LAUTECH Journal of Computing and Informatics (LAUJCI) – ISSN: 2714-4194
Volume 1– No.1, Sept. 2020 – www.laujci.lautech.edu.ng

Automated Lectures-Based Timetabling Generation Using Evolutionary Algorithm

Ismaila W. Oladimeji
Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, Ogbomoso, Nigeria,
woismaila@lautech.edu.ng

Abstract
A timetable management system is designed and created to handle as much course data as fed while ensuring the avoidance of redundancy. Every school year, institutions of education face the rigorous task of drawing up timetables that satisfies the various courses offered by the different department. The difficulty is due to the great complexity of the construction of timetables for lectures, due to the scheduling size of the lectures, the high number of constraints and criteria of allocation, usually circumvented with the use of little strict heuristics, based on solutions from previous years. Also, the former timetabling systems did not consider the requests of lecturers as par the time convenient to fix their classes. This work employed Genetic Algorithm to generate timetable for faculty of agriculture courses. The hard, soft and float constraints for the system were formulated. The float constraint was included in hard constraints (system A) and then in soft constraints (system B). The repair strategy is also used for initializing a random population. The system was run with different parameters settings to obtain optimum results.
The results of the systems are: SA produced total of 4 soft constraints; SB produced a total of 9 soft constraints while SC produced 15 soft constraints. Thus, the accuracies of SA, SB and SC timetabling systems are 95.8%, 89.45 and 87.2%. This established that the lecturers request should be part of strong constraints for a conflict free timetabling.

Keyword: Genetic Algorithm, Soft Constraints, float constraint, Hard Constraints, Repair Strategy

1. INTRODUCTION

University course timetabling problems are combinatorial problems, which consist of scheduling a set of courses within a given number of rooms and time periods. Every school year, institutions of education face the rigorous task of drawing up timetables that satisfies the various courses and their respective examinations being offered by the different department. The difficulty is due to the great complexity of the construction of timetables for lectures and examinations, due to the scheduling size of the lectures and examinations periods and the high number of constraints and criteria of allocation, usually circumvented with the use of little strict heuristics, based on solutions from previous years. The lecture timetabling problem is concerned with assigning a set of lecture to a limited number of timeslots, subject to a set of constraints. A critical factor in running a university or essentially an academic environment is the need for a well-planned and clash-free timetable. Back in the times when technology was not in wide use, academic timetables were manually created by the educational center staff (Moreira, 2008).

Nowadays, this process has been simplified by semi-automatic solutions based on timetable generation applications (e.g. Open Course Timetabling). A timetable management system is designed and created to handle as much course data as fed while ensuring the avoidance of redundancy. An educational timetable must meet a number of requirements and should satisfy the desires of all entities involved simultaneously as well as possible. Solutions to timetabling problems have been proposed since the 1980s (David, 1999). Research in this area is still active as there are several recent related papers in operational research and artificial intelligence journals. This indicates that there are many problems in timetabling that need to be solved in view of the availability of more powerful computing facilities and advancement of information technology. It is important to create a balanced lecture timetable which takes into consideration lecture times, lecture venues and courses offered in order to avoid conflicting instances of lecture schedules. Thus, all the timetabling systems developed did not factor the lecturers' requests for convenient lecture time in order to curb incessant lecture postponement and re-scheduling of lectures. In achieving this, there is need to test the lecturer's request (which was formulated as ***float constraint***) under hard or soft constraints.

The problem was first studied by (Gotlieb, 1962), who formulated a class-teacher timetabling problem by considering that each lecture contained one group of students, one teacher, and any number of times which could be chosen freely. Since then the problem is being continuously studied using different methods under different conditions. This inadequacy of classical methods has drawn the attention of the researchers towards the heuristic-based non-classical techniques. Worth mentioning non-classical techniques that are being applied to the problem are genetic algorithms (Colorni *et al*, 1992), neural network (Looi, 1992), and tabu search algorithm (Costa, 1994).. This work adopted Genetic Algorithm in scheduling lectures using a Faculty of Agriculture of an institution as a case study. Genetic algorithms (GAs) are numerical optimization algorithms that are as a result of both natural selection and natural genetics. The method which is general in nature is capable of being applied to a wider range of problems unlike most procedural approaches. Genetic algorithms help to solve practical problems on a daily basis. The algorithms are simple to understand and the required computer code easy to write. Concisely stated, a genetic algorithm is a programming technique that mimics biological evolution as a problem-solving strategy. Given a specific problem to solve, the input to the GA is a set of potential solutions to that problem, encoded in some fashion, and a metric called a fitness function that allows each candidate to be quantitatively evaluated (Ismaila, *et al* 2012). .

2. RELATED WORKS

A few worth mentioning Evolutionary Algorithms, used for the school timetabling problem, are those of Abramson & Abela, (1992); Piola, (1994) and Bufe *et al*, (2001). Similarly, EAs, used for the university class timetabling problem, are those of (Carrasco and Pato, 2001), (Srinivasan *et al*, 2002) and Datta *e tal*, (2006), modeled the university class timetabling problem as a multi-objective optimization problem, considering different class-structures, such as single-slot, multi-slot, split, combined, open, and group classes. Fang, 1994) investigates the use of genetic algorithms to solve a group of timetabling problems. He presents a framework for the utilization of genetic algorithms in solving of timetabling problems in the context of learning institutions. This framework has the following important points, which give you considerable flexibility: a declaration of the specific constraints of the problem and use of a function for evaluation of the solutions, advising the use of a genetic algorithm, since it is independent of the problem, for its resolution.

In Saptarini *et al*, authors apply genetic algorithms (GA) to prevent the violation of hard constraints and minimize the violation of soft constraints. The GA in this study distributing population in some groups. The distributed GA generate groups of population and then after each iteration, the migration between groups will be conducted based on given probability of migration. The study shown that the distributed GA succeeds to prevent violation of hard constraints, minimize the soft constraints violation and avoid the premature convergence. While researchers in Chilma *et al*, (2013).implemented a school day agenda that focused on the learning rhythms of students of elementary and secondary schools using a genetic algorithm. The genetic algorithm evolves through a process of mutation and selection and builds a total solution based on the best solutions for each group. Sixteen groups in a school are tested and the results of class schedule assignments are presented. The methodology developed achieves an improvement of 12% than traditional algorithms generate schedules. The algorithm shows high stability in the found solutions, ensuring an efficiency of over 92% in the result obtained in each cycles.

The researchers in Gröbner and Wilke, (2002) presented an approach to generalize all the timetabling problems, describing the basic structure of this problem. Gröbner proposes a generic language that can be used to describe timetabling problems and its constraints. The authors in (Ismaila, *et al*, 2012) employed genetic algorithm to develop a roster table for physicians/medical doctors of a hospital in Nigeria. The rostering algorithm involves non-cyclic scheduling process, establishment of hard and soft constraints and choosing of befitting genetic algorithm parameters for generating the medical doctor rostering table. The resultant schedules were analysed with soft constraints using degree of satisfaction nature. Fernandes, (2002) classified the constraints of class-teacher timetabling problem in constraints strong and weak. Violations to strong constraints (such as schedule a teacher in two classes at the same time) result in an invalid timetable. Violations to weak constraints result in valid timetable, but affect the quality of the solution (for example, the preference of teachers for certain hours). The proposed algorithm, evolutionary, has been tested in a university comprising 109 teachers, 37 rooms, 1131 a time interval of one hour each and 472 classes. The algorithm proposed in resolving the scheduling without violating the strong constraints in 30% of executions. Moreira, (2008) presented a solution method to the problem of automatic construction timetables for the exams. Among several mathematical models of representation, the final option was for a model matrix, which is justified by the benefits that this model presents when used in the algorithm solution. The method of solution is a meta-heuristics that includes a genetic algorithm. The model is directed to the construction of exam timetables in institutions of higher education. The results achieved in real and complex scenarios are satisfactory; the exam timetabling meets the imposed regulations. However, all the courses timetabling systems developed did not factor the convenience of lecturers in order to curb the incessant lectures postponement or cancelation/rescheduling of lectures. This is due to the fact that lecturers are involved not only in teaching but also in administrative and researches works. Hence the convenience of lecturers is factored into the constraints in this work.

3. MATERIALS

3.1 Genetic algorithm

Genetic algorithm (GA) is a well-known and frequently used evolutionary computation technique. This method was originally developed by Holland, (1975). The GA is inspired by the principles of genetics and evolution, and mimics the reproduction behavior observed in biological populations.

In GA, a candidate solution for a specific problem is called an *individual* or a *chromosome* and consists of a linear list of genes. GA begins its search from a randomly generated population of designs that evolve over successive generations (iterations). To perform its optimization-like process, the GA employs three operators to propagate its population from one generation to another.

- ***Selection:*** In which the GA considers the principal of "survival of the fittest" to select and generate individuals that are adapted to their environment.
- ***Crossover:*** It mimics mating in biological populations. The crossover operator propagates features of good surviving designs from the current population into the future population, which will have a better fitness value on average.
- ***Mutation:*** It promotes diversity in population characteristics. The mutation operator allows for global search of the design space and prevents the algorithm from getting trapped in local minima Dalila,*et al*, (2015).

The cycle described above is illustrated in Figure 1 (Moreira, 2008).

3.2 Constraints

Constraints are relations over problem variables that define the space of solutions by specifying restrictions on the values that the variables may take simultaneously. There are two major constraints viz: hard/strong constraints (are those constraints that must be satisfied by a timetable); and weak/soft constraint (are those constraints that are not necessary for a valid timetable but are required for a good timetable). However, a third constraint is introduced into this work termed "***float constraint***" which is considered as the constraint that is either weak or strong in formulating timetable constraints. Thus, the requests made by lecturers are considered as float constraint in this work.

4. METHODOLOGY

Figure 2 generally describes the methodology employed for generating feasible timetable solution in this work:

- ***Data Description***

The data was collected from Faculty of Agriculture of a renowned university which has six basic departments. The faculty offers the same courses including practical courses for second year and third year students but relatively different courses for final students. The faculty has eighty-five lecturers including forty-two Professor cadres. Thirty-four lecturers are involved in teaching 2^{nd}, 3^{rd} and 5^{th} year students from all the departments. The faculty is allocated one lecture theatre of about1000 seats capacity, three 80 seats capacity halls, one 250 lecture hall, one 150 lecture hall and two 60 capacity laboratories and a big farm land. Each course is being taken by at least three lecturers which cause venue or time clashes.

Analysis of input Data

(a) Course description: Each course is described by four fields viz: Course code, Number of slots, Course Name and Course property
(b) Room description such room is described by four fields viz: Room code, Name, Room type, Room property
(c) Slot description each slot is described by four fields namely Slot code, End time, Start time, Name
(d) Days need no specification and have been built in the code.
(e) Lecturer description: Each lecturer has name tagged.

- ***Constraints Formulation***

A timetable that satisfies the below stated constraints is a feasible timetable. That a timetable is feasible does not give the assurance that it is good enough to be used. There are many criteria used to ascertain the quality of a timetable, but the only real way to know whether a timetable is good one or not is if it is usable by the department, the faculty and the university.

Hard Constraints

- All courses are scheduled in blocks of more than one hour, these restrictions must be respected. (HC1)
- No room can occupy more than one lecture at the same time. (HC2)
- No lecturer should have two classes consecutively. (HC3)
- Specifically minimize the use of 1pm-2pm hour slots for tea break every weekday and to allow Muslim prayers on Friday and 3pm-6pm for sports on Wednesdays. (HC4)

Soft Constraints

- No student should have to take two lectures in an adjacent period except it is necessary.(SC1)
- Lectures should not be slated for practical and farm periods, (SC2)
- For each period in the timetabling, the resources demanded made by the events scheduled for the period must not exceed the resources available, i.e. it is important not to schedule more students in a room than there are desks. (SC3)
- As much as possible, evening lectures starting from 6pm to 9pm hours should be assigned to rooms with standby generators so as to minimize loss of lecture hours due to frequent power cuts. (SC4)

Float Constraint

- Lecturer request to lecture at a particular time is granted.

➢ ***Genetic Algorithm Timetable Model***

Timetables are randomly selected from the population and used for breeding. No favoritism is given to fitter timetables. A child timetable is breed by performing unity order based on the parents. This means that each parent has an equal chance of providing each gene.

- ***Representation in Genetic Algorithm***

The timetable for a single room is a two dimensional array as shown in table 1. The timetable for a faculty is therefore collection of room timetables; one for every room in the faculty. Times at which there is no class booked hold a null booking, which has a value of zero.

A Faculty timetable stores information about what days are booked in each room, at hour of the day, on any day of the week. For Genetic algorithms, each of these bookings (or null bookings) is one gene. A timetable also has fields which describe (decode) some aspect of this genetic information. A timetable has a field which stores its cost. It also has fields which stores the number of breaches of each type of hard constraint. A population (or colony) is a collection of timetables. A population is itself a structured type with a number of fields. It contains a pointer to the least costly timetable in the population, (which has, in turn, a pointer to the next least costly).

A colony of creatures is therefore a singly-linked list of structured types (creatures) timetable data (genes) in a three dimensional array. The timetables are kept in order least costly to most costly. This method of gene representation means that it is not possible to have two classes booked to the same room at the same time. As such, there is one less hard constraint to be considered when evaluating timetables.

A slot (timeslot) was used to represent each hour (we assume that time is in an hour), for every room of everyday. Also, we assume that classes cannot begin before 9am and should finish before or at 9pm (12 hours total) and working days are from Monday to Friday (5 days).

- ***Fitness Function***

For each chromosome produced, there must be a fitness value which must be assigned to it, in order to know if the chromosome can actually survive the test of time. In order to determine the fitness function, the following procedures are followed:

- Each class can have points between 0 and 3.
- If lecture is scheduled in classroom with enough available seats, the score is then increased.
- If the entity (student) has no other lecture scheduled for a particular time, the class score is incremented.
- If lecture scheduled for a particular time breaks rule at any time-space slot that it occupies, its score is unchanged for that rule.
- Total score of class schedule is sum of points of all classes.

  Fitness value is then calculated as:

  F(x) = (schedule_score/maximum_score) *3

  Where Maximum_Score = Number of classes *3

- ***Reproduction***

According to Darwin's theory, the best one should survive and create new offspring. The primary objective of the reproduction is to emphasize good solutions and eliminate bad solutions in a population. Therefore, overall fitness of population becomes better. Reproduction copies good chromosomes from current population to next generation population.

- ***Crossover***

Crossover (recombination) is inspired by the role of sexual recombination in the evolution of living things. The way crossover operator works are outlined below:

- Copy first timeslot of fittest chromosomes of two parents into offspring.
- Place the remaining lectures from (the first timeslot of second parent to the offspring which does not conflict with already placed lectures).
- Select each timeslot from parent chromosomes and place those lectures in the same timeslot of offspring which have lesser number of violations of third soft constraint.
- Select feasible timeslot and place remaining lectures in it.

- ***Mutation***

Mutation is a genetic operator that is used to alter one or more genes in a chromosome. The effect of mutation is to re-introduce divergence into a converging population. In the latter a Genetic Algorithm run the algorithm may be converging upon a local maximum mutating some chromosome in a way to avoid getting stuck in a local maximum.

- ***Repair Strategy***

A repair strategy is used which ensures that all classes appear exactly once. For this is done in two stages. Firstly, any class which appears more than once is (non-deterministic) altered such that they appear only once. Secondly, any class which did not appear at all are booked to a spare space (regardless of room size).
If this repair strategy is applied to an empty timetable, the result is a timetable with each booked to random time and place. As such, the repair strategy is also used for initializing a random population. The use of the repair strategy ensures that each class is booked exactly once. Hence, the number of hard constraints which must be considered when timetables are being evaluated is further reduced. It has so far been shown that the hard constraints "classrooms not be doubly booked" and "every class must be scheduled exactly once" have been by non-genetic means.

- **Evaluation**

The timetable is evaluated by examining the number of constraint violations and weighting the result. Let C be the set of constraints that must be adhered to. Then for each $c_j \epsilon C$, there exist an associated weighting w_j. Also for each $c_j \epsilon C$ let $v(v_j,t)$ be the number of times constraint c_j is violated in timetable t. then, the fitness of timetable t, f(t), is defined as:
$f(t) = \sum_j w_j \; v(c_t \; t)$……..(1)

All weightings are positive, so fitness will be 0 when all constraints are satisfied (if that is possible) and greater than 0 otherwise. Appropriate checks for weights will lead to reasonable trade-offs between the different constraints that hopefully will guide evolutionary algorithm towards good timetables.

5. Implementation and Results Discussion

The system was developed using C++ programming language. The interfaces of the system that accommodates the program input section is shown in figure 3 while figure 4 was designed for courses input section. The system implementation was divided into three with respect to constraints viz; system A (with float constraint as strong constraint); system B (with float constraint as weak constraint) and system C (absence of float constraint).
Appendix A shows the system output samples which comprises of two schedules, the table to the left is for $2^{nd}/3^{rd}$ year while the table to the right is 5^{th} year students. We assume that the system will produced the same results if combined in the one table.
Assumptions: there is assumption that there is no clashes of any form with any course from other faculties.

Parameters Settings
The parameters chosen for this problem are as follows:

Population size: 500
Initialization: Random
Selection: Proportional to fitness rank
Crossover probability: 0.65
Mutation probability: 0.5
Repair strategy: Survival of the best 10% of current population
Stopping condition: 4000-20000 generations

As the value of fitness and generation increase, the courses are placed on the timetable and all necessary precautions are put in place in order to make sure that all constraints both hard and soft are not violated, especially the latter constraint (hard). Also, the existing faculty timetable implementation was monitored for 2015/2016 session in order to note the anomalies like undue postponements, re-scheduling or cancelation of lectures.

The results of the generated timetables by the algorithms are shown in table2, table3 and table4. The results of the systems are: SA produced total of 4 soft constraints; SB produced a total of 9 soft constraints while SC produced 15 soft constraints. However, the SA implementation resulted in re-scheduling of two (2) courses in 500level, system SB implementation resulted in re-scheduling of five (5) courses in (two courses in 200 level, two courses in 300 level and one course in 500 level) while system SC implementation resulted in re-scheduling of seven courses (three courses in 200 level, two courses in 300 level and two courses in 500 level) and two postponements against existing manual courses timetable that has eight re-scheduling and six postponements of courses at various levels. Thus, the accuracies of SA, SB and SC timetabling systems are 95.8%, 89.45 and 87.2% respectively while the existing manual timetable with 70.2% accuracy.

6. CONCLUSION

Timetabling problem being the hard combinatorial problem that it is would take more than just the application of only one principle. The timetabling problem may only be solved when the constraints and allocations are clearly defined and simplified thoroughly. In Genetic Algorithm, few periods are used regardless of the number of clashes. Although the manual time-table used by the faculty has undergone various manipulations over the years, it still suffer from some identified factors such as: lecturers having lectures consecutively because more than one lecturers handle a course; students having classes consecutively especially carried over courses; faculty lectures clashes with university classes; small venues allocated to large class etc. some of these were relatively dealt with by the developed GA system. This study was carried out as is to reduce the intense manual effort being put into creating and developing university timetables. The timetable automation system currently is a conceptual work in progress but has the capability to generate near optimal timetables. Thus, the accuracies of SA, SB and SC timetabling systems are 95.8%, 89.45 and 87.2%. This established that the lecturers request should be part of strong constraints for a conflict free timetabling.

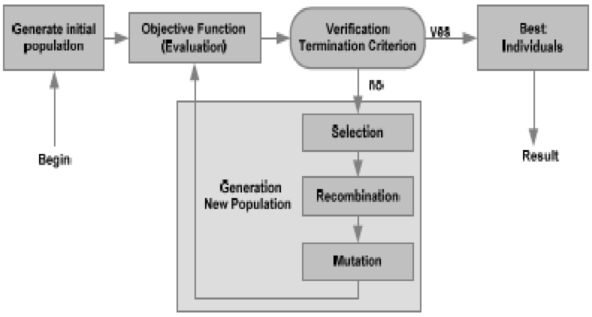

Fig. 1. Basic structure of the GA.

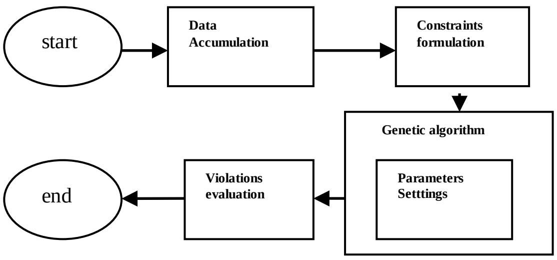

Figure 2: Work flow of the Timetabling System

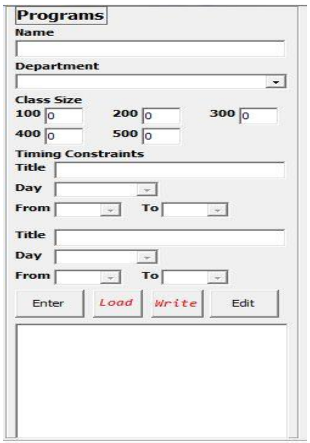

Figure 3: Program Input Section

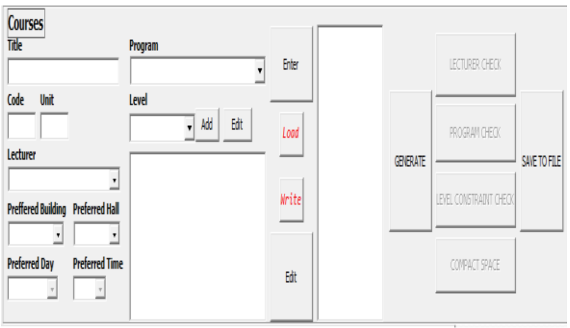

Figure 4: Course Input Section

Table 1: Example of a single room timetable

Time/Days	Monday	Tuesday	Wednesday	Thursday	Friday

Table 2: Results of generating Timetable (System A)

GA	SC 1	SC 2	SC 3	SC4	Sum of SCs
Run1	2	1	3	3	11
Run2	1	2	2	2	7
Run3	1	-	2	1	4

Table 3: Results of generating Timetable (SB)

GA	SC 1	SC 2	SC 3	SC 4	SC 5	Sum of SCs
Run1	2	3	3	2	2	12
Run2	1	2	2	2	1	10
Run3	1	2	2	1	1	9

Table 4: Results of generating Timetable (SC)

GA	SC 1	SC 2	SC 3	SC 4	Sum of SCs
Run1	4	4	5	5	18
Run2	3	4	6	4	17
Run3	4	2	4	5	15

7. REFERENCES

[1] Abramson D. & Abela J. (1992). "A Parallel Genetic Algorithm for Solving The School Timetabling Problem." In Proceedings of the 15th Australian Computer Science Conference, Hobart, 1-11.

[2] Bufe M., Fischer T., Gubbels H., Hacker C., Hasprich O., Scheibel C., Weicker K., Weicker N., Wenig M., & Wolfangel C. (2001). Automated solution of a highly constrained school timetabling problem - preliminary results. EvoWorkshops, Como-Italy.

[3] Carrasco M.P.& Pato M.V.(2001). "A Multiobjective Genetic Algorithm for the Class/Teacher Timetabling Problem." In Proceedings of the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science, Springer, 2079, 3-17.

[4] Chilma S.., C. Gomez and Á. G. Aguirre (2013). Class Schedule Assignment Based on Students Learning Rhythms Using A Genetic Algorithm, Ingeniería Ciencia, vol.9 no.17.

[5] Colorni A., Marco Dorigo, Vittorio Manniezzo (1992). "A Genetic Algorithm to Solve the Timetable Problem" Journal of Computational Optimization and Applications, 1, 90-92.

[6] Costa D., (1994). "A Tabu Search Algorithm For Computing An Operational Timetable." European Journal of Operational Research, 76(1), 98-110.

[7] Datta D., Deb K., & Fonseca, C.M.(2006). Multi-Objective Evolutionary Algorithm For University Class Timetabling Problem, In Evolutionary Scheduling, Springer-Verlag Press.

[8] David A. Coley (1999). An Introduction to Genetic Algorithms for Scientists and Engineers, 1st ed. World Scientific Publishing Co. Pte. Ltd.

[9] Dalila C., Imane H. and Amine N. (2015). Multimodal Score-Level Fusion Using Hybrid GA-PSO for Multibiometric System, Informatica 39 (2015) 209–216.

[10] Fang H. L. (1994). "Genetic Algorithms in Timetabling Problems." PhD Thesis, University of Edinburgh.

[11] Fernandes C. (2002). "Infected Genes Evolutionary Algorithm for School Timetabling." WSES International Conference.

[12] Gröbner M. andWilke P (2002). "A General View on Timetabling Problems." 4th International Conference on the Practice and Theory of Automated Timetabling - PATAT'02.

[13] Gotlieb C.C. (1962). "The Construction of Class-Teacher Timetables." Proceedings of IFIP Congress, North-Holland Pub. Co., Amsterdam, 73-77.

[14] Holland J. H., (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and Artificial Intelligence. U Michigan Press.

[15] Ismaila W. O, Adeosun O. O., Adigun A. A., Agunbiade A. D. (2012). "Genetic Algorithm-Based Physicians Non-Cyclic Rostering", International Journal of Emerging Technology and Advanced Engineering. 2(11), 776-781.

[16] Looi C. (1992). "Neural Network Methods In Combinatorial Optimization" Journal of Computers and Operations Research, 19(3/4), 191-208.

[17] Moreira J. (2008) "A system for Automatic Construction of Exam Timetable Using Genetic Algorithms" Journal of Algorithm and Computations 6(9), 1-18.

[18] Piola R., (1994). "Evolutionary Solutions To A Highly Constrained Combinatorial Problem." In Proceedings of IEEE Conference on Evolutionary Computation (First World Congress on Computational Intelligence), Orlando, Florida, 1, 446-450

[19] Srinivasan D., Seow T.H., & Xu J.X. (2002). "Automated time table generation using multiple context reasoning for university modules." In Proceedings of IEEE International Conference on Evolutionary Computation (CEC'02), 1751-1756.

[20] Saptarini N.G., Suasnawa I W and Ciptayani P. I. (..).Senior high school course scheduling using genetic algorithm, Journal of Physics: Conference Series, Volume 953, conference 1.

Ismaila W. Oladimeji has first degree in Computer science from Department of Mathematics and Computer Science, Federal University of Technology, Minna in 1998. He got his Master of Technology and PhD in Computer science from Department of Computer Science, Federal University of Technology, Akure in 2004 and 2012. He currently lectures as a Senior lecturer at Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria. His areas of research are Soft Computing, Modeling and Simulation, Computer Networks and Cyber-security. He has supervised about eighty-five graduates and six postgraduate students. He has published over thirty papers in both national and international journals. He is an associate member of National Computer Society and Computer Professionals of Nigeria.

LAUTECH Journal of Computing and Informatics (LAUJCI) – ISSN: 2714-4194
Volume 1– No.1, Sept. 2020 – www.laujci.lautech.edu.ng

Appendix A

Fitness: 1.000000, Generation: 15584

<table>
<tr><th>Room: Timetable
Lab: N Seats: 24</th><th>MON</th><th>THU</th><th>WED</th><th>THR</th><th>FRI</th></tr>
<tr><td>9 - 10</td><td></td><td rowspan="2">AEC 201
PROF.olaniyi
(AGN LAB)
/1800/
RSLPG</td><td></td><td></td><td></td></tr>
<tr><td>10 - 11</td><td rowspan="2">APH 303
PROF.aderinola
(AGN LAB)
/1300/
RSLPG</td><td></td><td rowspan="2">CSE 201
DR.amoo
(GHANA
HOUSE I)
/1200/
RSLPG</td><td rowspan="2">CPS 205
DR.olabayo
(FAG 1000LT)
/1300/
RSLPG</td></tr>
<tr><td>11 - 12</td><td rowspan="2">APH 301
DR.amoo
(GHANA
HOUSE I)
/1300/
RSLPG</td><td rowspan="3">ANB 201
DR.rafiu
(150LT)
/1700/
RSLPG</td></tr>
<tr><td>12 - 13</td><td></td><td rowspan="2">CPS 306
PROF.aderinola
(AGN LAB)
/1000/
RSLPG</td><td></td></tr>
<tr><td>13 - 14</td><td rowspan="2">APH 201
PROF.okunade
(BEE LAB)
/1700/
RSLPG</td><td></td><td rowspan="2">APH 203
DR.olaniyi
(GHANA
HOUSE II)
/1300/
RSLPG</td></tr>
<tr><td>14 - 15</td><td rowspan="2">APH 201
PROF.okunade
(BEE LAB)
/1800/
RSLPG</td><td rowspan="2">AER 201
PROF.aderinola
(AGN LAB)
/1300/
RSLPG</td><td></td></tr>
<tr><td>15 - 16</td><td rowspan="3">ANB 201
DR.rafiu
(150LT)
/1300/
RSLPG</td><td rowspan="2">CPS 201
DR.adetunji
(ADEOJO LH)
/2000/
RSLPG</td><td rowspan="3">ANB 201
DR.rafiu
(150LT)
/1800/
RSLPG</td></tr>
<tr><td>16 - 17</td><td rowspan="2">AER 201
PROF.aderinola
(AGN LAB)
/2000/
RSLPG</td><td rowspan="2">CPS 203
DR.amoo
(GHANA
HOUSE I)
/1300/
RSLPG</td></tr>
<tr><td>17 - 18</td><td rowspan="2">APH 201
PROF.akingbade
(GHANA
HOUSE I)
/1700/
RSLPG</td></tr>
<tr><td>18 - 19</td><td></td><td rowspan="2">AEC 303
PROF.olabiyi
(FAG 1000LT)
/1700/
RSLPG</td><td></td><td rowspan="2">APH 203
DR.olaniyi
(GHANA
HOUSE II)
/2000/
RSLPG</td></tr>
<tr><td>19 - 20</td><td rowspan="2">AER 201
DR.babajide
(BEE HALL)
/1700/
RSLPG</td><td rowspan="2">CEP 203
DR.babajide
(BEE HALL)
/1800/
RSLPG</td><td rowspan="2">FPT 201
PROF.aderinola
(AGN LAB)
/2000/
RSLPG</td></tr>
<tr><td>20 - 21</td><td></td><td></td></tr>
</table>

<table>
<tr><th>Room: Timetable
Lab: N Seats: 60</th><th>MON</th><th>THU</th><th>WED</th><th>THR</th><th>FRI</th></tr>
<tr><td>9 - 10</td><td></td><td rowspan="2">ANB 303
DR.olabayo
(FAG 1000LT)
/1000/
RSLPG</td><td rowspan="2">APH 201
PROF.okunade
(BEE LAB)
/2000/
RSLPG</td><td rowspan="2">AER 301
DR.babajide
(BEE HALL)
/1500/
RSLPG</td><td></td></tr>
<tr><td>10 - 11</td><td></td><td rowspan="2">AER 201
DR.babajide
(BEE HALL)
/1800/
RSLPG</td></tr>
<tr><td>11 - 12</td><td rowspan="2">APH 203
DR.olaniyi
(GHANA
HOUSE II)
/1700/
RSLPG</td><td rowspan="2">APH 201
PROF.akingbade
(GHANA
HOUSE I)
/1800/
RSLPG</td><td></td><td></td></tr>
<tr><td>12 - 13</td><td rowspan="2">APH 201
PROF.okunade
(BEE LAB)
/1300/
RSLPG</td><td rowspan="2">ANB 201
Dr.ojediran
(FAG 1000LT)
/2000/
RSLPG</td><td rowspan="2">AEC 201
PROF.olabiyi
(FAG 1000LT)
/1700/
RSLPG</td></tr>
<tr><td>13 - 14</td><td rowspan="2">CEP 301
DR.olabayo
(FAG 1000LT)
/1500/
RSLPG</td><td rowspan="2">AEC 201
PROF.olaniyi
(AGN LAB)
/1700/
RSLPG</td></tr>
<tr><td>14 - 15</td><td rowspan="2">CEP 201
PROF.olabiyi
(FAG 1000LT)
/1700/
RSLPG</td><td></td><td></td></tr>
<tr><td>15 - 16</td><td rowspan="2">ANB 201
Dr.ojediran
(FAG 1000LT)
/1700/
RSLPG</td><td rowspan="2">AEC 301
DR.amoo
(GHANA
HOUSE I)
/1300/
RSLPG</td><td rowspan="2">CPS 301
DR.babajide
(BEE HALL)
/1700/
RSLPG</td><td rowspan="2">CPS 304
PROF.aderinola
(AGN LAB)
/2000/
RSLPG</td></tr>
<tr><td>16 - 17</td><td rowspan="2">GNS 209
DR.babajide
(BEE HALL)
/1700/
RSLPG</td></tr>
<tr><td>17 - 18</td><td></td><td rowspan="2">APH 203
DR.olaniyi
(GHANA
HOUSE II)
/1800/
RSLPG</td><td></td><td rowspan="2">ANB 301
PROF.aderinola
(AGN LAB)
/1300/
RSLPG</td></tr>
<tr><td>18 - 19</td><td rowspan="3">ANB 201
DR.rafiu
(150LT)
/2000/
RSLPG</td><td></td><td rowspan="2">FPT 301
DR.amoo
(GHANA
HOUSE I)
/1000/
RSLPG</td></tr>
<tr><td>19 - 20</td><td rowspan="2">CPS 201
DR.adetunji
(ADEOJO LH)
/1800/
RSLPG</td><td rowspan="2">APH 305
DR.olabayo
(FAG 1000LT)
/1200/
RSLPG</td><td></td></tr>
<tr><td>20 - 21</td><td></td><td></td></tr>
</table>

R- Room overlapping **S**- Seats **L**- Agric lab **P**- Lecturer overlapping **G**- Student group overlapping
GREEN color indicates that requirement is satisfied RED color indicate that requirement is not satisfied
HATCHED area indicates overlapping of classes at same time and place