Exam Center Allotment Problem

Nikitha Reddy
Electronics and Communication Engineering
IIT Guwahati
Hyderabad, India
nikithareddy2309@gmail.com

Shreya Goel
Electronics and Communication Engineering
IIT Guwahati
Guwahati, India
shreya.goel@iitg.ac.in

Abstract

This work proposes a model for the Exam Center Allotment problem which involves determining an allotment of students to centers so as to optimize the total number of centers in use while ensuring that students' preferences are taken into consideration. In addition, several superiority constraints have to be followed. This single objective allotment problem can be formulated as an mixed integer linear programming problem and can be solved to optimality. In this work, a mathematical model is proposed to address the Exam Center Allotment problem along with a detailed example.

I. Introduction

For any large scale exam, centers are allotted over a huge geographical area. For the ease of students, preferences are taken regarding their choice of cities within a particular zone. The allotment of centers to students, taking into account their preferences and at the same time minimising the total number of centers to be used, along with other constraints, makes it an optimisation problem. This paper proposes a model to solve this optimisation problem using a mathematical approach. The formulation is a single objective Mixed Integer Linear Programming(MILP) problem.

The rest of the paper is organized as follows. Section 2 defines the problem under consideration. Section 3 develops the mathematical model for the Exam Center Allotment Problem. In Section 4, a computational example is presented to demonstrate how the formulation works. Section 5 provides closing remarks.

II. Problem Description

The exam center allotment problem can be defined as follows. Given :

- The list of cities
- The centers available in each city
- The list of students to be allotted centers
- The class of each student
  For example
  - class 1 - PWD Female
  - class 2 - PWD Male
  - class 3 - Female
  - class 4 - Male
- The capacity of each center
- The rating of each center with respect to all the other centers present in that city
- Preferences of each student for cities

The objective is to determine an assignment of students to centers so as to optimize the total number of centers in use.

In addition, there are several requirements that must be met as follows:

- Each student should be assigned to exactly one center.
- Number of students allotted to a center should be no more than it's capacity.
- Students should only be allotted a center in a city for which they have filled preference.
- A center with higher rating value is superior to other centers in that city with lower rating value and is considered to be more suitable for special classes of classes, like PWD(Person with Disabilities). So, inferior centers shouldn't be allotted to students when seats in superior centers are available.
- A student with lower class value is considered at more priority than the one with higher class value and is considered to have special needs and hence should be allotted their best preference whenever possible.
- Prioritized class students should be given superior centers whenever possible.
- The sum of preferences of all students corresponding to their allotted city should be minimised. As far as possible, all students should be assigned their best preference.

  The center allotment is to be done separately for each zone, in cases when cities are divided into zones.

III. Mathematical Model

The proposed formulation requires the following indices, sets, parameters, and variables

Indices and Sets:

- s $\in$ S, represents students.
- c $\in$ C, represents cities.
- e $\in$ E, represents exam centers available in all cities.

Parameters:

- center_vs_city(e,c)
  - Binary Parameter.
  - Represents whether or not center e is in city c.
  - Equals 1, if center e is in city c and 0 otherwise.
- class(s)
  - Positive Integer.
  - Represents class of student s.
  - A student with lower class value is at more priority than a student with higher class value.
- capacity(e)
  - Non negative Integer.
  - Represents capacity of center e, i.e., the maximum number of students that can be allotted to center e.
- rating(e)
  - Positive Integer.
  - Represents rating of center e.
  - A center with higher rating value is superior to a center with lower rating value.
- citywise_preference_table(s,c)
  - Non negative Integer.
  - Represents preference of student s for each city c.
  - Lower value indicates higher preference.
  - A preference value of 0 indicates that the student has not mentioned this city in his/her preferences.
- center_preference_table(s,e)
  - Non negative Integer.
  - Represents preference of student s for each center.
  - Preference given to a center e is nothing but preference given to the city in which it is.
  - Lower value indicates higher preference.

$$center\_preference\_table(s,e) = \sum_{c \in C} citywise\_preference\_table(s,c) \times center\_vs\_city(e,c)$$

- center_vs_center_superiority(e, ee)
  - Binary Parameter.
  - Represents whether or not centers e and ee are in the same city and center e is superior than center ee.
  - Equals 1, if center e and center ee belongs to same city and center e is superior than center ee and 0 otherwise.
    if:
    $$\sum_{c \in C} center\_vs\_city(e,c) \times center\_vs\_city(ee,c) = 1$$
    and
    $rating(e) - rating(ee) > 0$,
    $center\_vs\_center\_superiority(e,ee) = 1$

    else: $center\_vs\_center\_superiority(e,ee) = 0$
- city_class_preference_comparison(s, ss, c)
  - Binary Parameter.
  - Represents whether or not student s belongs to prioritized class than student ss and has given lower or equal preference for city c as student ss and both have mentioned this city in their preferences.
  - Equals 1, if class value of student s is lower than student ss, and preference value given to city c by student s is lower or equal to that of student ss and 0 otherwise.

    if
    $citywise\_preference\_table(s,c) \leq citywise\_preference\_table(ss,c)$
    $and\ citywise\_preference\_table(s,c) > 0$
    $and\ citywise\_preference\_table(ss,c) > 0$
    $and\ class(s) < class(ss)$:
    $city\_class\_preference\_comparison(s,ss,c) = 1$

    else :
    $city\_class\_preference\_comparision(s,ss,c) = 0$
- number_of_centers_in_city_of_this_center(e)
  - Non negative Integer.
  - Represents total number of centers in the city in which this center is present.

$$number\_of\_centers\_in\_city\_of\_this\_center(e) = \sum_{c \in C} \{ \sum_{ee \in E} center\_vs\_city(ee,c) \} \times center\_vs\_city(e,c)$$

Variables:

- student_vs_allotted_center(s, e)
  - Binary Variable.
  - Represents whether student s is allotted to center e.
  - Equals 1, if student s is allotted to center e and 0 otherwise.
- student_vs_allotted_city(s, c)
  - Binary Variable.
  - Represents whether student s is allotted to a center in city c.
  - Equals 1, if center allotted to student s is in city c and 0 otherwise.
- number_of_centers_in_use
  - Non Negative Integer.
  - Represents total number of centers in use after completion of allotment.
- is_center_in_use(e)
  - Binary Variable.
  - Represents whether or not any student is allotted to the center e.
  - Equals 1, if number of students allotted to the center e is greater than 0, and 0 otherwise.
- should_be_vacant(e)
  - Binary Variable.
  - Represents whether superiority constraint requires center e to be vacant or not.
  - Equals 1, if centers superior than center e have some vacancy and 0 otherwise.

- student_vs_allotted_city_center(s, c, e)
  - Binary Variable.
  - Represents whether center e of city c is allotted to student s.
  - Equals 1, if center e is in city c and is allotted to student s, and 0 otherwise.
- preference_city_allotted(s)
  - Positive Integer.
  - Represents preference of student s for the center he/she is allotted.
  - Lower bound of this variable is set to 1 to ensure that a student should not be allotted a center in any city beyond the cities which he/she has mentioned in his/her preferences.

$$preference\_city\_allotted \geq 1$$

- rating_center_allotted(s)
  - Positive Integer.
  - Represents rating of the center allotted to student s.

Constraints:

- Each student should be allotted exactly 1 center.
  Constraint 1 -

$$\sum_{e \in E} student\_vs\_allotted\_center(s, e) = 1, \;\; \forall s \in S$$

- Number of students allotted to a particular center should not exceed its capacity.
  Constraint 2 -

$$\sum_{s \in S} student\_vs\_allotted\_center(s, e) \leq capacity(e), \;\; \forall e \in E$$

- Inferior centers (centers with lower rating) should not be allotted students when superior centers (centers with higher rating) are available.
  - To convert this constraint to the mathematical form, we maintain a helper variable named as should_be_vacant and keep on updating it.
  - To update the variable should_be_vacant, we bound it from both sides as follows.
    Constraint 3.1 -

$$\begin{aligned} & small\_value \times \\ & (capacity(e) - \sum_{s \in S} student\_vs\_allotted\_center(s, e)) + \\ & center\_vs\_center\_superiority(e, ee) \leq \\ & should\_be\_vacant(ee) + 1, \;\; \forall e, ee \in E \end{aligned}$$

    Constraint 3.2 -

$$\begin{aligned} & (1 + small\_value) \times should\_be\_vacant(ee) \leq \\ & small\_value \times \\ & (capacity(e) - \sum_{s \in S} student\_vs\_allotted\_center(s, e)) + \\ & center\_vs\_center\_superiority(e, ee), \;\; \forall e, ee \in E \end{aligned}$$

  - The above equations set should_be_vacant(e) of a center e to 1, if there is any vacancy in any of the centers which have a rating value greater than center e, and belong to the same city in which center e lies.
  - For example, if there is a center e such that it lies in same city as center ee and center e has a higher rating than center ee, i.e

$$center\_vs\_center\_superiority(e, ee) = 1$$

    and center e has some vacancy left, i.e

$$capacity(e) - \sum_{s \in S} student\_vs\_allotted\_center(s, e) > 0$$

    then according to this, equation of Constraint 3.1 reduces to,

$$small\_value + 1 \leq should\_be\_vacant(ee) + 1$$

  - Here small_value lies between 0 and 1. Hence the above constraint, forces should_be_vacant(ee) to go to 1.
  - For all other cases, should_be_vacant(ee) will be forced to go to 0 by constraints 3.1 and 3.2 combined.
  - Now this should_be_vacant variable can be used to restrict filling inferior centers whenever there are vacant superior centers. Mathematical equation for this can be formed as follows.
    Constraint 3.3 -

$$\begin{aligned} & \sum_{s \in S} student\_vs\_allotted\_center(s, e) \leq \\ & (1 - should\_be\_vacant(e)) \times large\_value, \;\; \forall e \in E \end{aligned}$$

  - The above equation makes sure that the number of students allotted to a center e is greater than 0 only when should_be_vacant(e) is 0. If should_be_vacant(e) is 1, number of students allotted to center e is forced to go to 0.
- As decision variables, we have both student_vs_allotted_center(s,e) and student_vs_allotted_city(s,c). Hence we need to assure that these two outputs support each other for the solution to be feasible. For this we need to make a constraint to keep these 2 variables linked to each other, so that one variable gets updated accordingly when the other variable changes.
  - To achieve this, we maintain a variable that represents the center wise component of student_vs_allotted_city(s,c) i.e student_vs_allotted_city_center(s,c,e).
  - student_vs_allotted_city_center(s,c,e) represents whether center e is in city c and is allotted to student s.
  - To update this variable we bound it from both sides as follows. This sets its value accordingly whenever there is change in other variables.

    Constraint 4.1 -

$$\begin{aligned} & student\_vs\_allotted\_center(s, e) + center\_vs\_city(e, c) - 1 \leq \\ & student\_vs\_allotted\_city\_center(s, c, e), \;\; \forall s \in S, e \in E, c \in C \end{aligned}$$

Constraint 4.2 -

$$student\_vs\_allotted\_city\_center(s, c, e) \times 2 - 1 \leq$$
$$student\_vs\_allotted\_center(s, e) + center\_vs\_city(e, c) - 1,$$
$$\forall s \in S, e \in E, c \in C$$

- The above equations sets student_vs_allotted_city_center(s,c,e) to 1, only when center e lies in city c and center e is allotted to student s.
- In the above case,

$$student\_vs\_allotted\_center(s, e) = 1$$

and

$$center\_vs\_city(e, c) = 1$$

- Hence equation 7 reduces to,

$$1 \leq student\_vs\_allotted\_city\_center(s, c, e)$$

- This forces student_vs_allotted_city_center(s, c, e) to 1.
- For all other cases, student_vs_allotted_city_center(s, c, e) will be set to 0.
- Now, student_vs_allotted_city(s,c) can be simply calculated by taking the summation of student_vs_allotted_city_center(s, c, e) for all e.
  Constraint 4.3 -

$$student\_vs\_allotted\_city(s, c) =$$
$$\sum_{e \in E} student\_vs\_allotted\_city\_center(s, c, e), \ \forall s \in S, c \in C$$

- Hence student_vs_allotted_city(s, c) is set to 1 only when there is a center in city c which is allotted to student s.

- Prioritized class students should be given more priority and should be allotted their best preference possible.
  - To express this constraint mathematically, we first need to identify the preference given by a student to the city which contains the center he/she is allotted to.
  - For this, we introduced a new variable preference_city_allotted(s) which can be calculated as follows.
    Constraint 5.1 -

$$citywise\_preference\_table(s, c) +$$
$$\{number\_of\_preferences\_taken \times$$
$$(student\_vs\_allotted\_city(s, c) - 1)\} \leq$$
$$preference\_city\_allotted(s), \ \forall s \in S, c \in C$$

Constraint 5.2 -

$$preference\_city\_allotted(s) +$$
$$(student\_vs\_allotted\_city(s, c) - 1) \times large\_value \leq$$
$$citywise\_preference\_table(s, c) +$$
$$\{number\_of\_preferences\_taken \times$$
$$(student\_vs\_allotted\_city(s, c) - 1)\}, \ \forall s \in S, c \in C$$

  - The above equations set preference_city_allotted(s) equal to citywise_preference_table(s, c) for the city c that is allotted to student s.
  - For example, if student s is allotted to city c, then student_vs_allotted_city(s, c) = 1. Hence according to above 2 equations, value of preference_city_allotted(s) has to become equal to citywise_preference_table(s, c).
  - Now as we have the variable preference_city_allotted(s), we can write the equation for the above mentioned constraint as follows,

    Constraint 5.3 -

$$\frac{0.1((preference\_city\_allotted(s) - citywise\_preference\_table(s, c))}{number\_of\_preferences\_taken}$$
$$+ city\_class\_preference\_comparision(s, ss, c) -$$
$$student\_vs\_allotted\_city(s, c) + student\_vs\_allotted\_city(ss, c) \leq 2,$$
$$\forall s, ss \in S, c \in C$$

  - The above equation ensures that if student s is prioritized than student ss and has a higher or equal preference(a lower or equal preference value) for city c as compared to ss and preference of student s for city c allotted to him is higher than his preference for current city c into consideration, then student s should also be allotted to city c if student ss is.
    In the case when student s is not allotted to city c and all the other conditions hold, equation becomes,

$$0.1 \times (some\, positive\, value) + 1 - 0 + 1 \leq 2$$

    For the equation to satisfy, either student s has to be allotted to city c or student ss should be reassigned some other city.

- Prioritized student should be given superior center if available.
  - To express this constraint mathematically, we first need to identify the rating of the center that is allotted to a student i.e rating_center_allotted(s).
    Constraint 6.1 -

$$rating(e) +$$
$$number\_of\_centers\_in\_city\_of\_this\_center(e) \times$$
$$(student\_vs\_allotted\_center(s, e) - 1) \leq$$
$$rating\_center\_allotted(s), \ \forall s \in S, e \in E$$

    Constraint 6.2 -

$$rating\_center\_allotted(s) +$$
$$(student\_vs\_allotted\_center(s, e) - 1) \times large\_value \leq$$
$$rating(e) +$$
$$number\_of\_centers\_in\_city\_of\_this\_center(e) \times$$
$$(student\_vs\_allotted\_center(s, e) - 1), \ \forall s \in S, e \in E$$

  - The above equations set the value of rating_center_allotted(s).
  - For example if student s is allotted to center e, then student_vs_allotted_center(s, e) = 1. Hence according to above 2 equations, value of rating_center_allotted(s) becomes equal to rating(e).

- Now as we have the values of rating_center_allotted(s), the above constraint can be formulated as follows.
  Constraint 6.3 -
  $$\{(class(ss) - class(s))\times$$
  $$(rating\_center\_allotted(ss) - rating\_center\_allotted(s))\times$$
  $$small\_value\}+$$
  $$student\_vs\_allotted\_city(s, c)+$$
  $$student\_vs\_allotted\_city(ss, c) \leq 2,$$
  $$\forall s, ss \in S, c \in C$$
- The above equation ensures that if two students s and ss are allotted centers in the same city c, and if student s is prioritized than ss, then student s is allotted a center with higher or equal rating as compared to ss.

- One of the main objectives of this problem is to reduce the number of centers in use.
  - For this, we first need to check if a center is in use or not. We introduce a variable is_center_in_use(e) for this.
    Constraint 7.1 -
    $$\sum_{s\in S} student\_vs\_allotted\_center(s, e) \leq$$
    $$is\_center\_in\_use(e) \times large\_value, \ \ \forall e \in E$$
    Constraint 7.2 -
    $$is\_center\_in\_use(e) \leq$$
    $$\sum_{s\in S} student\_vs\_allotted\_center(s, e)), \ \ \forall e \in E$$
  - The above equations set the value of is_center_in_use(e) for all centers.
  - For example, If number of students allotted to a center e is greater than 0, then according to the constraint 7.1, is_center_in_use(e) is forced to set 1. Similarly if number of students allotted to a center e is equal to 0, then according to the constraint 7.2, is_center_in_use(e) is forced to go to 0.
  - Now, we can calculate number_of_centers_in_use by simply adding is_center_in_use(e) for all centers. Constraint 7.3 -
    $$number\_of\_centers\_in\_use = \sum_{e\in E} is\_center\_in\_use(e)$$

Objective Function:

The objective is to reduce the total number of centers in use i.e , the value of number_of_centers_in_use. In order to make sure that students are allotted their best preferences possible, we add a penalty to the objective function.

- Penalty is to be given whenever a student is allotted anything except his first preference.
- The penalty has to be more for prioritized class students. Hence, penalty is scaled by a value λ which varies as per the class of students, superior students having larger λ value given by,
  $$\lambda = ((number\_of\_classes + 1) - class(s)) \times \alpha$$
  where α is a large value.
- Hence the final objective function becomes,
  $$objective\_function = number\_of\_centers\_in\_use+$$
  $$\sum_{s\in S}(preference\_city\_allotted(s) - 1)\times$$
  $$((number\_of\_classes + 1) - class(s)) \times \alpha)$$

Scalar Values:

- The α value used in objective function calculation has to be 1 order higher than the total number of centers over all cities and can be calculated as follows,
  $$\alpha = 10 \times total\ number\ of\ centers\ over\ all\ cities$$
- The value of large_value mentioned in the constraints should be chosen such that it obeys the following conditions.
  - The large_value variable used in constraints 3.3 and 7.1 has to be bounded by the maximum capacity over all centers by the following equation.
    $$large\_value \geq max\{capacity(e)\}$$
  - The large_value variable used in constraints 5.2 has to be 1 order higher than the total number of preferences taken from each student.
    $$large\_value = 10 \times \{number\ of\ preferences\ taken\}$$
  - The large_value variable used in constraint 6.2 has to be 1 order higher than the maximum number of centers in any city.
    $$large\_value = 10 \times \{maximum\ number\ of\ centers\ in\ any\ city\}$$

  A single $large\_value$ variable can be used for all the constraints by choosing the maximum value among the above mentioned constraints on $large\_value$.

- The value of small_value mentioned in the constraints should be chosen such that it obeys the following conditions.
  - The small_value variable used in constraints 3.1 and 3.2 has to be bounded by the maximum capacity over all centers by the following equation.
    $$small\_value \leq 1/(max\{capacity(e)\} \times 10)$$
  - The small_value variable used in constraint 6.3 has to be bounded by the maximum number of centers in a city and the total number of classes of students by.
    $$small\_value = \frac{1}{(\{total\ no.\ of\ classes\} \times \{max\ no.\ of\ centers\ in\ any\ city\} \times 10)}$$

  A single $small\_value$ variable can be used for all the constraints by choosing the minimum value among the above mentioned constraints on $small\_value$.

IV. Example

In this section, one example of exam center allotment problem is demonstrated to show the working of the proposed approach.

Consider an example when we have 4 cities in a zone. Distribution of centers within the cities is as follows :

Center vs City				
	City 1	City 2	City 3	City 4
Center 1	1	0	0	0
Center 2	0	1	0	0
Center 3	0	0	0	1
Center 4	0	0	1	0
Center 5	0	0	1	0
Center 6	0	0	1	0

A value of 1 in the table indicates that center(i) is in city(j). So, Center 1 is in City 1, Center 3 in City 4 and so on.

Let say 8 students are to be allotted seats in these centers. Their classes are as follows :

class of students	
Student 1	1
Student 2	2
Student 3	2
Student 4	3
Student 5	3
Student 6	4
Student 7	4
Student 8	4

class 1 student is more prioritized than class 2, class 2 to class 3 and so on.

The capacities of each center are as follows :

Capacity of centers	
Center 1	1
Center 2	3
Center 3	2
Center 4	1
Center 5	1
Center 6	2

The center ratings are as given below :

Rating of centers	
Center 1	1
Center 2	1
Center 3	1
Center 4	1
Center 5	2
Center 6	3

Center 6 is superior to center 5, center 4 to center 3 and so on, since superior centers have higher rating.

The preferences of students for cities are listed below :

Students preferences for cities				
	City 1	City 2	City 3	City 4
Student 1	1	2	3	0
Student 2	1	0	3	2
Student 3	0	2	1	3
Student 4	1	3	0	2
Student 5	1	3	0	2
Student 6	0	1	3	2
Student 7	2	0	1	3
Student 8	2	0	3	1

A preference value of 0 indicates that, this particular student should not be allotted a center in that city.
Lower preference value denotes better choice. So, student 1 wants a center in city 1, if not available, then in city 2, if not then in city 3 and not at all in city 4.

The final allotment by the model proposed is given below :

Student vs Center allotted						
	Center1	Center2	Center3	Center4	Center5	Center6
Student1	1	0	0	0	0	0
Student2	0	0	1	0	0	0
Student3	0	0	0	0	0	1
Student4	0	1	0	0	0	0
Student5	0	0	1	0	0	0
Student6	0	1	0	0	0	0
Student7	0	0	0	0	0	1
Student8	0	0	0	0	1	0

The optimal solution has an objective value of 11005. Total number of centers used = 5. It can be verified that this is the most optimal allotment taking into account the constraints.

V. Conclusions

In this paper, we have introduced a novel mathematical framework to address the Exam Center Allotment problem which is formulated as a preference-constrained generalized assignment problem.

The proposed model allows for the identification of a zone wise exam center allotment of students considering their preferences, so as to reduce the number of centers that are being used. It also takes priority among students and centers into account, so that prioritized class students are given higher priority and superior centers are chosen over inferior centers. The resulting mathematical formulation results in a mixed-integer linear programming problem and can be solved to optimality. A computational example is also presented to demonstrate the effectiveness of the proposed approach.

The mathematical formulation proposed is scalable to any number of students, cities and centers available. It is also generalised for any number of classes of students into which students are divided. The classification of students into classes and the decision of priority among them is left to the user.

The proposed model can be used to allot exam centers for any large scale exams including the Joint Entrance Exam(JEE) for engineering which sees participation from lakhs of students.

REFERENCES

[1] Stacy L. Janak, Martin S. Taylor, and Christodoulos A. Floudas, "Novel and Effective Integer Optimization Approach for the NSF Panel-Assignment Problem: A Multiresource and Preference-Constrained Generalized Assignment Problem" Ind. Eng. Chem. Res. 2006, 45, 258-265

[2] Chauhan S.S., Kotecha P. (2020) Single-Level Production Planning in Petrochemical Industries Using Novel Computational Intelligence Algorithms. In: Bennis F., Bhattacharjya R. (eds) Nature-Inspired Methods for Metaheuristics Optimization. Modeling and Optimization in Science and Technologies, vol 16. Springer, Cham. https://doi.org/10.1007/978-3-030-26458-1_13

[3] Covers of GAMS User's Guide and Solver Manual