A Tabu Search-based University Lectures Timetable Scheduling Model

Adewale O. Sunday
Department of Computer Science
School of Computing
Federal University of Technology
Akure, Nigeria

Ibam E. Onwuka
Department of Information Systems
School of Computing
Federal University of Technology
Akure, Nigeria

Izundu Kingsley E.
Department of Computer Science
School of Computing
Federal University of Technology
Akure, Nigeria

ABSTRACT
Scheduling University Courses is regarded as a Non-deterministic Polynomial-time hardness (NP-hard) problem. This is because no universal constraint works for all universities. While some will have constraints similar, they might differ in their resource values - length of days, time slots, and rooms. Several literatures have been able to address several constraints, using various optimization methods - genetic algorithms, tabu search and so on. The result often time works but lacks adoptability due to their non-inclusiveness of some resource parameters - day and dynamic time-slot. In this research, we address the various constraints related to the Federal University of Technology Akure (FUTA) using mathematical model that includes the necessary resource parameters. We adopt Tabu search diversification approach to implement a scheduling system that satisfies the constraints defined.

General Terms
Polynomial-time hardness (NP-hard) problem, Optimisation, Tabu Search, Scheduling.

Keywords
Timetable, Tabu Search, Constraint, Diversification, Economy Code Protocol (ECP), Mathematical Model, Comfort Adjustment Factor (CAF).

1. INTRODUCTION
University timetabling problem has been considered in different researches as it directly affects the education quality and should satisfy existing rules and constraints [1].

Timetable is an event table used to specify who will participate, who will be held where and when such an event occurs [2]. According to [3], the course timetabling problem can be divided into five sub problems: teacher assignment, class-teacher timetabling, course scheduling, student scheduling and classroom assignment.

The timetabling problem arises in many real-life circumstances. A general timetabling problem includes many events such as but not limited to exam scheduling, course scheduling, meetings and training, computer jobs and so on. There is a large amount of successful research on Timetabling Problem. For different timetabling problems, these researches have been done by various meta-heuristic approaches. Timetabling using Tabu Search Diversification involves scheduling of courses to satisfy faculty requirements, consider limited resources such as room availability and time slots.

In this work, Federal University of Technology Akure is considered as a real case. We specifically focus on undergraduate level. There are a number of rules that should not be violated. Also, there are some constraints that should be satisfied. Some of them are common in timetabling problems whereas others are specifically imposed in the case. We have had timetables which, neither satisfied lecturers, students nor schedulers. For example, students wish to attend classes one or two days in a week and have more time for personal studies, some students who take audited courses would want to partake fully in the audited courses as well as their level courses. Lecturers wish to specify the day and time they would conveniently offer their lectures to students. While Schedulers or those in charge of preparing the timetable painstakingly sorts it to each respective departments even when they manually succeed with the master timetable sheet.

The objectives of this research therefore are:

- Construct a mathematical model that addresses various scheduling constraints applicable to FUTA.
- Design a system using Tabu Search approach that addresses the constraints.
- Evaluate the performance.

1.1 Constraints
The Timetabling Problem has some constraints. While preparing course schedule, lot of constraints arise. These constraints can be divided into two types depending on priority. One is hard constraints which cannot be avoided and another is soft constraints. Constraints are varying from institution to institution but most of the time constraints are similar. In this work, we describe different types of constraints and find the feasible solution.

1.2 Hard Constraint
In Hard Constraint, the following are observed:

- Resource allocated to a joint course/labwork must be different from resource of the respective department courses. i.e., intra-class and interclass of same school.
- Resource allocated to a joint course/labwork must be different from resource of other department courses.
- Resource allocated to the parent course e.g. CSC101 (1) must be different to resource allocated to the other CSC101 (2)

1.3 Soft Constraint
In Soft Constraint, the following are observed:

- As much as possible, resource allocation should consider Economy Code Protocol (ECP) capacity/population constraint.

- Lecturers have the choice of choosing a start time and finish time as well as day and room for lectures.

Addressing the hard Constraint would make it possible to meet our objective of achieving non clashing schedules. Also the soft constraint as much as possible should be satisfied. Though, due to the stochastic values of population number and lecture room capacity, we might not be able to provide a very optimal satisfaction as regards the soft constraint; in this case manual scheduling of such resources is the last resort.

In the proposed model, we also considered resource limitation and need to utilize resources; we considered the total population of students taking a course to the lecture room capacity to be used. Not being bias; we deduced an adjustment function which will take care of this so that we would not have to allocate a lecture room of capacity 1000 to student whose population is 30% of that capacity. This is considered unfair in resource management.

In this work, we present a simple but effective approach to solve this timetabling problem by using Tabu Search diversification technique. In Section 2, we describe some related published research works in the area of discuss. Section 3, contains the system modeling, Section 4 contains the system model implementation and evaluation while Section 5 is the conclusion. In conclusion, the system will take care of when to assign a course or laboratory work, where to use for such event based on constraints and which lecturer handles such event; the research is focused on maximizing class attendance experience and enhancing education quality.

2. RELATED WORKS

This section gives concise description of some relevant related research works reviewed to have relevant tools and data need for this work.

[4], generated an interactive timetable. It was suggested that interactive timetable (computer allocating activities and user interfering with the process) was more convenient than a full automated scheduler. In order to satisfy the need for an interactive timetable, an ad-hoc algorithm was designed. The algorithm provided some (partial) solution at each step. The research insisted more on interactivity of the algorithm which meant two things: the difference between the solutions provided by the algorithm in two consecutive steps should be minimal and the user can interfere with the solution process. The user was able to change parameters of activities and resources, add or remove dependencies, and allocate or detach any activity from the time slot. The algorithm was tested using real data from autumn semester 2001 at the Faculty of Mathematics and Physics, Charles University.

[5], used an efficient tabu search approach for the timetabling problem. Three satisfaction criteria were stated i.e. organizational, pedagogical and personal costs. The research considered school timetabling problem to be a very difficult combinatorial optimization problem. In other to achieve the satisfaction criteria, tabu search heuristic was adopted. The research considered diversification strategy an important aspect in the design of a TS algorithm. Since the use of a tabu list is not enough to prevent the search process from becoming trapped in certain regions of the search space, two main approaches towards the diversification strategy involved the use of adaptive relaxation and random restart. In adaptive relaxation the costs involved in the objective function are dynamically changed to bias the search process to newly, unvisited, regions of the search space. In random restart a new solution was generated and no previous information was utilized. In conclusion, the research employed a TS algorithm that uses an informed diversification strategy, this took into account the history of the search process to bias the selection of diversification movements.

The research result supports our adopted method of scheduling. One limitation with the research was that it did not consider inter-classes and intra-classes scheduling.

In [6], a timetabling solver based on forward search algorithm was developed. This was purposely to create and modify course timetabling at Purche University that better meet student course demand and allowed students to be assigned to the constituent course sections in a way that minimized conflicts. The algorithm was similar to local search methods; however in contrast to classical local search techniques, it operates over feasible, though not necessarily complete solutions. One limitation was that some classes were left unassigned. Hence the soft-constraint which was to minimize total number of student conflicts among students enrolling in several classes was not implemented. The research result contradicted [5], who proved tabu search as a local method to provide much recognized quality in timetabling system. With this contradiction it could be agreed that even though tabu search is a well acceptable local search method, the use of tabu list is not enough to prevent the search process from being trapped in a certain region of the search space. Therefore our research adopts diversification strategy according to [5].

[7], focused more on lecturer's workload in order to create a balance between lecturer's workload as a teaching staff of the university. They used the *Tridharma Perguruan Tinggi* issued by Education Ministry of Indonesia as reference in defining activities of lecturers which lead to some calculations to obtain optimal university timetable. According to the research paper, the rule of *Tridharma Perguruan Tinggi* consisted of three dharma. They are Dharma of Education and Teaching, Dharma of Research and Dharma of Public Dedication.

[8] present A PSO algorithm to solve a Real Course+Exam Timetabling Problem. They observed that a University course and exams timetabling is a very well-known constrained problem with the problem becoming more difficult when we have to face a dynamic timetabling problem where new courses and exams can appear during the semester. They propose a particle swarm approach to simultaneously solve both problems as well as a local search procedure to handle the dynamism. Technical University Federico Santa Marıa was used in their study where two main problems are identified: static problem and dynamic problem. In the static problem, a problem solution must satisfy the following hard constraints:

- Lectures having students in common can not take place at the same time-slot;
- Lectures must take place in a classroom suitable for them in terms of facilities and capacity;
- Two lectures of the same course scheduled in consecutive time-slots must be assigned to the same classroom;
- Two lectures cannot take place at the same time-slot in the same classroom; and
- Each lecture must be assigned to a classroom in its corresponding time-slot

In the dynamic problem, during the semester new requirements must be satisfied, as exams and new courses, thus the solution found for the static problem must be altered and address two observed main situations:

• There is an available classroom that satisfy the constraints, so it can be assigned to this new activity.

• When a classroom is not available, some re-assignments are required in order to update the original course

planning.

They presented mathematical models and two-phased algorithms, particle swarm algorithm for the static problem and the dynamic problem used a local-search approach that works on the solution found by the particle swarm algorithm. They used used a real problem data obtained from the UTFSM which considered around 950 lectures, 49 classrooms and 48 time-slots in a week. They reported that the particle swarm approach has shown to solve very quickly the initial static problem, optimizing the use of small classrooms, to let the biggest ones available for exams. And therefore, finding a classroom with a big capacity for an exam becomes easier for the local search procedure during the dynamic phase of the timetabling. However, their work did not show how audited courses in each school/faculty are handled and also how inter-classes and intra-classes scheduling are handled.

In [2], a mathematical model for determining timetabling that minimizes the number of students with conflicting schedules was looked into. Three procedures were considered in other to achieve the above objective, which were;

i. Apply filling rules that can be based on some type of criteria like teacher assignment, or in some pedagogic features.

ii. Evaluate how many students could not attend a course, although he or she already has the prerequisites this may happen because two courses that a student could attend is sharing one or more slots in timetable.

iii. Return to (i) and apply different rules.

The result from the research provided a computation system which was developed by following the procedures above, this computation system proved to be another approach towards achieving our objective and we adopt it in the form of an algorithm. One limitation with the research was that the mathematical model for problem of assignment of course which could have led to building the computational system was not addressed while building the system, as in our research we would consider mathematical model towards building a non-clashing timetable system.

[9] considered two algorithms, Genetic Algorithm (GA) and Tabu search algorithm in solving their class timetabling scheduling problem. The research considered GA due to its ability to handle complex search space with high probability of success in finding the optimal solution and Tabu search is used due to its memorized ability to prevent from searching previous visited area.

[10] present A Review of Distributed Multi-Agent Systems Approach to Solve University Course Timetabling Problem. Their aim is to analyze a new approach to solve university course timetabling problem called an approach based on multi agent systems (Cooperative Search) in addition to briefly study approaches based on operational researches, metaheuristic methods and intelligent novel methods. They observed that constraints in UCTTP problem are classified into two classes of hard and soft constraints. Hard constraints must be satisfied in the problem completely so that the generated solution would be possible and without conflict; no violation is allowed in these constraints. Soft constraints are related to objective function; objective function is to maximize the number of satisfied soft constraints. Unlike hard constraints, soft constraints are not necessarily required to satisfy; but as the number of these satisfied constraints increases, the quality of solutions of objective function increases.

They presented a formal definition of UCTTP problem as: *n*: the number of events *E*={*e1, e2,* ... , *en*}, *k*: the number of timeslots *T*={*t1, t2,* … , *tk*}, *m*: the number of rooms *R*={*r1, r2,* … , *rm*}, *L*: the number of rooms' features *F*={*f1, f2* , ... , *fl*} and *s*: the set of students *S*={*s1, s2,* ... , *ss*}. They presented the Penalty Function (PF) for Hard and Soft constraints. They identified approaches used in the study of UCTTP as follows: The first definition of timetabling presented as three sets of: (1) teachers, (2) classrooms and (3) timeslots. And the approaches used in solving the UCTTP problem includes: (1) Operational Researches (OR) based techniques including graph colouring theory based technique, IP/LP method and Constraint Based Satisfaction(s) technique (CPSs); (2) Meta-heuristic approaches also including Case Base Reasoning method (CBR), population based approaches and single solution based approaches where the population based approaches includes Genetic Algorithms (GAs), Ant Colony Optimization (ACO), Memetic Algorithm (MA), Harmonic Search Algorithm (HAS) and single solution algorithms also includes Tabu Search Algorithm (TS),

Variable Neighborhood Search (VNS), Randomized Iterative Improvement with Composite Neighboring

algorithm (RIICN), Simulated Annealing (SA) and Great Deluge Algorithm (GD); 3) multi criteria and multi

objective approaches; 4) intelligent novel approaches such as hybrid approaches, artificial intelligence based approaches, fuzzy theory based approaches and 5) distributed multi agent systems approach. They reported how Tabu search algorithm has been applied by [11] for the first time to allocate students to courses and also balance the number of students in one submitted group where the first phase is to generate a set of solutions for one student, the second one is to combine a set of solutions with applying Tabu search with local strategies and the third phase is to allocate rooms and improve the allocation, of course without modifying the initial allocation of courses to timeslots.

Due to time consuming and complexity involved with timetabling, [1] employed a mathematical model approach. Hence, they devised a binary model to develop timetable for an Iranian University. The model assigns each course to the most eligible lecturer; developed based on the real constraints of the case and was solved using the GAMS software.

[12] present An Intelligent Bio-Inspired Algorithm for the Faculty Scheduling Problem. They observed that the manual assignment of courses is a very tedious and time-consuming task that the scheduling committee frequently faces in every department. In order to solve this timetabling problem, they proposed a novel approach using the Bees Algorithm (BA), which is inspired from bees' foraging behaviour, hybridized with Demon algorithm and Hill Climbing for more extensive search. The scheduling process took into consideration all constraints and variables associated with scheduling courses, according to the requirements of the Computer Science department in their college.

They reported that the schedules produced from the algorithm outperformed the manual schedules in terms of achieving the objective function and satisfying the constraints. And that the hybridized version produced better results than the standard BA version without hybridization. According to them, the hybridized algorithm is designed for faculty scheduling, but can be further generalized to solve various timetabling problems. The limitations of their work are: the test case carried out using data from a single department is not sufficient enough considering that the university is made up various departments and offers many courses, even though they reported that their work can be further generalized to solve various timetabling problems, they failed to describe how it could be achieved.

Finally, our research adopts diversification as proposed by [5], to be used with tabu search in order to expand the search space and derive scheduling slots as much as possible.

2.1 Tabu Search

Tabu Search {TS) method is a general heuristic procedure for guiding search to obtain good solutions in complex solution spaces. Its rules are sufficiently broad that it is often used to direct the operations of other heuristic procedures. One of the main components of TS is its use of *flexible (adaptive)* memory, which plays an essential role in the search process. With roots going back to the late 1960's and early 1970's, TS was proposed in its present form a few years ago by [13] in his paper "Future Paths for Integer Programming and Links to Artificial Intelligence (1986)". It has now become an established optimization approach that is rapidly spreading to many new fields. TS helps to solve problems in a wide area of fields such as resource planning, telecommunications, VLSI design, financial analysis, scheduling, space planning, energy distribution, molecular engineering, logistics, pattern classification, flexible manufacturing, waste management, mineral exploration, biomedical analysis, environmental conversation and scores of other problems [13].

The TS method was partly motivated by the observation that human behaviour appears to operate with a random element that leads to inconsistent behaviour given similar circumstances. The TS method operates in this way with the exception that new subjects are not chosen randomly. Instead the TS proceeds according to the supposition that there is no point in accepting a new (poor) solution unless it is to avoid a path already investigated. This ensures that new regions of problem's solution space will be investigated with the goal of avoiding local minima and which should ultimately lead to the desired solution.

The TS begins by seeking a local minimum. To avoid retracing the steps used, the method records recent moves in one or more Tabu lists. The original intent of the lists was not to prevent a previous move from being repeated, but rather to ensure it was not reversed. The Tabu lists form the nucleus of the Tabu search memory. The role of the memory can change as the algorithm proceeds. At initialisation the goal is to make a broad examination of the solution space, known as 'diversification- expanding a search onto other nearest neighbour solution space', but as candidate locations are identified the search is more focused to produce local optima solutions in a process of 'intensification- digging dip within a neighbour solution space'. In many cases the differences between the various implementations of the Tabu method have to do with the size, variability, and adaptability of the Tabu memory to a particular problem domain [13].

2.2 Tabu Search Algorithm

[14] provides a good algorithm for the general tabu search technique as described below:

```
Initialisation
        S: = initial solution in X
        nbiter:= 0          {current iteration}
        besiter := 0        {iteration when the best solution
                             has been found}
        bestsol:= s         {best solution}
        T: = 0
        Initialise the aspiration function A
        While (f(s) > f*) and (nbiter-bestiter<nbmax) do
                nbiter:= nbiter +1
                generate a set V* of solutions s in n(s)
                which are either not tabu or such that
                A(f(s))>=f(s)
                choose a solution s* minimising f over V*
                update the aspiration function A and the
                tabu list T
                if f(s*) <f(bestsol) then
                        bestsol:=s*
                        bestiter:=nbiter
                        s:=s*
```

For each solution (s) this method requires the definition of a neighbourhood V(s), consisting of solution reachable in one step from (s). The basic step is to move from the current solution (s) to the best solution s*V(s). A tabu list (T) is used to avoid cycling as it stores descriptions of the last move or solution, while finding (s*) in V(s). T is scanned to avoid so called tabu moves that could bring the search back to a previous iteration. The procedure stops after a maximum number of iterations, until the best solution is found or until the solution space is exhausted.

In an optimisation problem the search space (S) is minimised by the objective function (f). A function N which depends on the structure of a specific problem is assigned to each feasible solution (s) which belongs to S in its neighbourhood [N (s) subset of S] each solution s' Є N(s) is called a neighbour of S.

TS is based on selected concepts that unite the fields of artificial intelligence and optimisation. In addition to Simulated Annealing (SA) and Genetic Algorithms (GA), TS was evaluated in the widely reference report by the Committee on the next Decade of Operations Research (Condor, 1998) to be "extremely promising" for the future treatment of practical applications [13].

3. SYSTEM MODELING

This chapter explains the method used in the design of the University Course Timetable (UCT) System, it describes the architecture of the system which takes the form of an expert system, the mathematical model answers the question of how many allocatable resources can be obtained given a set of known scheduling parameters.

3.1 System Architecture

The system architecture gives a conceptual view of the system with its components and their relationships.

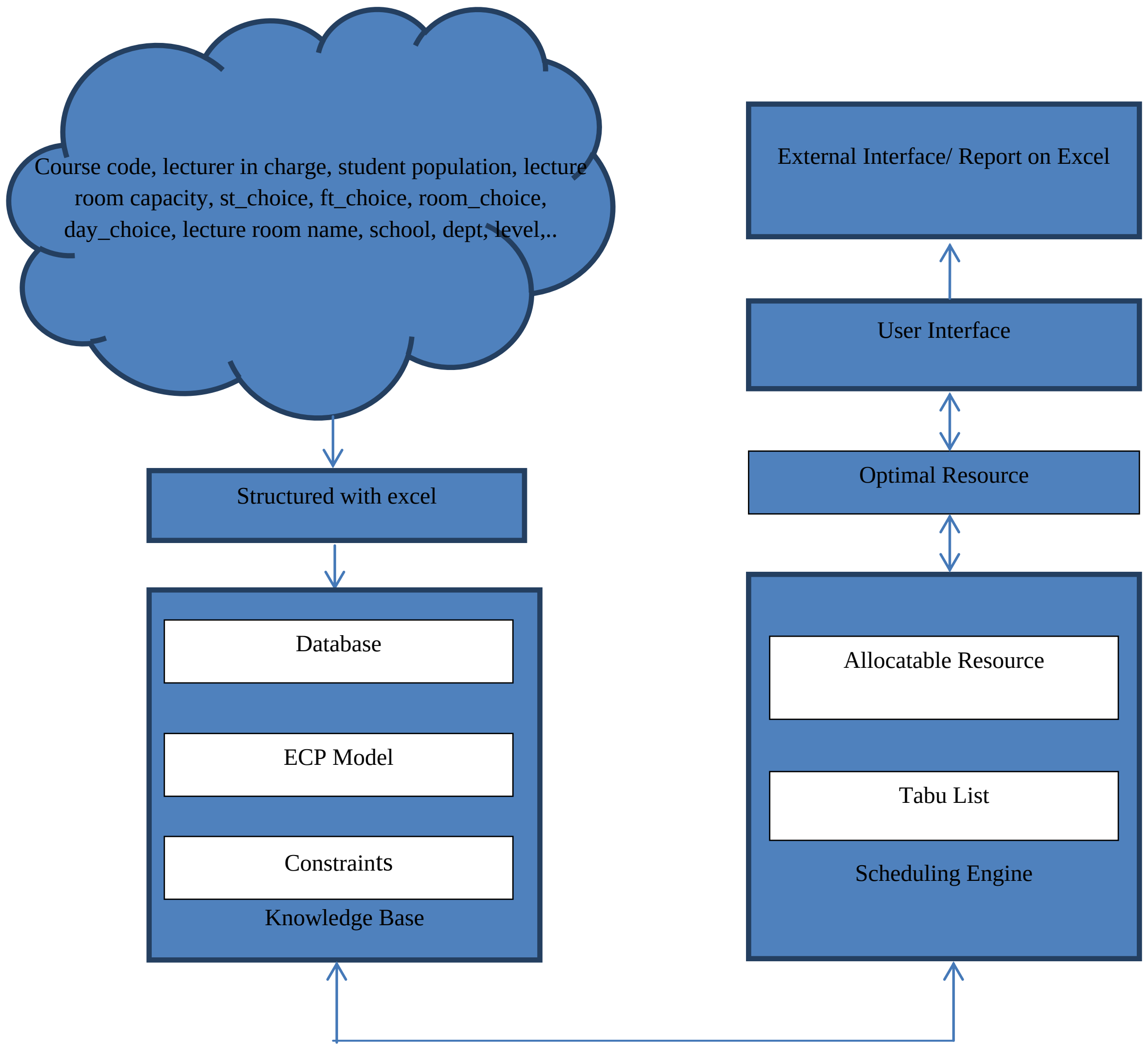

Course code, lecturer in charge, student population, lecture room capacity, st_choice, ft_choice, room_choice, day_choice, lecture room name, school, dept, level,..
Structured with excel
Database
ECP Model
Constraints
Knowledge Base
External Interface/ Report on Excel
User Interface
Optimal Resource
Allocatable Resource
Tabu List
Scheduling Engine

Figure 1: Architecture of the University Timetable System

3.2 The Mathematical Model

$Rq_{m:}$ manual request structure

$Rq_{a:}$ automatic request structure

Ct: Constraint set

Ra_m: Allocatable Resource Manual

$Ra_{a:}$ Allocatable Resource set for Automatic

Ro_m: Optimal Resource

Rq_m =[id, school, dept, pop, lic, course_code, st_choice, ft_choice, room_choice, day_choice]

Rq_a=[id,school,dept,pop,lic,course_code,st_choice,ft_choice]

Ra_m=[st_allocatable,ft_allocatable,room_allocatable, day_allocatable]

Ra_a=[st_allocatable1,ft_allocatable1,room_allocatable1,day_allocatable1,st_allocatable2, ft_allocatable2,room_allocatable2,day_allocatable2, …,st_allocatable_n,ft_allocatable_n, room_allocatable_n,day_allocatable_n]

Ct: [constraint1, constraint2, constraint3, constraint4]

$$Rq_m\ Ro_m = \begin{Bmatrix} 1, if\ \mathrm{RamCt} \\ 0, otherwise \end{Bmatrix} \quad (1)$$

$$Rq_a\ Ro_m = \begin{Bmatrix} 1, if\ \mathrm{Raa[i]Ct} \\ 0, otherwise \end{Bmatrix} \quad (2)$$

Deriving the total number of allocatable resources given scheduling parameters such as: room, day, start time and finish time

$$Resource\ R = (room, day, st, ft) \quad (3)$$

where st: start time and $ft: finish\ time$

Where number of room is denoted as N

Day = {mon, tue, wed, thur, fri, sat} = 6

Time_slot = {7-8,8-10,10-12,12-1,2-4,4-6} or{8-10,10-12,12-2,2-4,4-6} or{hourly slots}. Dynamic hence cannot be determined. Denote number of slot as T

Ra =Total number of unique resource allocatable to a number of courses

Given that i.e. [ETF, Mon, 8, 10] --- (*) is a resource, we can derive other number of unique resource that makes (*) allocatable to a request

Applying permutation on R:P(R) to derive all possible arrangements:

$diff\ room: N-1$

$diff\ day: 6-1=5$

$diff\ time: T-1$

$diff\ room\ and\ day: 5(N-1)$

$diff\ room\ and\ time: (T-1)(N-1) = TN-T-N+1$

$diff\ day\ and\ time: 5*(T-1) = 5T-5$

Summing up the above we derive the total unique resource allocatable to courses as:

$$Ra=(N-1)+5+T-1+5(N-1)+TN-T-N+1+5T-5 \qquad (4)$$

Rearranging we have:

$$Ra=5T+N(5+T)-6 \qquad (5)$$

$$Ra \geq 5T+N(5+T)-6$$

$$T=r1c1, r2c2, r3c3, \ldots, rncn \qquad (6)$$

$$\sum_{r=1,c=1}^{n} RaiciECP = Ro \qquad (7)$$

Hence,

$$ro1 \neq ro2 \neq ro3 \neq rno \begin{cases} 1\ if\ Ra \geq 5T+N(5+T)-6 \\ 0, otherwise \end{cases} \qquad (8)$$

3.3 Algorithm

x: set of allocatable resource [(room,cap,day,st,ft)]

y: set of parent_course_info //day or time must be different from the days or time of the common group having this course. ecp ensures comfortability

z: set of other_course_info //can have diff day, diff time, diff room with others not in common group

```
count=0
for i in x:
    for j in y:
        Constraint check:
        if (i[day] != j[day] or (i[st]>=j[ft] or i[ft]<=j[st]) and i[capacity]-j[population]>=0 and i[capacity]-j[population] <= n% of i[capacity])
    Constraint check:
    for k in z:
     If (((i[room] != k[room] and i[day]!=k[day]) or
    (i[room] != k[room] and (i[st]>=j[ft] or i[ft]<=j[st])) or
    (i[day]!=k[day] and (i[st]>=j[ft] or i[ft]<=j[st])) or
    (i[st]>=j[ft] or i[ft]<=j[st]) or
     (i[day]!=k[day]) or
    (i[room] != k[room]))
     Count+=1
        Intensification Test:
    If(count=len(z)-1):
        Update y(lecture_room=i[lecture_room],day=i[day],start_time=i[start_time],finish_time=i[finish_time])
Diversification Test:
If(i=len(x)-1):
Message (“Could not find optimal resource, please re-enter new start-time,finish_time)
```

4. SYSTEM IMPLEMENTATION

Table 1 shows a sample of the data gathered from Federal University of Technology Akure (FUTA), which is used as a case study for the University Course Timetable preparation. The table shows data for 100 and 200 levels from 6 departments within 3 schools/faculties that are considered.

Table 1. Sample of case study data used

School	Department	Level
SOS	CSC	100/200
	BIO	
SEET	AGE	
	MNE	
SAAT	FST	
	FWT	

4.1 Analysis of data collected

1. Total number of schools: 3
2. Total number of department: 6
3. Total level for each department: 2
4. Total number of courses: 108
5. Total number of lecture rooms: 21
6. Total number of labs: 17

4.2 Properties of data gathered

1. One or more department in a particular school offer one or more joint courses / labwork
2. 200L offer courses related to 100L i.e. audited courses in each school
3. Some courses are offered more than once in a week

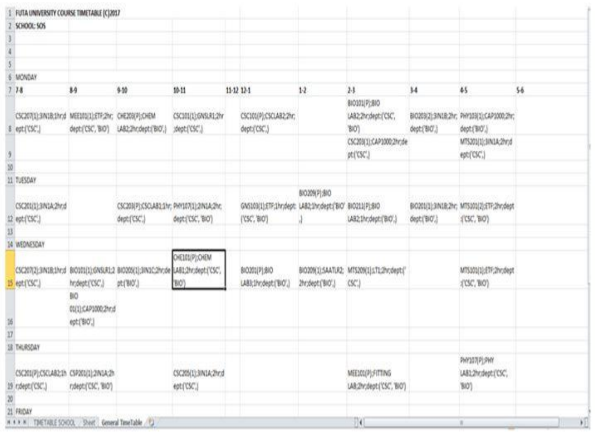
4.3 Technology used

Table 2. Technology used in the development of the UCT model

Input template	Programming Language (Interpreted)	Module for Batch Upload and Analysis	Database (Light weight)	Output format
Excel .xlsx	Python Qt4 (Anaconda 3.4)	openpyXL	Sqlite3	Excel .xlsx

4.4 Output

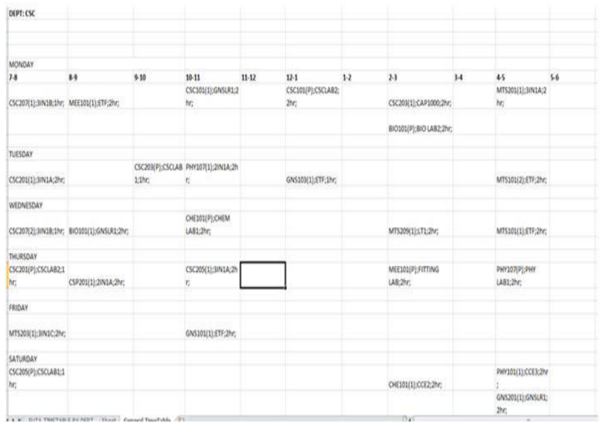
Figures 2 to Figures 6 show output screens of the timetables generated by the UCT model using the case study data.



1	FUTA UNIVERSITY COURSE TIMETABLE (C)2017										
2	SCHOOL: SOS										
6	MONDAY										
7	7-8	8-9	9-10	10-11	11-12	12-1	1-2	2-3	3-4	4-5	5-6
8	CSC207(1);3IN1B;1hr;dept:('CSC',)	MEE101(1);ETF;2hr; dept:('CSC', 'BIO')	CHE203(P);CHEM LAB2;2hr;dept:('BIO',)	CSC101(1);GNSLR1;2hr ;dept:('CSC',)		CSC101(P);CSCLAB2;2hr; dept:('CSC',)		BIO101(P);BIO LAB2;2hr;dept:('CSC', 'BIO')	BIO203(2);3IN1B;2hr; dept:('BIO',)	PHY103(1);CAP1000;2hr; dept:('BIO',)	
9								CSC203(1);CAP1000;2hr;dept:('CSC',)		MTS201(1);3IN1A;2hr;dept:('CSC',)	
11	TUESDAY										
12	CSC201(1);3IN1A;2hr;dept:('CSC',)		CSC203(P);CSCLAB1;1hr; dept:('CSC',)	PHY107(1);2IN1A;2hr; dept:('CSC', 'BIO')		GNS103(1);ETF;1hr;dept: ('CSC', 'BIO')	BIO209(P);BIO LAB2;1hr;dept:('BIO',)	BIO211(P);BIO LAB2;1hr;dept:('BIO',)	BIO201(1);3IN1B;2hr; dept:('BIO',)	MTS101(2);ETF;2hr;dept :('CSC', 'BIO')	
14	WEDNESDAY										
15	CSC207(2);3IN1B;1hr;dept:('CSC',)	BIO101(1);GNSLR1;2hr;dept:('CSC',)	BIO205(1);3IN1C;2hr;dept:('BIO',)	CHE101(P);CHEM LAB1;2hr;dept:('CSC', 'BIO')		BIO201(P);BIO LAB3;1hr;dept:('BIO',)	BIO209(1);SAATLR2; 2hr;dept:('BIO',)	MTS209(1);LT1;2hr;dept:('CSC',)		MTS101(1);ETF;2hr;dept :('CSC', 'BIO')	
16		BIO 01(1);CAP1000;2hr;dept:('BIO',)									
18	THURSDAY										
19	CSC201(P);CSCLAB2;1hr;dept:('CSC',)	CSP201(1);2IN1A;2hr;dept:('CSC', 'BIO')		CSC205(1);3IN1A;2hr;dept:('CSC',)				MEE101(P);FITTING LAB;2hr;dept:('CSC', 'BIO')		PHY107(P);PHY LAB1;2hr;dept:('CSC', 'BIO')	
21	FRIDAY										

TIMETABLE SCHOOL | Sheet | General TimeTable

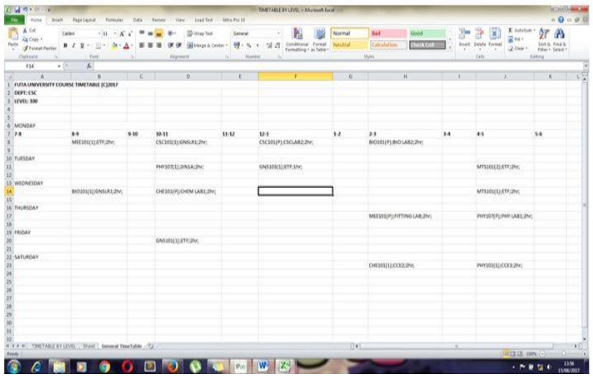
Figure 2: Timetable output by school/faculty



DEPT: CSC										
MONDAY										
7-8	8-9	9-10	10-11	11-12	12-1	1-2	2-3	3-4	4-5	5-6
CSC207(1);3IN1B;1hr;	MEE101(1);ETF;2hr;		CSC101(1);GNSLR1;2hr;		CSC101(P);CSCLAB2;2hr;		CSC203(1);CAP1000;2hr;		MTS201(1);3IN1A;2hr;	
							BIO101(P);BIO LAB2;2hr;			
TUESDAY										
CSC201(1);3IN1A;2hr;		CSC203(P);CSCLAB1;1hr;	PHY107(1);2IN1A;2hr;		GNS103(1);ETF;1hr;				MTS101(2);ETF;2hr;	
WEDNESDAY										
CSC207(2);3IN1B;1hr;	BIO101(1);GNSLR1;2hr;		CHE101(P);CHEM LAB1;2hr;				MTS209(1);LT1;2hr;		MTS101(1);ETF;2hr;	
THURSDAY										
CSC201(P);CSCLAB2;1hr;	CSP201(1);2IN1A;2hr;		CSC205(1);3IN1A;2hr;				MEE101(P);FITTING LAB;2hr;		PHY107(P);PHY LAB1;2hr;	
FRIDAY										
MTS203(1);3IN1C;2hr;			GNS101(1);ETF;2hr;							
SATURDAY										
CSC205(P);CSCLAB1;1hr;							CHE101(1);CCE2;2hr;		PHY101(1);CCE3;2hr;	
									GNS201(1);GNSLR1; 2hr;	

FUTA TIMETABLE BY DEPT | Sheet | General TimeTable

Figure 3: Timetable output by department

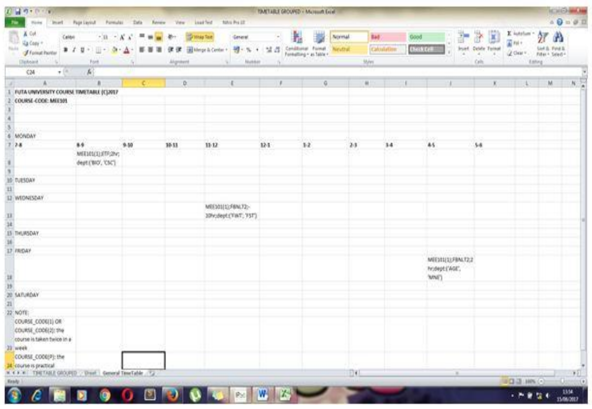


TIMETABLE BY LEVEL - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K
1	FUTA UNIVERSITY COURSE TIMETABLE (C)2017										
2	DEPT: CSC										
3	LEVEL: 100										
6	MONDAY										
7	7-8	8-9	9-10	10-11	11-12	12-1	1-2	2-3	3-4	4-5	5-6
8		MEE101(1);ETF;2hr;		CSC101(1);GNSLR1;2hr;		CSC101(P);CSCLAB2;2hr;		BIO101(P);BIO LAB2;2hr;			
10	TUESDAY										
11				PHY107(1);2IN1A;2hr;		GNS103(1);ETF;1hr;				MTS101(2);ETF;2hr;	
13	WEDNESDAY										
14		BIO101(1);GNSLR1;2hr;		CHE101(P);CHEM LAB1;2hr;						MTS101(1);ETF;2hr;	
16	THURSDAY										
17								MEE101(P);FITTING LAB;2hr;		PHY107(P);PHY LAB1;2hr;	
19	FRIDAY										
20				GNS101(1);ETF;2hr;							
22	SATURDAY										
23								CHE101(1);CCE2;2hr;		PHY101(1);CCE3;2hr;	

TIMETABLE BY LEVEL | Sheet | General TimeTable

Figure 4: Timetable output by level

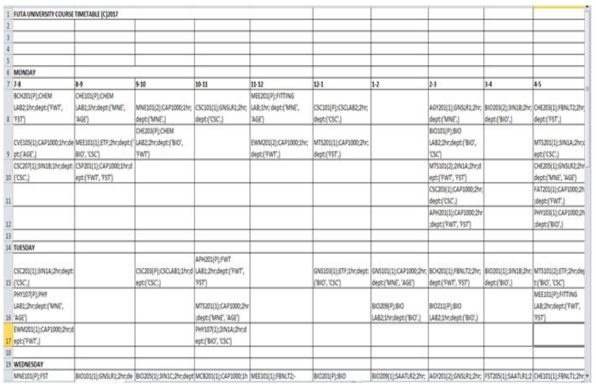


TIMETABLE GROUPED - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K
1	FUTA UNIVERSITY COURSE TIMETABLE (C)2017										
2	COURSE-CODE: MEE101										
6	MONDAY										
7	7-8	8-9	9-10	10-11	11-12	12-1	1-2	2-3	3-4	4-5	5-6
8		MEE101(1);ETF;2hr; dept:('BIO', 'CSC')									
10	TUESDAY										
12	WEDNESDAY										
13					MEE101(1);FBNLT2;-10hr;dept:('FWT', 'FST')						
15	THURSDAY										
17	FRIDAY										
18										MEE101(1);FBNLT2;2hr;dept:('AGE', 'MNE')	
20	SATURDAY										
22	NOTE;										
23	COURSE_CODE(1) OR COURSE_CODE(2): the course is taken twice in a week										
24	COURSE_CODE(P): the course is practical										

TIMETABLE GROUPED | Sheet | General TimeTable

Figure 5: Timetable output for a particular course



1	FUTA UNIVERSITY COURSE TIMETABLE (C)2017									
6	MONDAY									
7	7-8	8-9	9-10	10-11	11-12	12-1	1-2	2-3	3-4	4-5
8	BCH201(P);CHEM LAB2;1hr;dept:('FWT', 'FST')	CHE101(P);CHEM LAB1;1hr;dept:('MNE', 'AGE')	MNE101(2);CAP1000;1hr; dept:('MNE',)	CSC101(1);GNSLR1;2hr; dept:('CSC',)	MEE201(P);FITTING LAB;1hr; dept:('MNE', 'AGE')	CSC101(P);CSCLAB2;2hr; dept:('CSC',)		AGY201(1);GNSLR1;2hr; dept:('MNE',)	BIO203(2);3IN1B;2hr; dept:('BIO',)	CHE203(1);FBNLT2;2hr; dept:('FST',)
9	CVE105(1);CAP1000;1hr;dept:('AGE',)	MEE101(1);ETF;2hr;dept:('BIO', 'CSC')	CHE203(P);CHEM LAB2;2hr;dept:('BIO', 'FWT')		EWM201(2);CAP1000;1hr; dept:('FWT',)	MTS201(1);CAP1000;2hr; dept:('FST',)		BIO101(P);BIO LAB2;2hr;dept:('BIO', 'CSC')		MTS201(1);3IN1A;2hr;dept:('CSC',)
10	CSC207(1);3IN1B;1hr;dept:('CSC',)	CSP201(1);CAP1000;1hr;dept:('FWT', 'FST')						MTS101(2);2IN1A;2hr;dept:('FWT', 'FST')		CHE205(1);GNSLR2;2hr dept:('MNE', 'AGE')
11								CSC203(1);CAP1000;2hr; dept:('CSC',)		FAT201(1);CAP1000;2h ;dept:('FWT',)
12								APH201(1);CAP1000;2hr ;dept:('FWT', 'FST')		PHY103(1);CAP1000;2h ;dept:('BIO',)
14	TUESDAY									
15	CSC201(1);3IN1A;2hr;dept:('CSC',)		CSC203(P);CSCLAB1;1hr;dept:('CSC',)	APH201(P);FWT LAB1;2hr;dept:('FWT', 'FST')		GNS103(1);ETF;1hr;dept:('BIO', 'CSC')	GNS101(1);CAP1000;2hr; dept:('MNE', 'AGE')	BCH201(1);FBNLT2;2hr; dept:('FWT', 'FST')	BIO201(1);3IN1B;2hr; dept:('BIO',)	MTS101(2);ETF;2hr;dept:('BIO', 'CSC')
16	PHY107(P);PHY LAB1;2hr;dept:('MNE', 'AGE')			MTS201(1);CAP1000;2hr ;dept:('MNE', 'AGE')			BIO209(P);BIO LAB2;1hr;dept:('BIO',)	BIO211(P);BIO LAB2;1hr;dept:('BIO',)		MEE101(P);FITTING LAB;2hr;dept:('FWT', 'FST')
17	EWM201(1);CAP1000;2hr;dept:('FWT',)			PHY107(1);2IN1A;2hr;dept:('BIO', 'CSC')						
19	WEDNESDAY									
	MNE101(P);FST	BIO101(1);GNSLR1;2hr;de	BIO205(1);3IN1C;2hr;dept	MCB201(1);CAP1000;1h	MEE101(1);FBNLT2;-	BIO201(P);BIO	BIO209(1);SAATLR2;2hr;	AGY201(2);GNSLR1;2hr;	FST205(1);SAATLR1;2	CHE101(1);FBNLT1;2hr;

Figure 6: Timetable output for all schools/faculties

5. SYSTEM EVALUATION

Based on scalability, the system performed better, since it was able to provide results of timetable that satisfied course scheduling, practical work as well as considered Direct Entry cases. In situation where two or more classes (inter or intra) have lectures in any day or week, the system was able to allocate resource for that.

More so, considering input of data, the system reduced the time of input to the barest minimum by adopting a one-time batch upload of data into the scheduling system, and this made it easier to focus on only scheduling part and reduces stress of keying data manually. Resource optimization was also observed in the case of ECP (Economy Code Protocol) which ensured that allocated resource capacity is not too much compared to the population of students.

The performance values of the system were obtained from the mathematical model of deriving total number of allocatable resources given available scheduling parameters. The number of timeslot used was an average of 5.

Table 3. Performance evaluation of the model

Performance Metrics	**UCT Model**	**Existing System**
Schedule with ECP	70%	0%
Schedule with manual	30%	100%
Total number of rooms Rq	38	-
Total unique resource for rooms Tr	399	-
Total number of rooms	29	-

used Rt		
Total unique resource for rooms used Tu	309	-
Total number of courses scheduled on Tu	108	-
Resource available on used rooms	201	-
Percentage usage of resource	34.95	-
Percentage availability of resource	65.05	-
Number of clash	0	Inconsistent
Output layout	5	1

It is seen that due to ECP, some room capacity against population did not conform to the protocol earlier described in this chapter that is why we weren’t able to achieve a 100% scheduling with ECP. For this reason, we used manual scheduling to bypass ECP check and schedule courses to random class without considering ECP.

While existing system has manual scheduling to be 100%, it seems slow and the resource allocated does not provide comfortability in learning as students are crammed in a low capacity room.

The output type generated by UCT model is of 5 categories which are; timetable for all, timetable by school, timetable by department, timetable by Level, timetable by course. For the existing system, it was just timetable for all. Sorting this output was tiresome and could take weeks to have sorted into other four (4) generated by UCT model.

With the mathematical model deduced for generating the number of unique resource given available parameters, we were able to determine how much resource is available even after scheduling and how much resource is used. The model was able to meet its goal of scheduling courses using the minimum number of resource.

6. CONCLUSION

We were able to achieve our objective of providing optimal resource to several requests involving course scheduling, practical work and Direct Entry (DE) cases using tabu search diversification technique. Results were very explanatory and well represented in excel format. Resources were optimized using the ECP model. The system will aid the timetable committee in performing their task more efficiently. However, due to stochastic start time and finish time, the system was not able to determine during scheduling, what slots are available so that the user can easily choose that slot to schedule for courses. Generation of several data structures that involve different timeslot is therefore suggested so that a search for free timeslots can be made available when needed.

7. REFERENCES

[1] Alizera R. K., Mehrdad N. K. 2015. A mathematical model for University Course Scheduling, A case Study. International Journal of Technical Research and Applications, 20-25.

[2] Anibal T., Per M. G. and Alexandar K. 2013, A Mathematical Model for Determining Timetables that minimizes the Number of Students with Conflicting Schedules. In Proceedings of the European Modeling and Simulation Symposium, 619-624.

[3] Carter, M. W., and Laporte, G. 1998. Recent developments in practical course timetabling. In Burke, E., and Carter, M., eds., Practice and Theory of Automated Timetabling II, 3–19. Springer-Verlag LNCS 1408

[4] Thomas M. and Roman B. 2001. Interactive Timetabling. Accessed at https://www.researchgate.net/publication/220482439_Interactive_Timetabling, on May 27, 2017.

[5] Haroldo G., Luiz S., and Marcone J. 2004. An Efficient Tabu Search Heuristic for the School Timetabling Problem. In Proceedings of the Third International Workshop on Experimental and Efficient Algorithms, WEA 2004, Angra dos Reis, Brazil, May 25-28.

[6]] Keith M., Tornaj M. P. and Hana R. 2010. System Demonstration of Interactive Course Timetabling. In Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling, 573-577. Queen's University Belfast.

[7] Lintang Y., B., and Vega V. 2011. University Timetabling Algorithm Considering Lecturers Workload. In Proceedings of the sixth International Multi-Conference on Computing in the Global Information Technology, Luxembourg, 31-37.

[8] Elizabeth M., Mar´ıa-Cristina R. and Leopoldo A. 2011. A PSO algorithm to solve a Real Course+Exam Timetabling Problem. In Proceedings of the International conference on swarm intelligence. Cergy, France, June 14-15.

[9] Premlata A. S. and Leena R. 2014. Hybrid Genetic Algorithm and Tabu Search Algorithm to Solve Class Timetabling Scheduling Problem. International Journal of Research Studies in Computer Science and Engineering (IJRSCSE) Vol. 1, Issue 4, 19-26.

[10] Hamed B. and Aminu H. 2014. A Review of Distributed Multi-Agent Systems Approach to Solve University Course Timetabling Problem. ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 5, No.11.

[11] Alvarez, R., Crespo, E., & Tamarit, J. M. 2002. Design and Implementation of a Course Scheduling System Using Tabu Search. European Journal of Operational Research, Vol. 137, 512-523.

[12] Sarah A., Fai A., Hawazen A., Sarah A., Maha A. and Manar H. 2018. An Intelligent Bio-Inspired Algorithm for the Faculty Scheduling Problem. International Journal of Advanced Computer Science and Applications, Vol. 9, No. 5.

[13] Glover F. and Laguna M. 1997. Tabu Search. Kluwer Academic Publishers, Boston.

[14] Hertz A. (1991). Tabu search for large scale timetabling problems. European Journal of Operational Research, Vol. 54, Issue 1, 39-47