MAXIMAIZING USE OF POSTGRADUATE LECTURE HALLS IN COVENANT UNIVERSITY USING BRANCH-AND-BOUND METHOD

[1]Chinke R.I,[2]Agarana M.C, [3]Onosu G. and [4]AsiteOvakporoye

[2]Department of Mathematics, Covenant University, Nigeria.
[1,3,4]Department of Computer and Information Sciences, Covenant University, Nigeria.

ABSTRACT

Most postgraduate students in Covenant University often encounter the challenge of finding lecture rooms to use for their lectures. It is either the classes are already occupied by other postgraduate students or undergraduate students. There are eight lecture rooms in the College of Entrepreneurship Studies which have been assigned to the postgraduate students to have their lectures, but never specified which department would be having which class and at what time.

This paper attempts to provide a mathematical solution to the problem by modelling the problem as an integer programming problem. The model schedules students, classes and departments in such a way that when a department is to use a certain class for a period of time another department would not disturb them. This way each department knows the lecture duration for the classes assigned to them. The resulting model was solved using branch and bound method. It was observed that the Experimental results indicate that the proposed model is tractable using the branch-and-bound method.

Keywords*: Integer programming, Scheduling, Postgraduate students, branch-and-bound*

1. INTRODUCTION

Covenant University is a Christian based Private institution affiliated Living Faith Church, located in Canaan Land, Ota, Ogun State, Nigeria.The University was established on the 21st of October 2002 by Bishop David Oyedepo who is also the Founder of Living Faith Church Worldwide (Wikipedia, 2015). Covenant University is a growing, vision driven University, which is committed to pioneering excellence at the cutting edge of learning. (University, 2017) However, the School of Postgraduates Studies have a challenge of clashing classes in the classrooms provided for postgraduate students to have their lectures and students not optimizing the capacity of the lecture halls. Due to the growing number in the University a large variety of courses are available for students to study and each lecture halls available are limited, it is not appropriate for courses to be assigned to lecture halls with more capacity than the number of students registered for it (Ahmed & Fadi , 2016).From the perspective of Covenant University School of Postgraduate studies, class scheduling seems to be a difficult problem. Courses are assigned to a lecturer given time slot(s) during the week and a classroom to hold the lectures (Winch & Yurkiewicz, 2016). One course can begiven to a lecturer and a room at a given time. The number of registered students and characteristics of the course should be considered (Winch & Yurkiewicz, 2016).Viewing this from the students perspective, class scheduling is not as difficult as creating a schedule of classes for the departments, it also involves balancing various desires and constraints (Winch & Yurkiewicz, 2016). In most Universities, the students naturally spend a significant amount of time deciding the classes to take each semester, pondering over other alternative schedule that would accommodate their job schedule, internship and weighing the benefits and detriments of the various lecturers and courses (Winch & Yurkiewicz, 2016) . This challenge can be solved using linear integer programming, linear programming is a mathematical technique used for either maximizing or minimizing a linear function of several variables, which could be time, output or cost. (Wikipedia, 2016). Linear programming models are known to be continuous and the decision variables are allowed to be in fraction. The problem usually referred to as (Linear) integer programming problem, it is called a mixed integer program when some variables are restricted to be an integer and called Pure Integer program when all the decision variables must be integers. If the constraints are of a network nature, then an integer solution can be obtained by ignoring the integrality restrictions and solving the resulting linear program. Generally, variables will be fractional in the linear-programming solution and further measures must be taken to determine the integer-programming solution. Integer-programming models arise in nearly every area of application of mathematical programming (Wikipedia, 2016).Integer programming model are regarded by the fact that most of the variable are meant to be integers. For the purpose of this research we would be using the branch and bound model of integer programming as a method of solution to the scheduling problem faced by the School of Postgraduate studies (Alberto, 2013). The branch and bound method uses an intelligent search procedure to get an optimal or close-to-optimal solution to integer programming problems (Alberto, 2013). The branch and bound indirectly counts all the likely solutions to an integer program. Branch and bound main focus is to 'divide and conquer'. Considering that the large problem is difficult to solve directly, it is then divided into lesser sub-problems so the problems can be solved. The branching is done by dividing the set of possible solutions into lesser and lesser subsets (Baskaran, Bargiela, & Qu, 2014). The fathoming is done by providing a bound for the top best feasible solution in the subset or disposing the subset if the bound shows that it has no optimal solution (Baskaran, Bargiela, & Qu, 2014).

Presently, postgraduate students have to go to their college to research for an available class to use for their lecture. The number of classes available would be noted and number of departments would also be noted and this information will be gotten from School of Postgraduate studies. For example, if there are three courses with 6, 10 and 15 students respectively, and three classrooms with capacities of 10, 20 and 50 respectively, all three courses can fit in any of the classrooms. However, it would be more efficient to assign the larger course to the larger classroom this would help the students learning experience and give room for more students to attend the class, reduce the chances of cheating when conducting exams. (Ahmed & Fadi , 2016)

S/N	DEPARTMENT	LECTURE HALL	SITTING CAPACITY	USERS	NO OF LECTURES IN A WEEK	LOWEST ATTENDANCE	HIGHEST ATTENDANCE	NO OF HRS PER WEEK PER DEPARTMENT	TOTAL NO OF HR OF HALL AVAILABLE/WEEK
1	BUSINESS ADMIN	LH1	58	BA, SC, ACC, CIS,	7	10	12	14	43
2	COMP. INFO SYS	LH2	58	CIS, BA, BIO SC,	6	13	20	12	44
3	BIO SCIENCES	LH3	60	IR, SC,	8	14	15	16	42
4	ECONOMICS	LH4	60	CIS, BIO SC	5	6	9	10	43
5	SOCIOLOGY	LH5	64	BA, LNG, IR,	6	5	8	12	45
6	ACCOUNTING	LH6	64	BA, IR	6	8	10	12	45
7	LANGUAGES	LH7	64	CIS, BA,	5	3	3	10	45
8	INTERNATIONAL RELATIONS	LH8	64	LNG,	7	5	8	14	45

2. Formulation of Model

Table1

(Abbreviations used and their meanings: SC- Sociology, IR- International Relations, ACC- Accounting, CIS- Computer and Information Systems, BIO- Biological Sciences, BA- Business Administration, LNG- Languages, Econs- Economics)

The official branch and bound formulation for this problem follows:

- **Meaning of a node in the branch and bound tree**: A lecture hall in use, full or partial.
- **Node selection policy**: Global best value of the bounding function
- **Variable selection policy**: Choose the next department in a natural order from 1 to 8.
- **Bounding function:** For unassigned lecture halls, choose the biggest class with properly functioning lights, projector and AC, even if the class has been used previously.
- **Terminating Rule**: when all the classes have been assigned to departments for a particular period of time. E.g. all eight lecture halls have been given to various departments for *am-10am
- **Fathoming:** All the lecture halls have been assigned and no two departments have been given the same lecture hall to use at the same time.

The problem will consist of 4 variables. The following constraints are generated:

- Let x1 represent average number of students in Lecture Hall 1 and Lecture Hall 2 when used for lectures in a week.
- Let x2 represent average number of students in Lecture Hall 3 and Lecture Hall 4 when used for lectures in a week.
- Let x3 represent average number of students in Lecture Hall 5 and Lecture Hall 6 when used for lectures in a week.
- Let x4 represent average number of students in Lecture Hall 7 and Lecture Hall 8 when used for lectures in a week.

The optimization function is expressed as follows:

Maximize: Z = x1 + x2 + x3 + x4

Such That, x1 + x2 <= 58

x1 + x2 + x3 <= 60

x3 + x4 <= 64

x1, x2, x3, x4 >= 0, Integers

In summary the example needs a total of 5 variables, 5 constraints, and 1 objective function. One solution is identified by the ILP solver and is summarized in Table 2. Clearly, the solution is the optimal solution since it yields the largest objective function value and assigns the smaller department, i.e. Sociology, to the smaller Lecture Hall, i.e. Lecture Hall 1, and the larger department, i.e. Computer and Information Sciences, to the larger classroom, i.e. Lecture Hall 7. CIS can only fit in Lecture Hall 7.

TABLE 2

Variables	result
	122
x1	58
x2	0
x3	2
x4	62
st	0

3. MODEL RESULTS

In this section, we evaluate the use of LPSolve algorithms. Lp_solve is a Mixed Integer Linear Programming (MILP) solver. It is an open license linear (integer) programming solver based on revised simplex method and the branch-and-bound method for the integers. This solver can handle an unlimited number of departments and lecture halls, it allows the user to enter preferences for each department or lecture hall, as long as the LPSolver accepts the input format generated.

Table 3

Variables	value	from	till
objective	122	122	122
R1	1	0	60
R2	0	58	122
R3	1	2	+inf
x1	0	-inf	+inf
x2	0	-inf	58
x3	0	-inf	+inf
x4	0	-inf	+inf
st	-1	-2	58

SUMMARY OF RESULTS USING LPSolve.

Optimal solution 122 after 3 iter.

lp_solve version 5.5.2.5 for 32 bit OS, with 64 bit REAL variables. In the total iteration count 3, 0 (0.0%) were bound flips. There were 0 refactorizations, 0 triggered by time and 0 by density.

... on average 3.0 major pivots per refactorization.

The largest [LUSOL v2.2.1.0] fact(B) had 4 NZ entries, 1.0x largest basis.

The constraint matrix inf-norm is 1, with a dynamic range of 1.

Time to load data was 0.011 seconds, presolve used 0.014 seconds.

SUMMARY OF RESULTS USING THE BRANCH-AND-BOUND METHOD.

Max Z = x1 +x2 + x3 + x4

Such That x1 + x2 <= Capacity of halls, 58

x1 + x2+ x3 <= Capacity of halls, 60

x3 + x4 <= Capacity of halls, 64

x1, x2, x3, x4 >= 0, integer

STAGE1: Get feasible integer value set of all the variables. Look for possible values all the variables can take.

x1 = 0, 1, 2, 3

x2 = 0, 1, 2, 3

x3 = 0, 1, 2, 3

x4 = 0, 1, 2, 3

STAGE 2: First stage Branch-and-Bound

Solve the 4 linear programs involving integer values of x1 = 0, 1, 2, 3 disregarding the integer constraints on x2, x3 and x4.

PROBLEM 1: with x1 = 0

Max Z = x1 +x2 + x3 + x4

0 + x2 <= 58, x2 = 58

0 + 58 + x3 <= 60, x3 = 2

2 + x4 <=64, x4 = 62

Z = 0 + 58 + 2 + 62 = 122

PROBLEM 2: with x1 = 1

Max Z = x1 +x2 + x3 + x4

1 + x2 <= 58, x2 = 57

1 + 57 + x3 <= 60, x3 = 2

2 + x4 <=64, x4 = 62

Z = 1 + 57 + 2 + 62 = 122

PROBLEM 3: with x1 = 2

Max Z = x1 +x2 + x3 + x4

2 + x2 <= 58, x2 = 56

2 + 56 + x3 <= 60, x3 = 2

2 + x4 <=64, x4 = 62

Z =21 + 56 + 2 + 62 = 122

PROBLEM 3: with x1 = 3

Max Z = x1 +x2 + x3 + x4

3 + x2 <= 58, x2 = 55

3 + 58 + x3 <= 60, x3 = 2

2 + x4 <=64, x4 = 62

Z = 3 + 55 + 2 + 62 = 122

Since all branches have the same figures I would pick x1 = 0 as the upper bound.

STAGE 3: SECOND GATE

PROBLEM 1: with x1= 0, x2 =0

Max Z = x1 +x2 + x3 + x4

0 + 0 + x3 <= 60, x3 = 60

60 + x4 <=64, x4 = 4

Z = 0 + 0 + 60 + 4 = 64

PROBLEM 2: with x1= 0, x2 =1

Max Z = x1 +x2 + x3 + x4

0 + 1 + x3 <= 60, x3 = 59

59 + x4 <=64, x4 = 5

Z = 0 + 1 + 59 + 5 = 65

PROBLEM 3: with x1= 0, x2 =2

Max Z = x1 +x2 + x3 + x4

0 + 2 + x3 <= 60, x3 = 58

58 + x4 <=64, x4 = 6

Z = 0 + 2 + 58 + 6 = 66

PROBLEM 4: with x1= 0, x2 =3

Max Z = x1 +x2 + x3 + x4

0 + 3 + x3 <= 60, x3 = 57

57 + x4 <=64, x4 = 7

Z = 0 + 1 + 59 + 5 = 67

Therefore, problem 4 (where x1 = 0, x2 = 3) gives the highest value is identified as the upper bound.

STAGE 4: THIRD GATE

PROBLEM 1: with x1 = 0, x2 =3, x3 = 0

Max Z = x1 +x2 + x3 + x4

= x3 + x4 <= 64

= 0 + x4 = 64, x4 = 64

Z = 0 + 3 + 0 + 64 = 67

PROBLEM 2: with x1 = 0, x2 =3, x3 = 1

Max Z = x1 +x2 + x3 + x4

= x3 + x4 <= 64

= 1 + x4 = 64, x4 = 63

Z = 0 + 3 + 1 + 63 = 67

PROBLEM 3: with x1 = 0, x2 =3, x3 = 2

Max Z = x1 +x2 + x3 + x4

= x3 + x4 <= 64

= 2 + x4 = 64, x4 = 62

Z = 0 + 3 + 2 + 62 = 67

PROBLEM 4: with x1 = 0, x2 =3, x3 = 3

Max Z = x1 +x2 + x3 + x4

= x3 + x4 <= 64

= 3 + x4 = 64, x4 = 61

Z = 0 + 3 + 3 + 61 = 67

Thus, since all the problems have the same values I would pick x1 =0, x2 = 3, x3 = 3.

STAGE 5: FOURTH GATE

PROBLEM 1: with x1 = 0, x2 = 3, x3 = 3, x4 = 0

Max Z = x1 +x2 + x3 + x4

0 + 3 + 3 + 0 = 6

PROBLEM 2: with x1 = 0, x2 = 3, x3 = 3, x4 = 1

Max Z = x1 +x2 + x3 + x4

0 + 3 + 3 + 1 = 7

PROBLEM 3: with x1 = 0, x2 = 3, x3 = 3, x4 = 2

Max Z = x1 +x2 + x3 + x4

0 + 3 + 3 + 2 = 8

PROBLEM 4: with x1 = 0, x2 = 3, x3 = 3, x4 = 3

Max Z = x1 +x2 + x3 + x4

0 + 3 + 3 + 3 = 9

Since the fourth terminal branch gives the highest integer value, it gives the optimal solution of the initial problem. Thus, the optimal solution can be written as:

x1 = 0

x2 = 3

x3 = 3

x4 = 3

Z = 9

4. CONCLUSIONS

In this paper, we present an Lp_solve based approach and branch-and-bound to generate university departmental schedules for postgraduate students. We showed how to formulate the university schedule as an Lp_solve problem and as a branch-and-bound method. The Lp_solve models are solved using Boolean satisfiability based Lp_solve. The goal is to find a schedule that satisfies the university's postgraduate rules, yet optimizes the use of the existing facilities such as maximizing the capacity to enrollment ratio in a Lecture hall. The approach was tested on different cases with various sizes and showed promising results. The approach is complete and will find the best possible schedule, or will indicate that no schedule exists that meets the current university rules.

5. References

Ahmed , W., & Fadi , A. A. (2016). Solving the University Class Scheduling Problem. *ccse*, 1-5.

Alberto, J. (2013). Graphical Method Branch and Bound Method Meeting 13 Lecture 7. In J. Alberto, *Integer Programming:* (pp. 1-30).

Baskaran, G., Bargiela, A., & Qu, R. (2014). Integer Programming: Using Branch and Bound to Solve the Nurse Scheduling Problem. *Proceedings of 2014 International Conference on Artificial Intelligence & Manufacturing Engineering* (pp. 203-209). Dubai: ICAIME 2014.

University, C. (2017, April 12). *AboutUs*. Retrieved from Covenant University: www.covenantuniversity.edu.ng/about-us

Wikipedia. (2015, April 5). *Covenant University*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/Covenant_University

Wikipedia. (2016, 12 10). *linearprogramming*. Retrieved from wiki: wikipedia.com/linearprogramming

Winch, J. K., & Yurkiewicz, J. (2016). STUDENT CLASS SCHEDULING WITH LINEAR PROGRAMMING. *One Pace Plaza*, 1-3.