Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.4, No.7, 2014

Academic Scheduling Problem Made Easy Through Optimization Technique

Bolanle Oladejo[*] Fagbemi Babatunde
Computer Science Department, University of Ibadan, Ibadan, Nigeria
*fb.oladejo@ui.edu.ng

Abstract

The increase in the population of students in the higher institute of learning and crave for literacy has brought about the expansion of academic activities in our Universities. This has caused clashes in the use of limited available resources of each institution. In the past, ways have been sought to consider how the resources could be maximized, but to no avail. Sequel to this, this work aimed at developing an automated timetabling system which solved the problem faced in examinations scheduling. The optimization technique used was Genetic Algorithm (GA) for the scheduling. Automated exam invigilation timetabling system was developed and tested. A Faculty of Science in a reputable institution of learning in Nigeria was selected as a case study. The use of GA was suitable for solving problem(s) such as, under-utilization of available resources, clashes in scheduled courses' examination, faced by timetable scheduler, due to its optimization approach. GA is efficient for maximizing limited available resources. Considering the setbacks associated with the manual system of exam timetable construction, such as clashes of carry-over courses, insecurity of files, poor planning and waste of resources in terms of materials and time to effect correction on timetable: the automated system is more efficient with respect to resource optimization and online accessibility to generated timetable.

Keywords: Academic Activities, Available Resources, Automated Timetabling System, Genetic Algorithm, Optimization

1. Introduction

The University is regarded as a system where universal knowledge is been acquired and serve as a citadel for creating innovation. In this system, several activities occur in its operations - either in administrative or academic tasks. If we were asked to determine the most important part of university life, one thing we probably would not think of is the timetable. This is a mysterious abstract object with no real existence of its own, but exerting absolute authority on the way the university carries out its everyday business. Being part of a university (as a member of staff or as a student) involves an implicit acceptance of the fact that your movements and thoughts will be greatly influenced by the timetable for the duration of an academic session. Thus, timetable becomes a background, indeed a routine around which The University is regarded as a system where universal knowledge is been acquired and serve as a citadel for creating innovation. In this system, several activities occur in its operations - either in administrative or academic tasks. If normal life activities can be carried out.

The application of computer systems in virtually every sphere of human activities -communication, entertainment, medical science, sports, education, decision making, project management, telecasting; has brought about efficiency and effectiveness in such activities. Organizations such as the tertiary institutions and other institutes of learning use timetable to schedule lecture-rooms and invigilation of examinations for future event in a way that makes use of available resources.

Several problems are faced in the allocation of spaces and invigilation while scheduling limited resources for several events. The growing number of students and limited available spaces in a University Faculty pose a lot of problem in allocation of space during the creation of examination timetable. Among these problems are: wastage of space resources during examination period, poor management of resources, inefficient communication system between the invigilators and the timetable committee; duplication of an invigilator's name in different invigilation centers at the same time of the day, insecurity of information due to handling of files used in the manual preparation of timetable. Furthermore, other problems faced in the allocation of the available resources in the manual system also includes; limited classrooms/spaces, a department with small class size of students occupying large classroom thereby wasting available space, and clashes in the time fixed for different course in different departments but handled by same lecturer/invigilator. Thus, this work on Computerization of exam invigilation timetabling system was designed and implemented to solve these problems.

The paper is organized into five sections. Section 1 addresses the introduction, section 2 – the theoretical background; research methodology is detailed in section 3, while section 4 is on implementation of Genetic Algorithm in Scheduling Information system. The paper is concluded in section 5.

2. Theoretical Background

Before the invention of computer system, information were stored on paper. This made it difficult to access information that have been stored long ago and posed difficulty in decision making. According to Whisler et al, (1958), "humans have been storing, retrieving, manipulating and communicating information since the Sumerians in Mesopotamia, where they developed writing in about 3000 BC, but the term *information technology* in its modern sense first appeared in 1958".

"Information technology (IT) is the application of computers and telecommunications equipment to store, retrieves, transmit and manipulate data, often in the context of a business or other enterprise" (Daintith, 2009). IT is versatile and virtually applicable to every field of human endeavour such as education, health, industry, project management,… to mention a few. The different areas of application of IT is often characterized by scheduling of required tasks.

In the context of computer science, scheduling is the method by which threads, processes or data flows are given access to system resources in order to achieve load balancing and effective sharing of system resources. This is related to process scheduling in Operating Systems (OS). Howbeit, educational system is not left out of scheduling problem as related to allocation of limited available resources or infrastructures such as lecture theatres, laboratories, conference centres etc. to several events. By and large, we can relate how the OS schedule system resources to processes thereby showing the processes of how resources are been scheduled in the timetable.

Process Scheduling

In OS process scheduling, several processes are involved in utilizing available system resources. Consequently, there is a demand for the OS to schedule limited available resources among several processes competing for such resources. Figure 1 and Figure 2 show how the OS schedule the processes in different states in the system.

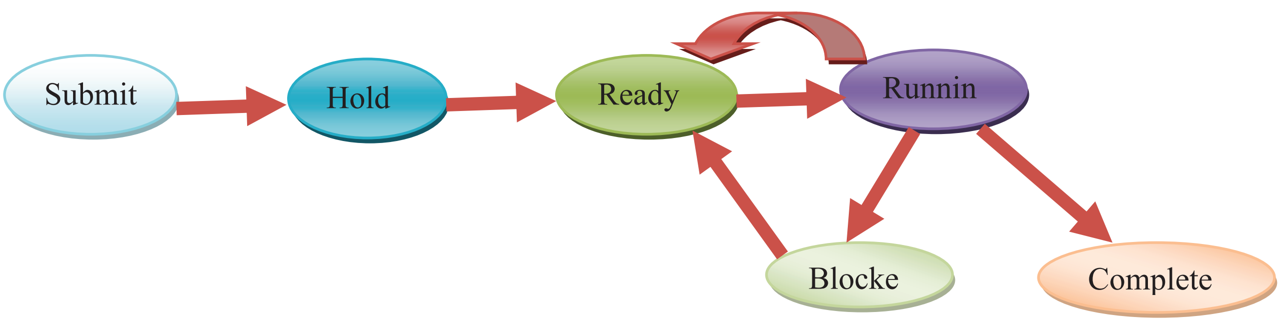

Figure 1. Process State for Scheduling Resources

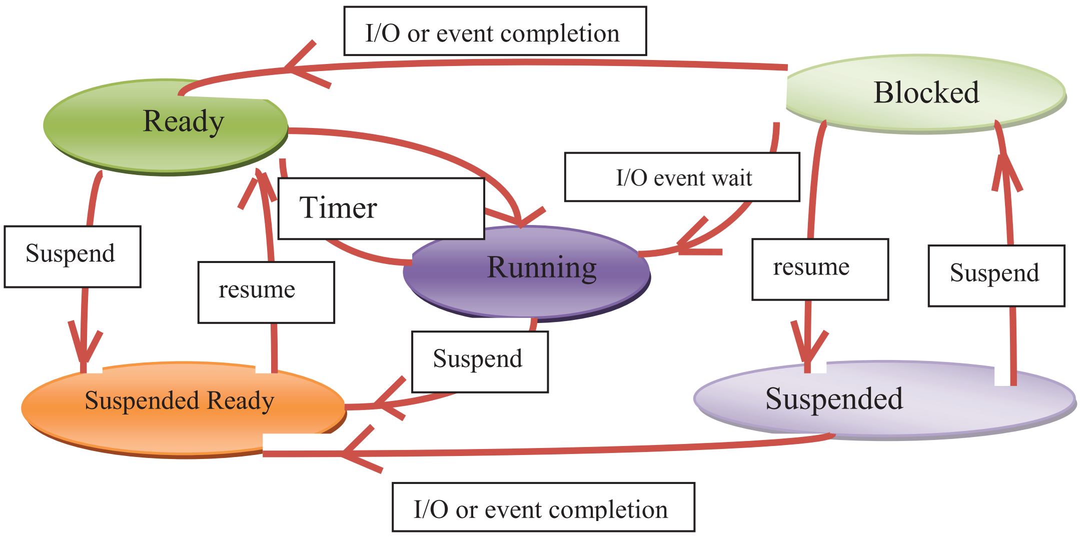

Figure 2. Extended Process State for Scheduling Resources

SUBMIT STATE: Is when the users' job is submitted to the system and the operating system must respond to the request. Likewise in the educational context for planning timetable for exam events, the jobs include the courses, students and staff.

HOLD STATE: When the user's job has been converted into machine readable form, but resources must be assigned to get to the ready state. In the context of the exam timetabling system, resources such as courses, student and staff have not been scheduled by the system.

READY STATE: This is when the jobs are on the queue waiting for the processor because; there are usually more processes than the processor. Therefore, the job must wait for it turns.

RUNNING STATE: When the job already has the attention of the processor as a resource and is been executed. This is when the GA schedules the resources, to generate the timetable.

BLOCKED STATE: When the process is waiting for some event to happen before it can be processed. This could be I/O completion event or could be waiting for a device that is been used by another process. In the context of the exam timetabling, resource such as venue, time or the course must have been scheduled.

SUSPENDED STATE: When the system is performing poorly or responding to fluctuation node.

COMPLETED STATE: When a process runs to completion and releases it resource. This is when the timetable have been generated for it to be uploaded.

In this theoretical background, optimization techniques such as the graph coloring and genetic algorithm are discussed as relating to this work.

2.1 Optimization Techniques

Educational timetabling is a major administrative activity for a wide variety of institutions.

A timetabling problem can be defined to be the problem of assigning a number of events into a limited number of time periods.

"Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives" Wren (1996).

This work concentrates on university timetabling problems. Such problems can be divided into two main categories: course timetabling and exam timetabling. These problems are subject to many constraints that are usually divided into two categories: "hard" and "soft"

(Burke et al, 1997).

The last ten years have seen a significant amount of research; for a survey. More recently, researchers have used different optimization techniques such as: simulated annealing, tabu search, genetic algorithm, memetic, integer programming approach in optimizing limited resources, improving efficiency and decision making, some of which are discussed below.

2.1.1 Graph Coloring Approach

Generally most timetable software is based on graph delegation. In a particular problem, dots on the graph will represent events and lines will correlate with those events. Two adjacent dots (contains conflict) if linked with the same line (common edge).

The graph can be separated by coloring the dots, where no adjacent dots (contains conflict) have the same color. This separation is equivalent with the total time occupied to schedule one set of subject, which either one conflicted or inherits the same time schedule.

The difficulty of creating the timetable can be represented using graph. The subject will be represented by nodes or dots with constraints with other subjects are represented by the line. Each separation inside the graph will consist of subjects that can be scheduled at a similar timeslot.

Figure 3 represents a simple model in developing timetable, utilizing the graph coloring approach. Each subject is drawn as a single point from the departments and each pairing subject cannot be scheduled at the same time. It will be represented by a line splitting between the two points. Furthermore, each point (subject) is colored particularly to show no pairing points connected by a single line that shares the same color.

All colors represents time differences, therefore no conflicting subject will be scheduled in pair.

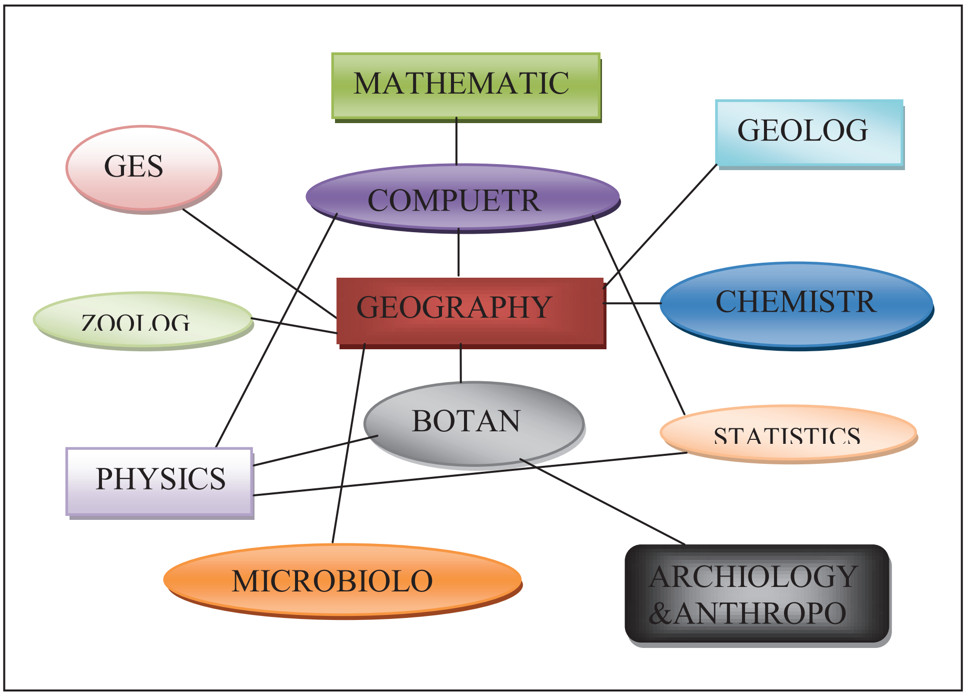

Figure 3. Graph Colorization for School Time Schedule

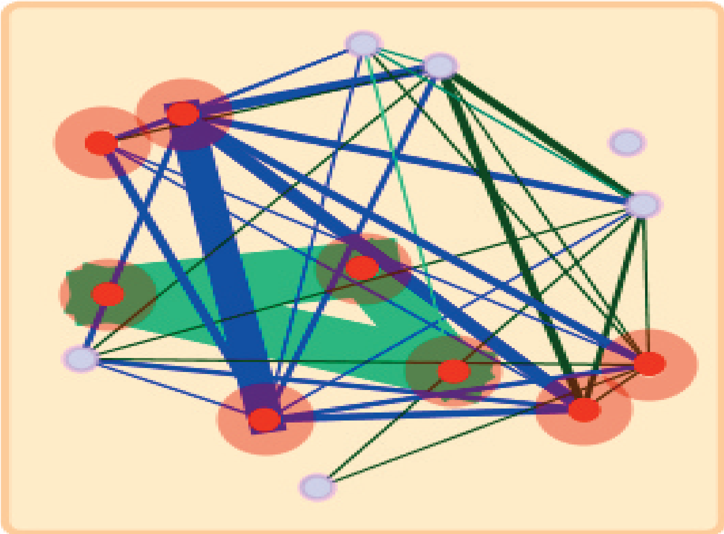

Figure 4. Three perspectives of a single department in the Complexity Graph (Source: Burke et al, 2004)

Figure 4 models the application of graph colouring technique for a number of departments where each node depicts each department in the faculty showing the exam nodes; the same department highlighting exams which clash with a number of other exams and the same department showing the clashes between exams. The optional line thickness is turned on to indicate the number of students involved in the clash which can help to distinguish clashes involving the same students.

2.1.2 Genetic Algorithm (GA)

Genetic Algorithms (GA) are powerful general purpose optimization tools which model the principles of evolution (Davis L. 91). They are often capable of finding globally optimal solutions even in the most complex of search spaces. They operate on a population of coded solutions which are selected according to their quality then used as the basis for a new generation of solutions found by combining (crossover) or altering (mutating) current individuals.

Algorithms are based on the principles of natural genetics and survival of the fittest. GA searches for solutions by emulating biological selection and reproduction. In a GA, the parameters of the model to be optimized are encoded into a finite length string, usually a string of bits. Each parameter is represented by a portion of the string. The string is called a chromosome or individual, and each input variable (feature) is called a gene. GA is a robust search and optimization technique using probabilistic rules to evolve a population from one generation to the next. In some cases, GA have come up with solutions that baffle the programmers who wrote the algorithms in the first place. Before a genetic algorithm can be put to work on any problem, a method is needed to encode potential solutions to that problem in a form that a computer can process. One common approach is to encode solutions as binary strings: sequences of 1's and 0's, where the digit at each position represents the value of some aspect of the solution. Another, similar approach is to encode solutions as arrays of integers or decimal numbers, with each position again representing some particular aspect of the solution. This approach allows for greater precision and complexity than the comparatively restricted method of using binary numbers only and often "is intuitively closer to the problem space" (Fleming and Purshouse 2002).

Burke and Newall 1999 used genetic algorithms to schedule exams among university students. The timetable problem in general is known to be NP-complete, meaning that no method is known to find a guaranteed-optimal solution in a reasonable amount of time. In such a problem, there are both hard constraints - two exams may not be assigned to the same room at the same time - and soft constraints - students should not be assigned to multiple exams in succession, if possible, to minimize fatigue. Hard constraints must be satisfied, while soft constraints should be satisfied as far as possible.

There are three genetic operators, viz–a-viz:

- **Reproduction**: It is an operator which allocates in the population P(t+1), an increasing number of copies of the individuals with the fitness function above the average in the population P(t).

- **Crossover**: A genetic operator with a probability Pc, independent of the individuals on which its applied; it takes two randomly selected individual (parents) as input and combines them to generate two siblings.
- **Mutation**: An operator that causes the change of allelic value of a randomly selected gene; for instance, if the alphabet were {0, 1}, an allelic value of 0 will be modified to 1 and 1 to 0.

3. Modelling of Exam Invigilation System

This section discusses Hierarchy Input, Process and Output (HIPO), activity diagram, and system architecture of Exam Invigilation System. It also looks into the logic design using flowchart for genetic algorithm

3.1. Hierarchy Input, Process and Output (HIPO) Chart for Exam Invigilation Timetabling System

Hierarchy plus Input-Process-Output (HIPO) is a graphic technique that can be used to describe a system. HIPO is a tool for planning and/or documenting a computer program. Its model consists of a hierarchy chart that graphically represents the program's control structure and a set of IPO (Input-Process-Output) charts that describe the inputs to, the outputs from, and the functions (or processes) performed by each module on the hierarchy chart. Figure 5 models the HIPO chart for automation of examination invigilation scheduling information system.

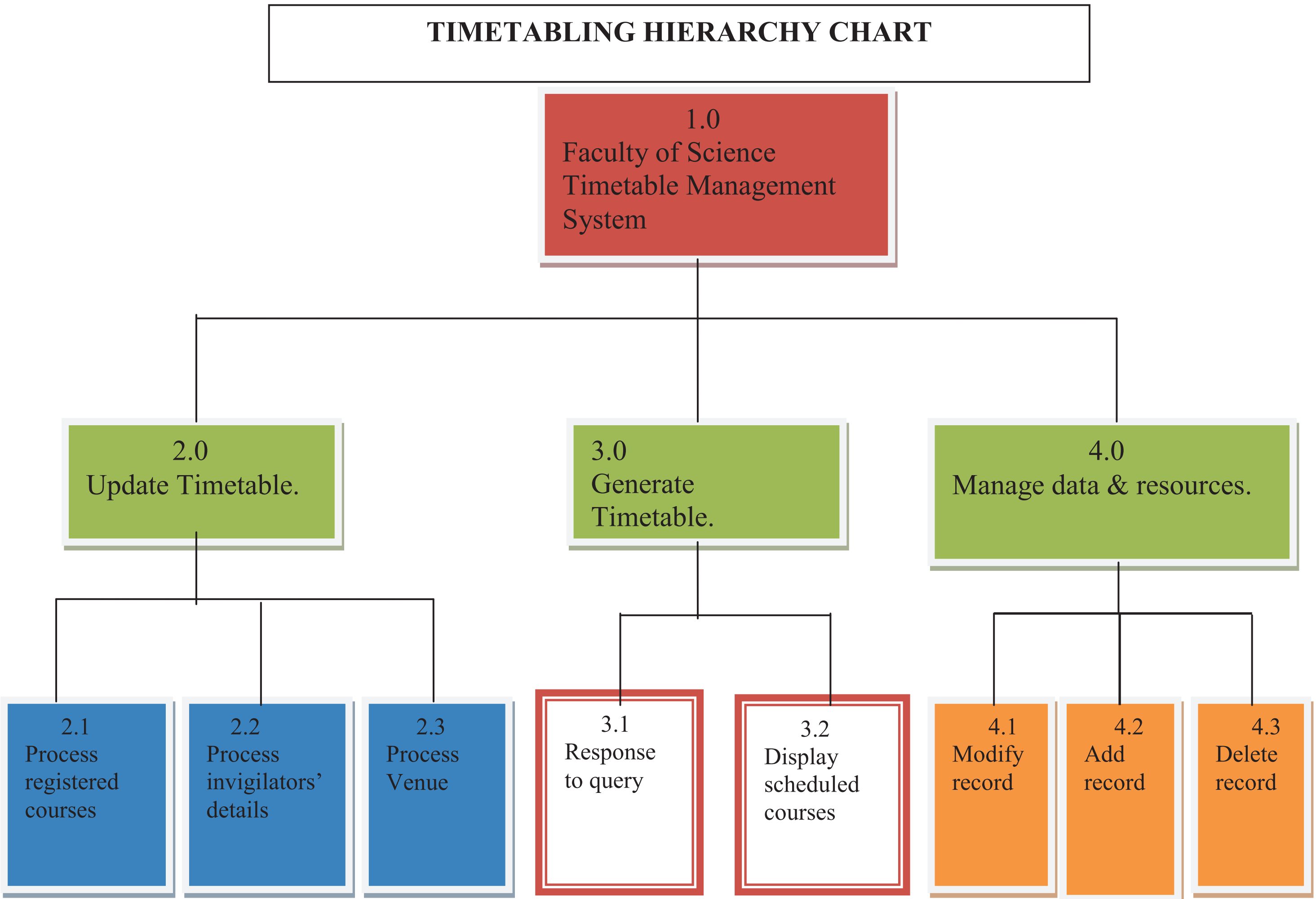

Figure 5. HIPO Diagram for Examination Scheduling System

3.2 Activity Diagram for Exam Invigilation Timetabling System

An activity diagram describes how activities are coordinated to provide a service. The service can be at different levels of abstraction. Activity diagram in this work depicts how users interact with the system and the activities.

It describes how the events for realizing the system functions relate to one another.

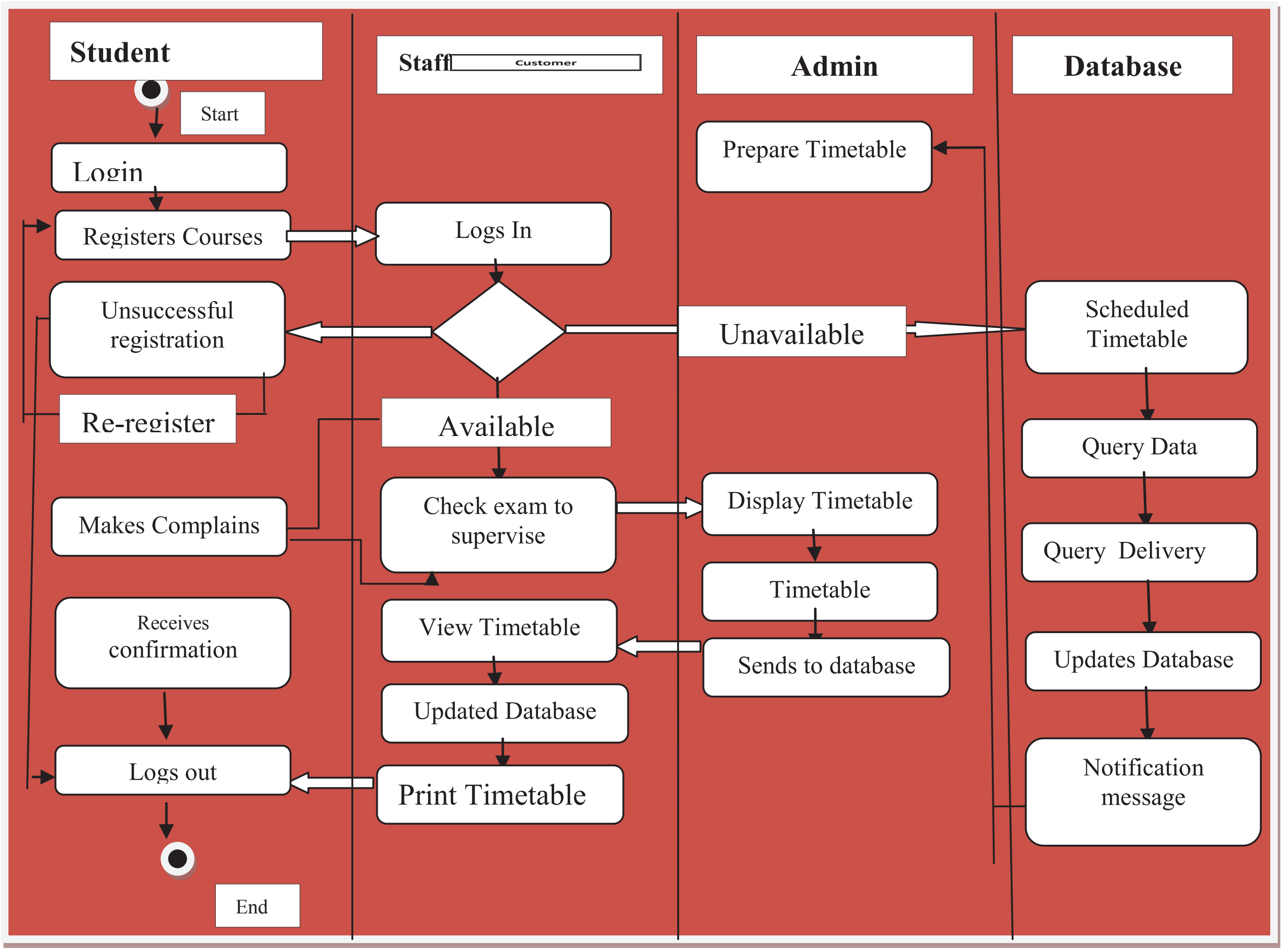

Figure 6. Activity Diagram for the Exam Invigilation Timetable System

Figure 6 is the Activity diagram showing the movement of data and requests from Staff and Students. It depicts how their requests/query are being processed and transformed to view the scheduled timetable. Resources database is being updated for manager's usage and future report generation. The start point of the activity diagram is the point where students login and register for session courses. The registration data is stored into the database, and the system confirms the registration. The students then check for the venue for their exams, if the timetable is available, the system confirms it by displaying the timetable and the students can print it.

However, it is a different scenario if the timetable is unavailable. The student and/or staff then informs the member of the exam committees who then directs the complaints to the Administrator. The administrator then queries the database and uploads it for students and staff to access it.

The UML model in figures 6 and 7 described the functional and non-functional requirements for the exam invigilation timetabling system. The system structure is further modeled in order to specify the interaction of the actors with the system.

3.3. System Architecture for Exam Timetabling System

The system has been designed with a completely web-based interface using the Enterprise Edition of Java (J2EE). MySQL has been used to enable relational database. Since timetabling is done at the departmental level (each department builds its own timetable), the GA needs to be able to work with multiple problem instances at the same time. Each problem is modeled as a constraint satisfaction and optimization problem (CSOP) and solved using the iterative forward search algorithm (Muller 2005).

The system architecture shows the overall structure of the system.

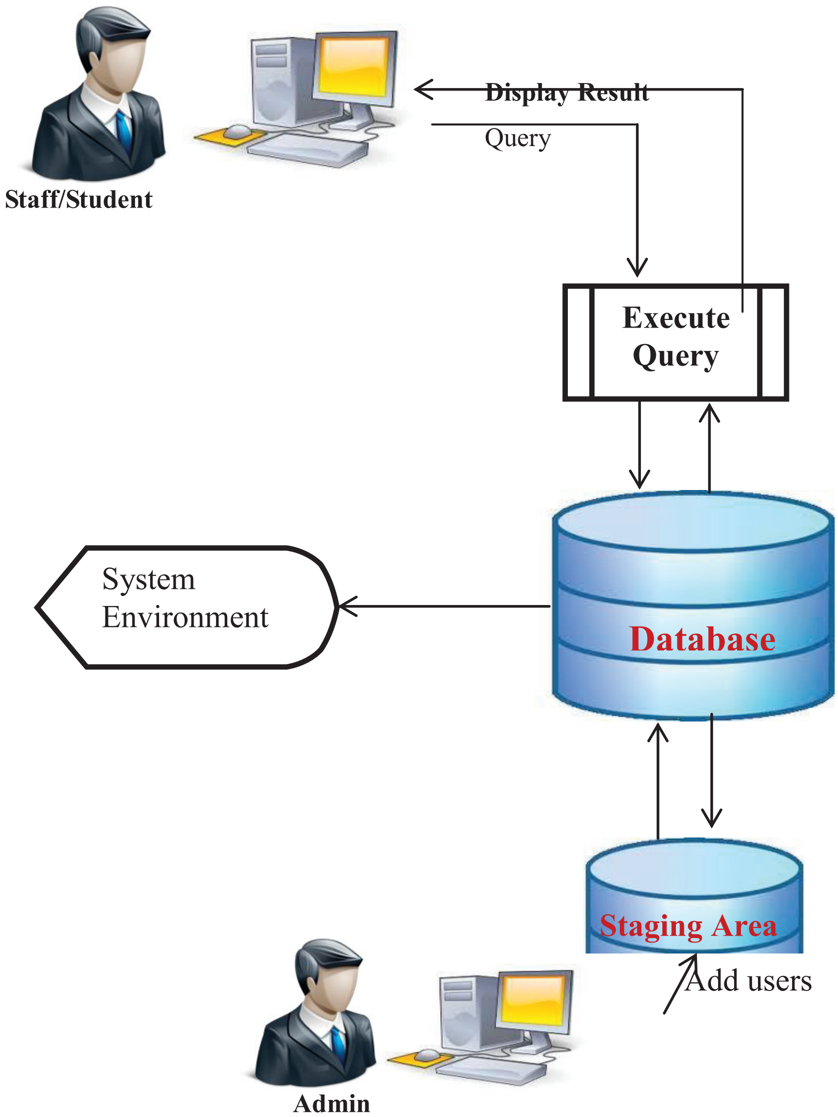

Figure 7. System Architecture of the exam invigilation System

In Figure 7, the user queries or adds data to the database. The genetic algorithm is used to categorize the data based on either the user is a student or staff. After which the result is displayed. Also, the administrator is in charge of updating the database from his end based on uploading of new timetable, adding users and assigning of resources.

3.4 Logic Design and Pseudocode of GA for Exam Invigilation Timetable System

The algorithm below shows how a simple genetic algorithm operates for the exam timetable.

t = 0

Initialize P (t)

Evaluate P (t)

while Termination condition is not satisfied do

Selection P(t + 1) from P(t) Perform Crossover on P(t + 1) Perform Mutation on P(t + 1) Evaluate P(t

+ 1)

Replace P(t) with P(t + 1)

t = t + 1

end while

(Source: David E.G., 1989).

Analyzing the algorithm above, P represents the population path to each resource (Staff, Students, Venue), t represents the resources (Staff, Students, Venue). After each resource is scheduled, the population path to that resource is destroyed to prevent rescheduling that resource, by using the operators: Selection, performing Crossover, Mutation and Evaluation.

The GA pseudo code is represented in a flowchart form in Figure 8, in order to simplify its understanding of operations.

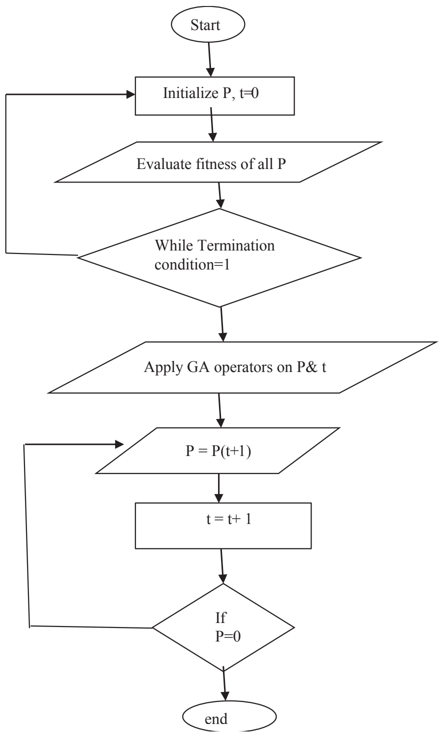

Figure 8. Flowchart for a GA Operation

Figure 8 gives a representation of the flow of how the GA operations are performed on the available resources and the population path toward the generation of the exam timetabling system.

4. Results and Discussion

We implemented the exam scheduling system as specified by the system models and the GA logic design. Figures 9 - 12 show the interfaces of the forms through which information is been sent to the resources database. The system ensures that there is no duplication of records in the databse when the administrator is scheduling the available resources and in the generation of the timetable.

4.1. User Authentication

The user authentication portal is the interface where the system only allows valid users' to the system to gain access to it.

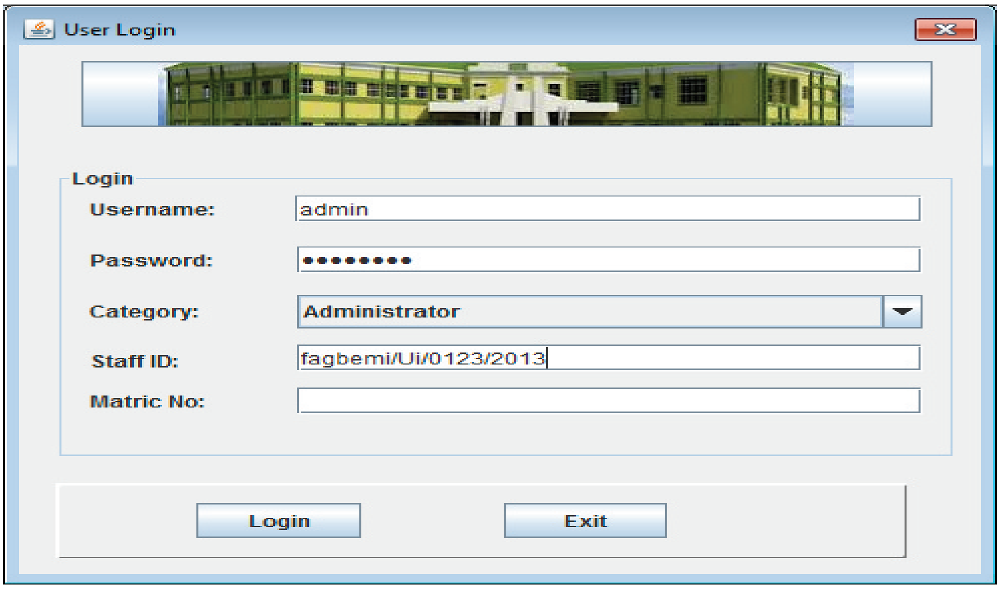

Figure 9. Login portal

Figure 9 secures the information in the system from unauthorized users' through the use of a username and password. The username and password would be created by the administrator for the users' in the faculty to access the exam invigilation system, which can then be changed later. The system authenticates valid users, after the correct username and password had been supplied. Otherwise, it prevents such user from gaining access to the system.

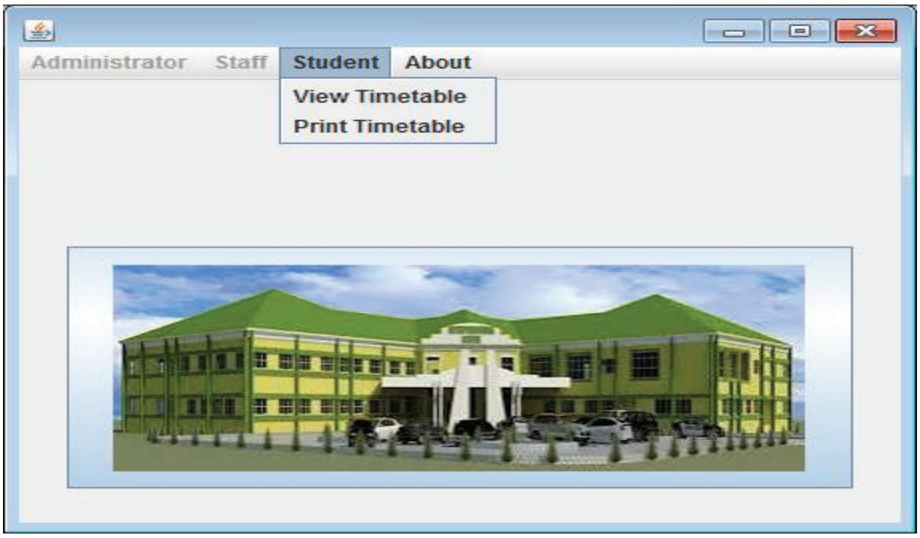

Figure 10. Navigation Panel

As depicted in Figure 10, valid user of the system performs basic operation(s) with the system. When a user selects 'Student' in the category section, the system disables other menu function, such as the Administrator menu and Staff menu, so as to prevent the Student from accessing other users' information and functions. The same constraint applies to other users, say, admin, or staff.

4.2 GA-driven Scheduled resources for Exam Invigilation Timetable

Figure 11 is the generated timetable, after the resources have been scheduled.

2013/2014 Students' time table

course code	course title	semester	venue name	date	time
csc233	Discrete Structure	2	Comp KDL	2014/03/30	8:00-11:00
csc221	Structured Programming	2	Comp LL	2014/03/30	12:00-3:00
csc232	Structured Programming	1	Comp KDL	2014/03/28	8:00-11:00
csc234	Industrial Training	2	Comp KDL	2014/03/27	8:00-11:00
csc302	Info. Management Systems	2	MTH LLT	2014//03/28	12:00-3:00
MAT213	Abstract ALGEBRA	1	SLLT	2014/03/31	3:00-6:00
csc101	Intro. to Computer Science	1	CCLT	2014/03/31	8:00-11:00
MAT222	Calculus	2	Comp. LT	2014/04/02	8:00-11:00
STA311	Distributive Statistics	1	MTH LLT	2014/04/02	12:00-3:00
STA111	Probability I	1	SSLT	2014/04/04	12:00-3:00
STA231	Statistical Inference II	2	SLLT	2014/04/04	8:00-11:00
MAT223	ANALYSIS II	2	MTH SLT	2014/04/03	12:00-3:00
MAT141	Linear ALGEBRA	1	MTH LLT	2014/04/03	8:00-11:00

Figure 11. Output for the Examination Timetable

Students view the interface on generated timetable after navigating to the Students menu bar. The timetable contains the course code, course title, semester in which it will be taken, venue for the examination, date and time.

2013/2014 Lecturer's time table

course code	course title	lecturer in charge	supporting lecturer 1	supporting lecturer 2	venue name	date	time
csc233	Discrete Structure	frew	srew	zew	Comp KDL	2014/03/30	8:00-11:00
csc221	Structured Program...	Adeyemo	Woods	Adeniji	Comp LL	2014/03/30	12:00-3:00
csc232	Structured Program...	Olajimi	Komolafe	Adeniji	Comp KDL	2014/03/28	8:00-11:00
csc234	Industrial Training	Macolo	Ojo	Adeleke	Comp KDL	2014/03/27	8:00-11:00
csc302	Info. Management ...	DR. osunade	Ademola	Olajire	MTH LLT	2014//03/28	12:00-3:00
MAT213	Abstract ALGEBRA	Zubai	Igwe	Omorege	SLLT	2014/03/31	3:00-6:00
csc101	Intro. to Computer ...	Oladejo	Komolafe	Jide	CCLT	2014/03/31	8:00-11:00
MAT222	Calculus	Igwe	Agbor	Femi	Comp. LT	2014/04/02	8:00-11:00
STA311	Distributive Statistics	Prof. Iyaniwura	Oluwaseun	Obisesan	MTH LLT	2014/04/02	12:00-3:00
STA111	Probability I	Angela	Alawodo	Olubusoye	SSLT	2014/04/04	12:00-3:00
STA231	Statistical Inference II	Yaya	Angela	Olufemi	SLLT	2014/04/04	8:00-11:00
MAT223	ANALYSIS II	Ayoola	Akinola	Ugbebor	MTH SLT	2014/04/03	12:00-3:00
MAT141	Linear ALGEBRA	Onasanya	Emeka	Omolona	MTH LLT	2014/04/03	8:00-11:00

Figure 12. Scheduling Examination Invigilation for staff

Figure 12 is the generated timetable for invigilators which they access after navigating to the Staff menu bar through the authentication portal. The timetable contains the course code, course title, lecturer in charge, supporting lecturers' to assist in supervising the examination, venue for the examination, date and time.

5. Conclusion

Timetable generation using Genetic Algorithm produces an optimized timetable. However, as the number of resources increases, the search space also increases, this has a negative impact on the speed of program execution. This work facilitates easy scheduling of allocation of space to events as well as easy and efficient processing and preparation of the invigilation timetable which consequently prevents collision or clashes of courses. Without gainsay, the work helps to optimize allocation of space or limited resources to academic activities such as examination event.

REFERENCES

[1] Burke E.K, Eckersley B.J. McCollum, Petrovic S, and Qiu R., (2004), "*Analysing Similarity in Examination Timetabling". 5th International Conference on the Practice and Theory of Automated Timetabling*. Pittsburgh, PA USA.

[2] Burke E.K., Jackson, K., Kingston, J. H., and Weare R. F., (1997),"Automated university timetabling: The state of the art";The Computer Journal, 40(9):565–571.

[3] Burke, E. Elliman, D.Weare, R., (2000), Automated scheduling of University Exams, Department of Computer Science, University of Nottingham.

[4] Burke, E. K. and Newall, J. P., (1999), "A multi-stage evolutionary algorithm for the timetable problem." *IEEE Transactions on Evolutionary Computation*, 3(1):63–74.

[5] Burke, E. K., Elliman, D. G., and Weare, R. F., (1994: 27(1):1–18), "A university timetabling system based on graph colouring and constraint manipulation." *Journal of Research on Computing in Education*.

[6] Burke, E. K., Newall, J. P., and Weare, R. F., (1998b), "A simple heuristically guided search for the timetable problem" In *International ICSC Symposium on Engineering of Intelligent Systems (EIS '98)*, pages 574–579: ICSC Academic Press, Canada/Switzerland.

[7] Daintith, John, ed., (2009), "Information Technology", *A Dictionary of Physics*, Oxford University Press, retrieved 1 August 2012.

[8] Muller, T.; Rudov´a, H.; and Bart´ak, R., (2005: 126–146). "Minimal perturbation problem in course timetabling". In Burke, E., and Trick, M., eds., Practice and Theory of Automated Timetabling V. Springer-Verlag LNCS 3616.