An Integer Linear Programming Model of a University Course Timetabling Problem.

V.O. Oladokun, Ph.D.* **and S.O. Badmus, B.Sc.**

Department of Industrial and Production Engineering,
University of Ibadan, Ibadan, Nigeria.

*E-mail: vo.oladokun@mail.ui.edu.ng

ABSTRACT

In this study, the combinatorial problem of university course timetabling of an Engineering Faculty of a Nigerian university (the University of Ibadan) was addressed. The problem of assigning lecturers, rooms, and courses to fixed timeslots, normally, a week, while satisfying a number of problem-specific constraints was modeled as an Integer Linear Programming (ILP).

The problem constraints have been divided into hard constraints and soft constraints. While the hard constraints constitute the problem constraints, the minimization of the violation of the soft constraints constitutes the objective function. Being an NP-Hard problem, a heuristic was developed and implemented manually.

Ten (10) solutions were generated manually from 10 runs with the newly developed search technique. An analysis of the performance of the solutions shows that the proposed heuristic is promising.

(Keywords: course timetabling, integer linear programming, ILP, heuristic)

INTRODUCTION

Course timetabling is the process of assigning courses, rooms, students, and lecturers to a fixed time period, normally a working week, while satisfying a given number of constraints. Many types of timetabling problems are NP-Hard problems characterized by some complex constraints (Cooper and Kingston, 1996; Jihad et al., 2005). These constraints are divided into Hard and Soft (Werra, 1985). While the hard constraints must be satisfied, the soft constraints are to be satisfied as much as possible but may be violated to arrive at a workable timetable. The violation of the hard constraints results in an infeasible solution or unacceptable schedule.

The literature in the area of school timetabling is extensive, though uneven. Techniques for solving timetable problems can be classified into four categories: heuristic, combinatorial, graph theoretical, (Miner et al, 1995) and mathematical programming (Werra, 1985; Daskalakiet et al., 2004). The heuristics approaches include the genetic algorithm (Peter and Dave, 1994), Tabu Search (Mushi, 2006; White et al., 2004), simulated annealing (White and Xie, 2001), evolutionary algorithms, and others.

Terminologies: The following terminologies are used in this paper:

1. **Course**: A discrete administrative unit of instruction from teacher(s) to students for which grade can be acquired. Courses are generally identified by a unique code such as TIE 312, and by a title which may not be unique.

2. **Timetabling Period:** This is the period of calendar time for which a course timetable will apply and is normally equal to the calendar length of course sections.

3. **Timeslots**: A timeslot, or period, is a continuous length of time during which instruction is offered. It can range from twenty minutes to four hours or more.

4. **Event**: An assignment of lecturer, room, and course to an hour timeslot. A course can be several hours depending on the number of hours set on the curriculum.

5. **Time Block**: A lecture with more than one consecutive event

6. **Lecturer**: A university staff employed to pass instruction to student.

7. **Students' Group**: Members of the same class.

8. **Course Group**: Set of Students' Group offering a particular course

9. **Room Capacity**: This is the maximum number of students a room can accommodate comfortably at a time.

10. **Course Weight (Unit)**: It is the relative amount of importance attached to every course. This determines the number of contact hour per week.

METHODOLOGY

This study aims at optimizing the allocation of limited resources available for lecturing. The resources include: lecturers, rooms, time, and other teaching facilities like sporting facilities, laboratory facilities, and so on. A two phase approach has been used in tackling this problem.

PHASE I: Modeling the problem mathematically

PHASE II: Solving the resulting model using a suitable algorithm

Assumptions

1. The decision of who takes which course is predetermined.

2. Timeslots occupied by courses from other departments are known before drawing the faculty timetable.

3. No lecturer preferences are given.

4. Students-course assignments are known and predetermined.

Decision Variables

Let i be the period index, i = 1,2,3, ..., Q; j the course index, j = 1,2,3 ..., N and k the room index, k = 1,2,3,..., M.

Let $x_{ijk} = 1$ if course j is scheduled for period i at room k and 0 otherwise.

Model Parameters

G	Set of all departmental courses (G_1, G_2, G_3, ...,G_8)
CS_a	Set of courses taken by a student group, ***a***
A	Set of students group (a_1, a_2, a_3, ...,a)
CT_b	Set of courses taken by Teacher, ***I***
T	Set of teachers
I	Set of periods
R	Set of rooms
RG_a	Set of rooms explicitly meant for courses, j
I_D	Set of periods of day D
B_k	Capacity of room, k
P_j	Size of course j
d_j	Duration of course j
R_α	Lecture rooms
R_β	Laboratory
Q	Number of periods
a	Number of student groups
b	Number of lecturers
M	Number of rooms
I_E	Set of pre-assigned period
I_L	Set of lunch periods
I_T	Set of lecturer time preferences
I_J	Set of Jumat periods
I_S	Set of special

Problem Constraints

Timetabling constraints are divided into two categories: hard constraint and soft constraints. The following are the hard constraints in this case:

1) A student group, ***a*** cannot take more than one course at a time, ***i***

2) A lecturer, ***l*** cannot take more than one course at a time, ***i***

3) A room must not be assigned for more than one course at a time, ***i***

4) Every course must have total number of allotted hours equivalent to its allocated weight, d_j

5) A course must not be scheduled more than once in a day

6) Room capacity constraints must be respected (i.e. no room can take more than its capacity).

The following are the soft constraints:

1) Minimize the utilization of early morning and late evening lectures

2) Avoid lectures during the special periods such as Jumat service periods and lunch

3) Avoid periods already pre-assigned to borrowed courses from other faculties

4) Respect lecturers preferences

5) Minimize the allocation of courses to rooms outside the course room group.

The following groups were identified:

1) Room group

2) Students' group

3) Lecturer's group

4) Time slots

Mathematically the constraints are:

Hard constraints: A student group, ***a*** cannot take more than one course at a time, ***i***:

$$\sum_{j \in CS_a} \sum_{k=1} x_{ijk} \leq 1 \quad \forall i = 1.2.3 \ldots Q. \forall a = 1.2.3 \ldots p \qquad \textbf{(1)}$$

A lecturer, ***l*** cannot present more than one course at time, ***i***:

$$\sum_{j \in CL} \sum_{k=1}^{M} x_{ijk} \leq 1 \quad \forall i = 1,2,3 Q. \forall b = 1,2,3 q. \forall a = 1,2,3 p.., \qquad \textbf{(2)}$$

A room must not be assigned for more than one course at a time, ***i***:

$$\sum_{j=1}^{N} x_{ijk} \leq 1 \quad i = 1,2,3,\ldots Q, \ \forall k = 1,2,3,\ldots M \qquad \textbf{(3)}$$

The total time allotted to a course must be equal to the weight assigned to it:

$$\sum_{i=1}^{Q} \sum_{k=1}^{M} x_{ijk} = d_j, \ \forall j = 1,2,3,\ldots N, \qquad \textbf{(4)}$$

Every course must have a total number of allotted hours equivalent to its allocated weight, d_j and must be contiguous:

$$\sum_{t=i-d_j+1}^{i} x_{tjk} = d_j \ . \forall j \in G, \forall k \in RG_a . \forall i \in I_D \qquad \textbf{(5) \& (6)}$$

Room capacity constraints must be respected (i.e., no room can take more than its capacity):

$$P_j \leq B_k, \ j = 1,2,3,\ldots, N, \ k = 1,2,3,\ldots, M \qquad \textbf{(7)}$$

Soft constraints: Special period assignment: Let $f_{s1}(x)$ be the function representing this assignment. We have:

$$\sum_{j=i}^{N} \sum_{k=1}^{M} \sum_{i \in I_s} x_{ijk} = f_{s1}(x), \ I_s = \textit{set} \text{ of early morning and late evening time slots}$$

If we let λ_{s1} be the assigned weight/penalty, then we minimize:

$$\lambda_{s1} f_{s1}(x) \qquad \textbf{(i)}$$

the minimum occurs when $f_{s1}(x) = 0$.

Jumat period assignment: Let $f_{J2}(x)$ represent such assignments.

$$\sum_{j=i}^{N}\sum_{k=1}^{M}\sum_{l\in I_J} x_{ijk} = f_{J2}(x),\ \mathrm{I_J} = set \text{ of}$$

Friday Muslim prayer times

If we let λ_{J2} be the assigned weight, then we minimize:

$\lambda_{J2}f_{J2}(x)$ **(ii)**

$f_{J2}(x) = 0$ is the best value.

Lunch period assignment: Lunch periods are to be respected for students and lecturers recess. Assigning lectures to these periods attracts a light weight penalty, λ_{L3}.

$$\sum_{j=i}^{N}\sum_{k=1}^{M}\sum_{l\in I_L} x_{ijk} = f_{L3}(x),\ \mathrm{I_L} = set \text{ of lunch timeslots}$$

then we minimize

$\lambda_{L3}f_{L3}(x)$ **(iii)**

The best value of $f_{L3}(x) = 0$

Pre-assigned periods: Assigning faculty courses to period pre-assigned to borrowed courses is highly discouraged and courses a heavy penalty, λ_{E4}. This translates to:

$$\sum_{j=i}^{N}\sum_{k=1}^{M}\sum_{l\in I_E} x_{ijk} = f_{E4}(x),\ \mathrm{I_E} = set \text{ of}$$

pre - assigned timeslots

Hence we minimize

$\lambda_{E4}f_{E4}(x)$ **(iv)**

Objective function

Combining the above four soft constraints, we obtain:

$f(x) = \lambda_{s1}f_{s1}(x) + \lambda_{J2}f_{J2}(x) + \lambda_{L3}f_{L3}(x) + \lambda_{E4}f_{E4}(x)$

Model

Minimize: $f(x) = \lambda_{s1}f_{s1}(x) + \lambda_{J2}f_{J2}(x) + \lambda_{L3}f_{L3}(x) + \lambda_{E4}f_{E4}(x) + \lambda_{T5}f_{T5}(x)$

Subject to equations 1-6 above.

Selection of weight for the objective function

The weights used above are valued based on experiences as shown in Table 1.

Table 1: Penalty Cost Assignment.

Weight	Value	Description
λ_{s1}	100	Early morning and late evening classes
λ_{J2}	1000	Friday Muslim Prayer
λ_{L3}	50	Lunch
λ_{E4}	1000	Pre-assignment
λ_{T5}	100	Lecturers' Preferences

Parameterization

The model defined above is a generic form of the problem under study. It is applicable to any system with characteristics similar to the one under study. To implement this model, the faculty of technology at the University of Ibadan was used. For this case study, N = 215 courses, Q = 54 timeslots, M = 19 rooms, b= 80 lecturers, and a = 32 students groups. This results in (54 x 215 x 19) variables.

The following are the values for the parameters whose notations were defined above:

G = {1, 2, 3,...,215}
R = {1, 2, 3,...,19}
I = {1, 2, 3,..., 54}-{11,22,33,44}
A = {1, 2, 3,..., 32}
T= {1, 2, 3,..., 80}
I_1 = {1,2,3, ..., 10}
I_2 = {12,13,14,...,21}
I_3 = {23,24,25,...,32}
I_4 = {34, 35, 36,...,43}
I_5 = {45,46,47,..., 54}
I_s = {1,10,12,21,23,32,34,43,45,53}
I_J = {49,50,51}
I_L = {6,17,28,39,50}
I_E = { Not available}

Implementation

A heuristics called knock-out has been developed. This heuristic approach uses the formulated model for implementation. Knock-out is only a solution generation procedure capable of generating reasonable solution which can be tested with the hard constraints for feasibility.

Knock-out search algorithm

STEP 0 initialization: generate a row vector of length QxNxM having all its elements being 1

STEP 1 Identify all the course group by students and lecturer

STEP 2 Select a "1" from the vector equivalent to a particular X_{tpq}

STEP 3 Turn all 1's of the X_{ijk} for which, i = t, k = q but j $\neq$ p to zero

STEP 4 Turn all 1's of the X_{ijk} for i = t, j = p and k is not equal to q to zero

STEP 5 Turn all 1's of the X_{ijk} for **i** = **t** and $(j, p) \in$ **G**, where **G** represents a course set

STEP 6 Keep the X_{tpq} as a valid solution

STEP 7 Select another non-zero variable X_{tpq} and go to STEP 3

STEP 8 If no other X_{ijk} can be knocked out, stop and report solution

The solution from this algorithm ensures the satisfaction of the first four hard constraints stated earlier in this work. The only short coming of this solution is its inability of assuring contiguity timeslots (i.e. time blocks). We can therefore test the acceptability of any solution generated by this algorithm by determining the number of course assignment completely satisfied.

RESULTS

The performance of each solution is rated on the following scale:

1. **Effectiveness**: this is a measure of number of hours scheduled as against the expectation.

Complete schedule	100
Half schedule	50
Unscheduled	0

2. **Contiguity**: this is a measure of whether the duration assigned is continuous (i.e., time without gap and in the same room).

	Time	Room
Contiguous	100	100
Discontinuous	0	0

Solution	**1**	**2**	**3**	**4**	**5**
Rating	350	350	450	650	450
Solution	6	7	8	9	10
Rating	700	750	700	450	650

From the above data, the performance of individual solutions can be seen at glance. It can be inferred that solution 7 produces a better schedule of the courses with a total rating of **750**. The durations for the courses are: $d_1 = 2$, $d_2 = 1$, $d_3 = 2$, $d_4 = 2$

CONCLUSION

The study has demonstrated how to model and solve course timetabling problem using the faculty of technology, University of Ibadan as a test case. This work uses a special search heuristic called knock-out to find feasible solutions to the small size problem. A performance rating was carried out on the derived solution in 10 runs using some criteria like *effectiveness*, *contiguity in time* and in *room*.

However, other search techniques available in the literature such as Genetic algorithm, tabu search, simulated annealing, etc. can be tested with the model. Adopting this model and implementing it with a very good search algorithm as the one described promises to bring improvement to resource allocation problem inherent in course timetabling of a university.

REFERENCES

1. Cooper, T.B. and Kingston, J.H. 1996. "The Complexity of Timetabling Construction Problems". *Springer Lecture Note in Computer Science*. 1153:283-295.

2. Daskalaik, S., Birbas T., and Hausos, E. 2004. "An Integer Programming Formulation for a Case Study in University Timetabling." *European Journal of Operations Research*. 153:117-138.

3. Jihad, M.J., Mohamed, L.R., and Ahmad, M.H. 2005. "Modeling the Timetabling Problem using Goal Programming". *Journal of Computer Science* (special issue):67-71.

4. Miner, S., Elmohamed, S., and Hon, W. 1995. "Optimisation of Timetabling Solutions Using Graph Coloring". In: *Timetabling Techniques*. North-East Parallel Architecture Centre (NPAC).

5. Mushi, A.R. 2006. "Tabu Search Heuristics to University Course Timetabling Problem". *African Journals for Science and Technology (AJST)*. Science and Technology Series. 7(1):34-40.

6. Peter, R. and Dave, C. 1994. "Applications of Genetic Algorithms". *AISB Quarterly*. T.C. Fogarty (ed). 89:23–30.

7. Werra, D. 1985. "An Introduction to Timetabling". *European Journal of Operation Research*. 19:151-162.

8. White, G.M. and Xie, S.B. 2001. "Examination Timetabling and Tabu Search with Longer Memory". In: *The Practice and Theory of Automated Timetabling III: Lecture Notes in Computer Science*. E. Burke and W. Erban (eds.). 2079: 85-103. Springr-Verlag: Berlin, Germany.

9. White, G.M., Xie, S.B., and Zonjie, S. 2004. "Using Tabu Search with Longer Term Memory and Relaxation to Create Examination Timetables". *European Journal of Operation Research*. 153:80-91.

ABOUT THE AUTHORS

V.O. Oladokun, lectures in the Department of Industrial and Production Engineering, Faculty of Technology, University of Ibadan. He graduated with a B.Sc. (Hons) in Mechanical Engineering from the Obafemi Awolowo University (OAU), Ile-Ife. He has an M.Sc. and Ph.D. in Industrial Engineering from the University of Ibadan. He is a Member, Nigerian Society of Engineers; and a Member, Nigerian Institution of Engineering Management. His research interests include enterprise model development and production system optimization in developing economies.

S.O. Badmus. Works as an Industrial Engineer with a multinational manufacturing company in Nigeria. He has a B.Sc. Degree In Industrial and Production Engineering from the University of Ibadan, Nigeria and was the top graduating student in his class.

SUGGESTED CITATION

Oladokun, V.O. and S.O. Badmus. 2008. "An Integer Linear Programming Model of a University Course timetabling Problem". *Pacific Journal of Science and Technology*. 9(2):426-431.

Pacific Journal of Science and Technology