



Group Student-Project Allocation Using Constraint Optimization Techniques

L. Yahaya*, A. Y. Tambuwal and Z. Sani

Department of Mathematics, Usmanu Danfodiyo University, P.M.B. 2346 Sokoto, Nigeria

*Corresponding Author: E-Mail: yahaya.lmd@gmail.com, Mobile Phone: +2348064431262

Received: August, 2016

Revised and Accepted: November, 2016

Abstract

The allocation of projects to final year students in various disciplines is a major challenge to many universities in Nigeria and elsewhere. In most cases manual method has been the pre-dominant for this activity. However, many problems arise which among other things include project duplication and delay in project activities. Other works concentrated on project allocation with both students and supervisors preferences, which may not be suitable for departments with small number of academics. In order to minimize the chances for these problems to occur, this study presents a constraint optimization approach to solving the student project allocation problem for the Computer Science programme of the Department of Mathematics, Usmanu Danfodiyo University Sokoto. The problem was modelled using constraint optimization (COP) techniques and its prototype was implemented on a Java platform which concentrated largely on students preferences on projects only, to ensure that a maximum number of students got their first choice projects. Upon testing the model, the results obtained showed that 85 out of the 95 students (90%) got allocation in the first round, from which 67 students (73%) got their first choices from the 24 projects supplied by 12 supervisors, which showed maximum allocation of students' best choices.

Keywords: project allocation; constraint; optimization; group projects.

1.0 INTRODUCTION

In universities, students undertake an individual or group final year projects every year as a requirement for their undergraduate programmes. The final-year project is an academic activity that engages a student in a research work in a certain field of study; which may be graded after successful completion and it carries certain percentage from the student's overall scores in the final semester. However, the benefits of this activity are to offer a bridge to independent learning for students and test their ability to work within a limited time constraint. This activity is entirely carried out under supervision of a lecturer (supervisor) assigned to undertake the responsibility. At completion, a student may be examined by his supervisor, a panel of

examiners during oral presentation and an external examiner. The examining procedure varies from one university/department to another and likewise the mode of undertaking a final year project. Some departments or universities allow students to carry out a group work while some allow for individual project only. Meanwhile, the major challenge of this activity is the allocation procedure.

The project allocation is a generic problem that is common to almost all universities particularly in Nigeria. Sometimes, students tend to duplicate projects due to inefficient allocation procedure. Despite the fact that several works on projects allocation have been done, there is no generic method of tackling the problem. The allocation process also varies from one university or department to another.

Some of the existing models dwell entirely on automation without given any specific allocation algorithm. Moreover, several methods (Kazakov, 2001; Anwar and Bahaj, 2003; Abraham, 2003) of projects allocation were proposed to solve the problem. Though, some (Abraham *et al.*, 2006; Manlove and O'Manley, 2008; Moussa and Abu El-Atta, 2011) of them provide reasonable solutions but still have some deficiencies (such as randomness, biasness and limited application). One common aspect to most of the models is that list of projects topics are supplied by supervisors from which students make their choices. Upon all, the allocation procedure ranges from manual paper work, to internet-based.

The internet or web-based student-project allocation method involves activities within the confines of internet connection. Lists of projects are submitted by supervisors through the web, students can view the project lists via the same medium and make their choices. The work of Ademola *et al.* (2013) is an example of this approach.

Based on the manual aspect, some universities or departments ask each student to get a suitable topic of his/her choice, seek the consent of his/her supervisor and then proceed with the work, which might eventually leads to projects duplication, as students find their own project topics and meet their supervisors independently. Some universities or departments adopt the same method but conduct proposal defence. Some universities try to avoid such problems by asking each student to submit a list of (three or more) project choices in the order of their preference, after which the project coordinator assigns project topics to all students manually with the aim of satisfying as many students as possible. This procedure is also a complex task for a human.

Meanwhile, the underlying problem is one of the constraint satisfaction (CSP) and constraint optimization (COP) problems to which a variety of computational techniques are applicable. Project allocation is a multi-constraint problem that is similar to timetabling and scheduling problems. According to Almu and Muhammad (2013), such problems are mainly classified as constraint satisfaction problems (CSPs) in nature where the main goal is to satisfy all problem constraints of the domain in question.

The work described in this research paper explored the constraint optimization techniques to formalize the student project allocation problem with a specific attention to allocating project topics to students in groups taking into account their preferences over projects they choose.

2.0 PREVIOUS WORKS

Several models were proposed to improve on the existing methods especially the manual approaches. The models presented are sufficiently generic to be applied to many other situations. Some of the models include the student-project allocation with students preferences over projects and supervisors preferences over students (SPA) presented by Abraham (2003); Abraham *et al* (2006). With this approach, students have preferences over the projects and their supervisors have preferences over them (students) based on their academic performance. This means students who performed poorly in their previous session(s) may not be allocated to supervisors. The next model was student-project allocation with preferences over projects by both students and supervisors (SPA-p) presented by Manlove and O'Manley (2008). Supervisors would check through the list to see those students that selected projects they offered and select students possibly based on their research areas. By this method, it will be difficult to have a best allocation. Moussa and Abu El-Atta (2011) presented a model called student-project allocation with students' preferences over projects and supervisors preferences over pairs, i.e. both projects and student (SPA-(s, p)). Both students and supervisors have preferences over projects, while supervisors have preferences over students too. The preference by supervisors over projects may deprive some of the students from being allocated their preferred project choices. Pudaruth *et al.* (2013) presented a framework for the allocation of final year projects for the Computer Science and Engineering Department at the University of Mauritius. The system allocates projects to students in order to maximise the number of students who gets their first choices in their preference list and to keep the load of supervisors and examiners within a reasonable range. The result obtained when the system was deployed in the department is that the percentage of students who obtained their first choice was 82% on 30 projects proposed by 15

supervisors for 11 teams. This model is based on the fact that supervisors are allocated to supervise only projects for which they have shown interest. The major concern of this approach is that projects are not supplied by supervisors; there would be possibility for students who chose projects that are not related to the research area of any of the supervisors to be left unassigned to any supervisor. Kwanashie *et al.* (2014) proposed a model for students' projects allocation problem based on individual or group projects offered by a supervisor. In this context, they modelled the SPA as a network flow problem and applied a GREEDY-MAX-SPA algorithm to obtain maximum optimal matching or assignments (i. e. to maximize the number of students that are assigned to the higher ranked projects). In addition, this approach allows for supervisor's minimum and maximum number of students he/she can supervise. In addition to the problems associated with supervisors' preferences, the application of these models would be limited to universities with adequate number of academics only. Because when such preferences exist in grossly understaffed universities, as the case may be in most Nigerian universities, it would not guarantee that all students would get supervisor allocation. Therefore, there is a need to come up with an approach that minimizes supervisors' overhead associated with presenting large number project lists for students to make choices.

3.0 GROUP PROJECT ALLOCATION

In our model termed Group Student-Project Allocation Tool (GSPAT), supervisors respectively submit their lists of projects through the project coordinator and students respectively make and submit back their lists of projects choices. Therefore, every student can be allocated any of the project(s) he/she submitted. One important difference between this model and the other models is that all students work in groups based on the project capacity constraints. This approach is applicable to situations where there are insufficient academics, as large number of students is clustered in groups to work on few projects.

Projects allocation problems are multi-constrained optimization problems that have huge search space. Their complexity increases with respect to the number of students

involved. These types of problems are well known as NP-hard (Adamuthe and Bichkar, 2012), their complexity exponentially increases with respect to size of problem instance. Constraint optimization techniques are relevant to dealing with such problems.

3.1 Constraint Optimization Techniques

The term constraint optimization problem (COP) is a technique used in which the whole problem consists a set V of variables defined by a corresponding set D of possible values (the variable's domain) and a set C of constraints and then plus an objective function. The task is to find an optimal solution with respect to the objective function (Apt and Wallace, 2007). The objective or cost function has to be maximized or minimized during solution process depending upon the nature of the problem. Problems of this nature are usually solved by backtracking approaches.

Backtracking, first named by D. H. Lehmer, is a well-known algorithm design technique for the solution of a class of problems that deal with searching for a set of feasible solutions or an optimal solution satisfying some constraints (Horowitz *et al.*, 2008).

3.2 Problem Formulation

We now seek to allocate project topics to students in groups taking into account their preferences on projects, to maximize the number of students that would be allocated their first project choices and finally to minimize the number of projects to be carried out. Therefore, we employed the COP approach to formulate the problem as shown in figure 3.1:

- i. $S = \{s_1, s_2, s_3, \dots, s_n\}$ is the set of final-year students.
- ii. $P = \{p_1, p_2, p_3, \dots, p_m\}$ is the set of submitted project choices by the students respectively. Where n and m are positive integers, and $n \leq m$.
- iii. $C = \{c_1, c_2, c_3, \dots, c_a\}$ is the set of constraints to be satisfied during projects allocation.
- iv. $L = \{l_1, l_2, l_3, \dots, l_b\}$ is the set of supervisors supplying project topics. Where b is a positive integer and $b < n$.
- v. $f(s_1, s_2, s_3, \dots, s_n)$ is the objective function for maximising students' best choices.
- vi. T_s stands for the total number of students who got their first choice projects allocated.
- vii. $X_{s,p}$ stands for the binary decision variable for s th student and p th project.
- viii. $Y_{s,p}$ stands for the binary decision variable that assumes the value of 1 if p th project is selected by s th student and 0 otherwise.

Figure 3.1: The GSPAT Model Formulation.

The model of Figure 3.1 was adopted from the work of Almu and Muhammad (2013) for lectures timetable scheduling (a CSP). Our model (a COP) includes an objective function plus some modifications in order to suit the project allocation problem. The solution to this formulation is an assignment of projects to students and students to their supervisors at the same time maximizing students' best project choices allocation thereby taking into account all constraints. Therefore, a student can be allocated any project (one) supplied by any supervisor.

$$1, \text{ if student } s \text{ is allocated to project } p \quad X_{s,p} = 1 \quad (1)$$

Where $X_{s,p}$ is a decision variable for the s th student and the p th project, s represents a student and can take a value from 1 up to S , S is the total number of final-year students for a given session; p represents a project and can take a value from 1 up to P , P is the total number of projects supplied.

Now, looking at the objective function, it is to maximize the number of students to be given their first choice projects. This can also be formulated as follows:

$$\text{MAX. } f = T_s \quad (2)$$

3.3 Constraints

The constraints are usually categorized into hard and soft constraints. Set of hard constraints demands the highest priority to be processed while maximum soft constraints satisfaction shoots up the overall quality of solution (Shaikh *et al*, 2011).

Hard constraints are those that must be satisfied. Violation of these constraints (also called as conflicts) will cause the solution to be infeasible (or inconsistent). For example, allocating more than one project to the same student is a violation of a hard constraint. The following hard constraints are considered for this work:

- i. Each student will be allocated to at most one project.

$$\sum_{p=1}^P X_{s,p} = 1 \quad \forall s \in S \quad (3)$$

- ii. Each project topic can be allocated to at most five (5) students.

$$\sum_{s=1}^S X_{s,p} \leq 5 \quad \forall p \in P \quad (4)$$

Soft constraints are those that are desirable in order to produce a good quality assignment (solution) but violations (if they exist) are allowed to satisfy the hard constraints. Because violating such constraints does not

cause the solution to be inconsistent. The soft constraint here is that each student can be allocated to project only from the list(s) he chooses.

$$\sum_{p=1}^P (X_{s,p} \times Y_{s,p}) = 1 \quad \forall s \in S \quad (5)$$

Where

$$Y_{s,p} = \begin{cases} 1, & \text{if project } p \text{ is one of the } s \text{ choices} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

4.0 IMPLEMENTATION

The implementation was carried out on a laptop computer with a hard disk capacity of approximately 500GB, installed memory (RAM) of 4.00GB, Intel(R) microprocessor of 2.40GHz, Java 7 programming language (jdk 1.7 version) and a Java IDE (JCreator LE 4.50 version). The Java was chosen because of its advantages over other programming languages which include easy learning, platform independent, security, robustness, object-oriented and multithreaded features.

The implementation of this model involved coding/programming the various components with the Java programming language on the JCreator IDE.

5.0 TESTING

The testing of this prototype was carried out on the platform with the hardware and software configuration mentioned in section 4 above. Valid dataset comprising 95 students and 12 supervisors with 108 project topics (for 2014/2015 academic session) were obtained from the Project Coordinator, Computer Unit of the Department of Mathematics, Usmanu Danfodiyo University Sokoto. Each supervisor supplied 2 projects (reducing the total number to 24) and random choices were made. The students' choices were fed to the system through the GUI. Upon entering all the required data correctly, the system executes

and displays the result which comprises of three components: they are the students' choices, the projects and the supervisor allocations. The result obtained shows maximum number of students was allocated their first project choices.

The realized prototype has a graphical user interface (GUI) through which it accepts the total number of projects to be involved in the allocation process, total number of supervisors to be involved in the project supervision activities, the total number of students intending to undertake an individual final year project. Other requests include the maximum number of projects that a student could choose, the maximum number of projects supplied by each supervisor, which are also to be entered through the same field. The maximum number of projects supplied by a supervisor is his capacity constraint, i.e. the supervisor's upper bound. The supervisor's load would depend upon how many students were allocated his projects by the system. The requests also include project capacity (i.e. how many students are to take a project), which are five for each of the projects since it a group projects. Moreover, It would also prompt the user with a request for supervisors' capacity (i.e. the maximum number of students that each supervisor is willing to supervise).

Then, there is a space for entering all project lists for all supervisors respectively. They should be entered as they are ordered by each supervisor. This would be followed by respective ordered students' project choices in another separate field.

6.0 RESULTS

After entering all the required information correctly, there is a PROCESS button on the GUI that when clicked the system would execute and produce the project and supervisor allocations, which are the expected results. The screenshot of Figure 6.1 displays the data entry and the results obtained on the system GUI:

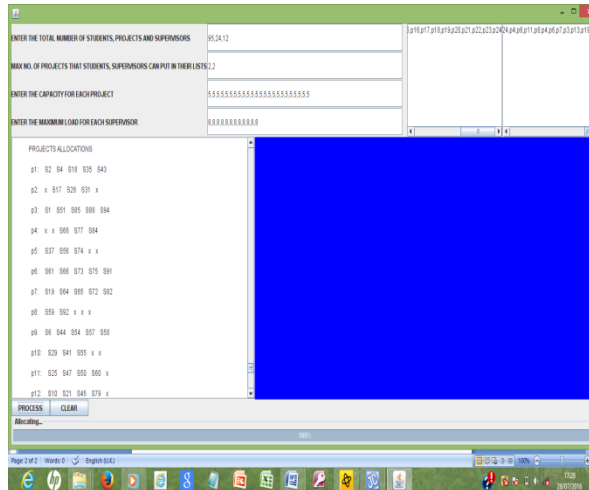


Figure 6.1: A Screenshot showing the Data Entry and the Project Allocations.

The result shows that only 10 students were not allocated to any of the projects. This was because the system works based on first come first serve basis and it gives priority to first choice projects. However, all the 24 projects were allocated, as shown in table 6.1 below.

Table 6.1: Overall Student-Project Allocations.

Total No. of Students	Allocated Students	Percentage (%)	Total No. of Projects	Allocated Projects
95	85	90%	24	24

Each of the projects was allocated to at least one student and at most five students. There are variations on the number of students allocated due to competition on some popular project titles. This can be shown as follows:

Table 6.2: Group Student-Project Allocations

Projects	No. of Students
P1	5
P2	3
P3	5
P4	3
P5	3
P6	5
P7	5
P8	2
P9	5
P10	3
P11	4
P12	4

P13	5
P14	1
P15	1
P16	2
P17	5
P18	3
P19	5
P20	3
P21	3
P22	2
P23	3
P24	5
TOTAL	85

The projects were supplied by 12 lecturers in which each of them supplied 2. Therefore, each lecturer was allocated to students who made their choices from his project list. The maximum number of students that each lecturer can supervise was limited to 8 students from two project topics, as shown in table 6.3 below.

Table 6.3: Student-Supervisor Allocations.

Lecturer	No. of Allocated Students
L1	8
L2	8
L3	8
L4	7
L5	8
L6	8
L7	6
L8	3
L9	8
L10	8
L11	5
L12	8
TOTAL	85

7.0 CONCLUSION AND FUTURE WORK

Project allocation is a generic problem that is common to almost all universities particularly in Nigeria. Sometimes, students tend to duplicate projects due to inefficient projects allocation process. Despite the fact that several works on projects allocation have been done, there is no generic method of tackling the problem. This paper presents a constraint optimization approach to develop a prototype using Java platform that would optimally solve project allocation problems. When tested with

95 students, 24 projects supplied by 12 lecturers, 85 students got project allocation which is approximately 90%. Students were optimally allocated to supervisors as shown in table 6.3. The applications of this model need not be limited to assigning students to projects only but may extend to other scenarios, such as the assignment of employees to posts in an organization or institution where available posts are offered by various departments. The major contributions of this research are as follows:

- i. Group work minimizes the number of projects involved in the allocation process thereby decreasing the size of the problem instance.
- ii. Eliminating supervisors' preferences on either students or project topics makes it flexible to be useful in academic institutions with insufficient academics (especially in Nigeria).
- iii. When project topics are supplied by supervisors (not by students), the chance of students to bring in project topics that were already done from other universities is blocked.

Next moment we focus on implementing this prototype using Chocó Solver or any other constraint programming (CP) library, such as ILOG Solver. CP libraries have built-in facilities that provide substantial support for solving problems of this nature therefore; using any of the libraries mentioned would help to compare performance with the conventional programming languages such as the Java used.

8.0 REFERENCES

- Abraham, D. J. (2003). *Algorithmic of Two-Sided Matching Problems*. Master's Thesis, 12-89, University of Glasgow, Department of Computing Science.
- Abraham, D. J., Irving, R. W. and Manlove, D. F. (2003). The Student-Project Allocation Problem. Technical Report TR-2003-141 of the Computing Science Department of Glasgow University.
- Abraham, D. J., Irving, R. W., and Manlove, D. F. (2006). Two Algorithms for the Student-Project Allocation Problem. *Journal of discrete algorithms*, 29 (6) 474-484.
- Adamuthe, A. C. and Bichkar, R. S. (2012). Personnel Scheduling: Comparative Study of Backtracking Approaches and Genetic Algorithms. *International Journal of Computer Applications*, 38 (10)0975 – 8887.
- Ademola, A., Adewale, A., and Ike, D. U. (2013). Design and Development of a University Portal for the Management of Final Year Undergraduate Projects. *International Journal Of Engineering And Computer Science*, ISSN:2319-7242(2) 2911-2920.
- Almu, A. and Muhammad, A. B. (2013). Automating Lectures Timetable Using Constraint Satisfaction Technique. *Continental Journal of Applied Sciences*, 8 (3) 23-30.
- Anwar, A. A. and Bahaj, S. A. (2003). Student-Project Allocation Using Integer Programming. *IEEE Transactions on Education*, 46 (3), 359-367.
- Apt, K. R. and Wallace, M. (2007). *Constraint Logic Programming Using ECLiPSe* (2nd ed.), (pp 14-182). Cambridge: Cambridge University Press.
- Horowitz, E., Sahni, S., and Rajasekaran, S. (2008). *Fundamentals of Computer Algorithms*. Universities Press.
- Kazakov, D. (2001). *Coordination of Students Projects Allocation*. Computer Science Department, University of York.
- Kwanashie, A., Irving, R. W., Manlove, D. F., and Sing, C. T. S. (2014). *Profile-Based Optimal Matchings in the Student/Project Allocation Problem*. School of Computing Science, University of Glasgow.
- Manlove, D. F. and O'Manley, G. (2008). Student-Project Allocation with Preferences over Projects. *Journal of Discrete Algorithms*, 6 (4) 69-80.

Moussa, M. I. and Abu El-Atta, A. H. (2011). A Visual Implementation of Student Project Allocation. *International Journal of Computer Theory and Engineering*, 3(2) 1793-8201.

Pudaruth, S., Bhugowandeen, M., and Beepur, V. (2013). A Multi-Objective Approach for the Project Allocation Problem. *International Journal of Computer Applications*, 20(69), 0975 – 8887.

Shaikh, A. A., Bukhari, A. H., and Memon, Z. A. (2011). Solving University Scheduling Problem Using Hybrid Approach. *Mehran University Research Journal of Engineering & Technology*, 30(4)613-624.