Transition From Observation To Knowledge To Intelligence (TOKI)

Editors
Dr. Victor ODUMUYIWA, Dr. Olufade ONIFADE,
Prof. Amos DAVID & Prof. Charles UWADIA
Transition from Observation to Knowledge to Intelligence

3rd Biennial International Conference on Transition from Observation to Knowledge to Intelligence (TOKI)
15-16 August 2019
University of Lagos, Nigeria

Editors

Dr. Victor ODUMUYIWA
Dr. Olufade ONIFADE
Prof. Amos DAVID
Prof. Charles UWADIA
Technological Discourse and Timetable Systems: Framework Development

AWORINDE, Halleluyah Oluwatobi
Department of Computer Science & Information Technology
Bowen University, Iwo, Nigeria.

SALU, John Ayodele
Syssoft Consultants Limited, Lagos, Nigeria.

OKEDIGBA, Temilola Oluwafunto
Department of Computer Science & Information Technology
Bowen University, Iwo, Nigeria

OLADELE, Femi
Department of Accounting
Bowen University, Iwo, Nigeria.

Abstract. One significant activity in tertiary institutions, which is still largely manual is the timetable system although, it remains a critical aspect of every institution. Poor coordination of the timetable for teaching and/or assessment can significantly undermine quality assurance and may lead to a deadlock among every member of the university community. Given this condition, we set out to advance (change) the technological discourse, which approaches timetabling by a manual chess-board-like game with an automated program that integrates three (3) validated approaches which are the Genetic Algorithm, Tabu Search and North-West Corner Method, evolved and fused into the Halsal chameleon algorithm on a flexible or generic week structure. In this study, we document the development and framework of an optimized timetable management system, in changing the technological discourse in a private tertiary institution in Nigeria. The purpose is to analyze the effectiveness and/or efficiency, as well as the sustainability of the new system, in following the goal of becoming a world class university in the 21st century.

Keywords: HALSAL Chameleon algorithm, Constraints, Week flexibility, Time Table, Tabu Search, Genetic Algorithm, North-West Corner Method
1. Introduction

The operational and functional areas (admission processing, human resource management, lecture delivery, assessment, result processing and database management, etc.) of many tertiary institutions in Nigeria still align with manual systems. Technological landscapes are however changing within and outside our jurisdictions and this makes the digital divide more prominent (Wang and Shih, 2009; Sabi et al., 2016; Bankole and Bankole, 2017; Zhang, 2017). This situation may be attributable to the inadequacy and/or non-existence of many support infrastructure required to initiate, operate and sustain technological requirements. However, technology continues to become more pervasive, and its acquisition cost continues to drop, it leaves interested organizations with reduced cost even at adoption albeit an incidence of high cost when there is need for exclusive manpower training. Technology use affords many benefits such as enhanced efficiency, effectiveness and productivity, yet critical fear factors include information overload (Wessels, 2004), trust (Akinwunmi, Olajubu and Aderounmu, 2015), overtrust (Hardre, 2016) threats to security and safety (Worrell, Bush and Di Gangi, 2014) complacency and phobia (Ahadiat, 2005; Humphrey and Beard, 2014; Kearns, 2016; Orr, Williams and Pennington, 2009) among others. Moderating factors for the use of technology include experience, voluntariness of use, gender and age (Venkatesh et al., 2016). This places organizations especially tertiary institutions in a dicey position, given the complexities and peculiarities of the academe; hence, the rationale behind this research.

2. Literature review

2.1. Overview of timetable and timetabling systems

Traditionally, timetable has been constructed by hand and then modified as time goes on (a process known as local repair). The solution to timetabling problem can be approached from a heuristic approach which is the transformation of a goal based on human way of solving it
in order to ascertain the most effective method in achieving the goal (Barbara, 2002).

Using technology to solve the timetable problem has graduated from using a single heuristic technique to the combination of several heuristic techniques. Techniques such tabu searching, simulated annealing, and genetic algorithm among others have been combined to enhance productivity in solving several problems including time table scheduling. The direct heuristics algorithm focuses on the idea to simulate the human way of solving the problem by attempting to arrange classes suitable to the lecturer, such that no lecturer will be billed for two classes at the same time slot. This heuristic approach can solve various constrictions such as courses that have various time, joint courses or performed at one time, the distribution of courses in a week and availability of lecture theatre required. This approach is limited to the scope of the method it is carved from; it cannot perform beyond the scope and cannot deliver less than the method. The genetic algorithms (GAs) are adaptive methods which may be used to solve search and optimization problems. They are based on the genetic processes of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection and “survival of the fittest”. By mimicking this process, genetic algorithms are able to "evolve" solutions to real world problems, if they have been suitably encoded (Liu and Ceder, 2017). Genetic algorithm has basic processes which cover the initialization, choice, matching and birth of a new one, these processes are initialization, evaluation, selection, crossover, mutation and repetition. The simulation of evolution allows survival of better individuals and extinction of inferior ones.

Evolution’s goal is to find better individuals in each generation. The process of evolution is maintained by selection, crossover and mutation (Franco, 2016).

Tabu search algorithm makes extensive use of local search as it proceeds through the search space it avoids local minima (the major problem associated with local search algorithms) by modifying the set of neighbours around the currently selected solution. For this the Tabu search can be applied to both the timetabling problem and the grouping
sub problem. Tabu search technique is one of the popular local search method based on neighbourhood search algorithm, which can virtually be applied to any kind of optimization problem (Deeksha et al, 2015). There are various kinds of moves which are OUT-IN move, INTRA or SIMPLE move and SWAP move (Çağdas and Gülüşum, 2007). An OUT-IN move moves one unscheduled lessons to table. This move favours minimizing unscheduled lessons as it has the highest weight among criteria and is quite effective (Hooshmand, Behshameh, Hamid, 2013). An INTRA move changes the period of a scheduled lesson in the table to a new period inside the table that is, it can be obtained from the table by changing the assignment of one lesson from one period to another period (Çağdas and Gülüşum, 2007). SWAP move can be obtained by interchanging the periods assigned to the two lessons based on meeting the requirements of the constraints on the two lessons (Çağdas and Gülüşum, 2007).

Simulated annealing (SA) is a probabilistic local search technique for finding solutions to optimization problems, it simulates the behaviour of metal atoms during the process of annealing (a treatment involving extremes of temperature). An initial allocation is made in which elements are placed in a randomly chosen period. The initial cost and an initial temperature are computed. In order to implement the SA algorithm a neighbourhood structure must be defined. This is the key component of any simulated annealing method, three algorithms are handled in different combinations. In each iteration of the algorithm, neighbourhood searching is performed once to find out the next possible solution set (Aycan and Ayav, 2008).

The application of the North-West Corner method to the time tabling problem aims at providing the best possible movement to the grid array of the time table, it is a method used to solve transportation problems of demand and supply. In the North-West Corner method, the largest possible allocation is made in the upper-left hand corner of the tableau followed by adjacent feasible cells (Ahmed et. al., 2016).
2.2. Empirical evidence

Timetabling automation has been on the front burner in universities for as far back as the 19th century and maybe earlier. In an article published in 1992, we see reviews of articles dating back even to 1966 in search of an efficient technology-driven timetabling system, obviously using the mainframe (Fahrion and Dollansky, 1992). This means that timetable continues to influence technological discourse in the university system. Basically, the timetable construction problem is an assignment problem (Fahrion and Dollansky, 1992) that has no exact solution, although it is possible to find an efficient heuristic algorithm to solve the problem. Diverse methods have been used to solve timetabling problems. Logic programming techniques was used to construct a university faculty timetable (Fahrion and Dollansky, 1992). The heuristic priority scheme was used to ensure that “feasible instructor/room assignments under additional consideration of the faculty’s education plan and certain a priori fixed assignment options” were efficiently programmed (Fahrion and Dollansky, 1992).

In Germany, the relation-algebraic specification method was used to propose a solution to timetabling problems in a university-owned secondary school (Berghammer and Kehden, 2010). In Egypt a genetic algorithm method was used (Abdelhalim and El Khayat, 2016a) and another study used RFID technology based on the genetic algorithm (Abdelhalim and El Khayat, 2016b). In Malaysia, an intelligent timetabling system using graph colouring methods was developed to solve a University’s timetabling problem (Ayob et al., 2011). In the transportation industry, a bi-objective, bi-level integer programming model using a “novel deficit function (DF)-based sequential search method combined with network flow and shifting vehicle departure time techniques is proposed to achieve a set of Pareto-efficient solution” (Liu and Ceder, 2017). The simulated annealing was also used considering hard constraints of time tabling parameters such as a number of subjects, number of lecture or tutorial sessions, number of classrooms, number of teachers, number of students and number of workloads (Basir, Ismail and Norwawi, 2013).
Many of the reviewed studies use one approach to solving the timetabling problem. We however note that a singular approach may not be sufficient to produce an all-encompassing solution, so we develop a framework using the multi-approach system. The algorithm used for this project is a hybrid developed algorithm, called the HALSAL Chameleon algorithm. The algorithms being adopted to build the HALSAL Chameleon algorithm are genetic algorithm which is combined with its evolved form which is the nested genetic algorithm, tabu searching algorithm and North-West corner method algorithm.

3. Development

The Software methodology adopted for this project is the agile process and the model applied is the Incremental development model (Humphrey and Beard, 2014). It is the best approach based on the changing desires of human behaviour in respect to the unpredictable decisions or choices in the academic system, to get the general view of students and lecturers using data evaluation form as a method of review that generates statistics to grade the performance of the application, using those reviews to upgrade and repair each version created. The initial input of courses is set to be the list of courses available i.e. the list of courses that have not been assigned a venue and have not been allocated to a period based on the unit of each course input. The data structure of the list of available courses is the FIFO (first in first out), this data structure takes in the full list and starts picking each course from the beginning of the list. When a course is selected it goes through the constrain check, the algorithm uses the nested-genetic algorithm to search for a suitable venue as a partner on the constraint of the size of the students taking the course in respect to the capacity of the venue, location of the venue, territory of the venue, type of venue as regards the type of course selected, and availability of the venue, which if all constraints are well met the course evolves from a single course to a selected course merged with a venue which then proceeds as the merged course removed from the list of available courses as well as the venue
is removed from the list of available locations, else the course remains
in the list of available courses and the selection is moved to the next
available course on the queue.

The algorithm takes the next step to place the selected course into
an available period with the use of the nested-genetic algorithm and
north-west corner method it searches starting from the top-left corner
of the table and looks for a suitable partner on the constraint of the
availability of the period, availability of the lecturer, clashing courses
for students in related departments in respect to their level, course
occurrence in relation to the period occurrence and the designed week
structure of the time-table, if all constraints are met on a selected period,
the algorithm uses the nested-genetic algorithm to merge a course +
venue with a selected period, which then evolves the selected course to
a course + venue + period, after this the algorithm proceeds to the next
available course and goes through the process as described above. On
account where a selected course doesn’t find a suitable period, the
course regresses to its initial state by disintegrating the venue from the
course and returning each item to its available state. The process is
repeated in rounds where each round has a degree of constraint check
in ascending order until the best timetable format possible is produced.

When the timetable is generated, the algorithm applies the Tabu
searching algorithm for amendments on unallocated courses (if any)
and allocated courses depending on the user’s preference and a
constraint check based on the type of move applied on the Tabu search.

The time taken for the process to be complete is relatable as done
manually in the sense that, the larger the data the more time taken to
produce its result and vice-versa. Using the HALSAL chameleon
algorithm time taken on a process to fit a course to a period is done in
less than 0.5 seconds, which when accumulatively compared to the
manual process which may take weeks to provide an adequate fully-
functional time table, will take the Time Shuffle application minutes to
produce.

The HALSAL Chameleon algorithm was applied as the back-bone
for the operation in a standalone desktop application developed in c#
using the Microsoft Visual Studio, and runs on the .NET Framework on the Windows operating system platform.

4. Result and implementation

This part covers the implementation and results obtained from the operation. The results of the Time Shuffle system are arranged into three for the university timetable, i.e., academic timetable, examination timetable, and invigilation schedule for the examination timetable. The Time Shuffle treats each university data as a structure, and each structure holds the full hierarchical step of the university academic system, starting from the faculties, the departments, staffs, venues, and courses, including its properties, using the data in the structure to generate the timetable as shown in Figure 1.

The software will generate a full educational institution academic, examination and invigilation schedule timetable showing each course’s
period, the day of the week it occurs, lecturer taking the course, departments taking the course, and faculty the course’s department belongs to, based on the input system which will be open for modification and automated suggestions (Murray and Müller, 2007). The academic time table is generated based on the data put into the structure as shown in Figure 2, it puts into consideration the various constraints for the academic time table, such as the course occurrence, lecturer constraints, department and department’s attendance constraints, resource constraints and course constraints. The Examination time table is generated putting into consideration the resources available, course constraints, resource sizes in respect to course size, department constraint and period constraint (Ramón, Enric, and José, 1997) as shown in Figure 3. Indicating the number of weeks you want each examination to hold for. The invigilation schedule is generated based on the present examination time table generated as shown in Figure 4, using the hierarchical representation of the staffs in academic system each rank of staff has a maximum number of schedules they can invigilate.
Figure 2: The generated academic time table for the university
Figure 3: The generated Examination time table for the university

<table>
<thead>
<tr>
<th>DAY</th>
<th>VENUE</th>
<th>9:00 am - 11:30 am</th>
<th>12:00 pm - 2:30 pm</th>
<th>3:30 pm - 6:30 pm</th>
</tr>
</thead>
<tbody>
<tr>
<td>MONDAY</td>
<td>GENLAB I</td>
<td>PHY 461(205), FST 212(205)</td>
<td>* Eucalyptus C. NCIT(T)</td>
<td>Operana O (RAIT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Kereko J (AMAT)</td>
<td>Ficus esa E (OBRAT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Anfich E (RAIT)</td>
<td>Anfich E (RAIT)</td>
</tr>
<tr>
<td></td>
<td>GENLAB II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHI</td>
<td>ARA 422(205)</td>
<td>FHS 318(205)</td>
<td>* Osunye E (NCIT)</td>
<td>Atele A (AMAT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Anfich E (RAIT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mirus E (NCIT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Kafe F (OCT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: The Invigilation schedule for the examination time table
4 System Evaluation

The system was subjected to series of testing in order to validate its performance compared to existing systems using metrics such as simplicity, usability, time taken, accuracy, compatibility, and integration. Thirty-six (36) respondents (students, staff and faculty) evaluated the system and rated it on star-rating basis alongside their remarks and comments. A full statistic of all evaluations gathered gives a picture of the overall performance of the system with respect to views of the evaluators.

The result is graphically represented in this work using a bar chart and tabular structure which were as well embedded in the system and shown in figures 5 and 6 respectively.

Figure 5: Evaluation Statistics Chart board

Figure 5 above show that a total of 36 respondents evaluated the system by responding to varying questions. Based on the star ratings from 1 (lowest) to 5(highest), the result indicated that 24(66.67%) respondents rated the system 5stars and 12(33.33%) respondents gave
a 4-star rating. We can presume that the project will enjoy wide acceptability among the university community from the results.

Figure 6: Tabular Representation of the Evaluation Statistics

Figure 6 gives a tabular breakdown of the individual respondent’s assessment of the system on the basis of the various metrics put in place to evaluate the system after which the system on the overall automatically generates the corresponding star rating of the work after summation of the various grades given under each of the metrics. The tabular representation of the evaluation statistics equally corresponds with the bar chart presented in figure 5 above.

5 Conclusions

The Time Shuffle system has been designed to perform beyond the limits of other automated systems, and have been able to improve on other automated systems weaknesses, in the following ways such as:

**Week Structure Flexibility:** Time Shuffle system will provide a flexible week structure to adapt to any universities academic system, i.e. enabling its users to decide what day of the week they do or do not want courses to be allocated or what period of the day they do or do not want courses to be allocated.

**Resource Consideration:** Time Shuffle system will provide a flexible resource constraint to adapt easily to any university’s resource
capability. On the issue of unequal venue sizes, Time Shuffle will enable each course to be merged to the perfect match or most optimized venue based on its size to make sure there is an efficient use of the venue resources.

**Course Placing Consideration:** Time Shuffle system will take the placing constraint factors into consideration, factors such as maximum course occurrences for academic timetable generation which by default will be set to two consecutive course occurrences in each day based on the standard stress-level and learning capability of the student and lecturer but can be adjusted for flexibility of the university’s system also other factors such as the lecturers handling multiple courses and courses handled by multiple lecturers.

**In-Built Tutorial Accessibility:** Time Shuffle system will hold an in-built tutorial access to allow first time users or new users to understand the working, operations and applications of the time table without the need of an expert. In other words, the Time Shuffle system will be able to be self-taught.

**Custom Handling of Data:** Time Shuffle uses a custom platform for holding data i.e. serialization of the objects into serialized data, which will provide a level of security given that the saved data or information will only be viewed or accessed in the Time Shuffle system. **Multiple Hands on Deck:** Time Shuffle will provide a platform for team work, depending on the university’s system, Time Shuffle will allow each department or faculty to control their data independently and also work with the time table committee by exporting their data in a custom format to the time table committee to import and generate the time table. It will also support single collation of data based on the user’s choice.

**In-Built Evaluation Statistics:** Time Shuffle will generate and provide an evaluation form for each user to view and give their opinions about the performance, operation, accuracy and other factors of the system, including their remarks on what should be improved in the system.
Independent and Light: Time Shuffle will be able to work efficiently without the use of the internet or any external application such as a database manager, it can handle its own data and operations by installing it only. Also data or information saved weigh very little as the average size of a university’s data on it will not be more than 20MB.

Security on Structures: Time Shuffle will provide an optional knowledge-based mode of security via the use of passwords on each structure created, for protection and privacy of one’s work.

All in one package: Time Shuffle will provide the academic, examination and invigilation schedule in one application or system, whereas most automated systems have these capabilities separated in different systems or specialize in just one area.

Portability: Time Shuffle will be able to be easily moved from one system to another has the entire operation requires a double click for a one-time installation.

The automated time table organization and management system provides a more efficient, easy, and reliable generation of timetable results into the academic system of the university. As the world evolves there are other automated Timetabling systems developed, Time Shuffle system has been able to look into these systems, spot their flaws and build on that to produce an ultimate Timetabling system. With all factors and aspects of university academic hierarchical system taken into deep consideration Time Shuffle produces the best timetable results.

List of References


Kearns, L.R., The experience of teaching online and its impact on faculty innovation across delivery methods, The Internet and Higher Education. 31 (2016) 71–78. doi:10.1016/j.iheduc.2016.06.005.


Franco, B., Genetic algorithms overview, 2016.


Fahrion, R. & Dollansky, G., Construction of university faculty timetables using logic programming techniques, Discrete Applied
0166218X92902467-main.pdf?_tid=5024a9ce-786b11e7-b84d-
00000aab0f02&acdnat=1501778976_b51929298568c47d5cd29
3a54c27e1 (accessed August 3, 2017).

Berghammer, R. & Kehden, B., Relation-algebraic specification and
solution of special university timetabling problems, The Journal of

Abdelhalim, E.A. & El Khayat, G.A., A utilization-based genetic
algorithm for solving the university timetabling problem (UGA),

Abdelhalim, E.A. & El Khayat, G.A., An information visibility-based
university timetabling for efficient use of learning spaces (IVUT),

Ayob, M., Hamdan, A.R., Abdullah, S., Othman, Z., Zakree, M.,
Nazri, A., Razak, K.A., Tan, R., Baharom, N., Ghafar, H.A., Dali,
R.M. & Sabar, N.R., Intelligent Examination Timetabling Software,

Basir, N., Ismail, W. & Norwawi, N.M., A simulated annealing for
Tahmidi course timetabling, Procedia Technology. 11 (2013) 437–

Murray, K. & Müller, T., Automated system for university timetabling,

Ramón, Á.-V., Enric, C. & José, T.M., A tabu search algorithm to
schedule university examinations, Qüestiió. 21 (1997) 201–205.