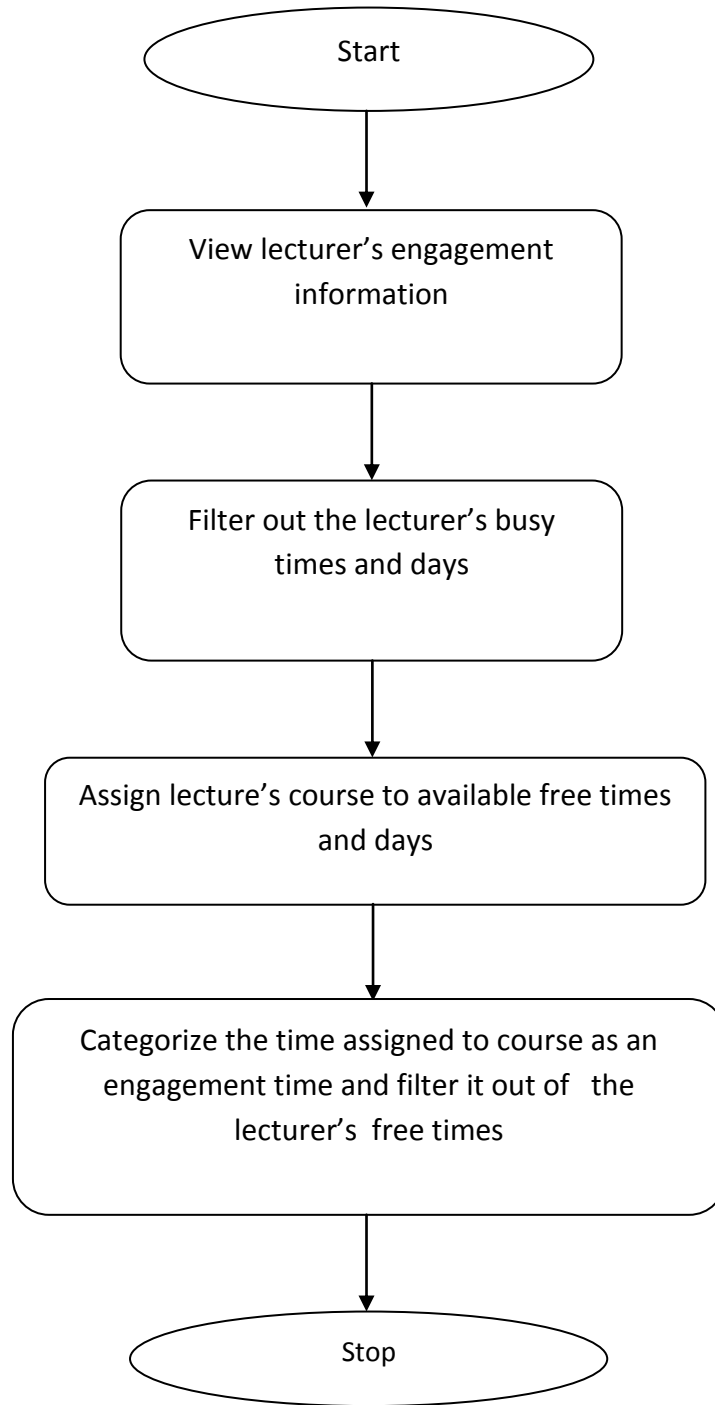


Figure 6: Flowchart showing lecturer's time schedule management procedure

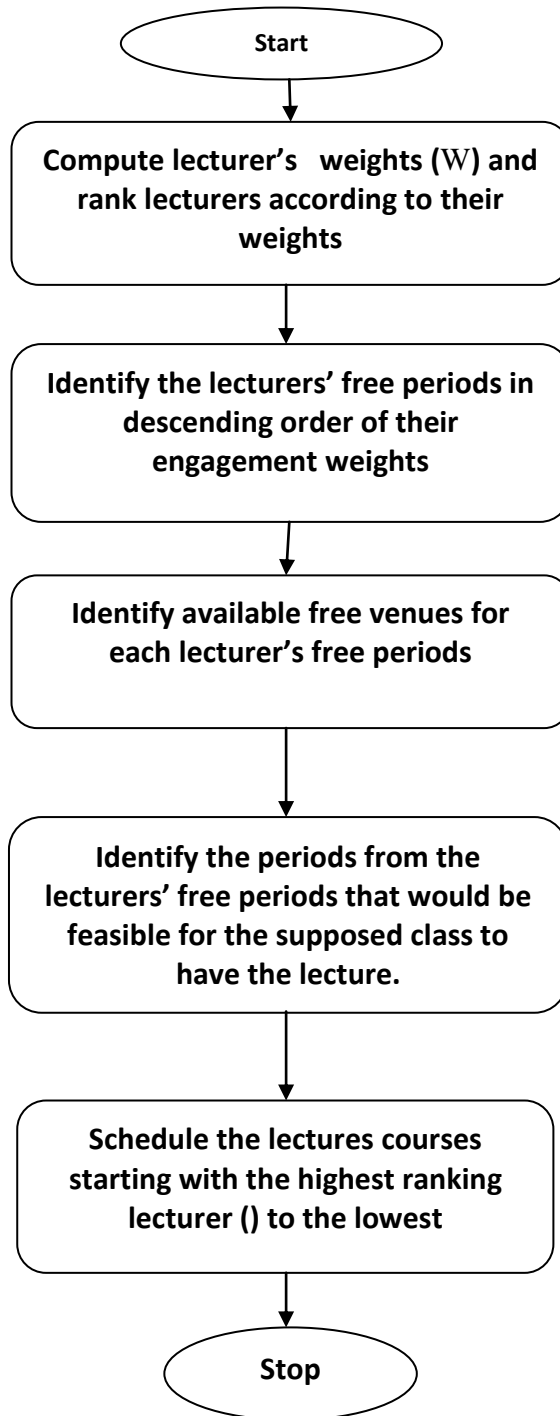


Figure 7 Flowchart for lecturer's weight computation

The matrix arrangement of the lecturers' ranks and appointments is given in Table 1. The values in Table 1 are generated from the lecturers' information and are used to compute the lecturers ranking weight which is used to rank the lecturers and hence resolve the scheduling challenge that occurs when two or more lecturers have limited time that must be allocated to them. The lecturer with higher ranking weight (W) is allocated time before the one with lower weight.

A lecturer may hold zero, one or more appointments at the same time. Each appointment adds to the workload of the

lecturer and hence reduces the lecturer's available time for lectures. The total weight assigned to a lecturer is affected by the total number (n) of appointments the lecturer holds at the same time. The total weight assigned to a lecturer is given by;

$$W = n(r) + \left(\sum_{i=1}^n(a_i)\right)$$

where r is the rank of the lecturer, n is the total number of appointments held by the lecturer, a_i is the weight of the i^{th} appointment where $i = 1,2,3,\dots,n$ and W is the ranking weight of the lecturer.



Table 1 Matrix arrangement of the lecturers' academic ranks and appointments

		Appointments (a_i)					
	Academic Rank (r)	No Appointment (a_0)	Timetable officer (a_1)	Course Adviser (a_2)	Exam Officer (a_3)	Head Of Department (a_4)	Dean of Faculty (a_5)
Graduate Assistant Lecturer	1.0	1.0	2.0	0.0	0.0	0.0	0.0
Assistant Lecturer	1.5	1.5	2.5	3.0	0.0	0.0	0.0
Lecturer II	2.0	2.0	3.0	3.5	4.0	0.0	0.0
Lecturer I	2.5	2.5	3.5	4.0	4.5	0.0	0.0
Senior Lecturer	3.0	3.0	4.0	4.5	5.0	5.5	6.0
Associate Professor	3.5	3.5	4.5	5.0	5.5	6.0	6.5
Professor	4.0	4.0	5.0	5.5	6.0	6.5	7.0

2.3 PROVISION FOR STUDENTS' ACTIVITIES CONSTRAINTS

Sometimes students' activities could constitute some constraints in the timetable scheduling process. Some of the students' activities include;

- Free lecture periods
- Students' union elections and activities
- Students' week

Most of these activities interrupt lectures and hence, increase the chances of the course lecturers' inability to complete their course outline in the semester. To salvage this situation, the

system has provided a platform where the days for some students' activities which are predefined at the beginning of a semester are identified and saved in the database. The days, times and venues for these activities are purposefully made unavailable for assignment in the lecture timetable. Moreover, the activities that are not predefined are equally planned for. The system provides a platform where the available times, venues, days, the students free periods and the course lecturer's free periods for a particular course affected by the activity are displayed. With the set of information, the courses affected can be rescheduled. Furthermore, the system also provides some value added services such as automated SMS alert, email alert and WhatsApp alert based on a student's subscription. The students' activities constraint management procedure is given in Figure 8 .

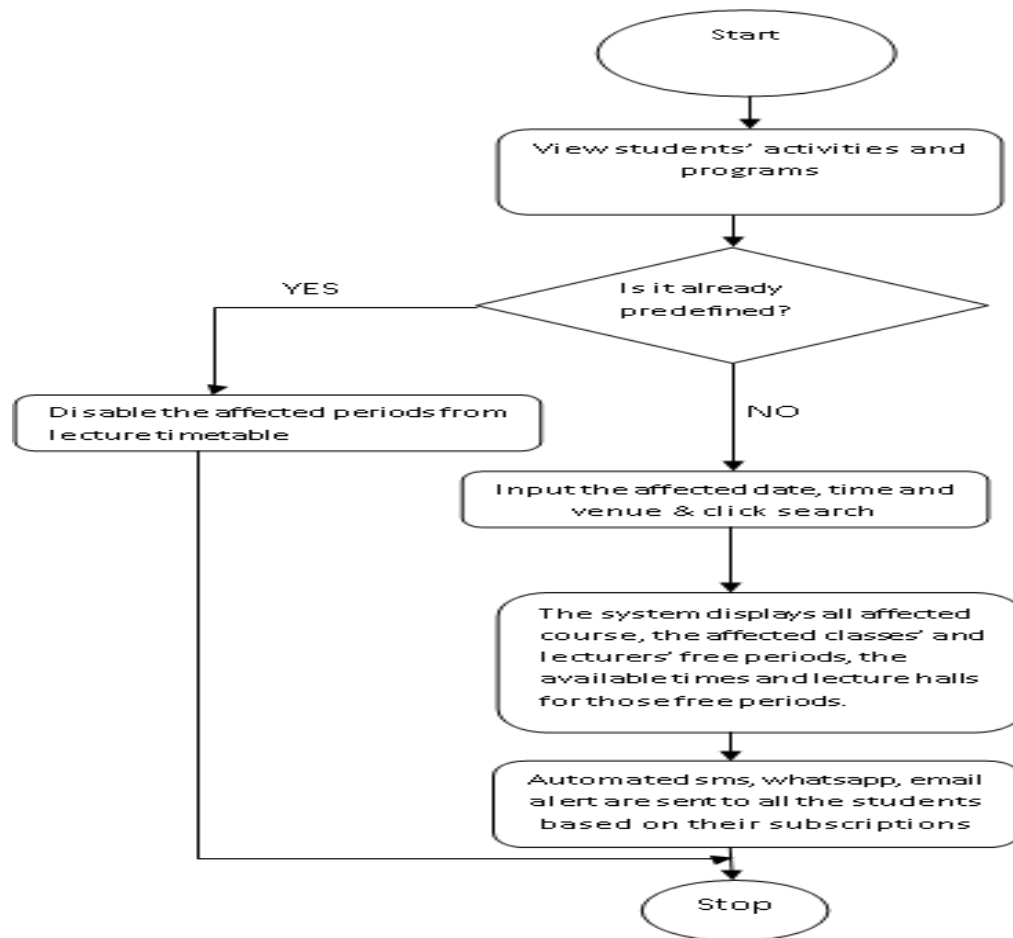


Figure 8 Students' Activities Constraint Management Procedure

2.4 ASSIGNMENT OF VENUES AND TIME TO COURSES

The required information pertaining to each lecture venue are:

- Venue name
- Location
- Capacity
- Coordinates
- Picture

This involves the actual venue and time allocations to courses. The following procedure is followed in allocating venues and time to courses :

- The venue information in the database is sorted by the lecture venue's capacity.
- The venues are matched with courses using the venues' capacities information and the value of the estimated class size for the course. At this point each course has a list of potential venues that can be allocated to it.
- Times are allotted to courses following first the course ranking starting with the course with the highest rank. For each course rank considered, the

list of lecturers offering the course of that rank are considered based on the lecture's ranking weight (W); starting with the lecturer with the highest ranking weight (W). The procedure is repeated until all the lecturers for the given course rank are scheduled (allocated time and venue) on the timetable.

The flowchart for the procedure is given in Figure 9. According to Figure 9, the repository of ranked courses realized from Figure.4 is matched with the repository of venues sorted by venues' capacities. This gives rise to a repository of courses with the list of potentially suitable venues. Time and venue are then allocated to each of the courses. Furthermore, the awful stress faced by students and lecturers in having lectures in venues very far away from their faculties or departments has been addressed. Only the possible catchment venues of a department or faculty are made available for assignment as soon as a course offered by the faculty or department is selected. Interestingly, this work has also considered the automation of anomalous alert capability which enables the system to trigger alerts when some anomalous cases arise. The anomalous situations include cases that the system cannot handle automatically. An example of such a situation is where the estimated class size is larger than all available catchment venues for any given course. Extremely important is the issue of clashes of courses especially amongst carryover students. In this work, the policy

that a student must pass a prerequisite course before offering the related course has been assumed. This has necessitated the adoption of a strategy whereby a course and its prerequisite

are taken simultaneously unless the same lecturer is teaching the courses.

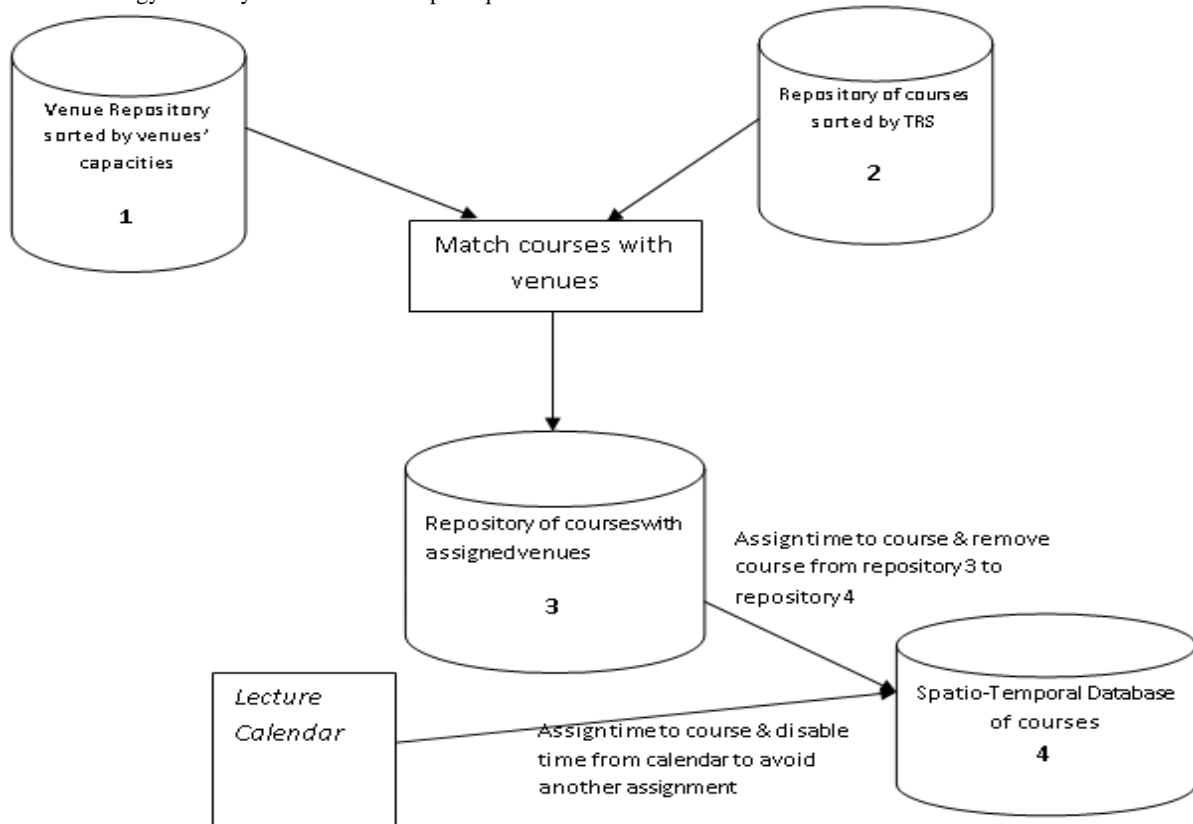


Figure 9: Procedure for assigning venues and time to courses

3. CONCLUSION

Mechanism for handling conflicts and constraints in university timetable management system is presented. The development of the relevant a set of algorithms and analytical expressions for the mechanism was presented. First, conflicts associated with courses, are handled by ranking the courses based on their total ranking scores (TRS) which is computed from such parameters as the class size for the course, the level of study and the number of different department that are offering the course. The course with higher total ranking score gets higher priority in course scheduling process. The conflicts associated with course lecturers are also handled by ranking the lecturers based on their status along with other vital parameters. Provision is also made to accommodate students' activities constraints. In the actual course scheduling procedure, the venues are first matched with the courses based on their carrying capacity which is enough to accommodate the course class size. Then the procedure for time allocation to courses on the timetable considers the course ranking first before the lecturers ranking. Importantly, the design constraints and decisions considered in this paper were based on information on the timetable issues at University of Uyo, Akwa Ibom state, Nigeria. In any case, the idea presented in this paper can be adapted or further developed and generalized to apply to other institutions.

4. REFERENCES

- [1] Oke, T. I., & Bukar, I. B. (2018). Exploring Alternative Sources of Funding Universal Basic Education for Sustainable Development in Nigeria. *KIU Journal of Humanities*, 2(2 (A)), 31-38.
- [2] Twumasi-Ampofo, K., Ofori, P. A., Tutu, E. O., Cobinah, R., Twumasi, E. A., & Kusi, S. (2017). Maintenance of government buildings in Ghana: the case of selected public residential buildings in Ejisu-Ashanti. *Journal of Emerging Trends in Economics and Management Sciences (JETEMS)*, 8(3), 146-154.
- [3] Oluwatumbi, O. S., & Olubunmi, A. V. (2017). Availability And Utilization Of Internet Facilities Among Undergraduate Students Of Colleges Of Education Nigeria. *British Journal of Education*, 5(9), 100-107.
- [4] Vrieling, R. O., Jansen, E. A., Hans, E. W., & van Hillegerberg, J. (2017). Practices in timetabling in higher education institutions: a systematic review. *Annals of Operations Research*, 1-16.
- [5] Ansari, A., & Bojewar, S. (2014). Genetic Algorithm to Generate the Automatic Time-Table—An Over View. *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, 2(11), 3480-3483.



- [6] Pillay, N. (2012). Hyper-heuristics for educational timetabling. In Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (pp. 316-340).
- [7] Sutar, S. R., & Bichkar, R. S. (2012). University timetabling based on hard constraints using genetic algorithm. *International Journal of Computer Applications*, 42(15), 3-5.
- [8] Petrovic, S., & Burke, E. K. (2004). Educational timetabling. *Handbook of scheduling: algorithms, models, and performance analysis*, 45-1.
- [9] Vrieling, R. O., Jansen, E. A., Hans, E. W., & van Hillegersberg, J. (2017). Practices in timetabling in higher education institutions: a systematic review. *Annals of Operations Research*, 1-16.
- [10] Ansari, A., & Bojewar, S. (2014). Genetic Algorithm to Generate the Automatic Time-Table—An Over View. *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, 2(11), 3480-3483.
- [11] Pillay, N. (2012). Hyper-heuristics for educational timetabling. In Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (pp. 316-340).
- [12] Sutar, S. R., & Bichkar, R. S. (2012). University timetabling based on hard constraints using genetic algorithm. *International Journal of Computer Applications*, 42(15), 3-5.
- [13] Petrovic, S., & Burke, E. K. (2004). Educational timetabling. *Handbook of scheduling: algorithms, models, and performance analysis*, 45-1.
- [14] Ugwu, O. O., Okafor, C. C., & Nwoji, C. U. (2018). Assessment of building maintenance in Nigerian university system: a case study of University of Nigeria, Nsukka. *Nigerian Journal of Technology*, 37(1), 44-52.
- [15] Eze, S. C., Chinedu-Eze, V. C., & Bello, A. O. (2018). The utilisation of e-learning facilities in the educational delivery system of Nigeria: a study of M-University. *International Journal of Educational Technology in Higher Education*, 15(1), 34.
- [16] Udidai, U. J., Essien, E. S., & Pisca, J. I. (2017). Institutional Activities and Accreditation of Higher Education Academic Programmes in the Era of Economic Recession in Cross River State, Nigeria.
- [17] Akinkunmi, T. (2016). Assessment of Maintenance Management Culture of Tertiary Institutions in Nigeria. *Assessment*, 8(6).
- [18] Hénard, F., & Roseveare, D. (2012). Fostering quality teaching in higher education: Policies and Practices. An IMHE Guide for Higher Education Institutions, 7-11.
- [19] Vandiver, B. (2011). The impact of school facilities on the learning environment. Capella University.
- [20] Henard, F., & Leprince-Ringuet, S. (2008). The path to quality teaching in higher education. Paris: OCED Publication.–2008.
- [21] Bullock, C. (2007). The relationship between school building conditions and student achievement at the middle school level in the Commonwealth of Virginia (Doctoral dissertation, Virginia Tech).
- [22] Lin, Y. P., Chang, T. K., Fan, C., Anthony, J., Petway, J. R., Lien, W. Y., ... & Ho, Y. F. (2017). Applications of information and communication technology for improvements of water and soil monitoring and assessments in agricultural areas—A case study in the Taoyuan irrigation district. *Environments*, 4(1), 6.
- [23] Oyier, C. R., Odundo, P. A., Lilian, G. K., & Wangui, K. R. (2015). Effects of ICT Integration in Management of Private Secondary Schools in Nairobi County, Kenya: Policy Options and Practices. *World Journal of Education*, 5(6), 14.
- [24] Forcada Matheu, N. (2005). Life cycle document management system for construction. Universitat Politècnica de Catalunya.
- [25] Berkhout, F., & Hertin, J. (2001). Impacts of information and communication technologies on environmental sustainability: Speculations and evidence. Report to the OECD, Brighton, 21.
- [26] Ramos, A. L. A., Matienzo, K. L. C., Casunuran, J. M. D., Nervida, C. M., Rosal, J. M. S., & Bederico, A. V. (2018). E-Vision: A Campus Locator Map Mobile Application using A* Algorithm. *International Journal of Computer Science and Software Engineering*, 7(1), 6-11.
- [27] Kulkarni, K. R., Yatish, C. H., Kamble, K. P., Kulkarni, A. A., & Bangi, S. C. (2017). Development of 2D Map and 3D Model of GIT Campus using GIS Technology.
- [28] Hall, C. S. J. S., & Hall, D. S. G. S. (2017). Campus map. *Lamp*, 99, 1.
- [29] Chowdhary, A., Kakde, P., Dhoke, S., Ingle, S., Rushiya, R., & Gawande, D. (2014). Timetable Generation System. *International Journal of Computer Science and Mobile Computing*, 3(2).
- [30] Doulaty, M., Derakhshi, M. F., & Abdi, M. (2013). Timetabling: A state-of-the-art evolutionary approach. *International Journal of Machine Learning and Computing*, 3(3), 255.
- [31] Murray, K., & Müller, T. (2006). Automated System for University Timetabling. In Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling (pp. 536-541).
- [32] Smith, S. F. (2005). Is scheduling a solved problem?. In *Multidisciplinary Scheduling: Theory and Applications* (pp. 3-17). Springer, Boston, MA.
- [33] Chand, A. (2004). A constraint based generic model for representing complete university timetabling data. In Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (pp. 125-150).