Chapter 1

Linear Programming and Simplex algorithm

1.1 The problem

A *linear programming* (LP) over variables $x_1, x_2, \ldots, x_n$ consists of the following

1. a linear function called *objective function* $c_1x_1 + c_2x_2 + \cdots + c_nx_n$.
2. a *set* of linear constraints of the form $a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$, or $a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b$ or $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$

The minimizing linear programming problem (similarly maximizing) is to find an $x_1, x_2, \ldots, x_n \in \mathbb{R}$ which satisfies all the linear constraints and such that the objective function is minimized (similarly maximized) at that point. A linear programming problem with any kind of constraint is called an LP in general form. Here are two interesting special forms of LP.

Definition 1.1. *A linear programming problem is in* standard form*, if the linear constraints are as follows*

$$\begin{aligned} \mathbb{A}\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

Definition 1.2. *A linear programming problem is in* canonical form*, if the linear constraints are as follows*

$$\begin{aligned} \mathbb{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

For the lecture we will assume (if not mentioned) that the objective is to minimize the objective function. That is, we want to minimize $\mathbf{c}^\mathsf{T}\mathbf{x}$.

We denote a column vector by

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

and a row vector by $\mathbf{a}^\mathsf{T} = (a_1, a_2, \ldots, a_n)$. We denote by $\mathbb{R}$ the reals and by $\mathbb{R}^n$ the set of all vectors $\mathbf{a}$ where $\mathbf{a}^\mathsf{T} = (a_1, a_2, \ldots, a_n)$ where $a_i \in \mathbb{R}$. Usually vectors will be denoted by $\mathbf{a}, \mathbf{b}$ or $\mathbf{c}$. The exception is: for a real number $m \in \mathbb{R}$ we denote by $\mathbf{m} \in \mathbb{R}^n$ the vector $\mathbf{m}^\mathsf{T} = (m, m, \ldots, m)$ of all m's. The rows of a matrix $\mathbb{A}$ is denoted as

$$\begin{pmatrix} -- & \mathbf{a}_1{}^\mathsf{T} & -- \\ -- & \mathbf{a}_2{}^\mathsf{T} & -- \\ & \cdots & \\ -- & \mathbf{a}_m{}^\mathsf{T} & -- \end{pmatrix}$$

where the i^{th} row vector is $\mathbf{a}_i{}^\mathsf{T}$. The columns of matrix $\mathbb{A}$ is denoted as

$$\begin{pmatrix} \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_n \\ \vdots & \vdots & & \vdots \end{pmatrix}$$

where the j^{th} column vector is $\mathbf{A}_j$. The j^{th} element of the i^{th} row vector is $a_{ij} = A_{ji}$. The identity matrix is denoted by $\mathbb{I}$.

1.2 Math background - Convex sets

We say that $\mathbf{z} \in \mathbb{R}^n$ is a convex combination of vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ if there exists a $\lambda \in [0, 1]$ such that $\mathbf{z} = \lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$.

Definition 1.3. *A set $X \subseteq \mathbb{R}^n$ is a* convex *set if for all $\mathbf{x}, \mathbf{y} \in X$, and all $\lambda \in [0, 1]$ we have that $\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} \in X$.*

Some examples of convex sets (also see Figure 1.1 and 1.2).

1. A line is a convex set.

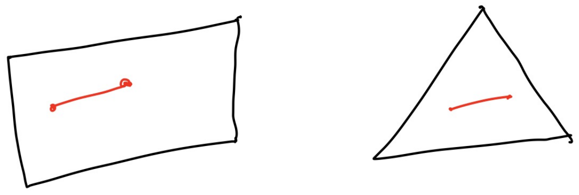

Figure 1.1: Convex sets

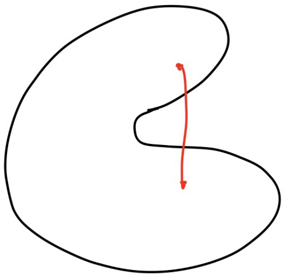
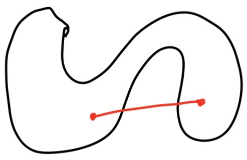

Figure 1.2: Not convex sets

2. A line segment

3. $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbb{A}\mathbf{x} = \mathbf{b}\}$

4. $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbb{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}\}$

Here are some properties of convex sets (try proving them).

Claim 1.1. *The following properties hold for convex sets.*

1. *Convex sets are closed under intersections.*

2. *A vector space is a convex set.*

A vector $\mathbf{x} \in X$ is an *extreme point* in the convex set X if $\mathbf{x}$ cannot be represented as a convex combination of two other points in X.

The *convex hull* of a set X is the least convex set Y which contains X (that is $X \subseteq Y$). For example, the convex hull of two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ is the line segment joining $\mathbf{x}$ and $\mathbf{y}$. Figure 1.3 and 1.4 shows the convex hull of points X.

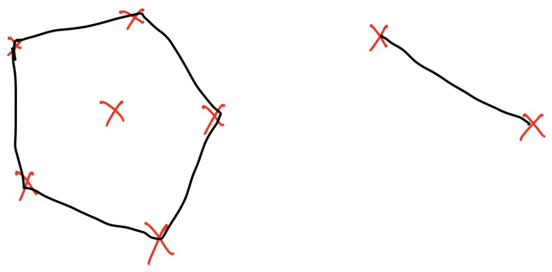

Figure 1.3: Convex Hull of the red points.

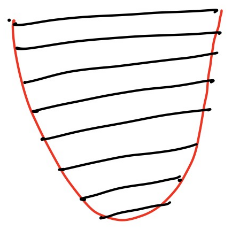

Figure 1.4: Convex Hull

1.3 Standard form LP - geometric view

1.4 Canonical form LP - algebraic view

Let $\mathbb{A}'$ be an $m \times n - m$ matrix. That is the matrix has m rows and $n - m$ columns. Let $\mathbf{c}' \in \mathbb{R}^{n-m}$ be a real vector. Here we have $n \geq m$.

Consider the LP in standard form. Find an $x \in \mathbb{R}^{n-m}$ where

$$\begin{aligned} \texttt{minimize}\ {\mathbf{c}'}^{\mathsf{T}}\mathbf{x}' & \\ \mathbb{A}'\mathbf{x}' &\leq \mathbf{b} \\ \mathbf{x}' &\geq \mathbf{0} \end{aligned}$$

This can be converted into an LP in canonical form as follows. For the i^{th} in-equality constraint ${\mathbf{a}'_i}^{\mathsf{T}}\mathbf{x}' \leq b_i$, we introduce a new variable x_{n-m+i} and add the following constraints

$$\begin{aligned} {\mathbf{a}'_i}^{\mathsf{T}}\mathbf{x}' + x_{n-m+i} &= b_i \\ x_{n-m+i} &\geq 0 \end{aligned}$$

This gives us the following LP in canonical form

$$\begin{aligned} \texttt{minimize } \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \mathbb{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Here $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{c}^{\mathsf{T}} = (\mathbf{c}'^{\mathsf{T}}, 0, 0, \ldots, 0)$ (This is an abuse of notation which means the last m elements of the vector $\mathbf{c}$ are 0s and the first elements correspond to that of $\mathbf{c}'$). The vector $\mathbf{x} \in \mathbb{R}^n$ and matrix $\mathbb{A}$ is of the following form.

$$\mathbb{A} = \left(\begin{array}{c|c} \\ \quad \mathbb{A}' \quad & \qquad \mathbb{I} \qquad \\ \\ \end{array} \right)$$

The set of all feasible points (called the feasible set) of this LP in the canonical form is

$$F_C = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbb{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq 0\}$$

This set is closely related to the feasible set of the LP in standard form. We now describe a mapping $f : F_S \to F_C$. Let $\mathbf{x}' \in F_S$. Now consider the vector $\mathbf{x} \in \mathbb{R}^n$ where x_j for a $j \leq n$ is given as

$$x_j = \begin{cases} x'_j \text{ if } j \leq n - m \\ b_i - \mathbf{a}_i'^{\mathsf{T}}\mathbf{x}' \text{ if } j = n - m + i \text{ and } i \leq m \end{cases}$$

The first claim is

Claim 1.2. *The above vector $\mathbf{x} \in \mathbb{R}^n$ belongs to F_C.*

Proof. This is an exercise. □

Therefore, the above mapping takes every vector $\mathbf{x}' \in F_S$ to a vector $\mathbf{x} \in F_C$. Let us represent this mapping by $f : F_S \to F_C$.

Claim 1.3. *Let $V \subseteq F_S$ be the set of extreme points in F_S. Then $f(V)$ is the set of extreme points in F_C.*

Proof. This is an exercise. □

We will now give an alternate (algebraic) definition of the extreme points of F_C. First a few definitions. For a vector $\mathbf{x} \in \mathbb{R}^n$ and a set $B \subseteq \{1, 2, \ldots, n\}$ we denote by $\mathbf{x}_{\overline{B}} = 0$ to mean $x_j = 0$ for all $j \notin B$. For example, consider $x = (2, 3, 0, 0)^{\mathsf{T}} \in \mathbb{R}^4$. Then $x_{\overline{\{1,2\}}} = 0$ since $x_3 = x_4 = 0$.

We say that $\mathbf{x} \in F_C$ is a *basic feasible point*, if there exists a set $B \subseteq \{1, 2, \ldots, n\}$ such that the set of column vectors $\{A_j \mid j \in B\}$ form a basis for the column space of $\mathbb{A}$ (denoted by $C(\mathbb{A})$) and $x_{\overline{B}} = 0$. In other words the vector b is represented as a positive linear combinations of a basis of $\mathbb{A}$. The set of all basic feasible point is called *basic feasible set* (BFS for short).

$$\text{BFS} = \{\mathbf{x} \in F_C \mid \mathbf{x} \text{ is a basic feasible point.}\}$$

The following assumption is going to be assumed henceforth.

Assumption I: The rows of $\mathbb{A}$ are independent. Note that the number of rows is represented by m.

Exercise 1.1. *Show that a basis for the column space of $\mathbb{A}$ has m vectors.*

Exercise 1.2. *Given a matrix where rows are dependent, give an equivalent LP with independent rows.*

From the above assumption, we note that the number of non-zeros in a vector $\mathbf{x} \in$ BFS is atmost m. The reason for at most m and not exact m is because it can happen that the co-efficient of a basis vector is also 0. Therefore the number of zeros is atleast $n - m$.

The next lemma is important.

Lemma 1.4. *The* BFS *form the extreme points of F_C.*

Proof. We need to show two directions for the proof.

First we show that an extreme point of F_C is a BFS.

We now show the other direction of the lemma. That is, BFS is an extreme point of F_C. This is left as an exercise. □

BFS are the most important points in the LP. We will soon see that the optimum occurs at one of the BFS.

Special Case: The feasible set F_C is bounded.

We say that a set $X \subseteq \mathbb{R}^n$ is *bounded* if there exists an $M \in \mathbb{R}$ such that for all $\mathbf{x} \in X$, we have that $\mathbf{x} \leq M^n$. Convex sets which are closed and bounded satisfy a nice property.

Claim 1.5. *Let X be a closed bounded convex set. Then, X is equal to the convex hull of its extreme points.*

Therefore, if F_C is bounded, we have that the convex hull of BFS gives us F_C. The following lemma captures the importance of BFS in bounded sets. It says, if there is an optimal solution to the LP, then it occurs at one of the points in BFS.

Claim 1.6. *Let the feasible set be bounded. If there is an optimal solution, then it occurs at a* BFS.

Proof. Let $\mathbf{x}$ be an optimal solution. That is $\mathbf{c}^\mathsf{T}\mathbf{x} = \min\{\mathbf{c}^\mathsf{T}\mathbf{y} \mid \mathbf{y} \in F_C\}$. In other words,

$$\forall \mathbf{y} \in F_C,\ \mathbf{c}^\mathsf{T}\mathbf{y} \geq \mathbf{c}^\mathsf{T}\mathbf{x} \tag{1.1}$$

We show that there exists an $\mathbf{x}^* \in$ BFS such that $\mathbf{c}^\mathsf{T}\mathbf{x}^* = \mathbf{c}^\mathsf{T}\mathbf{x}$. Let us assume $\mathbf{x} \notin$ BFS. From Lemma 1.4 we know that $\mathbf{x}$ is not an extreme points and hence can be written as a convex combination of vectors in $\mathbf{z}_i \in$ BFS (using claim 1.6). That is, $\mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{z}_i$ where $\sum_i \lambda_i = 1$. Let us assume without loss of generality, $\mathbf{c}^\mathsf{T}\mathbf{z}_1 \leq \mathbf{c}^\mathsf{T}\mathbf{z}_i$ for all $i \leq k$. Therefore

$$\mathbf{c}^\mathsf{T}\mathbf{x} \geq (\sum_{i=1}^{k} \lambda_i)\ \mathbf{c}^\mathsf{T}\mathbf{z}_1 = \mathbf{c}^\mathsf{T}\mathbf{z}_1 \tag{1.2}$$

From Equation 1.1 and 1.2 we have that $\mathbf{c}^\mathsf{T}\mathbf{z}_1 = \mathbf{c}^\mathsf{T}\mathbf{x}$ and therefore $\mathbf{z}_1 \in$ BFS is also an optimum point. $\square$

1.5 Simplex algorithm

With the intuition that the optimum occurs at one of the BFS, our algorithm will be to check all the BFS and return the optimal value. We have a better algorithm (which in the worst case is as bad as the naive algorithm) is very good in practise. The algorithm is called *Simplex* and the fundamental idea is to start from a BFS and keep going to a neighbouring vertex which is better in the objective function. Before we formally see the algorithm, let us define what are neighbouring BFS.

We say that two basis $B, B' \subseteq \{A_1, A_2, \ldots, A_n\}$ are *neighbours* if $|B \cap B'| = m-1$. That is, they share exactly $m-1$ columns. Each has one column which is not in the other. We say that $\mathbf{x}, \mathbf{y} \in$ BFS are *neighbours* if there exists two basis B, B' which are neighbours such that the basis associated with $\mathbf{x}$ and $\mathbf{y}$ are B and B' respectively. In other words, there are atleast $n-m-1$ locations where both $\mathbf{x}$ and $\mathbf{y}$ are zeros.

The Simplex algorithm is given in Algorithm 1.

Algorithm 1 Simplex algorithm

```
Find an initial BFS $\mathbf{x}$
while $\exists \mathbf{y}$ which is a neighbour of $\mathbf{x}$ and $\mathbf{c}^\mathsf{T}\mathbf{y} < \mathbf{c}^\mathsf{T}\mathbf{x}$ do
  $\mathbf{x} = \mathbf{y}$
end while
return $\mathbf{x}$
```

1.5.1 Neighbouring BFS

Our first algorithmic challenge is to identify a neighbouring $\mathbf{y} \in$ BFS given a $\mathbf{x} \in$ BFS where $\mathbf{c}^\mathsf{T}\mathbf{y} < \mathbf{c}^\mathsf{T}\mathbf{x}$. That is, a neighbour whose cost is lesser that at $\mathbf{x}$. Let B be the basis associated with vector $\mathbf{x}$ and N the rest of the column vectors of $\mathbb{A}$. We can always re-arrange the columns so that the first m columns is B. We also denote the various components of $\mathbb{A}, \mathbf{x}, \mathbf{c}$ as given below.

$$\mathbb{A} = \left(\begin{array}{c|c} \quad \mathbb{A}_B \quad & \quad \mathbb{A}_N \quad \end{array}\right), \mathbf{x} = \left(\begin{array}{c} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{array}\right), \mathbf{c} = \left(\begin{array}{c} \mathbf{c}_B \\ \hline \mathbf{c}_N \end{array}\right)$$

Therefore, we can write $\mathbb{A}\mathbf{x} = \mathbf{b}$ equivalently as $\mathbb{A}_B\mathbf{x}_B + \mathbb{A}_N\mathbf{x}_N = \mathbf{b}$ and the objective function as $\mathbf{c}^\mathsf{T}\mathbf{x} = \mathbf{c}_B^\mathsf{T}\mathbf{x}_B + \mathbf{c}_N^\mathsf{T}\mathbf{x}_N$. Also note that $\mathbf{x}_N = 0^{n-m}$.

Let $\mathbf{x}$ be a BFS corresponding to basis B. Our plan is to find a basis B' which is a neighbour to B. Let us consider that we want to include the column A_j in B'. This would mean we will have to remove some other column from B. Without loss of generality, let us assume $B = \{A_1, A_2, \ldots, A_m\}$ and $j > m$. Since B forms a basis we have that there exists α_is such that

$$A_j = \sum_{i=1}^{m} \alpha_i A_i \tag{1.3}$$

We also have that

$$b = \sum_{i=1}^{m} x_i A_i \tag{1.4}$$

Assumption: We will assume that $x_i > 0$ for all $i \leq m$. The tricky *degenerate* case of some of $x_j = 0$ is skipped in the lecture notes.

Consider the following equation (we fix the θ later): Eq. 1.4 - θ Eq. 1.3.

$$b = \sum_{i=1}^{m} (x_i - \theta\alpha_i)A_i + \theta A_j$$

The neighbour BFS to $\mathbf{x}$ is now given by $\mathbf{y}$ where y_i is

$$y_i = \begin{cases} x_i - \theta\alpha_i \text{ if } i \leq m \\ \theta \text{ if } i = j \\ 0 \text{ otherwise} \end{cases}$$

There are two cases to be considered now.
Case I: There is atleast one $\alpha_i > 0$. In this case, take θ as follows

$$\theta = min\{\frac{x_i}{\alpha_i} \mid \alpha_i > 0\}$$

Let $\theta = \frac{x_k}{\alpha_k}$. Then we have that $y_k = 0$ and all other $y_i \geq 0$. Moreover $\mathbf{y}$ is a BFS with basis $B' = B \backslash \{A_k\} \cup \{A_j\}$. We can now check if $\mathbf{c}^\mathsf{T}\mathbf{y} < \mathbf{c}^\mathsf{T}\mathbf{x}$ and return $\mathbf{y}$ as a "good" neighbour if the cost decreases. Otherwise we find another neighbour and continue this procedure.
Case II: All $\alpha_i \leq 0$. In this case y_is increases as we increase θ. We can therefore increase y_i to as big a number as we want. Note that this cannot happen in a bounded F_C. On the other hand, we can look at the cost function $\mathbf{c}^\mathsf{T}\mathbf{y}$. It either increases as θ increase or decreases. If it is increasing, we are not interested we can skip and go to find another neighbour of $\mathbf{x}$. On the other hand, if it is decreasing, we can decrease $\mathbf{c}^\mathsf{T}\mathbf{y}$ to as much as we want and therefore, there is no optimum value.

This gives an algorithm to find a neighbouring BFS.

1.5.2 Finding an initial BFS

1.5.3 Correctness of Simplex algorithm

Showing the following claim gives us the correctness of the Simplex algorithm.

Theorem 1.7. *Let* $\mathbf{x}$ *be a* BFS *such that for all neighbours* $\mathbf{y} \in$ BFS *we have* $\mathbf{c}^\mathsf{T}\mathbf{y} \geq \mathbf{c}^\mathsf{T}\mathbf{x}$. *Then* $\mathbf{x}$ *is an optimal solution.*

Proof. First we understand the relationship between $\mathbf{x}$ and its neighbours. Let B be the basis of $\mathbf{x}$ and let $\mathbf{z}$ be a neighbour with basis $B' = B \backslash \{A_k\} \cup$

$\{A_j\}$. We will use the matrix notation as given below.

$$\mathbb{A} = \left(\begin{array}{c|c} \mathbb{A}_B & \mathbb{A}_N \end{array} \right), \mathbf{x} = \left(\begin{array}{c} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{array} \right), \mathbf{c} = \left(\begin{array}{c} \mathbf{c}_B \\ \hline \mathbf{c}_N \end{array} \right)$$

Since $\mathbf{x}_N = \mathbf{0}$, we have that $\mathbb{A}_B \mathbf{x}_B = b$. Since A_j is in the basis for $\mathbf{z}$ we have that $\mathbb{A}_B \mathbf{z}_B + z_j A_j = b$. Subtracting, we have $\mathbb{A}_B \mathbf{x}_B - \mathbb{A}_B \mathbf{y}_B - z_j A_j = 0$. Therefore,

$$\mathbf{x}_B - \mathbf{z}_B = z_j \mathbb{A}_B^{-1} A_j$$

Now let us calculate the cost difference at both the places

$$0 \geq \mathbf{c}^\mathsf{T}\mathbf{x} - \mathbf{c}^\mathsf{T}\mathbf{y} = \mathbf{c}_B^\mathsf{T}(\mathbf{x}_B - \mathbf{z}_B) - c_j z_j = z_j \mathbf{c}_B^\mathsf{T} \mathbb{A}_B^{-1} A_j - c_j z_j$$

Since $z_j > 0$ we have the inequality for all $j \notin B$.

$$c_j - \mathbf{c}_B^\mathsf{T} \mathbb{A}_B^{-1} A_j \geq 0 \tag{1.5}$$

Now we show that $\mathbf{x}$ is an optimal point. Assume not. Let $\mathbf{y}$ has a better objective value. That is, $\mathbf{c}_B^\mathsf{T}\mathbf{y}_B + \mathbf{c}_N^\mathsf{T}\mathbf{y}_N = \mathbf{c}^\mathsf{T}\mathbf{y} < \mathbf{c}^\mathsf{T}\mathbf{x} = \mathbf{c}_B^\mathsf{T}\mathbf{x}_B$. Therefore,

$$\mathbf{c}_B^\mathsf{T}(\mathbf{y}_B - \mathbf{x}_B) + \sum_{j \notin B} c_j y_j < 0 \tag{1.6}$$

We know that $0 = \mathbb{A}(\mathbf{y} - \mathbf{x}) = \mathbb{A}_B(\mathbf{y}_B - \mathbf{x}_B) + \sum_{j \notin B} y_j A_j$. Therefore,

$$\mathbf{y}_B - \mathbf{x}_B = \sum_{j \notin B} -y_j \mathbb{A}_B^{-1} A_j \tag{1.7}$$

We can substitute the above equation in Eq. 1.6 giving us

$$\sum_{j \notin B} (c_j - \mathbf{c}_B^\mathsf{T} \mathbb{A}_B^{-1} A_j) y_j < 0$$

This gives us a contradiction from Eq. 1.5 and since $y_j > 0$ ($y_j = 0$ will not satisfy above inequality). $\square$

1.6 Matching using LP

A *bipartite graph* G is represented by $G = (U, V, E)$ where $U \cup V$ are the set of vertices and edges are denoted by $E \subseteq U \times V$. A *matching* $M \subseteq E$ of a bipartite graph G is a set of edges such that for any vertex v of G at most one edge incident on v is in E. In the *maximal matching problem*, given a bipartite graph, we interested in finding out a matching of highest cardinality. We reduce the problem to an LP as follows. Without loss of generality we assume cardinality of U is equal to cardinality of V (otherwise add some vertices with no edges) which is equal to n.

We introduce variables x_{ij} for all $i \in U$ and $j \in V$. We want a solution to the LP where $x_{ij} \in \{0, 1\}$ for all i and j. We can view any assignment of x_{ij}s to $\{0, 1\}$ as a set of edges $X = \{(i, j) \in E \mid x_{ij} = 1\}$. Consider the following LP.

$$\begin{aligned} \texttt{maximize} \sum_{(i,j)\in E} x_{ij} \\ \forall i \in U, \sum_{j\in V} x_{ij} = 1 \\ \forall j \in V, \sum_{i\in U} x_{ij} = 1 \\ \forall i, j, 0 \le x_{ij} \le 1 \end{aligned}$$

Unfortunately, the optimal solution to the LP may not have integer values for x_{ij}. The next claim shows that there is always an integer solution associated with a maximal matching.

Claim 1.8. *If $M \subseteq U \times V$ is a maximal matching, then there is an optimal solution with integer values and where $x_{ij} = 1$ for all $(i, j) \in M$.*

Proof. We give the assignment to x_{ij}s as follows. Let $\{u_1, u_2, \dots, u_k\} \subseteq U$ and $\{v_1, v_2, \dots, v_k\} \subseteq V$ be the set of vertices not matched in M. For the feasible point, make $x_{u_i v_i} = 1$. Note that, there is no edge between (u_i, v_i). Otherwise this would have been in the maximal matching. □

It is also easy to see that if we have an integer optimal solution, then we have a maximal matching. Now, we come to deal with the non-integer solution problem. We show that if we have a non-integer solution we can get one out of that. In practise this wont be necessary though. It is worth noting that due to some properties of the linear constraints in the maximal matching LP, the optimal solution is always integral (we skip this linear algebraic argument here). We give a method to extract an integer solution of same optimal value from a rational optimal solution.

Let there be K number of fractions in the solution. We reduce the number of fractions atleast by one without changing the optimal condition. Let $u \in U$ be a vertex where there is an edge with fractional value, say (u, v). Take that edge. This means that v has another edge which is also fractional. We take that edge leading to another vertex having another fractional edge. We continue doing this until we hit upon a cycle. Let us concentrate on the cycle. Assume the cycle starts from $u \in U$. We subtract θ (a very very small value) from x_{uv} (we assume (u, v) is the edge in the cycle) and add θ on the other edge starting from v in the cycle. This is continued. That is, for every edge from U to V we subtract θ whereas we add θ in the other direction. Note that θ is smaller than all x_ijs in the cycle.

Claim 1.9. *The optimal value of the LP does not change on this new assignment.*

Proof. Let us assume that the optimal value increases. A contradiction since x is an optimal solution and therefore we cannot increase the optimal value. So, let us assume that the optimal value decreases. Then, consider an alternate assignment where we replace θ by $-\theta$. Due to the previous reason, the optimal value should now increase. This leads to another contradiction. □

We can therefore take a θ such that one of x_{ij}s becomes an integer solution (either 0 or 1) and the rest remains inside the feasible region. This reduces the number of fractional solutions by atleast one. Continuing the process will lead to an optimal solution with all integer values.