A Model for Assigning Teaching Assistants to Courses

Mansi Goel

Department of Mathematics
School of Natural Sciences
Shiv Nadar University

March 26, 2021

Table of contents

1. Linear Programming
2. Integer Programming
3. TA Assignment Problem

Linear Program

Definition

A **linear program** is comprised of a set of variables, a system of linear equations and inequalities which constrain the variables, and an objective function of the variables which is to be maximized or minimized.

Linear Program

Definition

A **linear program** is comprised of a set of variables, a system of linear equations and inequalities which constrain the variables, and an objective function of the variables which is to be maximized or minimized.

$$
\begin{aligned}
\max \quad & 2x_1 + 4x_2 \\
\text{subject to} \quad & \\
& 3x_1 + 5x_2 \leq 15 \\
& 3x_1 + 2x_2 \leq 12 \\
& x_1, x_2 \geq 0.
\end{aligned}
$$

Shiv Nadar University

Solution of a Linear Program

A **half space** is a set $\{ x \in \mathbb{R}^n : Ax \leq \alpha \}$ or $\{ x \in \mathbb{R}^n : Ax \geq \alpha \}$ for any given $A \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$.

Then, we define a **polyhedron** as a finite intersection of half spaces.

The set of possible solutions of a linear program forms a polyhedron. The solution set of an LP is knows as the **feasible region**.

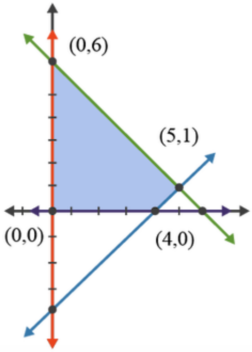
The constraints of a linear program can each be realized as half-spaces, and so their intersection forms a polyhedron. All feasible solutions are encapsulated by the polyhedron.

SHIV NADAR UNIVERSITY

For example,

$$\textit{constraints:} \begin{cases} x + y \geq 6 \\ x - y \geq 4 \\ x \geq 0 \\ y \geq 0 \end{cases}$$

$$\textit{Objective Function:} \quad C = 2x + y$$



The Simplex Method

The simplex method provides an algorithm to traverse a polyhedron to find an optimal solution. The central idea behind the simplex method is that optimal solutions are found among the vertices of the polyhedron. It methodically examines the value at each vertex to find the value that yields the highest/lowest objective value depending on whether it's a maximization or minimization problem.

The Simplex Method

The simplex method provides an algorithm to traverse a polyhedron to find an optimal solution. The central idea behind the simplex method is that optimal solutions are found among the vertices of the polyhedron. It methodically examines the value at each vertex to find the value that yields the highest/lowest objective value depending on whether it's a maximization or minimization problem.

This is done by moving from vertex to vertex of the polyhedron, and checking each adjacent vertex. If every neighboring vertex decreases the objective value or does not increase the objective value(maximization problem),the process finishes and the current vertex is the optimal solution.

Example

Question

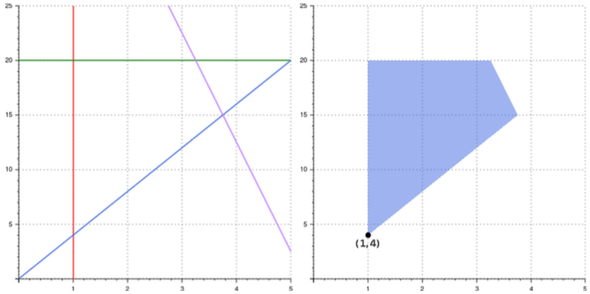
Consider a farmer who is trying to decide what animals she should buy to raise on her farm. She wants to buy cows and chickens, and would like to purchase as many total animals as she can with the money she has. Say cows cost $ 200 each and chickens cost $20 each, and the farmer has $1050 to spend. Assume our farmer wants at least one cow and at least four times as many chickens as cows, but her coop will hold no more than 20 chickens.

$$
\begin{aligned}
\max \quad & x + y \\
\text{s.t.} \quad 200x + 20y &\leq 1050 \\
x &\geq 0 \\
y &\geq 4x \\
y &\leq 20
\end{aligned}
$$

SHIV NADAR UNIVERSITY

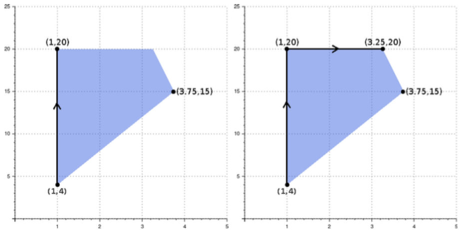
Example

Graphing the constraints;



Example

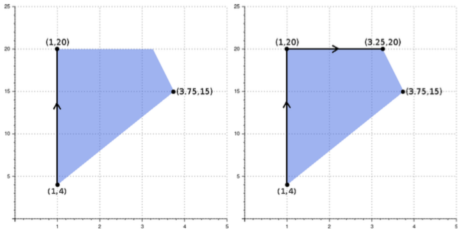
Starting with any arbitrary vertex, we find the optimal solution by traversing the vertices until we cannot improve our objective value by traveling to a new vertex.



- Starting with (1,4), now we can move upwards to (1,20) to get 21 or to the right diagonal to get 18.75. We choose to move up.

Example

Starting with any arbitrary vertex, we find the optimal solution by traversing the vertices until we cannot improve our objective value by traveling to a new vertex.

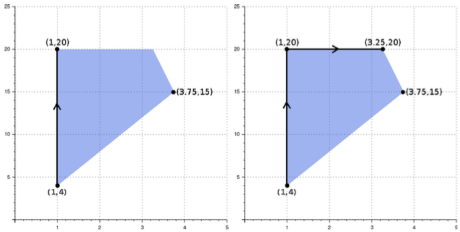


- Starting with (1,4), now we can move upwards to (1,20) to get 21 or to the right diagonal to get 18.75. We choose to move up.
- We again move to the right to get 23.25.

Shiv Nadar University

Example

Starting with any arbitrary vertex, we find the optimal solution by traversing the vertices until we cannot improve our objective value by traveling to a new vertex.

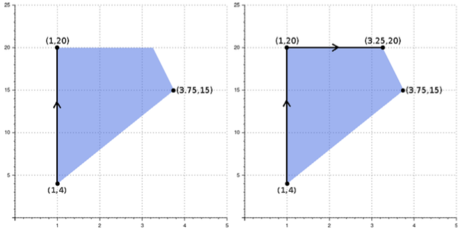


- Starting with (1,4), now we can move upwards to (1,20) to get 21 or to the right diagonal to get 18.75. We choose to move up.
- We again move to the right to get 23.25.
- We cannot move further as all neighbours have strictly lower objective function values.

SHIV NADAR UNIVERSITY

Example

Starting with any arbitrary vertex, we find the optimal solution by traversing the vertices until we cannot improve our objective value by traveling to a new vertex.



- Starting with (1,4), now we can move upwards to (1,20) to get 21 or to the right diagonal to get 18.75. We choose to move up.
- We again move to the right to get 23.25.
- We cannot move further as all neighbours have strictly lower objective function values.

Thus, our optimal solution is **(3.25,20)**

Example

From the example above, we observe that the farmer should purchase 3.25 cows and 20 chickens.

As one might expect, it is unrealistic for the farmer to purchase one quarter of a cow. Instead we must find some way to extract a **feasible integer solution**. This can be done using the **branch and bound method**.

Integer Programming

Definition

As observed in the above example, many real life problems when modelled as linear programming problems, require that some or all of the variables be integers. Such problems are called **integer linear programming problems**.

For Example,

$$\begin{array}{ll} \text{maximize} & 6x + 5y \\ \text{subject to} & 2x + 3y \leq 7 \\ & 2x - y \leq 2 \\ & x \geq 0,\ y \geq 0 \\ & x \text{ integer},\ y \text{ integer} \end{array}$$

Shiv Nadar University

Branch and Bound Method

Branch and Bound method is an efficient way to examine all possible integer feasible points and arrive at an integer optimal solution depending on which variables are constrained to be integers.

Branch and Bound Method

Branch and Bound method is an efficient way to examine all possible integer feasible points and arrive at an integer optimal solution depending on which variables are constrained to be integers.

We define an **associate LP** in which we allow the variables to take real values instead of integer values and we can arrive at an optimal integer solution by solving a sequence of real-valued linear programs.

Branch and Bound Method

- We solve the associate LP and if we achieve an optimal integer solution, we stop.

Branch and Bound Method

- We solve the associate LP and if we achieve an optimal integer solution, we stop.
- Otherwise, we branch our problem into two LPs based on the variable which is constrained to be an integer but has a fractional value.

Branch and Bound Method

- We solve the associate LP and if we achieve an optimal integer solution, we stop.
- Otherwise, we branch our problem into two LPs based on the variable which is constrained to be an integer but has a fractional value.
- Solve the two branched problems and compare it to the optimal value we currently have.

Branch and Bound Method

- We solve the associate LP and if we achieve an optimal integer solution, we stop.
- Otherwise, we branch our problem into two LPs based on the variable which is constrained to be an integer but has a fractional value.
- Solve the two branched problems and compare it to the optimal value we currently have.
- If we find a solution that gives a better objective value than our optimal solution, but is not integer, we repeat the process.

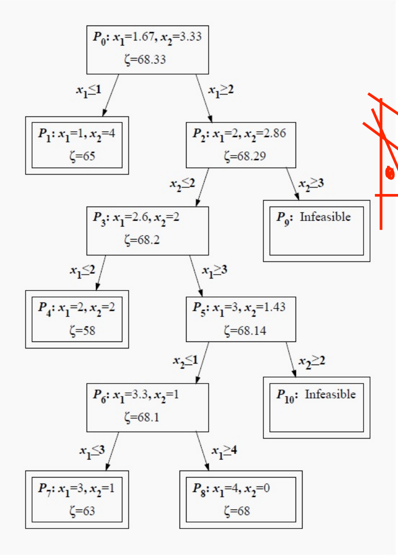
SHIV NADAR UNIVERSITY

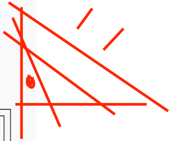
Branch and Bound Method

- We solve the associate LP and if we achieve an optimal integer solution, we stop.
- Otherwise, we branch our problem into two LPs based on the variable which is constrained to be an integer but has a fractional value.
- Solve the two branched problems and compare it to the optimal value we currently have.
- If we find a solution that gives a better objective value than our optimal solution, but is not integer, we repeat the process.
- All sub problems that yield worse objective values than the current optimal integer solution will be eliminated.

In this way, we eventually reach an optimal integer solution.

Branch and Bound Method





Branch and Bound Method

We now revisit the farmer problem example and we will now use branch and bound to attain an optimal integer solution.

Branch and Bound Method

We now revisit the farmer problem example and we will now use branch and bound to attain an optimal integer solution.

On solving the LP, we had got our solution as $x = 3.25$ cows, $y = 20$ chickens, and an objective function value of $x + y = 23.25$.

Branch and Bound Method

We now revisit the farmer problem example and we will now use branch and bound to attain an optimal integer solution.

On solving the LP, we had got our solution as $x = 3.25$ cows, $y = 20$ chickens, and an objective function value of $x + y = 23.25$.

Seeing this has a non-integer solution, we now apply branch and bound. Bounding our solution, we will now branch into two sub cases: $x = 3$ and $x = 4$.

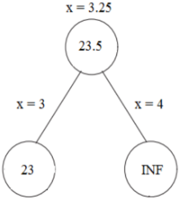
SHIV NADAR UNIVERSITY

Branch and Bound Method

With $x = 3$ we apply the simplex method again to get an objective function value of 23 with $y = 20$. Since both x and y are integral, we hold 23 as our current optimal integer solution while we test other cases.

Branch and Bound Method

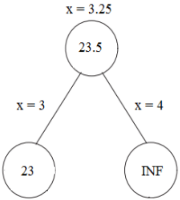
With x = 3 we apply the simplex method again to get an objective function value of 23 with y = 20. Since both x and y are integral, we hold 23 as our current optimal integer solution while we test other cases.
Likewise with x = 4 we apply the simplex method and find that the solution is infeasible. This is because when the farmer purchases 4 cows, she must also purchase at least 16 chickens, which costs her $ 1120 total, exceeding her budget.



Shiv Nadar University

Branch and Bound Method

With x = 3 we apply the simplex method again to get an objective function value of 23 with y = 20. Since both x and y are integral, we hold 23 as our current optimal integer solution while we test other cases.
Likewise with x = 4 we apply the simplex method and find that the solution is infeasible. This is because when the farmer purchases 4 cows, she must also purchase at least 16 chickens, which costs her $ 1120 total, exceeding her budget.



Thus, we have obtained our optimal solution as 3 cows, 20 chickens and 23 total animals on the farm.

Introduction

Every quarter, university students attend discussion sections led by graduate students in the hope of better understanding the content from the lectures.

The graduate students, called teaching assistants (hereafter referred to as TAs) who lead these sections are assigned by hand by the department.

Introduction

Every quarter, university students attend discussion sections led by graduate students in the hope of better understanding the content from the lectures.

The graduate students, called teaching assistants (hereafter referred to as TAs) who lead these sections are assigned by hand by the department.

The manual assignment process is usually carried out in two steps :

- The staff surveys all graduate students to find out what classes they would prefer to teach and when they are unavailable to teach.
- Match the graduate students to the available classes, trying to produce a fair and balanced assignment.

Introduction

Some of the disadvantages of a manually generated schedule include:

- Time spent by the staff

Introduction

Some of the disadvantages of a manually generated schedule include:

- Time spent by the staff
- Lack of consistency from year to year

Introduction

Some of the disadvantages of a manually generated schedule include:

- Time spent by the staff
- Lack of consistency from year to year
- Introduction of human bias

Introduction

Some of the disadvantages of a manually generated schedule include:

- Time spent by the staff
- Lack of consistency from year to year
- Introduction of human bias

A mathematical model can overcome these disadvantages and produce an impartial, consistent assignment in less time which could prove as an excellent resource for the staff responsible for assigning the TAs.

We now begin modelling our TA assignment problem.

Modelling Happiness

In order to define TA happiness, we considered five main components:

- Which courses the TA prefers to teach.

Modelling Happiness

In order to define TA happiness, we considered five main components:

- Which courses the TA prefers to teach.
- Which times of the day the TA prefers to teach.

Modelling Happiness

In order to define TA happiness, we considered five main components:

- Which courses the TA prefers to teach.
- Which times of the day the TA prefers to teach.
- Whether the TA prefers to teach sections from the same course.

Modelling Happiness

In order to define TA happiness, we considered five main components:

- Which courses the TA prefers to teach.
- Which times of the day the TA prefers to teach.
- Whether the TA prefers to teach sections from the same course.
- Whether the TA prefers to teach sections on the same day.

Modelling Happiness

In order to define TA happiness, we considered five main components:

- Which courses the TA prefers to teach.
- Which times of the day the TA prefers to teach.
- Whether the TA prefers to teach sections from the same course.
- Whether the TA prefers to teach sections on the same day.
- Whether the TA prefers to teach back-to-back sections.

Each of these components factors into our objective function in a different way, but contributes to the overall satisfaction of the TAs.

We will now model each of these components individually.

Course Preference

Each TA has their own mathematical interests that lead them to find certain undergraduate courses more enjoyable to teach than others. We define the following notations :

T : set of TAs

C : set of courses

$i \in$ T is a TA

k $\in$ C is a course

x_{ij} : binary assignment variable to represent whether TA i $\in$ T is assigned to section j $\in$ S.

$\star$: denotes a course that is preferred by that TA

$\diamond$: indicates indifference

$\otimes$: denotes a course that is not preferred by that TA

Shiv Nadar University

Course Preference

Now we give each TA-course pair a numerical value p_{ik} on a scale from 0 to 100 that represents how much TA i likes course k.

A score of 100 is awarded to the TA's favorite class, 0 to the TA's least favorite class,50 to all courses towards which the TA is indifferent. We assign a normalized value to every other course based on the total number of courses.

Course Preference

For example,

Diane
- $\star$ Combinatorics
- $\star$ Analysis
- $\star$ Algebra
- $\otimes$ Calculus

Diane gives Combinatorics a 100. Then, Analysis a 83.3 and Algebra a 66.6, since these two numbers evenly split the interval between 50 and 100 into three equal parts.

Course Preference

For example,

Diane
- ⋆ Combinatorics
- ⋆ Analysis
- ⋆ Algebra
- ⊗ Calculus

Diane gives Combinatorics a 100. Then, Analysis a 83.3 and Algebra a 66.6, since these two numbers evenly split the interval between 50 and 100 into three equal parts.

Now define Φ_{ik} the (positive) rank that TA i gives course k if i likes k and ϕ_{ik} the (negative) rank i gives k if i dislikes k. Then to account for indifference and normalizing, we let L_i the set of courses liked by i, D_i the set of courses disliked by i, and then formally define p_{ik} as follows:

$$p_{ik} = \begin{cases} 100 - \frac{50}{|L_i|}(\Phi_{ik} - 1); & i \text{ likes } k; \\ 50; & i \text{ is indifferent towards } k; \\ 0 + \frac{50}{|D_i|}(\phi_{ik} - 1); & i \text{ dislikes } k. \end{cases}$$

Shiv Nadar University

Time of Day Preference

As with courses, TAs have (often strong) preferences about what times of day they prefer to hold discussion sections. We can use our ranking model from above to assign numerical values to these preferences as well.For simplicity, we will divide a day into **Morning, Afternoon, and Evening**.

Time of Day Preference

As with courses, TAs have (often strong) preferences about what times of day they prefer to hold discussion sections. We can use our ranking model from above to assign numerical values to these preferences as well.For simplicity, we will divide a day into **Morning, Afternoon, and Evening**.

Again, we use $\star$ to denote preferred times to teach,$\otimes$ to denote disliked times to teach, and $\diamond$ to denote indifference.

Time of Day Preference

As with courses, TAs have (often strong) preferences about what times of day they prefer to hold discussion sections. We can use our ranking model from above to assign numerical values to these preferences as well.For simplicity, we will divide a day into **Morning, Afternoon, and Evening**.

Again, we use $\star$ to denote preferred times to teach,$\otimes$ to denote disliked times to teach, and $\diamond$ to denote indifference.

As before, define Ψ_{it} the (positive) rank that TA i gives time t if i likes t and ψ_{it} the (negative) rank i gives t if i dislikes t.These are calculated in the same way as $\Phi and \phi$.

We also account for indifference and normalizing the same way: given T_i the set of times liked by i, and τ_i the set of times disliked by i, we define the time preference pit as follows:

Shiv Nadar University

Time of Day Preference

$$p_{it} = \begin{cases} 100 - \frac{50}{|T_i|}\left(\Psi_{it} - 1\right); & i \text{ likes } t; \\ 50; & i \text{ is indifferent towards } t; \\ 0 + \frac{50}{|\tau_i|}\left(\psi_{it} - 1\right); & i \text{ dislikes } t; \end{cases}$$

Time of Day Preference

$$p_{it} = \begin{cases} 100 - \frac{50}{|T_i|}\left(\Psi_{it} - 1\right); & i \text{ likes } t; \\ 50; & i \text{ is indifferent towards } t; \\ 0 + \frac{50}{|\tau_i|}\left(\psi_{it} - 1\right); & i \text{ dislikes } t; \end{cases}$$

For example, Diane gives Evening a 100, Afternoon a 75 and 0 to Morning.

Diane

$\star$ Evening

$\star$ Afternoon

$\otimes$ Morning

Section Preference

A given section j has both a corresponding course k and a corresponding time of day t. So for a TA i, we have two values to look at when considering section preference, namely p_{ik} and p_{it}. These two values have different weights to different TAs.

Section Preference

A given section j has both a corresponding course k and a corresponding time of day t. So for a TA i, we have two values to look at when considering section preference, namely p_{ik} and p_{it}. These two values have different weights to different TAs.

Thus to define p_{ij}, the preference of TA i for section j, we have implemented a weight system with the following five choices:

$$p_{ij} = \begin{cases} p_{ik}; & i \text{ is only concerned with the course material of a section;} \\ \frac{2*p_{ik}+p_{it}}{3}; & i \text{ values the course material of a section higher than the time of day;} \\ \frac{p_{ik}+p_{it}}{2}; & i \text{ values the time of day and course material of a section equally;} \\ \frac{p_{ik}+2*p_{it}}{3}; & i \text{ values the time of day of a section higher than the course material;} \\ p_{it}; & i \text{ is only concerned with the time of day of a section..} \end{cases}$$

Section Preference

A given section j has both a corresponding course k and a corresponding time of day t. So for a TA i, we have two values to look at when considering section preference, namely p_{ik} and p_{it}. These two values have different weights to different TAs.

Thus to define p_{ij}, the preference of TA i for section j, we have implemented a weight system with the following five choices:

$$p_{ij} = \begin{cases} p_{ik}; & i \text{ is only concerned with the course material of a section;} \\ \frac{2*p_{ik}+p_{it}}{3}; & i \text{ values the course material of a section higher than the time of day;} \\ \frac{p_{ik}+p_{it}}{2}; & i \text{ values the time of day and course material of a section equally;} \\ \frac{p_{ik}+2*p_{it}}{3}; & i \text{ values the time of day of a section higher than the course material;} \\ p_{it}; & i \text{ is only concerned with the time of day of a section..} \end{cases}$$

In this way, we can gather p_{ij} for every TA i and section j but this is only part of the data we need; as we will see, it is not just the individual courses but rather the combination of courses that makes a TA teaching schedule good or bad.

Sections from The Same Course

One of the most important components to the happiness of several TAs is being able to teach sections from the same course. Presumably, the fewer distinct courses among sections a TA is teaching, the less time that TA must spend preparing material from different topics. We model this as follows :

Sections from The Same Course

One of the most important components to the happiness of several TAs is being able to teach sections from the same course. Presumably, the fewer distinct courses among sections a TA is teaching, the less time that TA must spend preparing material from different topics. We model this as follows :

Let C the set of courses, $i \in T$ a TA, $j \in S$ a section, $k \in C$ a course, and $l \in C$ the course of section j. Then we define δ_{kl} a simple binary indicator of whether l and k are the same section

Section from the Same Course

Now, we define a binary variable q_{ik} such that it is **1 if TA i is assigned to at least one section of course k** and 0 otherwise. To force this, we write:

$$q_{ik} \geq \sum_{j \in S} \frac{x_{ij}}{m} \delta_{kl}.$$

Section from the Same Course

Now, we define a binary variable q_{ik} such that it is **1 if TA i is assigned to at least one section of course k** and 0 otherwise.To force this, we write:

$$q_{ik} \geq \sum_{j \in S} \frac{x_{ij}}{m} \delta_{kl}.$$

Where m is the maximum number of units a TA can teach whch is included here as a normalizer to ensure that q_{ik} does not exceed 1.This helps to force q_{ik} into its proper position - if TA i is assigned to teach section j and section j is from course k, then $l = k$ so we have that $\delta_{kl} = 1$ and $x_{ij} = 1$. So the right-hand side is strictly greater than zero and since q_{ik} is binary, this means it must be set to 1.

Shiv Nadar University

Sections from the Same Course

Now we can define, z_i : total number of distinct courses that TA i is teaching sections from as:

$$z_i = \sum_{k \in C} q_{ik}.$$

Shiv Nadar University

Sections from the Same Course

Now we can define, z_i : total number of distinct courses that TA i is teaching sections from as:

$$z_i = \sum_{k \in C} q_{ik}.$$

Remark : Every TA wants a z_i of 1, but that is more difficult for a TA who is teaching 4 sections versus one who is teaching only 1 section.

Sections from the Same Course

Now we can define, z_i : total number of distinct courses that TA i is teaching sections from as:

$$z_i = \sum_{k \in C} q_{ik}.$$

Remark : Every TA wants a z_i of 1, but that is more difficult for a TA who is teaching 4 sections versus one who is teaching only 1 section.

So, we also need to account for the total number of sections that TA i is teaching, which we denote by μ_i. We want a function f that rewards the LP for giving few courses to TAs with many sections. We take our f to be:

$$f(\mu_i, z_i) = \mu_i - z_i + 1.$$

We add the 1 to ensure that f never gives a value of zero. Thus the lowest value f can return is 1, which happens when $\mu_i = z_i$.

Sections from the Same Course

Now we can define, z_i : total number of distinct courses that TA i is teaching sections from as:

$$z_i = \sum_{k \in C} q_{ik}.$$

Remark : Every TA wants a z_i of 1, but that is more difficult for a TA who is teaching 4 sections versus one who is teaching only 1 section.

So, we also need to account for the total number of sections that TA i is teaching, which we denote by μ_i. We want a function f that rewards the LP for giving few courses to TAs with many sections. We take our f to be:

$$f(\mu_i, z_i) = \mu_i - z_i + 1.$$

We add the 1 to ensure that f never gives a value of zero. Thus the lowest value f can return is 1, which happens when $\mu_i = z_i$.

Note : As μ_i increases, we would like to keep z_i low. Since μ_i is unchangeable data, but z_i depends on the assignments that the LP makes, the LP will attempt to keep it small; that is, prevent TAs from teaching sections from too many distinct courses.

Sections on the Same Day

We want to define a measure for the number of days a TA must teach on. Some TAs might prefer to teach all of their sections on the same day, while others would rather spread their sections throughout the week. We model this as follows:

Sections on the Same Day

We want to define a measure for the number of days a TA must teach on. Some TAs might prefer to teach all of their sections on the same day, while others would rather spread their sections throughout the week. We model this as follows:

Let D be the set of days of the week, j a section and $d \in D$ a day of the week. Then , we define v_{jd} the binary indicator of whether section j is held on day d.

Sections on the Same Day

We want to define a measure for the number of days a TA must teach on. Some TAs might prefer to teach all of their sections on the same day, while others would rather spread their sections throughout the week. We model this as follows:

Let D be the set of days of the week, j a section and d $\in$ D a day of the week. Then , we define v_{jd} the binary indicator of whether section j is held on day d.

Now, we define a binary variable y_{id} such that it is **1 is TA i is teaching on day d** and 0 otherwise. To force this, we write:

$$y_{id} \geq \sum_{j \in S} \frac{x_{ij}}{m} v_{jd}.$$

Shiv Nadar University

Sections on the Same Day

We want to define a measure for the number of days a TA must teach on. Some TAs might prefer to teach all of their sections on the same day, while others would rather spread their sections throughout the week. We model this as follows:

Let D be the set of days of the week, j a section and d $\in$ D a day of the week. Then , we define v_{jd} the binary indicator of whether section j is held on day d.

Now, we define a binary variable y_{id} such that it is **1 is TA i is teaching on day d** and 0 otherwise. To force this, we write:

$$y_{id} \geq \sum_{j \in S} \frac{x_{ij}}{m} v_{jd}.$$

If TA i is teaching section j, and section j is held on day d, then $v_{jd} = 1$ and $x_{ij} = 1$, so the RHS is strictly greater than zero, which forces y_{id} to be 1 since it is binary.

Shiv Nadar University

Sections on the Same Day

Now we define v_i : the total number of days on which TA i teaches as:

$$v_i = \sum_{d \in D} y_{id}.$$

Sections on the Same Day

Now we define v_i : the total number of days on which TA i teaches as:

$$v_i = \sum_{d \in D} y_{id}.$$

As with number of courses, we need to normalize v_i over the number of sections μ_i. We use the same function as before:

$$g(\mu_i, v_i) = \mu_i - v_i + 1.$$

Here, as before, g will take a higher value when a TA with many sections has few days.

Back to Back Sections

Some TAs prefer to have their sections close to each other while others would rather have them spaced out.This is not the same as having sections on the same day: a TA could prefer to have sections on the same day but not immediately back-to-back. We model this as follows:

Back to Back Sections

Some TAs prefer to have their sections close to each other while others would rather have them spaced out.This is not the same as having sections on the same day: a TA could prefer to have sections on the same day but not immediately back-to-back. We model this as follows:

First, for a TA i and sections j, $j^{'}$, we define $v_{jj^{'}}$ the binary indicator of whether sections j and $j^{'}$ are back-to-back. This can be calculated ahead of time simply using section data.

Shiv Nadar University

Back to Back Sections

Next, we want to identify back-to-back sections assigned to TAs. For this, define $\beta_{ijj'}$ the binary indicator of whether TA i is teaching back-to-back sections j and $j^{'}$. To force $\beta_{ijj'}$ to take proper values, we write:

SHIV NADAR UNIVERSITY

Back to Back Sections

Next, we want to identify back-to-back sections assigned to TAs. For this, define $\beta_{ijj'}$ the binary indicator of whether TA i is teaching back-to-back sections j and j'. To force $\beta_{ijj'}$ to take proper values, we write:

$$\frac{1}{2}\nu_{jj'}(x_{ij} + x_{ij'}) \geq \beta_{ijj'} \geq \frac{1}{2}\nu_{jj'}(x_{ij} + x_{ij'} - 1).$$

Back to Back Sections

Next, we want to identify back-to-back sections assigned to TAs. For this, define $\beta_{ijj'}$ the binary indicator of whether TA i is teaching back-to-back sections j and j'. To force $\beta_{ijj'}$ to take proper values, we write:

$$\frac{1}{2}\nu_{jj'}(x_{ij} + x_{ij'}) \geq \beta_{ijj'} \geq \frac{1}{2}\nu_{jj'}(x_{ij} + x_{ij'} - 1).$$

• j *and* j' are back to back, TA i is teaching both j *and* j' • So, $1 \geq \beta_{ij} \geq \frac{1}{2} \Longrightarrow \beta_{ij} = 1$	• j *and* j' are back to back but TA i is not teaching both j *and* j'. • So, $0 \geq \beta_{ij} \geq -\frac{1}{2} \Longrightarrow \beta_{ij} = 0$
• j *and* j' are not back to back but TA i is teaching both j *and* j' • So, $0 \geq \beta_{ij} \geq 0 \Longrightarrow \beta_{ij} = 0$	• j *and* j' are not back to back and TA i is not teaching j *and* j' • So, $0 \geq \beta_{ij} \geq 0 \Longrightarrow \beta_{ij} = 0$

Shiv Nadar University

Back to Back Sections

Now, to calculate the total number of pairs of back-to-back sections that are assigned to TA i, we add up all the $\beta_{ijj'}$ and call it w_{ij}:

$$w_i = \sum_{(j,j') \in SxS} \beta_{ijj'}.$$

Back to Back Sections

Now, to calculate the total number of pairs of back-to-back sections that are assigned to TA i, we add up all the $\beta_{ijj'}$ and call it w_{ij}:

$$w_i = \sum_{(j,j')\in SxS} \beta_{ijj'}.$$

Once again, we normalize over the number of sections a TA is teaching, μ_i. We chose a function h defined by:

$$h(\mu_i, w_i) = \mu_i - w_i.$$

h returns a higher value when a TA has many sections but few back-to-back sections. We no longer need to add 1 here since $\mu_i > w_{ij}$.

Modelling Fairness

Now that we have an idea of how to model "happiness", we have to incorporate fairness into our model.

Modelling Fairness

Now that we have an idea of how to model "happiness", we have to incorporate fairness into our model.

Fairness can be described in terms of whether or not a TA deserves to be assigned to his or her preferred section. The staff of the Math department assign each TA a **numerical rank** R_i**, ranging from 0 to 5**. When assigning these ranks, the staff takes into account factors like **prior performance, ability, and seniority**.

Modelling Fairness

To see why fairness is an important piece of our model, consider two TAs, Edgar and Frank where Frank has a higher ranking than Edgar.

Modelling Fairness

To see why fairness is an important piece of our model, consider two TAs, Edgar and Frank where Frank has a higher ranking than Edgar.

- Frank prefers course $y^{'}$, then y and then other courses.
- Edgar prefers only course y over other courses.
- However, Frank's schedule will not allow him to teach course $y^{'}$.

SHIV NADAR UNIVERSITY

Modelling Fairness

To see why fairness is an important piece of our model, consider two TAs, Edgar and Frank where Frank has a higher ranking than Edgar.

- Frank prefers course y', then y and then other courses.
- Edgar prefers only course y over other courses.
- However, Frank's schedule will not allow him to teach course y'.

If the model was based on happiness alone, Frank would have to teach some other course only because y' had a time conflict while giving his second favorite course(y) to Edgar, who, based on his ranking, did not deserve to teach course y as much as Frank.

The Objective Function

Now that we have modelled all of the factors, we will incorporate these in the objective function whose objective is to **maximise the satisfaction of the TAs**. Or, in other words, we want to maximize the number of TAs that are assigned to sections they prefer given their rank.

The Objective Function

Now that we have modelled all of the factors, we will incorporate these in the objective function whose objective is to **maximise the satisfaction of the TAs**. Or, in other words, we want to maximize the number of TAs that are assigned to sections they prefer given their rank.

Restating the factors modelled above:

- Section preference p_{ij}.

Shiv Nadar University

The Objective Function

Now that we have modelled all of the factors, we will incorporate these in the objective function whose objective is to **maximise the satisfaction of the TAs**. Or, in other words, we want to maximize the number of TAs that are assigned to sections they prefer given their rank.

Restating the factors modelled above:

- Section preference p_{ij}.
- Teaching sections from the same course, represented by f(μ_i ,z_i)

SHIV NADAR UNIVERSITY

The Objective Function

Now that we have modelled all of the factors, we will incorporate these in the objective function whose objective is to **maximise the satisfaction of the TAs**. Or, in other words, we want to maximize the number of TAs that are assigned to sections they prefer given their rank.

Restating the factors modelled above:

- Section preference p_{ij}.
- Teaching sections from the same course, represented by f(μ_i ,z_i)
- Teaching sections from the same day, represented by g(μ_i ,w_i)

The Objective Function

Now that we have modelled all of the factors, we will incorporate these in the objective function whose objective is to **maximise the satisfaction of the TAs**. Or, in other words, we want to maximize the number of TAs that are assigned to sections they prefer given their rank.

Restating the factors modelled above:

- Section preference p_{ij}.
- Teaching sections from the same course, represented by f(μ_i ,z_i)
- Teaching sections from the same day, represented by g(μ_i ,w_i)
- Teaching back-to-back sections, represented by h(μ_i ,v_i)

The Objective Function

As alluded before, TAs have different preferences, for example, some TAs prefer to teach sections on the same day, while others prefer not to. We account for these differences by including constants a_i, b_i, and c_i defined as:

The Objective Function

As alluded before, TAs have different preferences, for example, some TAs prefer to teach sections on the same day, while others prefer not to. We account for these differences by including constants a_i, b_i, and c_i defined as:

$$a_i = \begin{cases} 1 & \text{if TA } i \text{ prefers to teach sections of the same course} \\ 0 & \text{if TA } i \text{ is indifferent} \\ -1 & \text{if TA } i \text{ doesn't prefer to teach sections of the same course} \end{cases}$$

$$b_i = \begin{cases} 1 & \text{if TA } i \text{ prefers to teach sections on the same day} \\ 0 & \text{if TA } i \text{ is indifferent} \\ -1 & \text{if TA } i \text{ doesn't prefer to teach sections on the same day} \end{cases}$$

$$c_i = \begin{cases} 1 & \text{if TA } i \text{ prefers to teach sections that are back to back} \\ 0 & \text{if TA } i \text{ is indifferent} \\ -1 & \text{if TA } i \text{ doesn't prefer to teach sections that are back to back} \end{cases}$$

The Objective Function

As alluded before, TAs have different preferences, for example, some TAs prefer to teach sections on the same day, while others prefer not to. We account for these differences by including constants a_i, b_i, and c_i defined as:

$$a_i = \begin{cases} 1 & \text{if TA } i \text{ prefers to teach sections of the same course} \\ 0 & \text{if TA } i \text{ is indifferent} \\ -1 & \text{if TA } i \text{ doesn't prefer to teach sections of the same course} \end{cases}$$

$$b_i = \begin{cases} 1 & \text{if TA } i \text{ prefers to teach sections on the same day} \\ 0 & \text{if TA } i \text{ is indifferent} \\ -1 & \text{if TA } i \text{ doesn't prefer to teach sections on the same day} \end{cases}$$

$$c_i = \begin{cases} 1 & \text{if TA } i \text{ prefers to teach sections that are back to back} \\ 0 & \text{if TA } i \text{ is indifferent} \\ -1 & \text{if TA } i \text{ doesn't prefer to teach sections that are back to back} \end{cases}$$

Given all of the above preferences and constants, we now present the objective function:

$$\sum_{i \in T} R_i * \left(a_i * f(\mu_i, z_i) + b_i * g(u_i, w_i) + c_i * h(\mu_i, v_i) + \sum_{j \in S} x_{ij} p_{ij} \right).$$

Shiv Nadar University

Constraints

Not just any set of variables are feasible solutions for our problem. We need to introduce some constraints on the variables as follows:

Constraints

Not just any set of variables are feasible solutions for our problem. We need to introduce some constraints on the variables as follows:

1. For every section j, we have

$$\sum_{i \in T} x_{ij} = 1$$

which means that every section requires exactly one TA.

SHIV NADAR UNIVERSITY

Constraints

Not just any set of variables are feasible solutions for our problem. We need to introduce some constraints on the variables as follows:

1. For every section j, we have

$$\sum_{i \in T} x_{ij} = 1$$

   which means that every section requires exactly one TA.

2. We require that every TA, i, meets their teaching duties. To ensure each TA i teaches exactly μ_i sections we require:

$$\sum_{j \in S} x_{ij} = \mu_i$$

Constraints

Sections are divided into 3 categories - lower division UG, upper division UG and graduate. TAs in their first few quarters are restricted to teach lower division sections only. To formulate this constraints we use the notations - L_T - **set of TA only qualified to teach lower division sections**, U_T - **set of TA only qualified to teach UG sections**, U_S - **set of upper division UG sections**, G_S - **set of graduate sections**.

Constraints

Sections are divided into 3 categories - lower division UG, upper division UG and graduate. TAs in their first few quarters are restricted to teach lower division sections only. To formulate this constraints we use the notations - L_T - **set of TA only qualified to teach lower division sections**, U_T - **set of TA only qualified to teach UG sections**, U_S - **set of upper division UG sections**, G_S - **set of graduate sections**.

5. For TAs who were only allowed to teach lower division courses , we need:

$$\forall (i,j) \in L_T \times (U_S \cup G_S), x_{ij} = 0$$

Constraints

Sections are divided into 3 categories - lower division UG, upper division UG and graduate. TAs in their first few quarters are restricted to teach lower division sections only. To formulate this constraints we use the notations - L_T - **set of TA only qualified to teach lower division sections**, U_T - **set of TA only qualified to teach UG sections**, U_S - **set of upper division UG sections**, G_S - **set of graduate sections**.

7. For TAs who were only allowed to teach lower division courses , we need:

$$\forall (i,j) \in L_T \times (U_S \cup G_S), x_{ij} = 0$$

8. For TAs who were not allowed to teach graduate courses, we need:

$$\forall (i,j) \in (L_T \cup U_T) \times G_S, x_{ij} = 0$$

Shiv Nadar University

Constraints

9. For every TA i, we require that there is no overlap between the sections each TA teaches. For this, we define a binary indicator $\Omega_{jj'}$ which represents whether j and $j^{'}$ overlap. Now, we define another binary indicator $\xi_{ijj'}$ of whether TA i is teaching overlapping sections j and $j^{'}$. We force ξ to represent this by:

$$\frac{1}{2}\Omega_{jj'}\left(x_{ij} + x_{ij'}\right) \geq \xi_{ijj'} \geq \frac{1}{2}\Omega_{jj'}\left(x_{ij} + x_{ij'} - 1\right)$$

Now, to ensure this overlapping situation never occurs, we want, for every TA i:

$$\sum_{j,j' \in S \times S} \xi_{ijj'} = 0.$$

Shiv Nadar University

Size of the Model

Linear programs are most often measured by the number of variables they introduce and the number of constraints they enforce. By examining these numbers, one can get an idea of the scale of our model and how it might perform in practice.

Size of the Model

Linear programs are most often measured by the number of variables they introduce and the number of constraints they enforce. By examining these numbers, one can get an idea of the scale of our model and how it might perform in practice.

The Number of Variables:

- $x_{ij} \forall (i, j) \in T \times S$
- $q_{ik} \forall (i, k) \in T \times C$
- $y_{id} \forall (i, d) \in T \times D$
- $\beta_{ijj'} \forall (i, jj') \in T \times (S \times S)$

Size of the Model

Linear programs are most often measured by the number of variables they introduce and the number of constraints they enforce. By examining these numbers, one can get an idea of the scale of our model and how it might perform in practice.

The Number of Variables:

- $x_{ij} \forall (i, j) \in T \times S$
- $q_{ik} \forall (i, k) \in T \times C$
- $y_{id} \forall (i, d) \in T \times D$
- $\beta_{ijj'} \forall (i, jj') \in T \times (S \times S)$

So, a solution has to assign values to $|T| * \left(|S| + |C| + |D| + |S|^2\right)$ variables.

Size of the Model

The Number of Constraints

- $|T|$ constraints to ensure each TA meets his or her teaching duties.
- $|S|$ constraints to ensure each section has one TA.
- $|T| * |S|^2$ constraints ensure that no TA is assigned classes that occur at the same time.
- $|L_T| * (|U_S| + |G_S|)$ constraints ensure no TA unfit to teach Upper division and graduate level courses does so. Bound this above by $|T| * |S|$.
- $|U_T| * |G_S|$ constraints ensure no TA unfit to teach graduate courses does so. This can also be bounded above by $|T| * |S|$.
- We also have to take into account constraints due to TA schedule conflicts. Let this number of conflicts be Q.

Shiv Nadar University

Size of the Model

The Number of Constraints

- $|T|$ constraints to ensure each TA meets his or her teaching duties.
- $|S|$ constraints to ensure each section has one TA.
- $|T| * |S|^2$ constraints ensure that no TA is assigned classes that occur at the same time.
- $|L_T| * (|U_S| + |G_S|)$ constraints ensure no TA unfit to teach Upper division and graduate level courses does so. Bound this above by $|T| * |S|$.
- $|U_T| * |G_S|$ constraints ensure no TA unfit to teach graduate courses does so. This can also be bounded above by $|T| * |S|$.
- We also have to take into account constraints due to TA schedule conflicts. Let this number of conflicts be Q.

Then the total number of constraints is about
$|T| + |S| + |T| * |S|^2 + |T| * |S| + |T| * |S| + Q$.

SHIV NADAR UNIVERSITY

Conclusion

This paper gives an insight into how a real life problem can be modelled into a linear programming problem. However, a theoretical model is good on paper but it does not help the TAs or the staff unless it is made practically accessible.

For this purpose, this thesis has developed a web interface using SQLite and PHP. To solve the integer linear program, the SCIP (Solving Constraint Integer Programs) solver is used. The program has around about 20,000 variables and about 500,000 constraints. The software used allows the staff to reach an optimal assignment with less than 15 minutes of work.

The utilization of integer linear programming to solve the TA assignment problem gives one an idea of the power and versatility of integer linear programs.