Stable Matching Algorithm Approach to Resolving Institutional Projects Allocation and Distribution Optimization Problems

ISMAIL OLANIYI MURAINA

Department of Computer Science
Lagos State University of Education
Otto/Ijanikin Lagos
NIGERIA

Abstract: - The manual approach to resolving assignment issues involving the distribution or assignment of graduate students to lecturers for projects or thesis studies continues to be difficult and time-consuming for both the involved departments and the students. Even though it is evident that a student must be paired with a lecturer, there are significant problems with matching due to individual preferences, specializations, interests, prior student-teacher relationships, and other factors directly or indirectly related to the distribution process. This study uses a reliable matching technique to address issues with project allocation and distribution optimization problems, ten lecturers and ten graduate students participated. A matching algorithm is provided randomly, with students favouring a lecturer (although these roles are reversible). The lists of preferences for students and lecturers are input into the algorithm. Students and lecturers are split into two categories throughout the algorithm: those previously selected and those who have not yet been determined (free). Both instructors and students are initially accessible. The method chooses one student X randomly from the group of free students, provided that the group is not empty. Student X prefers a lecturer (let's say lecturer Y) who he rates among the top lecturers he has never previously picked. The Python programming language created and executed the stable matching algorithm. The stable matching algorithm makes the allocation faster, more accurate and timely compared to manual methods.

Key-Words: - Stable Machine, Algorithm, Optimization, Project Allocation, Optimization Problems, Matching Algorithm

Received: June 29, 2023. Revised: March 17, 2024. Accepted: May 21, 2024. Published: June 21, 2024.

1 Introduction

A With the help of technology, working in isolation is no more needed. Better decision is always made when people come in unison to share and combine ideas [11]. The distribution and allocation of students for project work should not be stressful. A student and institution should take project writing seriously because it enables students to develop a curious mind, always wanting to know why things happen the way they do; it also allows the student to be organised in his approach to solving the research problem, it causes the student to stop relying on rule-of-thumb but to be scientifically minded, and it also enables the student to be more organised with his work and do things in an orderly manner, It also helps pupils acquire a sense of proportion and the habit of tenaciously following goals to a successful conclusion, which is the realisation of results. Scheduling is one of the most important tools for resolving educational allocation/resource issues between teachers and students [2]. A project can take weeks, months, or even a semester. Writing projects necessitate a lecturer supervising one or more students per session (usually final-year students). The findings from scholars have established their concerns about institutions and the process of selecting students for particular courses of study[3]. Many students, particularly graduating students, find it inconvenient that they must do project work. They feel stressed by the cognitive weight of the educational system, and project work adds to it. Because the project is a requirement for students to graduate, they must complete it. For appropriate allocation and distribution, students and lecturers must allow for preferences; if students can prioritize the professor, they prefer to supervise them, and lecturers should be allowed to do the same. Appropriate allocation of projects will result in a high level of interest and understanding between students and professors during the project writing process. The stable matching method is one approach to solving student-lecturer allocation/distribution difficulties.

2 Related Literature

Using a stable matching algorithm has been acknowledged as a problem-solving strategy for various societal and organizational difficulties. Among the researchers are:

Many researchers have examined variants of the benchmark resource leveling and allocation optimization issues using various scheduling approaches, according to [4], because it makes scheduling techniques easier and faster. As educational projects get more complicated, multiple obstacles develop, such as appropriately deploying human and material resources or selecting alternative investments within a portfolio. Because answers based on intuition are risky due to the complexity, there is a search for less intuitive and more dependable approaches to handling educational challenges. An engineering approach to the problem may lead to operations research mathematical modeling to aid in timetabling solution discovery [5]. [6] also observed that the issue of class distribution among available teachers was time-consuming and attracted various constraints that must be met, such as preventing a teacher from teaching in two different locations at the same time and avoiding solutions which some teachers have more class hours than others.

In most cases, this is done manually, which is time-consuming and does not allow teachers to see different combinations of class assignments. Furthermore, it is frequently prone to errors. According to [7], an assignment problem is a particular type of linear programming that is a well-studied optimization problem in practically every job of life. The fundamental goal of using an optimization problem algorithm is to assign a given number of people to an equal number of positions on a one-to-one basis so that the overall cost of accomplishing that task is minimized or maximized for efficiency. According to [8], the essence of using an assignment problem is due to the fluctuating capacity of a human or machine to accomplish the assigned work or job. The use of classical Assignment Problems to solve real-world problems has some limitations. Stable Matching can undoubtedly help to alleviate these constraints. Over the years, certain scholars, such as [4], have employed a simple scheduling technique known as resource histogram to utilize the allocation and leveling of available resources. According to the findings, resource constraints substantially impact performance and lead the project to be extended beyond its scheduled term. [5] also employed a mathematical model (software) to solve a timetabling problem as valuable software to maximize the utilization of human resources. The software was built on a mathematical model that determined the timetable while limiting the number of students with conflicting schedules. [6] developed a decision support tool for assigning professors to university classrooms by appreciating the simple technique to address assignment challenges. The research comprised the creation of computer software that used object orientation notions to build a search technique called Beam Search, which investigates the combinatorial nature of the problem. The programming language utilized was Java, and the program included a graphical interface for inserting and manipulating data.

Similarly, [7] conducted a a study on the placement of the right individual for the correct position, which they found to be highly challenging due to ambiguity and imprecise information. To tackle the challenge, they adopted a fuzzy assignment problem approach. The findings obtained were compared to effectively apply the placement of the right candidate for the suitable job. Further research by [9] examined the criteria for assigning practice teachers to supervise prospective senior teachers during school experience and practice courses. The study's findings indicated some criteria for giving practice teachers, such as informal comments from students and faculty mentors, informal perspectives of school and university coordinators, and participant ideas about the assignment procedure. In 2021, [10] offered an optimal solution to a wide range of assignment problems, directly employing numerical illustration to determine the optimality of the model's output. The appealing model required simple arithmetic and logical computations to generate output.

The usual way of solving the general assignment problem by most institutions is to manually examine the full list of courses in some predefined order and for each course to a lecturer or lecturers of best-fitting [10]. In line with this, [11] studied departmental course distribution in a university system over a specific period. The distribution considered the limits imposed by lecturers' contact hours and teaching load per week, as well as individual lecturers' choices for courses offered in the department per semester. The goal was to create a framework to allocate methods to lecturers based on their choices optimally. The Gomory cutting plane algorithm was used to solve the course allocation problem, which was then implemented in MATLAB. Similarly, other scholars have used stable matching algorithms for similar allocation and distribution concerns [12], [8], [13]. For example, [12] provides a private stable matching algorithm based on the well-known Gale and Shapley method. Several independent parties, known as Matching Authorities, were used in the secret algorithm.

When Matching Authorities are truthful, the protocol correctly produces a stable match. It provides no information other than what can be learned from that match and the participants' preferences controlled by the adversary. The protocol's security and privacy were built on re-encryption mix networks and an additively homomorphic semantically secure public-key encryption technique. [8], like other academics, did research on the student-Project Allocation problem (SPA) to solve a generalization of the traditional Hospitals / Residents problem (HR). The study included a group of students, projects, and instructors. A lecturer is assigned a task, and both the lecturer and the project have capacity constraints. The results of stable Matching provided by the first algorithm are the best for all students, whereas the results produced by the second algorithm are the best for all instructors.

Furthermore, it was demonstrated that some structural insights involving the set of stable matchings in a given instance of SPA were quite general and could be applied to various settings other than student-project allocation. [13] employed stable Matching in economics to resolve conflicts of interest among selfish market agents in their study. Sub-problem agents might express their preferences over solution agents in the study, and vice versa. The Stable Matching Model (STM) produced a single solution for each sub-problem.

Similarly, [14] discovered that stable matching problems are the same as problems involving stable X-network setups. [12] said that the absence of any odd party is both a necessary and sufficient condition for the presence of a complete stable matching. A "stable partition," a new structure that generalized the concept of a whole stable matching, showing that every case of the stable roommate's dilemma has at least one such design. A stable partition also contained all of the odd parties. Finally, an O(n2) algorithm discovers one stable division, which yields all the uncommon parties. In the same vein, [12] examined stable matching problems from an algorithmic standpoint, with some problems derived from new stable matching models and others from existing stable matching models involving ties and incomplete lists, with additional natural constraints on the problem instance. The application of the stable matching algorithm in addressing matching-related issues has grown in popularity. [15] employed the stable matching algorithm to resolve difficulties in discovering a b-matching of the highest weight while observing lower quotas. They demonstrated in their study that obtaining such a maximum weight matching is an NP-complete task even if all students mark at most two projects and all projects have upper and lower quotas of three. Furthermore, it was demonstrated that if no upper percentage is greater than two, the issue is automatically polynomially tractable. In a general situation, an approximation approach that achieved the best possible ratio described in several forms was presented. Anytime we are crucial with time, accuracy, and error free of processing data, using machine learning tactic to handle the necessary procedures always prove encouraging and convenient than manual methods. The usual way of solving the general assignment problem by most institutions is to manually examine the full list of courses in some predefined order and for each course to find a corresponding shortlist of best-fitting lecturers, and then assign one of those lecturers to the course [16].

3 Methodology

The stable matching technique is used in this work to handle project allocation and distribution optimization problems. There were ten graduate students and ten lecturers engaged. A matching algorithm is provided at random, with students preferring a professor (these roles are reversible). The algorithm is fed lists of student and professor preferences as input. Students and professors are divided into two groups throughout the algorithm: those who have already been picked and those who have yet to be chosen (free). All students and lecturers are initially free. As long as the group of free students is not empty, the algorithm chooses one student X at random from the group of free students. Student X prefers a lecturer whom he ranks top among instructors whom he has never previously preferred (let us call him lecturer Y). The Python programming language was used to create and run the stable matching algorithm. The Gale-Shapley technique was modified to address the current issue (See Fig. 1).

```
Gale–Shapley algorithm to Find a Stable Matching
Initialize all students and lecturers to free
while there exists a free student s who still has a lecturer l to prefer for supervision
{
  l = s's highest ranked such lecturer to whom he has not yet preferred
  if lecturer is free
    (s, l) become engaged
  else some pair (s', l) already exists
    if l prefers s to s'
      (s, l) become engaged
      s' becomes free
    else
      (s', l) remain engaged
}
```

Fig. 1: Gale-Shapley Algorithm Adapted

4 Results

The hypothesized model for Matching was built in which all students were freely matched with all professors without regard for the students' or lecturers' preferences (See Fig. 2).

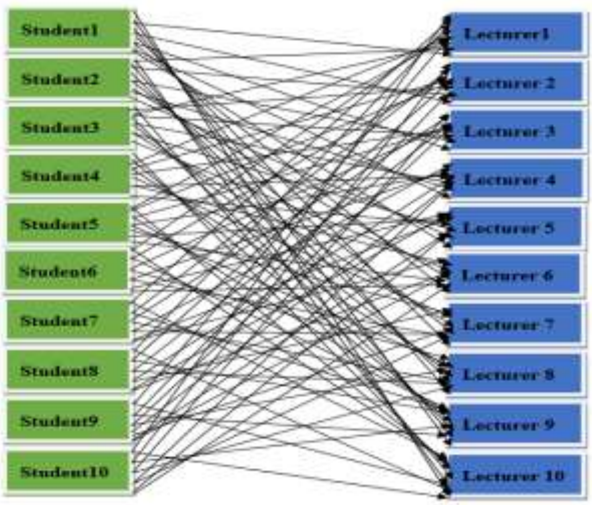


Figure 2: Hypothesized Model for Matching Students and Lecturers

In order to take preferences into account, the tables (Tables 1 and 2) indicate student and professor preferences.

Table 1: Table of Preferences for students

Stu	1st	2nd	3rd	4th	5th	6th	7st	8th	9th	10th

dents	Prefer	Prefer	Prefer	Prefer	Prefer	Prefer	Prefer	Prefer	Prefer	Prefer
Student1	Lecturer 1	Lecturer 2	Lecturer 3	Lecturer 4	Lecturer 5	Lecturer6	Lecturer 7	Lecturer 8	Lecturer9	Lecturer 10
Student2	Lecturer 2	Lecturer 3	Lecturer 5	Lecturer 1	Lecturer 4	Lecturer6	Lecturer 8	Lecturer 10	Lecturer9	Lecturer 7
Student3	Lecturer 7	Lecturer 2	Lecturer 3	Lecturer 1	Lecturer 10	Lecturer6	Lecturer 4	Lecturer 9	Lecturer8	Lecturer 5
Student4	Lecturer 4	Lecturer 3	Lecturer 8	Lecturer 9	Lecturer 5	Lecturer7	Lecturer 6	Lecturer 1	Lecturer2	Lecturer 10
Student5	Lecturer 2	Lecturer 3	Lecturer 1	Lecturer 5	Lecturer 4	Lecturer8	Lecturer 10	Lecturer 6	Lecturer7	Lecturer 9
Student6	Lecturer 1	Lecturer 3	Lecturer 2	Lecturer 10	Lecturer 9	Lecturer6	Lecturer 4	Lecturer 7	Lecturer8	Lecturer 5
Student7	Lecturer 2	Lecturer 6	Lecturer 10	Lecturer 5	Lecturer 4	Lecturer3	Lecturer 8	Lecturer 7	Lecturer9	Lecturer 1
Student8	Lecturer 4	Lecturer 6	Lecturer 8	Lecturer 3	Lecturer 1	Lecturer9	Lecturer 2	Lecturer 5	Lecturer7	Lecturer 10
Student9	Lecturer 3	Lecturer 10	Lecturer 7	Lecturer 6	Lecturer 1	Lecturer8	Lecturer 2	Lecturer 9	Lecturer4	Lecturer 5
Student10	Lecturer 10	Lecturer 8	Lecturer 6	Lecturer 2	Lecturer 1	Lecturer4	Lecturer 3	Lecturer 5	Lecturer7	Lecturer 9

According to the students' preferences table (Table 1), student one chooses lecturer 1 initially, and if lecturer 1 is unavailable, he prefers lecturer 2. It proceeds in this order to the final speaker he selects based on his preference - which could be based on interest, knowledge, area of specialization, or other instinctual reasons. Similarly, lecturers have preferences for the types of students they want to supervise based on previous relationships or observations of such lecturers during class activities (See Table 2).

Table 2: Table of Preferences for Lecturers

Students	1st Prefer	2nd Prefer	3rd Prefer	4th Prefer	5th Prefer	6th Prefer	7st Prefer	8th Prefer	9th Prefer	10th Prefer
Lecturer 1	Student1	Student2	Student3	Student5	Student4	Student6	Student8	Student7	Student9	Student10
Lecturer 2	Student3	Student1	Student2	Student4	Student5	Student6	Student8	Student7	Student10	Student9
Lecturer 3	Student3	Student1	Student2	Student4	Student5	Student6	Student8	Student9	Student7	Student10
Lecturer 4	Student1	Student2	Student3	Student4	Student5	Student6	Student9	Student10	Student8	Student7
Lecturer 5	Student1	Student2	Student3	Student4	Student5	Student6	Student7	Student9	Student8	Student10
Lecturer 6	Student2	Student1	Student3	Student4	Student6	Student5	Student1	Student8	Student7	Student9

							0			
Lecturer 7	Student4	Student3	Student2	Student1	Student5	Student6	Student8	Student9	Student7	Student10
Lecturer 8	Student1	Student3	Student2	Student4	Student6	Student5	Student8	Student7	Student9	Student10
Lecturer 9	Student10	Student7	Student8	Student6	Student4	Student3	Student2	Student1	Student5	Student9
Lecturer 10	Student6	Student4	Student1	Student2	Student3	Student7	Student9	Student8	Student10	Student5

The code was written in Python and yielded the following results.

```
.\Users\DR\PycharmProjects\pythonProje
Student7 is breaking with Lecturer6.
Student8 is breaking with Lecturer6.
Student9 is breaking with Lecturer8.


Success!
Allocation of Students to Lecturers:
===================================
Student1 <---> Lecturer1
Student2 <---> Lecturer2
Student3 <---> Lecturer7
Student4 <---> Lecturer4
Student5 <---> Lecturer3
Student6 <---> Lecturer10
Student7 <---> Lecturer5
Student8 <---> Lecturer8
Student9 <---> Lecturer9
Student10 <---> Lecturer6

Process finished with exit code 0
```

Fig. 3: Results generated from the codes

According to the allocation results, there were certain occasions when two or more students were assigned to a professor, which could generate confusion (see Fig. 4). When two or more students are unable to attach to a single professor, there should be a break between students and lecturers, and any possible matching should be considered.

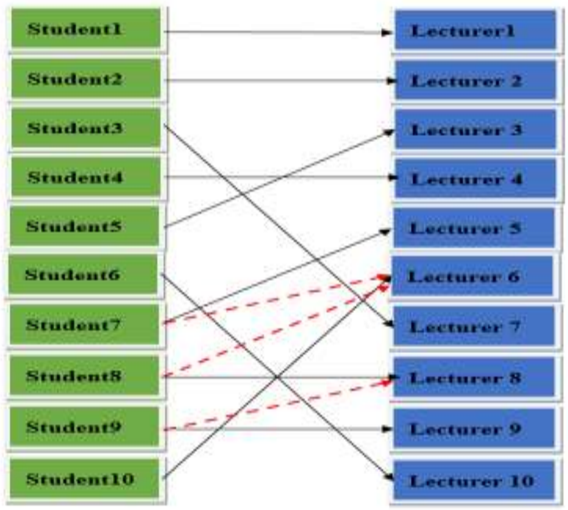


Fig. 4: Students that are to break for effective allocation

Finally, the best allocation and distribution were found. In this case, all 10 students were assigned to all instructors without any conflicts. As a result, Fig. 5 shows the final allocation with 100% optimization.

```
Success!
Allocation of Students to Lecturers:
===================================
Student1 <---> Lecturer1
Student2 <---> Lecturer2
Student3 <---> Lecturer7
Student4 <---> Lecturer4
Student5 <---> Lecturer3
Student6 <---> Lecturer10
Student7 <---> Lecturer5
Student8 <---> Lecturer8
Student9 <---> Lecturer9
Student10 <---> Lecturer6
```

Fig. 5: Successful allocation of students to lecturers

As a result, the results of the Python code allocation are supported by Fig. 6. This also means that the project allocation/distribution has been optimized.

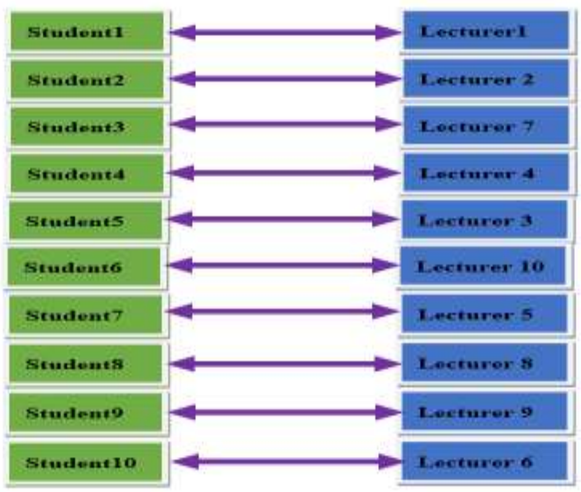


Fig. 6: Optimal Allocation Obtained

Expanded Optimization

Students	1st Prefer	2nd Prefer	3rd Prefer	4th Prefer	5th Prefer	6th Prefer	7st Prefer	8th Prefer	9th Prefer	10th Prefer
Student1	Lecturer 1	Lecturer 2	Lecturer 3	Lecturer 4	Lecturer 5	Lecturer6	Lecturer 7	Lecturer 8	Lecturer 9	Lecturer 10
Student2	Lecturer 2	Lecturer 3	Lecturer 5	Lecturer 1	Lecturer 4	Lecturer6	Lecturer 8	Lecturer 10	Lecturer 9	Lecturer 7
Student3	Lecturer 7	Lecturer 2	Lecturer 3	Lecturer 1	Lecturer 10	Lecturer6	Lecturer 4	Lecturer 9	Lecturer 8	Lecturer 5
Student4	Lecturer 4	Lecturer 3	Lecturer 8	Lecturer 9	Lecturer 5	Lecturer7	Lecturer 6	Lecturer 1	Lecturer 2	Lecturer 10
Student5	Lecturer 2	Lecturer 3	Lecturer 1	Lecturer 5	Lecturer 4	Lecturer8	Lecturer 10	Lecturer 6	Lecturer 7	Lecturer 9
Student6	Lecturer 1	Lecturer 3	Lecturer 2	Lecturer 10	Lecturer 9	Lecturer6	Lecturer 4	Lecturer 7	Lecturer 8	Lecturer 5
Student7	Lecturer 2	Lecturer 6	Lecturer 10	Lecturer 5	Lecturer 4	Lecturer3	Lecturer 8	Lecturer 7	Lecturer 9	Lecturer 1
Student8	Lecturer 4	Lecturer 6	Lecturer 8	Lecturer 3	Lecturer 1	Lecturer9	Lecturer 2	Lecturer 5	Lecturer 7	Lecturer 10
Student9	Lecturer 3	Lecturer 10	Lecturer 7	Lecturer 6	Lecturer 1	Lecturer8	Lecturer 2	Lecturer 9	Lecturer 4	Lecturer 5
Student10	Lecturer 10	Lecturer 8	Lecturer 6	Lecturer 2	Lecturer 1	Lecturer4	Lecturer 3	Lecturer 5	Lecturer 7	Lecturer 9
Student11	Lecturer 7	Lecturer 8	Lecturer 3	Lecturer 4	Lecturer 6	Lecturer1	Lecturer 5	Lecturer 2	Lecturer 10	Lecturer 9
Student12	Lecturer 6	Lecturer 2	Lecturer 8	Lecturer 10	Lecturer 5	Lecturer9	Lecturer 7	Lecturer 3	Lecturer 1	Lecturer 4
Student13	Lecturer 8	Lecturer 7	Lecturer 6	Lecturer 1	Lecturer 2	Lecturer4	Lecturer 10	Lecturer 9	Lecturer 3	Lecturer 5
Student1	Lecturer	Lecturer	Lecturer	Lecturer	Lecturer	Lecture	Lecturer	Lecturer	Lecturer	Lecturer

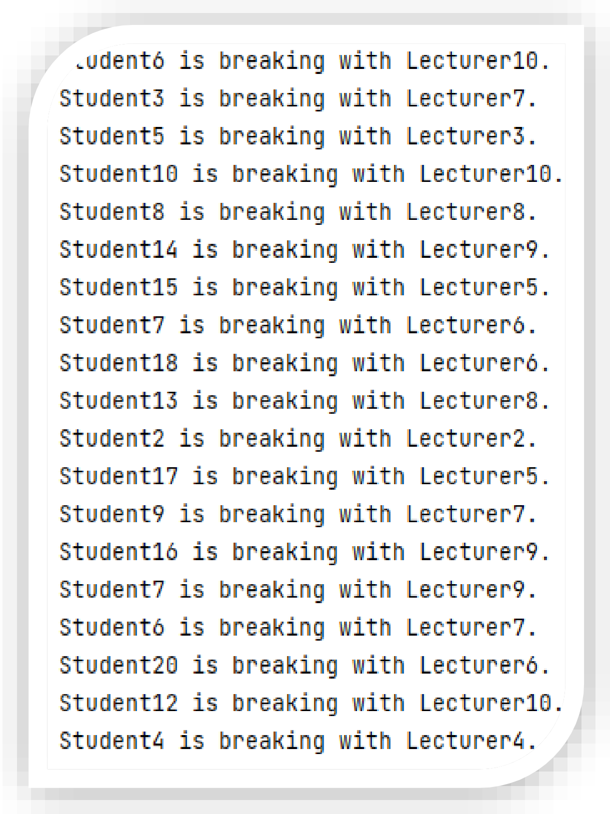
4	9	3	10	6	5	r8	7	1	2	4
Student15	Lecturer 5	Lecturer 1	Lecturer 10	Lecturer 4	Lecturer 6	Lecturer 9	Lecturer 8	Lecturer 2	Lecturer 3	Lecturer 7
Student16	Lecturer 3	Lecturer 8	Lecturer 7	Lecturer 9	Lecturer 5	Lecturer 2	Lecturer 10	Lecturer 1	Lecturer 4	Lecturer 6
Student17	Lecturer 2	Lecturer 10	Lecturer 5	Lecturer 6	Lecturer 7	Lecturer 1	Lecturer 3	Lecturer 9	Lecturer 4	Lecturer 8
Student18	Lecturer 1	Lecturer 4	Lecturer 6	Lecturer 7	Lecturer 10	Lecturer 3	Lecturer 2	Lecturer 8	Lecturer 5	Lecturer 9
Student19	Lecturer 4	Lecturer 1	Lecturer 2	Lecturer 3	Lecturer 9	Lecturer 5	Lecturer 6	Lecturer 8	Lecturer 7	Lecturer 10
Student20	Lecturer 5	Lecturer 6	Lecturer 3	Lecturer 10	Lecturer 9	Lecturer 8	Lecturer 1	Lecturer 2	Lecturer 7	Lecturer 4
Student21	Lecturer 9	Lecturer 4	Lecturer 1	Lecturer 2	Lecturer 3	Lecturer 8	Lecturer 5	Lecturer 10	Lecturer 7	Lecturer 6
Student22	Lecturer 4	Lecturer 6	Lecturer 3	Lecturer 5	Lecturer 9	Lecturer 7	Lecturer 8	Lecturer 2	Lecturer 10	Lecturer 1

Students	1st Prefer	2nd Prefer	3rd Prefer	4th Prefer	5th Prefer	6th Prefer	7st Prefer	8th Prefer	9th Prefer	10th Prefer
Lecturer 1	Student1	Student2	Student3	Student5	Student4	Student6	Student8	Student7	Student9	Student10
	Student11	Student13	Student16	Student15	Student14	Student17	Student12	Student20	Student22	Student21
	Student18	Student19								
Lecturer 2	Student3	Student11	Student2	Student4	Student5	Student6	Student8	Student7	Student10	Student9
	Student22	Student17	Student20	Student14	Student15	Student19	Student18	Student2	Student19	Student12
	Student13	Student1								
Lecturer 3	Student13	Student1	Student2	Student14	Student5	Student16	Student8	Student19	Student7	Student20
	Student3	Student1	Student12	Student4	Student15	Student6	Student18	Student9	Student17	Student10
	Student21	Student22								
Lecturer 4	Student2	Student1	Student4	Student3	Student15	Student16	Student19	Student10	Student8	Student7

	Student11	Student12	Student13	Student14	Student5	Student6	Student9	Student20	Student21	Student22
	Student18	Student17								
Lecturer5	Student5	Student9	Student13	Student4	Student1	Student16	Student17	Student21	Student18	Student10
	Student1	Student12	Student3	Student14	Student15	Student6	Student7	Student11	Student8	Student20
	Student19	Student22								
Lecturer6	Student2	Student1	Student3	Student14	Student16	Student15	Student20	Student18	Student7	Student9
	Student12	Student11	Student13	Student4	Student6	Student5	Student10	Student8	Student17	Student19
	Student22	Student21								
Lecturer7	Student14	Student13	Student2	Student1	Student15	Student6	Student8	Student9	Student17	Student20
	Student4	Student3	Student12	Student21	Student5	Student16	Student18	Student19	Student7	Student10
	Student11	Student22								
Lecturer8	Student21	Student13	Student2	Student14	Student6	Student15	Student8	Student17	Student9	Student20
	Student1	Student3	Student12	Student4	Student16	Student5	Student18	Student7	Student19	Student10
	Student21	Student22								
Lecturer9	Student10	Student7	Student8	Student16	Student4	Student3	Student2	Student1	Student15	Student19
	Student20	Student17	Student22	Student6	Student14	Student13	Student11	Student12	Student5	Student9
	Student18	Student21								
Lecturer10	Student16	Student4	Student21	Student12	Student13	Student22	Student19	Student18	Student10	Student5
	Student6	Student14	Student11	Student2	Student3	Student7	Student9	Student8	Student20	Student15
	Studen	Studen								

	t1	t1 7									

1	**Student6 is breaking with Lecturer10.**
2	Student3 is breaking with Lecturer7.
3	Student5 is breaking with Lecturer3.
4	Student10 is breaking with Lecturer10.
5	Student8 is breaking with Lecturer8.
6	Student14 is breaking with Lecturer9.
7	Student15 is breaking with Lecturer5.
8	Student7 is breaking with Lecturer6.
9	Student18 is breaking with Lecturer6.
10	Student13 is breaking with Lecturer8.
11	Student2 is breaking with Lecturer2.
12	Student17 is breaking with Lecturer5.
13	Student9 is breaking with Lecturer7.
14	Student16 is breaking with Lecturer9.
15	Student7 is breaking with Lecturer9.
16	Student6 is breaking with Lecturer7.
17	Student20 is breaking with Lecturer6.
18	Student12 is breaking with Lecturer10.
19	Student4 is breaking with Lecturer4. …


```
.udent6 is breaking with Lecturer10.
Student3 is breaking with Lecturer7.
Student5 is breaking with Lecturer3.
Student10 is breaking with Lecturer10.
Student8 is breaking with Lecturer8.
Student14 is breaking with Lecturer9.
Student15 is breaking with Lecturer5.
Student7 is breaking with Lecturer6.
Student18 is breaking with Lecturer6.
Student13 is breaking with Lecturer8.
Student2 is breaking with Lecturer2.
Student17 is breaking with Lecturer5.
Student9 is breaking with Lecturer7.
Student16 is breaking with Lecturer9.
Student7 is breaking with Lecturer9.
Student6 is breaking with Lecturer7.
Student20 is breaking with Lecturer6.
Student12 is breaking with Lecturer10.
Student4 is breaking with Lecturer4.
```

5 Discussion

There are hundreds of grudges among students in traditional project allocations between students and professors due to inadequate clues as a result of the lecturer assigned to such student. Due to time limits, the difficulty of distribution, and allocation loops, several students did not complete their projects or sought the assistance of an expert to assist them. The basic goal of using an optimization problem method is to assign a given number of people to an equal number of jobs on a one-to-one basis in such a way that the overall time/cost of doing that work is minimized or the total efficiency/profit of allocation is maximized. According to Akpan and Abraham (2016), the essence of using assignment problem is due to the fluctuating capacity of a human or machine to accomplish the assigned work or job. The use of classical Assignment Problems to solve real-world problems has some limitations. Stable Matching can certainly help to alleviate these constraints. The model was appealing and required extremely simple arithmetic and logical computations to produce results. It should be emphasized that the model was provided properly and accurately in order to achieve the desired output. Researchers such as [13] used stable Matching in economics to resolve conflicts of interest among selfish market actors, and [15] used stable matching algorithm to resolve challenges to identify a b-matching of highest weight while observing lower quotas. [11] conducted a study on departmental course allocation problem in a university system over a definite period of time, and [4] used a simple scheduling technique called resource histogram to utilize the allocation and leveling of the available resources.

6 Conclusion

It goes without saying that project allocation and distribution are crucial in final year project writing. As a result, it can be inferred that the approach (Stable Matching Algorithm) delivers an optimal solution to an Assignment Problem in less rounds. This strategy will be highly useful for departmental academic staff/decision makers dealing with allocation/distribution concerns because it takes less time and is very simple to grasp and execute. Future studies may investigate using another programming language to achieve similar outcomes.

References:

[1]. Muraina I O, Aiyegbusi E A& Lesi B O (2021). Ensuring Team Building Learning with the Use of Relevant Collaborative Tools. *Middle East International Conference on Contemporary Scientific Studies*-VI, September 20-22, 2021 Beirut, Lebanon. Pp 264-270

[2]. Muraina, I O (2023). Youth Empowerment, Green Skills Acquisition, and Environmental Needs & Sustainability: Analysis and Emphasis on Correlational and Influential Factors. *CUKUROVA 11th International Scientific Researches Conference*, August 22-24, 2023/ Adana, Turkey, 844-851

[3]. Muraina I O, Lesi B O, Oladapo W O & Hamzat I O (2021). Artificial Neural Network Model to Predict Students ‘Relevant Courses of Study on Getting Into Higher Institutions. *International Siirt Scientific Research Congress* 5-7 November 2021 SIIRT, Turkey. Pp 364-370

[4]. Eirgash, M. A. (2020). Resource Allocation and Leveling in Construction Management Projects with Resource Histogram. *American Journal of Engineering and Technology Management*. Vol. 5, No. 6, pp. 91-95. doi: 10.11648/j.ajetm.20200506.11

[5]. Azevedo, A. T; Kameyama, A; Amorim, J. A & Gustavsson, P M (2013a) A Mathematical Model for Determining Timetables that Minimizes the Number of Students with Conflicting Schedules. A Conference Paper. Retrieved from https://www.researchgate.net

[6]. Azevedo, Anibal Tavares de; Kameyama, Alexander; Amorim, Joni A & Gustavsson, Per M (2013b) A Mathematical Model for Determining Timetables that Minimizes the Number of Students with Conflicting Schedules. A Conference Paper. Retrieved from https://www.researchgate.net

[7]. Thakre, T. A; Chaudhari, O. K & Dhawade, N. R (January, 2018). Placement of Staff in LIC using Fuzzy Assignment Problem. *International Journal of Mathematics Trends and Technology (IJMTT)* – Volume 53 Number 4, Page 259 – 266

[8]. Abraham, David J; Irving, Robert & Manlove, David F (2007). Two Algorithms for the Student-Project Allocation Problem; Engineering and Physical Sciences Research Council grant, *Journal of Discrete Algorithms*, 5(1), 73-90, doi:10.1016/j.jda.2006.03.006

[9]. Göktaş, Ö. & Şad, S. N., (2014). Assigning the practice teachers for school experience and teaching practice courses: Criteria, challenges and suggestions. [in Turkish]. Hacettepe Üniversitesi Eğitim Fakültesi Dergisi [*Hacettepe University Journal of Education*], 29(4), 115-128.

[10]. Muraina I O, Zosu, S J & Agoi, M A (2021). Expert System Rule-Based Optimization in Assigning Courses of Study. *Latin American Conference on Natural and Applied Sciences* November

5-6, 2021/ Villahermosa, Mexico. Pp 390-394

[11]. Ekhosuehi, V. U (2016) University Course Allocation in A Department Using Linear Programming Techniques. https://www.researchgate.net/publication/308902884

[12]. Golle, P. (2022). *A Private Stable Matching Algorithm*; Palo Alto Research Center 3333 Coyote Hill Road, Palo Alto, CA 94304, USA

[13]. Li, Ke; Zhang, Qingfu; Kwong, Sam; Kwong Sam & Wang, Ran (2014). Stable Matching-Based Selection in Evolutionary Multiobjective Optimization; *IEEE Transactions on Evolutionary Computation*,18 (6), https://doi.org/10.1109/TEVC.2013.2293776

[14]. Subramanian, Ashok (1994). A New Approach to Stable Matching Problems, *HomeSIAM Journal on Computing* 23(4), https://doi.org/10.1137/S0097539789169483PDF

[15]. Arulselvan, Ashwin; Cseh, Ágnes & Matuschke, Jannik (2014). The Student/Project Allocation problem with group projects

[16]. Muraina, I O & Zosu, S J (2021). Artificial Intelligent Rule-Based Optimization for Course Allocation Challenges. *International Journal of Applied Physics*, 6, 42-48. Link: https://www.iaras.org/iaras/home/caijap/artificial-intelligent-rule-based-optimization-for-course-allocation-challenges

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The author equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US