Linear Programming Formulations for Schedule and Budget Optimization for Tutorial Center

Abstract—**Scheduling classes efficiently is one of most main problems that encounter any institution around the world. In this paper, a Linear programming model is proposed to solve the scheduling problem facing the Tutorial Center at the university by minimizing the total cost as well as increase the learning rate of students by maximizing the competency of the peer tutors. The data received from the tutorial center were tested on the proposed LP model, and the results show a promising performance. The center cost after optimization dropped down by 10%, and the load distribution between peer tutors as well as week days is balanced too.**

Index Terms—**Linear programming, scheduling, budget minimization.**

I. Introduction

The Tutorial Center at the university offers set of courses for any student who want to improve in any of the offered courses. Originally, the set of courses were small (Less than 10), in addition, the number of students asking for courses small too. Therefore, things have been done manually. As the number of students, as well, the sets of offered courses grow, the tutorial center needs an efficient method to schedule classes based on. The center is aiming with the optimization of its operation, to minimize the cost of peer tutors as the center is paying each peer tutor per hour of teaching. Moreover, balancing the distribution of classes' loads between peer tutors and weekdays is another objective of the center after schedule optimization. Before optimization, the operator of the center manually schedule classes instance by instance. For example, the operator will remember set of peer tutors, or write them down in excel file as shown in Fig.1, and schedule the new request randomly between the tutors. This lead to imbalance in the load distribution between peer tutors. The second issue is that the manual search is not fast nor accurate. In some instances, the operator will declare to the student that there are no matching peer tutors where in fact there are multiple tutors matching this student, but cannot be found in short period of time as the list of peer tutors, students, and courses grow. This load imbalance causes single point of failure as known in any system. Another challenge to the center is to answer the following question, given a set of peer tutors, set of scheduled sessions, how to distribute these tutors across the allocated sessions. In this project, the rating maximization is used to solve this issue. By maximizing the rating of allocated peer tutors in the schedule, we can make sure that the maximum happiness is obtained since the rating is taken from the students feedback after each session. Through the literature, three papers were the most useful one to this research. Since they gave insights into the proper way model the problem in order to solve it efficiently. In [1], the researcher tried to develop a Linear programming model that will optimize student learning by efficiently distributing faculties through different classes. The paper considered the faculty competency in each course, and tried to schedule the most competent faculty to each course. This idea is used in this paper by trying t maximize the ratings of final schedule by allocating peer tutors with high ratings. In [2], they used occupational matrix to build a software tool and mathematical model to help their school schedule classes more efficiently. The occupational matrix turned out to be useful in the proposed formulation as it gave insights into the proper way to transform the input textual data into $0-1$ matrices. Finally, some tried to propose an Integer Linear programming model to help in the scheduling task of their school. They managed to generate a university schedule that minimizes the capacity to enrollment ratio of any class[3]. The remaining sections are: data transformation, LP formulations, and results analysis. The paper tries to combine multiple approaches to obtain the most efficient method to solve the scheduling problem.

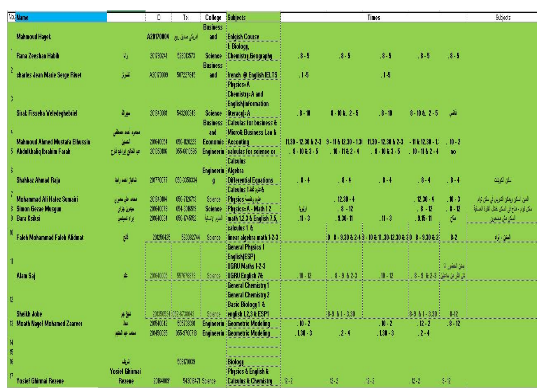

Fig. 1. Original Missy Schedule.

II. Data Processing

Data processing is essential part of any research. The data received from the Tutorial Center had many instances, as shown in TABLE.I, the center is offering 36 different courses, and it has 44 peer tutors willing to teach these courses for the different students. Currently, there are over 100 students looking to schedule some session for different courses..

TABLE I
NUMBER OF INSTANCES IN THE INPUT DATA

Instance	Number of Instances
Courses	36
Tutors	44
Students	100

For each student as well as for each peer tutor, the data received that specify the courses requirements and the available times has the shape shown in Fig.2. For peer tutors, it will be the courses that they can teach, and the timing across week that they can deliver a session. The main issue with this representation is that necessity for data transformation. Since solver will not be able to process textual data, and model formulations become practically difficult.


```
{
    "Courses":
        "Calculus I",
        "Physics I",
        "Chemistry I",

    "Sunday":
        [08, 12],
        [16, 18],

    "Monday":
        [10, 12],

    "Tuesday":
        [10, 14],
        [16, 18],

    "Wednesday":
        [09, 11],

    "Thursday":
        []
}
```


Fig. 2. Example of Input data.

a) Tutors and Student Courses: The first transformation is done with the courses data. Since the number of courses is relatively small, a matrix is formed that has the number of student or peer tutors as a row, and the number of courses as columns. C_s of size 100x$36$is the matrix that shows the courses that each student is asking for help in. Each row represent one of the student ac cross the 36 courses. If the student is asking for a course, the matrix at that row and the column corresponding to the course will be one, else it will be zero. Similarly, peer tutors will have their matrix C_p of size 44x36too.

$$c_s = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} ; c_p = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \quad (1)$$

b) Available Times: The second transformation is done with the available times data. The tutorial center works 5 days across the week, 14 hours per day. The matrix that represent the timing for peer tutors is t_p with the size of 44x70. Each row represent one of the peer tutors across the 70 $(14*5)$ time slots across the week.If the peer tutor is available at a time slot, the matrix at that row and the column corresponding to the time slot will be one, else it will be zero. Similarly, students will have their matrix t_s of size 100x70 too.

$$t_s = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} ; t_p = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \quad (2)$$

c) Tutors Ratings: The third data matrix is the ratings matrix r of size 44x36. It shows the ratings of each peer tutor in each one of the courses that he teach. Ratings are between 1 and 4.The objective is to maximize the overall ratings of the allocated schedule after optimization. If one session can be delivered be two peer tutors, ti make sense to choose the one with higher rating.

$$r = \begin{pmatrix} 1 & \cdots & \cdots & 4 \\ 3 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 2 & \cdots & 0 & 1 \end{pmatrix} \quad (3)$$

d) Student Demands: The last data matrix is the demand matrix d of size 100x36. It shows the demand of each students in each course. The demand is between 1 and 3. The objective of the optimization is to satisfy the demand of each student,but with the minimum cost. The demand matrix in this project as well as the ratings matrix has been randomly generated. It is worth to mention that some demands might cause the model to be infeasible. For instance, if a student is asking for 3 hours in course i, but this student, have only one hour matching peer tutor in term of timing. Then, the demand has to be reduced to one before optimizing the final schedule.

$$d = \begin{pmatrix} 1 & \cdots & \cdots & 2 \\ 3 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 2 & \cdots & 0 & 1 \end{pmatrix} \quad (4)$$

III. LINEAR PROGRAMMING FORMULATION

The objective of this formulation is to satisfy the demand from the students as well as distributing the different peer tutors across the courses such that we choose the best tutors. The best in terms of the their ratings. In this formulation, there is two objective functions, and a set of constraints for each one of these objective functions.

A. Decision Variables

Since there is two objective functions, the LP formulation has two main decision variables. The index i corresponds into one of the available 36 courses, and the index j corresponds to one of the available 70 time slots across the week. For each student s, there is a decision matrix X_{sij}, it is of size 36x70. This matrix can take either zero or one, zero at course i and time slot j means that we schedule a session of course i in the time j. Therefore, the full X_{sij} that combine all students is of size 100x36x70.
For each tutor p, there is a decision matrix Y_{pij}, it is of size 36x70. This matrix can take either zero or one, zero at course i and time slot j means that we allocate this tutor into one of the scheduled session of course i in the time j. Similarly, the full Y_{pij} that combine all students is of size 44x36x70.

$$i, \text{ for } i = 1, \ldots, 36. \tag{Course Index}$$

$$j, \text{ for } j = 1, \ldots, 70. \tag{Time Index}$$

The decision variables are:

$$X_{sij} \in \{0, 1\}, \text{ for } s = 1, \ldots, 100. \tag{Student}$$

$$Y_{pij} \in \{0, 1\}, \text{ for } p = 1, \ldots, 44. \tag{Peer Tutor}$$

B. Objective Functions

it is clear now that the model needs to minimize the number of scheduled classes such that the total demand is satisfied. Each peer tutor cost 25 AED per one hour of teaching. Afterward, the model will distribute the peer tutors across the optimized schedule by trying to maximize the overall rating of the schedule. The center will gain two benefits, it will have an optimized.

$$\min \quad 25 * \sum_{s=1}^{100} \sum_{i=1}^{36} \sum_{j=1}^{70} X_{sij} \tag{Schedule Cost}$$

$$\max \quad \sum_{p=1}^{44} \sum_{i=1}^{36} \sum_{j=1}^{70} Y_{pij} * r_{pi} * tp_{pj} \tag{Tutor Ratings}$$

C. Student Constraints

This section is for constraints of the first objective function. Since the matrix X_{sij} that combine all students is of size 100x36x70, the number of decision variables is huge, but, by pre solving, the number can be reduced significantly. Since any student s, will ask for three courses on average, and be available for 10 hours on average across the week. Then, all other 33 courses indices, as well as the other 60 time slots will be zero. hence, only 12.5% of the matrix X_{sij} is feasible to be one, the other 87.5% are zeros before optimizing the LP model. The pre solving is done simply by multiplication of $cs_{si} * ts_{sj}$.

$$X_{sij} \leq cs_{si} * ts_{sj} \tag{Solution Reduction}$$
$$\text{for } s = 1, \ldots, 100, i = 1 \ldots, 36, j = 1, \ldots, 70. \tag{5}$$

To avoid any time conflict, the model is constrained to schedule one class at any time slot. This is done by restricting the sum of all scheduled classes for any student at any time slot to be less than or equal to one.

$$\sum_{i=1}^{36} X_{sij} \leq 1 \tag{Time Conflict}$$
$$\text{for } s = 1, \ldots, 100, j = 1, \ldots, 70. \tag{6}$$

Sometimes, there might be shortage in supply, the optimized model needs to make sure that the number of peer tutors must be greater than half of the scheduled classes for any course at any time. In other word, the number of scheduled classes at any instance(i, j) can not exceed twice the number of peer tutors at that instance(i, j). The factor of two is chosen because the tutorial center is allowing for two students to take the same session.

$$\sum_{s=1}^{100} X_{sij} \leq 2 * \sum_{p=1}^{44} tp_{pj} * cp_{pi} \tag{Supply Satisfaction}$$
$$\text{for } i = 1, \ldots, 36, j = 1, \ldots, 70. \tag{7}$$

The demand satisfaction constraints is the one that will drive the minimization model from the zero solution. For any student s, the number of scheduled classes across the week for course i must exceed the demand for course i. Note that the sum is multiplied by $cp_{pi} * tp_{pj}$ just to count the time slots that have matching peer tutors teaching the course i.

$$\sum_{p=1}^{44} \sum_{j=1}^{70} X_{sij} * cp_{pi} * tp_{pj} \geq d_{si} \tag{Demand Satisfaction}$$
$$\text{for } s = 1, \ldots, 100, i = 1, \ldots, 36. \tag{8}$$

If we consider satisfying the demand for each course alone without looking into the big picture, then the model might not satisfy the total demand. The last constraints make sure that the number of scheduled hours is greater than or equal to the total demand for each student. Someone might have a good argument that this constraint is redundant since there are time conflict constraint, and demand satisfaction constraint.

$$\sum_{i=1}^{36} \sum_{j=1}^{70} X_{sij} \geq \sum_{i=1}^{36} d_{si} \tag{All Demand Satisfaction}$$
$$\text{for } s = 1, \ldots, 100. \tag{9}$$

D. Schedule Construction

The final optimized schedule is constructed by combining each two student allocated for course i and time slot j. Therefore, the $schedule_{ij}$ matrix is of size 36x70, and the cost of operation for the tutorial center is calculated based on this matrix. If the center allows for more than two students at the same class, then, the cost can be reduced further.

E. Peer Tutors Constraints

The maximization model for the peer tutors ratings is quite similar to cost minimization model in terms of constraints.

Since the matrix Y_{pij} that combine all peer tutors is of size 44x36x70, the number of decision variables is large, again, by pre solving, the number can be reduced significantly. Since any peer tutors p, can teach five courses on average, and be available for 15 hours on average across the week. Then, all other 31 courses indices, as well as the other 55 time slots will be zero. hence, only 20% of the matrix Y_{pij} is feasible to be one, the other 80% are zeros before optimizing the LP model. The pre solving is done simply by

$$Y_{pij} \leq cp_{pi} * tp_{pj} \qquad \text{(Solution Reduction)}$$
$$\text{for } p = 1, \ldots, 44, i = 1. \ldots, 36, j = 1, \ldots, 70. \qquad (10)$$

To avoid any time conflict, the model is constrained to schedule one class at any time slot. This is done by restricting the sum of all scheduled classes for any peer tutor at any time slot to be less than or equal to one.

$$\sum_{i=1}^{36} Y_{pij} \leq 1 \qquad \text{(Time Conflict)}$$
$$\text{for } p = 1, \ldots, 44, j = 1, \ldots, 70. \qquad (11)$$

The maximization model will try to drive the solution to infinity. The final schedule is built and optimized, and it is obvious that the number of scheduled tutors for course i and time j must be equal to the number of scheduled classes. The less than is added to give more flexibility to the model since any way the maximization model will satisfy the equality sign given solution will be feasible.

$$\sum_{p=1}^{44} Y_{pij} \leq schedule_{ij} \qquad \text{(Demand Satisfaction)}$$
$$\text{for } i = 1, \ldots, 36, j = 1, \ldots, 70. \qquad (12)$$

The aforementioned constrains are the constraints needed to build the optimal schedule for the tutorial center, more techniques can be added later on to distribute tutors more efficiently, and to force the scheduling model to combine student into the same slot(i, j) such that the cost when the final schedule is constructed is minimum. In the next section, the performance of the LP model across multiple metric is discussed to show that indeed this model although might not be the most optimal model in the world, helped the tutorial center.

IV. Results Discussion

This section discussed the different results obtained after optimizing the schedule for the tutorial center. As shown in Fig.3 The cost before the schedule optimization, was 5000 AED per week, and after the optimization the cost dropped down to 4500 AED per week. Each month, the center is saving 2000 AED of the budget, which is around 10% cost saving. Since the demand was randomly generated, it might not reflect the reality as anticipated. For example, in reality, most of the demand for a specific course i will come from the students taking the same course, even from the same section, hence, these students are mostly going to request the same course giving almost similar available times, this will result in many cases where many students can be allocated to same time slot. Moreover, if the constraint that each session must have at most two students is increased, or even considered as a parameter to optimize, then the cost can reduced even more.

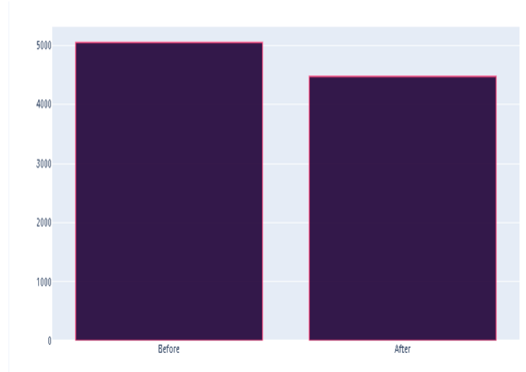

Fig. 3. Weekly Cost before and after Optimization.

The schedule after optimizing left 4% of the total demand unsatisfied as shown in Fig.4. Mostly, this is a result of the randomly generated demands, because the timing given by the student might not be enough compared to the randomly generated demand. This unsatisfied demands can also be manged manually given the fact there is an optimized schedule ready. Just create a new session that satisfy the unsatisfied demand.

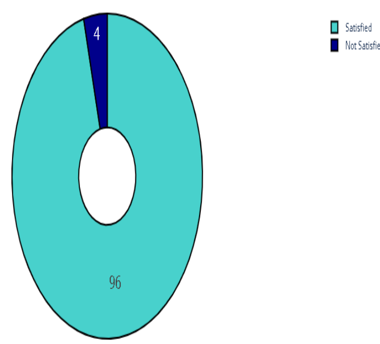

Fig. 4. Unsatisfied Demand percentage after optimization

Looking into the number of scheduled sessions each day of the week shown in Fig.5, the number is maximum at Sunday which is the objective as the students are more energetic to have tutorials at the beginning of the week. The load is decreasing each day which is anticipated as the number of tutors decreases too. The other advantage of such distribution, is the closeness between the first four days in terms of the number of sessions. This helps a lot when one of the days is a holiday, or the center is not working, the missed session can be covered back, or dismissed without huge burden on the center.

Furthermore, the optimized schedule distributes the teaching

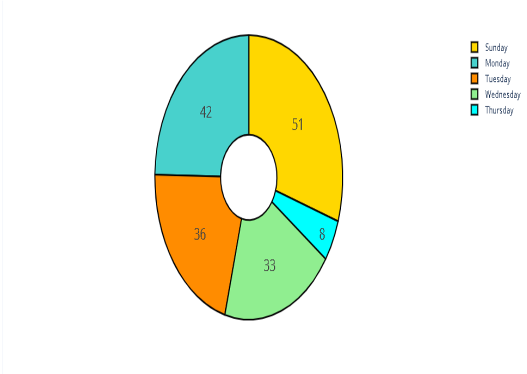

Fig. 5. Classes Distribution across week days.

loads between peer tutors equally except for some cases. This is clear in Fig.6 when each tutor is teaching $2-5\%$ of the total number of session. This helps in avoiding single point of failure, if one tutor is teaching 15%, then by missing this tutor, the center will have difficult times rescheduling missed sessions.

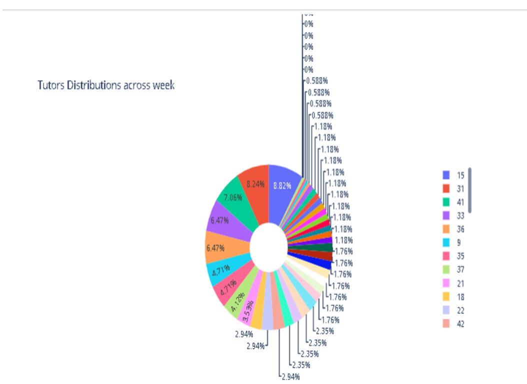

Fig. 6. Load distribution across peer tutors.

Fig.7 is a result of this benefits, it is clear that majority of peer tutors (over 85%) have some teaching sessions, which is one of the objective of the center is to help tutors financially since they are students at the end of the day. At the same time, students will have the chance to try different peer tutors, and choose the best for them. This is also helping the center attract other students.

CONCLUSION

This paper discussed a Linear programming model to optimize the scheduling cost for the Tutorial Center. The results showed that the model indeed reduced the operation cost by 10% as well as increase the satisfaction of the students with the schedule by maximizing the ratings of the allocated peer tutors. Finally, the load distribution over peer tutors and the classes distribution across the week is also balanced.

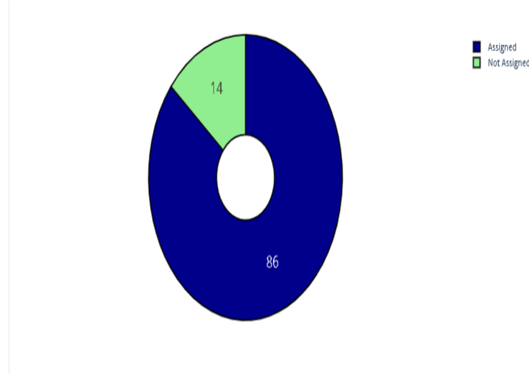

Fig. 7. Percentage of allocated peer tutors.

REFERENCES

[1] Elvira E. Ongy, "Optimizing Student Learning: A Faculty-Course Assignment Problem Using Linear Programming," Journal of Science, Engineering and Technology 5:1-14(2017).

[2] Ryan Wormald, Curtis Guimond, "Creating a More Efficient Course Schedule at WPI Using Linear Optimization", April 2012.

[3] Ahmed Wasfy and Fadi A. Aloul, "Solving the University Class Scheduling Problem Using Advanced ILP Techniques," Published in 2007.