Universiti Malaysia Terengganu Journal of Undergraduate Research *eISSN: 2637-1138*
Volume 3 Number 4, October 2021: 127-136

COURSE ALLOCATION AMONG LECTURERS USING PYTHON

LAI ANN NA AND MOHAMED SAIFULLAH HUSSIN*

Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia.

**Corresponding author: saifullah@umt.edu.my*

Abstract: Course allocation among lecturers describes the process of allotting a set of courses to a number of lecturers. The administrators who are responsible in the allotment of courses to lecturers at least once a year are supposed to assign the most suitable lecturer to teach the courses in an efficient and effective way. However, the process of course allocation among lecturers is being done manually in most of the educational institutions through a trial-and-error manner and the lecturers' years of teaching experience was not being considered during the allocation causing imprecision of the allocation made. Therefore, a random allocation of courses to lecturers using Microsoft Excel was done and the objective function of the solution obtained through the random allocation is compared to the objective functions of exact solutions obtained using OpenSolver and Python. The purpose of using Python is to automate the allocation of courses to lecturers in which a lecturer's years of teaching experience is being optimized even if there is occurrence of data changes. Besides that, the computational time used in obtaining the solutions using the three mentioned approaches are compared to show the difference in terms of efficiency and effectiveness of the allocation made. Python proved to be the most efficient and effective approach as compared to the other two approaches used for this course allocation problem as Python requires the least time and effort to obtain the optimal combination of lecturers and courses based on lecturers' years of experience.

Keywords: Python, course allocation, lecturers, years of experience, optimization.

Introduction

A university consists of multiple faculties or schools where faculties, in turn are often made up of several departments. Departments offer courses that include core courses and elective courses to students to meet the academic requirements of undergraduate and post-graduate students. The academic performance of students in a university depends on how well the student is being taught and also the capability of student coping with the course (Emeka, 2018). Lecturers who work at postsecondary institutions like universities are the biggest guide and also the most influential factor contributing to student achievement. Therefore, course allocation among lecturers is an important aspect to be focused on, and this can be seen from works that have been done previously (Amuji *et al.,* 2017; Badri, 1996; Gunawan & Ng, 2011 & Shohaimay *et al.,* 2016).

Course allocation involves the assignment of a certain number of lecturers to teach a set of courses offered. The administrators who are responsible in course allocation will need to allocate the courses to lecturers at least once a year due to the occurrence of changes in the number of lecturers, courses, classes and students. For instance, the administrators need to administer the allocation of courses again when any lecturer is not available due to maternity leave or retirement, or when courses are cancelled or developed due to low or high demand. Since the process of allocating teaching load that involves allocation of courses among lecturers is being done manually in most of the educational institutions through a laborious trial-and-error approach (Shohaimay *et al.,* 2016), therefore it will be time-consuming, non-effective and has bias tendency (Emeka, 2018) due to unequal teaching load.

Therefore, the objectives of this study are to identify the most effective and efficient method that can be used by administrators to automatically allocate courses to lecturers and to optimize the allocation of courses to lecturers according to their years of experience in teaching the courses using Python. The approaches are necessary to obtain the optimal combination between lecturers and courses within a short period of time so that the lecturers have more priority to teach the courses that they are experts in. This helps to increase the students' understanding as the courses will be taught by experienced lecturers (Helen, 2013). In this study, the lecturers and courses involved are from the Mathematics Programme which consists of Applied Mathematics and Financial Mathematics from the Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu.

Methods

The methods for this study are divided into data collection, random allocation of courses among lecturers using Microsoft Excel, allocation of courses among lecturers depending on the lecturers' years of teaching experience using OpenSolver and allocation of courses among lecturers depending on years of teaching experience using Python. The objective of this allocation is to maximize the years of teaching experience of lecturers in teaching the courses. This is based on the assumption that the longer a teacher teaches a certain subject, the better he gets in delivering that subject to his students. The solutions' objective function and computational time obtained from random allocation and allocation of courses among lecturers using OpenSolver and Python are then compared to show the difference in terms of efficiency and effectiveness of the allocation made.

There are some rules for this study on the allocation of courses among lecturers. The rules include:

1. The allocation of courses among lecturers must satisfy the minimum and maximum number of courses to be allocated to the lecturers.

2. The number of lecturers allocated to a course must be equal to the number of classes of that course. The number of classes is determined every semester by the Mathematics department.

3. Each lecturer cannot be allocated to the same course again if he or she has been allocated once so that the occurrence of a lecturer teaching all the classes for the same course can be avoided.

Data Collection

The data such as the total number of lecturers, total number of courses and total number of classes for each course, and also the minimum number of courses to be taught by each lecturer are obtained by interviewing the Head of Department (HOD) of Mathematics Programme in the Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu. Furthermore, the data on lecturers' years of experience in teaching the courses offered in the Mathematics Programme are collected using online questionnaire via Google Form.

There are 25 core courses with 63 classes, 11 elective courses with 11 classes and eight service courses with 14 classes. A total of 28 lecturers are involved in this allocation. The minimum number of courses to be taught by the lecturers in a year is three courses per person excluding the lecturers who have administrative posts. The lecturers who are involved in administrative jobs are required to be allocated to at least two courses per person in a year. The maximum number of courses each lecturer can be allocated in a year is not being set by the department, therefore the addition of a course for each lecturer from his or her minimum number of courses to be allocated would be considered as the maximum number of courses that can be allocated. For instance, a lecturer who teaches at least three courses per year would have a maximum of four courses that he or she can teach in a year.

Random Allocation of Courses among Lecturers Using Microsoft Excel

Random allocation is a fair allocation without considering any prior allocation factors that involved, in which it ensures that each course has the same opportunity to be assigned to any lecturer. Random allocation is done by using Microsoft Excel. There is a list of 44 courses with 88 classes to be assigned to 28 lecturers. RAND function is used to obtain a random number for each class. RANDgenerate numbers between 0 and 1.

Since RAND is a volatile function, therefore it will recalculate every time the worksheet changed. To avoid the changing of random numbers generated, the formula is converted to values by using paste special. Next, the RANK function is used to rank each class according to the random numbers generated. The rank is then divided by the number of classes with different courses each lecturer will be assigned to. The values obtained through the division between rank and the number of classes with different courses each lecturer need to be assigned are then round up using CEILING with multiple of 1. The values obtained will be between 1 to 28. Each lecturer would probably be assigned with two, three or four classes of different courses depending on lecturer's minimum and maximum number of courses per year.

Course Allocation among Lecturers Using OpenSolver

Course allocation among lecturers in this study is a problem of zero-one integer programming (Hillier & Lieberman, 2010 & Joshi & Moudgalya, 2004) in which each variable can only take the value of 1 (yes) or 0 (no). The allocation of courses among lecturers is a maximization problem in which it maximizes the teaching experience of lecturers. The mathematical model to solve the problem of courses allocation among lecturers using OpenSolver is:

$$\text{Maximize } Z = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\text{Subject to: } \sum_{i \in I} x_{ij} = b_i \text{ , for } j \in J$$

$$d_j^{min} \leq \sum_{j \in J} x_{ij} \leq d_j^{max}, \text{ for } i \in I$$

$$j_i \leq b_i$$

All variables must be 0 or 1.

Where I is all the courses offered, i

J is all the lecturers involved, j

x_{ij} is the decision variable,

b_i is the number of classes for course i ,

c_{ij} is the years of experience of lecturer in teaching course i,

d_j^{min} is the minimum number of courses to be allocated to lecturer j,

d_j^{max} is the maximum number of courses to be allocated to lecturer j,

The decision variable is:

$$x_{ij} = \begin{cases} 1 \text{ if course } i \text{ is allocated to lecturer } j \\ 0, \text{other} \end{cases}$$

The objective function of this model is the sum of the decision variable x_{ij} and years of experience c_{ij} for all courses i and lecturers j. This zero-one integer programming problem of allocation of courses among lecturers is then solved by using OpenSolver where it is a free, open source software that consists of COIN-OR CBC optimization engine which can help to solve Linear and Integer problems regardless of the size of the problem.

Course Allocation among Lecturers Using Python

Python programming language is a high-level programming language that is powerful to reflect people's way of thinking and their way of code implementation (Bogdanchikov *et al.,* 2013). Besides that, Python programming language is easier to read and also to write compared to other programming languages because it reduces the usage of extra keywords needed to write syntactically an error-free programme.

Besides that, Python has a lot of libraries. One of the libraries that is free, open and has the capabilities in calling GLPK, COIN CLP/ CBC, CPLEX and GUROBI to solve LP problems apart from developing MPS or LP files is called PuLP. PuLP and another free open-source library written in Python called Pandas has the capabilities of manipulating, analyzing and cleaning the data that will be used for this course allocation problem.

The data of the lecturers' years of teaching experience for each course, the minimum and maximum number of courses each lecturer need to be assigned to and the number of classes of each course are saved in an Excel comma-separated values (CSV) file. The Excel CSV file is imported into Python using Pandas and the process of slicing is carried out to select and save the sets of DataFrame needed as variables. The aim of this allocation problem is to maximize lecturers' years of teaching experience, therefore LpMaximize is used. Besides that, LpBinary is used to ensure all decision variables are binary. Only two symbols, "0" and "1" are used. The objective function and the constraints of course allocation, among lecturers, are then built up. Finally, this allocation problem is solved using model.solve and pulp.LpStatus [model.status].

Results and Discussion

The solutions obtained for course allocation among lecturers using MS Excel, OpenSolver and Python are shown in Table 1.

Table 1: Allocation of courses among lecturers using MS Excel, OpenSolver and Python

No.	Lecturer	Random Solution-MS Excel	Exact Solution-OpenSolver	Exact Solution-Python
1	Prof. Dr. Lazim	Real Analysis	Algebra	Algebra
		Mathematical Fluid Mechanics	Statistics	Statistics
		Engineering Statistics	Biostatistics	Biostatistics
2	Prof. Dr. Fatah	Linear Algebra	Calculus	Calculus
		Biostatistics	Topology	Topology
			Fuzzy Set Theory	Fuzzy Set Theory
3	Dr. Safiih	Algebra	Statistics	Statistics
		Computer Programming Language	Advanced Statistics	Financial Econometrics
		Modern Algebra	Introduction to Statistics	Advanced Statistics
		Optimization		Introduction to Statistics
4	Dr. Norizan	Calculus	Numerical Analysis	Advanced Calculus
		Advanced Statistics	Financial Forecasting	Financial Forecasting
		Linear Programing	Statistics for Chemistry	Statistics for Chemistry
5	Dr. Roslan	Stochastic Modelling	Real Analysis	Real Analysis
		Operational Research	Complex Variable Calculus	Complex Variable Calculus
		Engineering Mathematics	Graph Theory	Graph Theory

6	Dr. Zabidin	Statistics	Modern Algebra	Modern Algebra
		Graph Theory	Real Analysis	Real Analysis
		Biostatistics	Functional Analysis	Functional Analysis
7	Dr. Termimi	Advanced Calculus	Algebra	Algebra
		Numerical Analysis	Linear Algebra	Linear Algebra
		Real Analysis	Geometric Modelling	Advanced Statistics
8	Dr. Auni	Numerical Analysis	Algebra	Differential Equation
		Statistics	Differential Equation	Algebra
		Vector Calculus	Partial Differential Equation	Partial Differential Equation
9	Dr. Azwani	Investment Mathematics	Calculus	Calculus
		Optimization	Advanced Calculus	Advanced Calculus
		Advanced Numerical Analysis	Numerical Analysis	Numerical Analysis
10	Dr. Binyamin	Algebra	Linear Algebra	Linear Algebra
		Differential Equation	Optimization	Optimization
		Statistics for Chemistry	Financial Optimization	Financial Optimization
11	Dr. Che Imran	Advanced Calculus	Investment Mathematics	Investment Mathematics
		Geometric Modelling	Mathematical Theory in Insurance	Mathematical Theory in Insurance
		Statistics for Chemistry	Financial Risk Analysis	Financial Risk Analysis
12	Dr. Chee	Statistics	Statistics	Statistics
		Financial Econometric	Biostatistics	Biostatistics
		Partial Differential Equation	Statistics for Chemistry	Statistics for Chemistry
13	Dr. Fatimah	Calculus	Advanced Calculus	Advanced Calculus
		Computer Programming Language	Differential Equation	Differential Equation
		Engineering Mathematics	Mathematical Methods	Mathematical Methods
			Engineering Mathematics	Engineering Mathematics
14	Dr. Hanafi	Advanced Statistics	Statistics	Statistics
		Probability Theory in Finance	Advanced Statistics	Advanced Statistics
		Numerical Analysis	Stochastic Modelling	Stochastic Modelling
15	Dr. Hassilah	Calculus	Probability Theory in Finance	Probability Theory in Finance
		Linear Algebra	Investment Mathematics	Investment Mathematics
		Biostatistics	Introduction to Stochastic Differential Equation	Introduction to Stochastic Differential Equation

16	Dr. Ilyani	Computer Programming	Linear Algebra	Linear Algebra
		Linear Algebra	Differential Equation	Differential Equation
		Mathematics for Chemistry	Mathematics Fluid Mechanics	Mathematics Fluid Mechanics
17	Dr. Maharani	Linear Algebra	Advanced Statistics	Stochastic Modelling
		Mathematical Methods	Financial Derivatives	Advanced Numerical Analysis
		Financial Derivatives	Advanced Numerical Analysis	Numerical Modelling in Finance
18	Dr. Mohd. Nazri	Computer Programming Language	Real Analysis	Linear Algebra
		Introduction to Statistics Differential Equation	Financial Econometrics	Real Analysis
		Introduction to Statistics	Mathematics	Mathematics
19	Dr. Saifullah	Calculus	Computer Programming Language	Computer Programming Language
		Differential Equation	Linear Programming	Linear Programming
		Mathematical Theory in Insurance	Operational Research	Operational Research
			Optimization	Optimization
20	Dr. Aidya	Advanced Statistics	Calculus	Calculus
		Operational Research	Linear Programming	Linear Programming
		Numerical Modelling in Finance	Operational Research	Operational Research
			Mathematical for Chemistry	Mathematical for Chemistry
21	Dr. Fadhilah	Linear Programming	Linear Programming	Linear Programming
		Advanced Calculus	Operational Research	Operational Research
		Optimization	Optimization	Optimization
		Mathematics		
22	Dr. Ruwaidiah	Advanced Calculus	Advanced Calculus	Advanced Calculus
		Real Analysis	Vector Calculus	Vector Calculus
		Functional Analysis	Introduction to Stochastic Differential Equation	Introduction to Stochastic Differential Equation
23	Dr. Shalela	Algebra	Numerical Analysis	Numerical Analysis
		Probability Theory in Finance	Introduction to Calculus	Introduction to Calculus
		Topology	Mathematics for Chemistry	Mathematics for Chemistry

24	Dr. Syerrina	Advance Statistics	Computer Programming Language	Advanced Statistics
		Stochastic Modelling	Advanced Calculus	Numerical Analysis
		Operational Research	Advanced Statistics	Financial Derivatives
25	Dr. Ummu	Differential Equation	Calculus	Calculus
		Introduction to Stochastic Differential Equation	Computer Programming Language	Computer Programming Language
		Financial Optimization	Linear Algebra	Probability Theory in Finance
			Probability Theory in Finance	
26	Dr. Baini	Algebra	Algebra	Computer Programming Language
		Statistics	Numerical Modelling in Finance	Financial Derivatives
		Financial Forecasting	Biostatistics	Biostatistics
27	Pn. Azlida	Linear Programming	Computer Programming Language	Algebra
		Investment Mathematics	Stochastic Modelling	Computer Programming Language
		Fuzzy Set Theory	Financial Derivatives	Geometric Modelling
28	Pn. Siti Madhihah	Financial Derivatives	Engineering Mathematics	Engineering Mathematics
		Introduction to Calculus	Engineering Statistics	Engineering Statistics
		Biostatistics	Biostatistics	Biostatistics
		Mathematics for Chemistry		

From the results, the comparison of the objective function and the computational time in obtaining the solution for course allocation among lecturers using three different methods is shown in Table 2. The methods include random allocation, allocation of courses among lecturers using OpenSolver and allocation of courses using Python.

Table 2: Comparison of objective function and computational time

Methods	**Random**	**OpenSolver**	**Python**
Objective Function	64	419	419
Computational Time	1 minute 31 second	4 minute 19 second	9 second

From Table 2, the objective function of the solution obtained through random allocation is the least which is 64 whereas the objective function obtained using OpenSolver and Python is the most which is 419. This shows that the problem of course allocation among lecturers can be solved in the most effective way using OpenSolver and Python because the solutions obtained have successfully produced a desired or intended result which is to maximize

lecturers' years of experience in teaching the courses. Conversely, random allocation is not an effective approach to solve course allocation among lecturers because the allocation made did not assign the correct courses to the most suitable lecturer.

The computational time required to obtain the solution for course allocation problem is the least when the allocation is made using Python (9 seconds) followed by random allocation (1 minute and 31 seconds) and finally using OpenSolver (4 minutes and 19 seconds). Python proved to be the most efficient way to solve the problem of course allocation among lecturers because it performs the fastest compared to random allocation and allocation of courses among lecturers using OpenSolver.

The computational time is actually dependent on the size of the problem. If the size of the problem is small, then the computational time required is less and vice versa. Since the allocation made using Python requires the least computational time followed by random allocation and lastly OpenSolver, this sequence of efficiency level will remain the same regardless of the occurrence of data changes. For instance, a course is removed or an addition of new lecturer will not change the fact that Python is the most efficient approach to perform the allocation of courses as it can automatically read the changes of data in the CSV file into Python.

Since the objective of this study is to identify the most effective and efficient method that can help administrators to allocate courses to lecturers automatically where the lecturers' years of experience are optimized, therefore both aspects which include the effectiveness and efficiency of the allocation made are taken into account. Course allocation among lecturers using Python is the most effective and efficient approach to assign the most suitable lecturers who are experts to teach the courses based on their years of experience in teaching the courses.

Conclusion

In this study, one of the objectives is to identify the most effective and efficient method that can help administrators automatically allocate courses to lecturers. Therefore, three methods are used to solve the problem of course allocation among lecturers of the Mathematics Programme in the Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu. The methods consist of random allocation using MS Excel, allocation of courses among lecturers using OpenSolver and allocation of courses among lecturers using Python. Both process of allocating courses to lecturers using OpenSolver and Python is based on LP approach where the solutions obtained are exact solutions. The objective functions and computational time needed for the process of allocating courses to lecturers through the three mentioned approaches are also being compared. It is concluded that Python is the most effective and efficient method for administrators to automatically allocate courses to lecturers. The reason is that Python requires the least time to obtain the desired solution.

Moreover, this study is also aimed to optimize the allocation of courses among lecturers by maximizing lecturers' years of teaching experience. An online questionnaire was conducted to obtain the years of teaching experience of each lecturer for each course. All solutions obtained through these three approaches are feasible where all constraints are satisfied. The constraints include the minimum and maximum number of courses to be assigned to each lecturer. Besides that, the number of lecturers assigned to each course is ensured that it is identical to the number classes required for that particular course. The number of classes for each course is determined by the respective departments at UMT, and in this case, the Mathematics department. Finally, each lecturer

is not assigned to the same course again when he or she has been assigned once. The results obtained showed that the objective function of the allocation made using OpenSolver and Python is the most which is 419. This indicates that the courses would be assigned and taught by lecturers that have the most experience.

Although the objective function of the solution obtained using OpenSolver and Python is the maximum, the computational time of course allocation using OpenSolver is longer compared to Python. Therefore, Python would be recommended to administrators to perform the process of course allocation. This is because Python allows easier reallocation whenever there are changes in the data such as the number of lecturers available and the number of courses offered.

Despite the benefit of using Python over random allocation using MS Excel or OpenSolver, there is still a drawback when using it. When one needs to develop a new programme to solve a specific problem, writing the initial Python code will take the most time compared to other approaches. However, if the code written needs to be re-used again, then this can be done within a shorter time frame.

For future studies one can improve the current model by including more factors in the model, such as the teaching performance, research interest, or even considering students' preferences.

References

Amuji, H. O., Ugwuanyim, G. U., Ogbonna, C. J., Iwu, H. C. & Okechukwu, B. N. (2017). The usefulness of dynamic programming in course allocation in the Nigerian Universities. *Open Journal of Optimization.* 176-186.

Badri, M. A. (1996). A two-stage multiobjective scheduling model for [faculty-course-time] assignment, *European Journal of Operational Research.* 16-28.

Bogdanchikov, A., Zhaparov, M., & Suliyev, R. (2013). Python to learn programming. *Journal of Physics: Conference series 423 (2013) 012027*. doi: 10.1088/1742-6596/423/1/012027.

Emeka, A. G. (2018). An algorithm for efficient allocation of courses to lecturers for effective teaching. *International Journal of Engineering Science Invention (IJESI), 7*(9): 68-74.

Gunawan, A., & Ng, K. M. (2011). Solving the teacher assignment problem by two metaheuristics. *International Journal of Information and Management Sciences*. 73-86.

Helen, F. L. (2013). Why experienced teachers are important-and what can be done to develop them. *Scholars Strategy Network*. Retrieved from https://scholars.org/contribution/why-experienced-teachers-are-important-and-what-can-be-done-develop-them.

Hillier, F. S., & Lieberman, G. J. (2010). *Introduction to operational research.* Boston: McGraw-Hill.

Joshi, M. C., & Moudgalya, K. M. (2004). *Optimization: theory and practice.* United Kingdom, UK: Alpha Science International Ltd.

Shohaimay, F., Dasman, A., & Suparlan, A. (2016). Teaching load allocation using linear programming. *Business Management and Computing Research Colloquium (BMCRC).*