CSS.201.1 Algorithms

Instructor: Umang Bhaskar

TIFR 2024, Aug-Nov

Scribe: Soham Chatterjee

soham.chatterjee@tifr.res.in
Website: sohamch08.github.io

Contents

Chapter 1

Linear Programming

1.1 Introduction

Definition 1.1.1: Linear Program

A linear programming problem asks for a vector $x \in \mathbb{R}^d$ that maximizes or minimizes a given linear function, among all vectors x that satisfy given set of linear inequalities.

The general form of a maximization linear programming problem is the following: given $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $a_i \in \mathbb{R}^n$ for each $i \in [m]$ then

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & a_i^T x \leq b_i \quad \forall\, i \in [p], \\ & a_i^T x = b_i \quad \forall\, i \in \{p+1, \dots, p+q\}, \\ & a_i^T x \geq b_i \quad \forall\, i \in \{p+q+1, \dots, m\}, \\ & x_j \geq 0 \quad \forall j \in [k], \\ & x_j \leq 0 \quad \forall j \in [\{k+1, \dots, k+l\} \quad (\text{Some } x_j\text{'s are free}) \end{aligned}$$

The similar goes for minimization linear programming problem. For maximization problem we can always write the LP in the form

$$\begin{aligned} \text{maximize} \quad & c^T \hat{x} \\ \text{subject to} \quad & \hat{a}_i^T x \leq b_i' \quad \forall\, i \in [m], \\ & x_j' \geq 0 \quad \forall j \in [n] \end{aligned}$$

And then the LP is said to be in the ***canonical form***. What we can do is the following:

- For $i \in \{p+q+1, \dots, m\}$, we can replace $a_i^T x \leq b_i$ with $-a_i^T x \geq -b_i$
- For $i \in \{p+1, \dots, p+q\}$, we can replace with two constraints $a_i^T x \geq b_i$ and $a_i^T x \leq b_i$
- For $j \in \{k+1 \dots, k+l\}$, we can replace $x_j \leq 0$ with $-x_j \geq 0$
- For $j \in \{k+l+1 \dots, n\}$, we can replace the free x_j's with $x_j^+ - x_j^-$ all the equations where $x_j^+, x_j^- \geq 0$

This way we can always get a LP of that form. Now we can replace the $\hat{a}_i$ for $i \in [m]$ with a matrix $A \in \mathbb{R}^{m \times n}$ and replace the constraint $\hat{a}_i^T x \leq b_i', \forall\, i \in [m]$ with $Ax \leq b$

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b, \\ & x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b, \\ & x \geq 0 \end{aligned}$$

1.2 Geometry of LP

Definition 1.2.1: Feasible Point and Region

A point $x \in \mathbb{R}^n$ is *feasible* with respect to some LP if it satisfies all the linear constraints. The set of all feasible points is called the *feasible region* for that LP.

Feasible region of a LP has a particularly nice geometric structure. Before that we will first introduce some geometric terminologies used in the linear programming context:

Definition 1.2.2: Hyperplane, Polyhedron, Polytope

- **Line**: The set $\{x + \lambda d, \lambda \in \mathbb{R}\}$ is line for any $x, d \in \mathbb{R}^n$.
- **Hyperplane**: The set $\{x \in \mathbb{R}^n : a^x = b\}$ is a hyperplane for any $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.
- **Hyperspace**: The set $\{x \in \mathbb{R}^n : a^x \leq b\}$ is a hyperspace or half-space for any $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.
- **Polyhedron**: A polyhedron is the intersection of a finite set of half-spaces i.e. the set $\{x \in \mathbb{R}^n : Ax \leq b\}$ for any $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^m$.
- **Polytope**: A bounded polyhedron is called a polytope.

Now it is not hard to verify that any polyhedron is a convex set i.e. if a polyhedron contains two points then it contains the entire line segment joining those two points.

Lemma 1.2.1

Polyhedron is a convex set

Hence the feasible region of a LP creates a polyhedron in $\mathbb{R}^n$. And $c^T x$ is the hyperplane normal to the vector c and the objective of the LP is by moving the plane normal to the vector c for which point in the polyhedron the hyperplane $c^T x$ has the highest value. Since polyhedron can be unbounded there may not exists any point x where $c^T x$ is maximum.

Suppose we have a LP

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b, \\ & x \geq 0 \end{aligned}$$

Let P be the polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. Then given $x^* \in P$ if any constraint $a_i^T x^* = b_i$ then this constrain is said to be *tight* or *binding* or *active* at x^*. Now two constraints $a_i^T x \leq b_i$ and $a_j^T x \leq b_j$ are said to be linearly independent if a_i and a_j are linearly independent.

Definition 1.2.3: Basic Solution and Basic Feasible Solution

$x^* \in \mathbb{R}^n$ is a basic solution if n linearly independent constraints are active at x^* (Doesn't need to be feasible).

$x^* \in \mathbb{R}^n$ is a basic feasible solution if x^* is a basic solution and $x^* \in P$. The basic feasible solutions are also called *corners* of a polyhedron.

Theorem 1.2.2

Given a LP

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b, \\ & x \geq 0 \end{aligned}$$

Let P is the polyhedron $\{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$. Suppose P is non-empty and has at least one basic feasible solution then either the optimal value is $-\infty$ or there is an optimal basic feasible solution.

Theorem 1.2.3

If polyhedron P does not contain a line it contains at least one basic feasible solution (Hence if P is bounded it contains at least one basic feasible solution).

With this geometry in hand, we can easily picture two pathological cases where a given linear programming problem has no solution. The first possibility is that there are no feasible points; in this case the problem is called *infeasible*. The second possibility is that there are feasible points at which the objective function is arbitrarily large; in this case, we call the problem *unbounded*. The same polyhedron could be unbounded for some objective functions but not others, or it could be unbounded for every objective function.

Example 1.2.1

- **Maximum Matchings:** Given undirected graph $G = (V, E)$. Say variable x_e for each $e \in E$, $x_e = 1 \implies e$ in matching and $x_e = 0$ otherwise.

$$\begin{aligned}
\text{maximize} \quad & \sum_{e \in E} x_e \\
\text{subject to} \quad & \sum_{e \text{ incident on } v} x_e \le 1 && \forall\, v \in V, \\
& x_e \ge 0 && \forall\, e \in E, \\
& x_e \in \{0, 1\} && \forall\, e \in E
\end{aligned}$$

Observation. *M is a matching iff $\{x : x_e = 1$ if $e \in M, = 0$ otherwise$\}$ is a feasible solution*

- **Maximum $s - t$ Flow:** Given directed graph $G = (V, E)$ with vertices s, t and capacity c_e on edges. Say variable x_e for each edge and equal to flow on that edge. Then the LP of this problem:

$$\begin{aligned}
\text{maximize} \quad & \sum_{e \in out(s)} x_e \\
\text{subject to} \quad & \sum_{e \in in(v)} x_e - \sum_{c \in out(v)} x_e = 0 && \forall\, v \in V, v \ne s, t, \\
& c_e \ge x_e \ge 0 && \forall\, e \in E
\end{aligned}$$

We will now introduce a theorem without proof that for any LP with a polytope we can find a solution in polynomial time.

Theorem 1.2.4

Let $P = \{x \in \mathbb{R}^n : Ax \ge b\}$ be a polytope. Then we can find an optimal basic feasible solution for the LP $\min c^T x$ where $x \in P$ in polynomial time.

1.3 LP Integrality

For the LP for matchings in bipartite graphs $G = (L \cup R, E)$ we have:

$$\begin{aligned}
\text{maximize} \quad & \sum_{e \in E} x_e \\
\text{subject to} \quad & \sum_{e \text{ incident on } v} x_e \le 1 && \forall\, v \in V, \\
& x_e \ge 0 && \forall\, e \in E
\end{aligned}$$

We want $x_e \in \{0, 1\}$ i.e. we want to have integral solution for this LP

Question 1.1

LP's can give fractional solutions. When is solution integral?

Sufficient Condition: Every basic feasible solution of the feasible polytope is integral i.e. x^* is basic feasible solution $\implies x^* \in \mathbb{Z}^n$. If all basic feasible solution are integral then for all $I \subseteq [m]$ with $|I| = n$, $A_I^{-1} b_I$ is integral. Let $x = A_I^{-1} b_I$ Then j^{th} component $x_j = \frac{|A_I^j|}{|A|}$ (Cramer's Rule).

1.3.1 Totally Unimodular Matrix

Definition 1.3.1: Totally Unimodular Matrix (TUM)

A matrix $A \in \{0, 1, -1\}^{m \times n}$ is totally unimodular (TU) if every square submatrix of A has determinant $-1, 0, 1$.

Hence in the above LP is A is TU and b is integral then all basic feasible solutions are integral.

Lemma 1.3.1

Let A be TUM and $b \in \mathbb{Z}^n$ then $P = \{x : Ax \geq b\}$ is integral i.e. every basic feasible solution is integral.

Hence using Theorem 1.2.4 if the polytope is integral we can find optimal integral solution in polynomial time. We will now discuss properties of Totally Unimodular Matrix.

Lemma 1.3.2

$A \in \{0, 1, -1\}^{m \times n}$ is TU iff the following are TU:

(i) $-A$

(ii) A^T

(iii) $\begin{bmatrix} A & e_i \end{bmatrix}$, $\begin{bmatrix} A & -e_i \end{bmatrix}$

(iv) $\begin{bmatrix} A & I \end{bmatrix}$, $\begin{bmatrix} A & -I \end{bmatrix}$

(v) $\begin{bmatrix} A & A_i \end{bmatrix}$, $\begin{bmatrix} A & -A_i \end{bmatrix}$ where A_i is the i^{th} column of A.

Corollary 1.3.3

If A is TUM and $a, b, c, d \in \mathbb{Z}^n$ are integer vectors then the polytope $Q = \{x \in \mathbb{R}^n : a \leq Ax \leq b, c \leq x \leq d\}$ is integral.

Proof: We can combine the four inequalities in one inequality. Consider the matrix $\begin{bmatrix} A & -A & I & -I \end{bmatrix}^T$. Then the given polytope is

$$Q = \left\{ x \in \mathbb{Z}^n : \begin{bmatrix} A \\ -A \\ I \\ -I \end{bmatrix} x \leq \begin{bmatrix} b \\ -a \\ d \\ -c \end{bmatrix} \right\}$$

By Lemma 1.3.2, $\begin{bmatrix} A & -A & I & -I \end{bmatrix}^T$ is a TUM since A is TUM. Therefore the polytope Q is integral. ■

The following theorem lets us to give a necessary and sufficient condition to check if a given matrix is TUM. Again we will accept the following theorem without the proof since the proof is a little nontrivial.

Theorem 1.3.4

Let $A \in \{-1,0,1\}^{m\times n}$. Then A is TU iff every set $S \subseteq [n]$ can be partitioned into S_1, S_2 such that

$$\sum_{i\in S_1} A_i - \sum_{i\in S_2} A_i \in \{-1,0,1\}^m$$

where A_i is the i^{th} column of A. C

1.3.2 Integrality of Some Well-Known Polytopes

Now using this theorem we will show that the polytope for bipartite maximum matching is integral. The LP for bipartite maximum matching is given by:

$$\begin{aligned}
\text{maximize} \quad & \sum_{e\in E} x_e \\
\text{subject to} \quad & \sum_{e \text{ incident on } v} x_e \leq 1 \quad \forall\, v \in V, \\
& x_e \geq 0 \qquad\qquad\quad \forall\, e \in E
\end{aligned}$$

Lemma 1.3.5

The polytope for bipartite maximum matching is integral.

Proof: Let A be the matrix for the polytope. Now clearly from the construction of the polytope we have $A \in \{0,1\}^{n\times m}$ where $n = |V|$ and $m = |E|$. Now we will show that A^T is TUM. Let L and R are the two sets of vertices in the bipartite graph. Now suppose $S \subseteq L \cup R$. Then take $S_1 = S \cap L$ and $S_2 = S \cap R$. Then for any row $e \in E$, we have

$$\sum_{i\in S_1} A_i - \sum_{i\in S_2} A_i \in \{-1,0,1\}$$

Hence A^T is TUM and therefore by Lemma 1.3.2 A is TUM. Hence the polytope for bipartite maximum matching is integral. ■

Note:-

For general graphs this polytope is not integral. Consider the triangle graph K_3. Then the point $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$ is a feasible solution but not in the convex hull of the integral solutions $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$.

Lemma 1.3.6

The LP for $s-t$ max flow is

$$\begin{aligned}
\text{maximize} \quad & \sum_{e\in out(s)} x_e \\
\text{subject to} \quad & \sum_{e\in in(v)} x_e - \sum_{c\in out(v)} x_e = 0 \quad \forall\, v \in V, v \neq s,t, \\
& c_e \geq x_e \geq 0 \qquad\qquad\qquad\quad \forall\, e \in E
\end{aligned}$$

Then the max flow polytope is integral.

Proof: Let A be the matrix for the polytope. We will show that A is TUM. Given $S \subseteq V \setminus \{s,t\}$ take $S_1 = S$ and $S_2 = \emptyset$. By the first condition of the polytope for all vertices we already have satisfied the condition

$$\sum_{i\in S_1} A_i - \sum_{i\in S_2} A_i = 0 \in \{-1,0,1\}^m$$

Therefore the polytope is TUM and hence integral. ■

1.4 Duality

Suppose we have the following LP:

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 \\ \text{subject to} \quad & x_1 - x_2 \geq 3, \\ & 2x_1 + x_2 \geq 1, \\ & x_1, x_2 \geq 0 \end{aligned}$$

Suppose we want to have a lower bount on the optimal solution of the LP. Then we will try to find a linear combination of the constriants such that in the LHS we obtain some thing which is at most the objective function and on the RHS we get the lower bound. So let we multiply the first constraint with y_1, second with y_2. For now y_1, y_2 are unknowns. Then we have the following:

$$\begin{aligned} x_1 + 2x_2 &\geq (y_1 + 2y_2)x_1 + (-y_1 + y_2)x_2 \\ &= y_1(x_1 - x_2) + y_2(2x_1 + x_2) \geq 3y_1 + y_2 \end{aligned}$$

But we also have the conditions that the coefficients of x_1 and x_2 can not be more than the coefficients of x_1 and x_2 in the objective function respectively. So we have the following conditions:

$$\begin{aligned} y_1 + 2y_2 &\leq 1 \\ -y_1 + y_2 &\leq 2 \end{aligned}$$

So now we have found a maximization LP which gives us a lower bound on the optimal solution of the original LP:

$$\begin{aligned} \text{maximize} \quad & 3y_1 + y_2 \\ \text{subject to} \quad & y_1 + 2y_2 \leq 1, \\ & -y_1 + y_2 \leq 2, \\ & y_1, y_2 \geq 0 \end{aligned}$$

This is called the *dual* of the original LP. The original LP is called the *primal* of the dual. The primal and dual are related in a very nice way. The following theorem gives us the relation between primal and dual.

For every minimization LP there is a dual LP that provides a lower bound on the optimal value of the primal LP.

Note:-

If the Primal LP is unbounded then the dual LP is infeasible.

Lemma 1.4.1

Dual of Dual is the primal LP

1.4.1 Dualization of LP

If the primal LP is in canonical form then we have the following:

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b, \\ & x \geq 0 \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \text{minimize} \quad & b^T y \\ \text{subject to} \quad & A^T y \leq c, \\ & x \geq 0 \end{aligned}$$

$\boxed{\text{Primal}}$ $\boxed{\text{Dual}}$

But if the primal LP is not in the canonical form then we have two options: either we can convert the primal to the canonical form and the dualize it or we can directly dualize the primal LP. The following method gives us a way to dualize the primal LP without converting it tot the canonical form.

$$\begin{array}{ll}
\text{maximize} & c^T x \\
\text{subject to} & A_j x \geq b_j \quad \forall\, j \in [d], \\
& A_j x = b_j \quad \forall\, j \in \{d+1, \ldots, m\}, \\
& x_i \geq 0 \quad \forall\, i \in [k], \\
& x_i \text{ is free} \quad \forall\, i \in \{k+1, \ldots, n\}
\end{array}
\qquad \Longleftrightarrow \qquad
\begin{array}{ll}
\text{minimize} & b^T y \\
\text{subject to} & \sum_{j=1}^{m} A_{ji} y_j \leq c_i \quad \forall\, i \in [k], \\
& \sum_{j=1}^{m} A_{ji} y_j = c_j \quad \forall\, i \in \{k+1, \ldots, n\}, \\
& y_j \geq 0 \quad \forall\, j \in [d], \\
& y_j \text{ is free} \quad \forall\, i \in \{d+1, \ldots, m\}
\end{array}$$

$\boxed{\text{Primal}}$ $\boxed{\text{Dual}}$

So we have the following observations:

Observation. *In dualization of a LP which is not in canonical form*

Primal		*Dual*
Non-negative variables	$\Longleftrightarrow$	*Inequality constraints*
Free variables	$\Longleftrightarrow$	*Equality constraints*

1.4.2 Weak and Strong Duality

Now as the motivation for constructing the dual LP. We have the following theorem which proves the any feasible solution of the dual LP indeed gives a lower bound on the optimal solution of the primal LP.

Theorem 1.4.2 Weak Duality Theorem

If x, y are feasible solutions for the primal and dual LPs respectively and then $c^T x \geq b^T y$.

Proof: We have

$$b^T \leq \sum_{j=1}^{d} y_j(A_j x) + \sum_{j=d+1}^{m} y_j(A_j x) = \sum_{j=1}^{d} y_j A_j x = \sum_{j=1}^{m}\sum_{i=1}^{n} y_j A_{ji} x_i = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} A_{ji} y_j \leq \sum_{i=1}^{m} x_i c_i = c^x$$

Hence we have the theorem. ■

We also have a much stronger theorem which tells us that the optimal solutions of the primal and dual LPs are equal.

Theorem 1.4.3 Strong Duality Theorem

Let the primal and dual LP are feasible and x^*, y^* are the optimal solutions of the primal and dual LPs respectively. Then $c^T x^* = b^T y^*$.

Notice that if for any feasible solution y of the dual LP is $c^T x^* = b^T y$ then y must be the optimal solution of the dual LP.

1.4.3 Complementary Slackness

Question 1.2

Suppose we have optimal solutions x^*, y^* of the primal and dual LPs respectively. What can be said about which constraints are tight in the primal and dual?

Theorem 1.4.4 Complementary Slackness

Let x^*, y^* be the optimal solutions of the primal and dual LPs respectively iff:

(i) If $A_j x^* > b_j$ then $y_j^* = 0$.

(ii) If $A^{i^T} y^* < c_i$ then $x_i^* = 0$.

Proof: Suppose x^*, y^* are the optimal solutions of the primal and dual LPs respectively. Then by Strong Duality Theorem we have

$$\sum_{i=1}^{k} x_i \sum_{j=1}^{m} A_{ji} y_j + \sum_{i=k+1}^{n} x_i \sum_{j=1}^{m} A_{ji} y_j = \sum_{i=1}^{k} x_i c_i + \sum_{i=k+1}^{n} x_i c_i$$

So we have

$$\sum_{i=1}^{k} x_i \sum_{j=1}^{m} A_{ji} y_j = \sum_{i=1}^{k} x_i c_i$$

Hence either $x_i = 0$ or $\sum_{j=1}^{m} A_{ji} y_j = c_i$ for all $i \in [k]$. So $A^{i^T} y^* < c_i$ implies $x_i^* = 0$. Similarly we have $A_j x^* > b_j$ then $y_j^* = 0$. ∎

There is also a relaxed version of the complementary slackness theorem, Theorem 2.1.4 which is useful in practice. It is explained in the next chapter.

1.4.4 Max-Flow Min-Cut Theorem

So here using LP-duality we give another proof of Max-Flow Min-Cut Theorem. The LP for maximum flow is given by:

$$\begin{aligned}
\text{maximize} \quad & \sum_{e \in out(s)} x_e \\
\text{subject to} \quad & \sum_{e \in in(v)} x_e - \sum_{c \in out(v)} x_e = 0 \qquad \forall\, v \in V, v \neq s, t, \\
& c_e \geq x_e \qquad \forall\, e \in E, \\
& x_e \geq 0 \qquad \forall\, e \in E
\end{aligned}$$

We can convert this LP by adding edges of $in(s)$ and giving them capacity 0. So we have the modified LP:

$$\begin{aligned}
\text{maximize} \quad & \sum_{e \in out(s)} x_e - \sum_{e \in in(s)} x_e \\
\text{subject to} \quad & \sum_{e \in in(v)} x_e - \sum_{c \in out(v)} x_e = 0 \qquad \forall\, v \in V, v \neq s, t, \\
& c_e \geq x_e \qquad \forall\, e \in E, \\
& x_e \geq 0 \qquad \forall\, e \in E
\end{aligned}$$

For the first constraint we have the variables α_v and for the second constrain we have the variables β_e. So the dual of this LP is given by:

$$\begin{aligned} \text{minimize} \quad & \sum_{e\in E} c_e\beta_e \\ \text{subject to} \quad & -\alpha_u+\alpha_v+\beta_e \geq 0 \quad \forall\, e=(u,v)\in E, u,v\notin\{s,t\}, \\ & \alpha_v \geq 0 \quad \forall\, v\in V, v\neq s,t, \\ & \beta_e \geq 0 \quad \forall\, e\in E \end{aligned}$$

Now we can add $\alpha_s = 1$ and $\alpha_t = 0$ to the dual LP and obtain the modified dual LP:

$$\begin{aligned} \text{minimize} \quad & \sum_{e\in E} c_e\beta_e \\ \text{subject to} \quad & \beta_e \geq \alpha_u - \alpha_v + \quad \forall\, e=(u,v)\in E, u,v\notin\{s,t\}, \\ & \alpha_v \geq 0 \quad \forall\, v\in V, v\neq s,t, \\ & \beta_e \geq 0 \quad \forall\, e\in E, \\ & \alpha_s = 1, \\ & \alpha_t = 0 \end{aligned}$$

Now for the max-flow LP we already proved in Lemma 1.3.6 that the polytope is integral. By Lemma 1.3.2 the polytope for the dual is also integral. Let $x^*, (\alpha^*, \beta^*)$ be the optimal solution of the primal and dual LPs respectively. Now by Complementary Slackness we have the following:

$$x_e^* > 0 \implies \beta_e^* = \alpha_u^* - \alpha_v^* \qquad \text{and} \qquad \beta_e^* > 0 \implies x_e^* = c_e$$

Now $\alpha_s^* = 1$. Let $X = \{v\colon \alpha_v^* \geq 1\}$. Then $s\in X$ and $t\notin X$. Hence X is a $s-t$ cut. Now consider an edge (u,v) out of X. Then

$$\alpha_u^* \geq 1 \text{ and } \alpha_v^* < 1 \implies \beta_e^* > 0 \implies x_e^* = c_e$$

And for an edge $e=(u,v)$ in to X

$$x_e^* > 0, \alpha_u^* < 1, \alpha_v^* \geq 1 \implies \beta_e^* < 0$$

Hence for an edge e into X, $x_e^* = 0$. Hence maximum flow is equal to the $\sum\limits_{e\in out(X)} c_e$ and this is the minimum cut.

1.4.5 Maximum Bipartite Matching minimum Vertex Cover

The maximum bipartite matching problem is given by the following LP:

$$\begin{aligned} \text{maximize} \quad & \sum_{e\in E} x_e \\ \text{subject to} \quad & \sum_{e \text{ incident on } v} x_e \leq 1 \quad \forall\, v\in V, \\ & x_e \geq 0 \quad \forall\, e\in E \end{aligned}$$

The dual of the LP si given by

$$\begin{aligned} \text{minimize} \quad & \sum_{v\in V} y_v \\ \text{subject to} \quad & y_u + y_v \geq 1 \quad \forall\, (u,v)\in E, \\ & y_v \geq 0 \quad \forall\, v\in V \end{aligned}$$

Since in Lemma 1.3.5 we have proved the polytope for bipartite maximum matching is integral the polytope for the dual is also integral.

Definition 1.4.1: Vertex Cover

Given $G=(V,E)$ a vertex cover is a subset $C\subseteq V$ such that $\forall\, e\in E$ at least one of the endpoints of e is in C.

Then we have the following lemma:

Lemma 1.4.5

Let C be a vertex cover. Then there exists a dual feasible solution y such that $\sum_v y_v = |C|$.

Proof: Consider the vector $y \in \{0,1\}^{|V|}$ such that $y_v = 1$ if $v \in C$ and $y_v = 0$ otherwise. Then we have the lemma. ■

Lemma 1.4.6

Let y be an integral dual solution. Then $C = \{v : y_v \geq 1\}$ is a vertex cover.

Proof: For every edge $e = (u,v)$ we have $y_u + y_v \geq 1$. So either $y_u \geq 1$ or $y_v \geq 1$ as y is integral. Hence either $u \in C$ or $v \in C$. Hence every edge is covered by C and hence C is a vertex cover. ■

Note:-

In general graphs computing a minimum sized vertex cover in NP-hard. But since for bipartite graph the polytope is integral we can compute minimum weight vertex cover in polynomial time.

Chapter 2

Approximation Algorithms using LP

In this chapter we will study some approximation algorithms using linear programming to get better approximation ratios of the optimal solution.

2.1 Set Cover

Set Cover
Input: $\mathcal{U}$: Universe of all elements $u_1, \ldots, u_n$
$\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq \mathcal{U}$ forall $i \in [m]$
Function $c : \mathcal{S} \to \mathbb{Z}_+$
Question: Given $\mathcal{U}, \mathcal{S}$ and the function c find $T \subseteq [m]$ such that $\bigcup\limits_{i \in T} S_i = \mathcal{U}$ to minimize the total cost $c(T) = \sum\limits_{i \in T} c(S_i)$

Since the special case of Set Cover is basically the Vertex Cover problem we discussed earlier, we know that Set Cover is NP-hard.

Theorem 2.1.1
Set Cover is NP-hard.

Since we are going to find approximate solutions using LP let's first write the linear program for Set Cover:

$$\begin{aligned}
\text{minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\
\text{subject to} \quad & \sum_{S : u \in S} x_S \geq 1 \quad \forall\, u \in \mathcal{U}, \\
& x_S \geq 0 \quad \forall\, S \in \mathcal{S}
\end{aligned}$$

2.1.1 Frequency f-Approximation Algorithm

Let for any element $u \in \mathcal{U}$, f_u is the frequency of the element u in $\mathcal{S}$ i.e. $f_u = |\{S \in \mathcal{S} : u \in S\}|$. Then let $f = \max\{f_u : u \in \mathcal{U}\}$. Then we want to find a f-approximation algorithm for set cover.

Question 2.1: F

r vertex cover what is f?

For all $e \in E$ we have $f_e = 2$ since the elements of universe corresponds to the edges and sets corresponds to vertices and each edge is contained in exactly 2 sets. So $f = 2$.

Algorithm 1: f-Approximate Algorithm

Input: $\mathcal{U}, \mathcal{S}, c$

Output: $T \subseteq [m]$ such that $\bigcup_{i \in T} S_i = \mathcal{U}$ and $\sum_{i \in T} c(S_i)$ is minimized

begin
- $T \longleftarrow \emptyset$
- $\hat{x} \longleftarrow 0^{|\mathcal{S}|}$
- Let x^* is the optimal solution of the LP for Set Cover problem
- **for** $S_i \in \mathcal{S}$ **do**
  - **if** $x^*_{S_i} \geq \frac{1}{f}$ **then**
    - $T \longleftarrow T \cup \{i\}$
    - $\hat{x}_{S_i} \longleftarrow 1$
- **return** T

Lemma 2.1.2

$\hat{x}$ is a feasible solution.

Proof: For all $e \in \mathcal{U}$ there are at most f sets containing e. Thus at most f terms in the summation in *LHS* of the first constraint for each $e \in \mathcal{U}$ Thus in x^* at least one such term is $\geq \frac{1}{f}$. ■

Lemma 2.1.3

$\sum_{S \in \mathcal{S}} c(S)\hat{x}_S \leq f \cdot \sum_{S \in \mathcal{S}} c(S)x^*_S$

Proof: In $\hat{x}$ if $\hat{x}_S = 1$ that means $x^*_S \geq \frac{1}{f}$. Therefore we have the lemma. ■

Hence with this algorithm we can get a f-approximation for Set Cover problem. But this is not good enough since one element can be in too many sets and then it doesn't give a good approximation. In the next section we will see a new way of getting the same approximation ratio.

2.1.2 Frequency f-Approximation Algorithm through Dual Fitting

First let's wrote the dual of the LP for Set Cover problem:

$$\begin{array}{llll} \text{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c(S)x_S & & \\ \text{subject to} & \displaystyle\sum_{S : u \in S} x_S \geq 1 & \forall\, u \in \mathcal{U}, \\ & x_S \geq 0 & \forall\, S \in \mathcal{S} \end{array} \quad \Longleftrightarrow \quad \begin{array}{llll} \text{maximize} & \displaystyle\sum_{u \in \mathcal{U}} y_u & & \\ \text{subject to} & \displaystyle\sum_{u \in S} y_u \leq c(S) & \forall\, S \in \mathcal{S}, \\ & y_u \geq 0 & \forall\, u \in \mathcal{U} \end{array}$$

$\boxed{\text{Primal}}$ "Covering Problem" $\qquad$ $\boxed{\text{Dual}}$ "Packing Problem"

Both the primal and dual are feasible. Let x, y are feasible solutions of the primal and dual respectively. Then by Weak Duality we have

$$\sum_{S \in \mathcal{S}} c(S)x_S \geq \sum_{u \in \mathcal{U}} y_u$$

Let x^*, y^* are the optimal solutions of primal and dual respectively. Then by [Complementary Slackness]

$$x_S^* > 0 \implies \sum_{u\in S} y_u^* = c(S) \qquad y_u^* > 0 \implies \sum_{S:u\in S} x_S^* = 1$$

Theorem 2.1.4 Relaxed Complementary Slackness
Suppose x, y are feasible solutions of the primal and dual respectively and the satisfy the following conditions:

1. If $x_j > 0$ then $\frac{1}{\alpha}\cdot c_j \le {A^j}^T y \le c_j$ where $\alpha \ge 1$.
2. If $y_i > 0$ then $b_i \le A_i^T x \le \beta \cdot b_i$ where $\beta \ge 1$.

Then
$$c^T x \le \alpha\beta \cdot b^T y \le \alpha\beta \cdot c^T x^* = \alpha\beta \cdot \text{OPT}$$

Proof: x, y are the feasible solutions of the primal and dual respectively. Suppose first d constraints of the primal are equalities and rest are inequalities and similarly first l constraints of the dual are equalities and rest are inequalities. Then we have

$$c^T x = \sum_{j=1}^m c_j x_j \le \sum_{j=1}^m \left(\alpha {A^j}^T y\right) x_j = \alpha \sum_{j=1}^m \sum_{i=1}^n A_{ij} y_i x_j = \alpha \sum_{i=1}^n \left(\sum_{i=1}^m A_{ij} x_j\right) y_i \le \alpha \sum_{i=1}^m \beta \cdot b_i y_i = \alpha\beta \cdot b^T y$$

Hence we have $c^T x \le \alpha\beta \cdot b^T y \le \alpha\beta \cdot c^T x^* = \alpha\beta \cdot \text{OPT}$. ■

To show a f-approximation algorithm for set cover we will first find x, y feasible primal, dual solution which satisfies:

1. x is integral.
2. x satisfies the first condition of Relaxed Complementary Slackness with $\alpha = f$.

Algorithm 2: Dual Fitting Algorithm for Set Cover

Input: $\mathcal{U}, \mathcal{S}, c$
Output: $T \subseteq [m]$ such that $\bigcup\limits_{i\in T} S_i = \mathcal{U}$ and $\sum\limits_{i\in T} c(S_i)$ is minimized

```
begin
    Initialize U' ⟵ U, C ⟵ ∅
    while ∃ u ∈ U' do
        Increase y_u until for some S ∈ 𝒮 such that u ∈ S we have Σ_{u'∈S} y_{u'} = c(S)
        C ⟵ C ∪ { S ∈ 𝒮 : Σ_{u∈S} y_u = c(S) }
        for S ∈ C do
            U' ⟵ U' \ S
    return C
```

From C we con construct x by $x_S = 1$ if $S \in C$ and otherwise $x_=0$ for all $S \in \mathcal{S}$. Now we have the observations:

Observation. *After the algorithm terminates we have:*

1. *At the beginning of the loop if $u \in \mathcal{U}$, $y_u = 0$.*
2. *If $x_S = 1$ and $u \in S$ then y_u is not increased.*
3. *$x_S \in \{0,1\}^{|\mathcal{S}|}$ is integral.*

Lemma 2.1.5

1. x is feasible at the end of the algorithm.
2. y is feasible at every iteration of the while loop

Proof: The algorithm terminates when $\mathcal{U}' = \emptyset$. That means all the elements of the universe are covered. Hence the set C output after the algorithm terminates is indeed a set cover. Hence x is a feasible solution.

At the start of the algorithm $y = 0^{|\mathcal{U}|}$. Hence y is feasible. Now suppose at any iteration y is feasible. If the algorithm goes through another iteration then there exists an element in $\mathcal{U}'$ which is not covered. Let $u \in \mathcal{U}'$ which is not covered. Hence $y_u = 0$. Since in the previous iteration y was feasible we have $\sum\limits_{S:u\in S} y_u \leq c(S)$. Now we increase y_u to the point we achieve the equality $\sum\limits_{u'\in S} y_{u'} = c(S)$ for all $S \in \mathcal{S}$. Therefore even after updating y_u all the constraints of dual are satisfied. Hence y is a feasible solution after another iteration of the while loop. Therefore y is feasible at every iteration of the while loop. ■

Lemma 2.1.6

x, y satisfy the Relaxed Complementary Slackness conditions.

Proof: If for any $S \in \mathcal{S}$, $x_S > 0$ then we have $\sum\limits_{u\in S} y_u = c(S)$ by the construction of C in the algorithm. Therefore

$$x_S > 0 \implies \sum_{u\in S} y_u = c(S)$$

Hence $\alpha = 1$.

Now let for some $u \in \mathcal{U}$, $y_u > 0$. Since f is the maximum frequency of any element of the universe we have $f \geq \sum\limits_{S:u\in S} x_S \geq 1$. Therefore

$$y_u > 0 \implies f \geq \sum_{S:u\in S} x_S \geq 1$$

Hence $\beta = f$. ■

Therefore by Relaxed Complementary Slackness C is an f-approximate solution for the set cover problem. In the next sections we will show how to get a better approximation ratio.

2.1.3 $O(n \log n)$-Approximation Algorithm through Randomized Rounding

Algorithm 3: $n \log n$-Approximate Algorithm

Input: $\mathcal{U}, \mathcal{S}, c$
Output: $T \subseteq [m]$ such that $\bigcup\limits_{i\in T} S_i = \mathcal{U}$ and $\sum\limits_{i\in T} c(S_i)$ is minimized

begin
  $\hat{x} \longleftarrow 0^{|\mathcal{S}|}$
  Let x^* is the optimal solution of the LP for Set Cover problem
  for $S \in \mathcal{S}$ **do**
    Set $\hat{x}_S \longleftarrow 1$ with probability x^*_S.
  return $\hat{x}$

From the construction of $\hat{x}$ we have $\mathbb{E}\left[\sum\limits_{S\in\mathcal{S}} c(S)\hat{x}_S\right] = \sum\limits_{S\in\mathcal{S}} c(S)x_S^*$. Now suppose we fixed an element $u \in \mathcal{U}$. Then

$$\mathbb{P}[u \text{ is not covered}] = \prod_{S:u\in S} \mathbb{P}[S \text{ is not selected}] = \prod_{S:u\in S} (1 - x_S^*) \leq \prod_{S:u\in S} e^{-x_S^*} = \exp\left[-\sum_{S:u\in S} x_S^*\right] \leq e^{-1}$$

Hence to reduce the probability of not covering an element of $\mathcal{U}$ we repeat the algorithm multiple times. Hence we have the updated algorithm:

Algorithm 4: $n\log n$-Approximate Algorithm

Input: $\mathcal{U}, \mathcal{S}, c$
Output: $T \subseteq [m]$ such that $\bigcup\limits_{i\in T} S_i = \mathcal{U}$ and $\sum\limits_{i\in T} c(S_i)$ is minimized

begin
  Let x^* is the optimal solution of the LP for Set Cover problem
  for $i \in [2\log n]$ **do**
    $C_i \longleftarrow \emptyset$
    for $S \in \mathcal{S}$ **do**
      Put S in C_i with probability x_S^*.
  $C \longleftarrow \bigcup\limits_{i=1}^{2\log n} C_i$
  return C

Again now we fix an element $u \in \mathcal{U}$. Now we will calculate the probability that u is not covered in the union of all C_i's.

$$\mathbb{P}[u \text{ is not covered by } C] = \mathbb{P}[u \text{ is not covered by } C_i \text{ for all } i \in [2\log n]] \leq e^{-2\log n} = \frac{1}{n^2}$$

Hence the probability that e is covered is at least $1 - \frac{1}{n^2}$. Therefore

$$\mathbb{P}[\exists\ e \in \mathcal{U} \text{ is not covered by } C] \leq \sum_{u\in\mathcal{U}} \frac{1}{n^2} = \frac{1}{n}$$

Hence $\mathbb{P}[C \text{ is a set cover}] \geq 1 - \frac{1}{n}$. Now we have to bound the cost of C. By Markov's inequality we have

$$\mathbb{P}\left[c(C) \geq 6\log n \sum_{S\in\mathcal{S}} c(S)x_S^*\right] \leq \frac{1}{3}$$

$$\mathbb{P}\left[C \text{ is not a set cover OR cost of C} \geq 6\log n \sum_{S\in\mathcal{S}} c(S)x_S^*\right] \leq \frac{1}{n} + \frac{1}{3} \leq \frac{1}{2}$$

Therefore

$$\mathbb{P}\left[C \text{ is set cover AND } c(S) \leq 6\log n \sum_{S\in\mathcal{S}} c(S)x_S^*\right] \geq 12$$

Hence with probability at least $\frac{1}{2}$ we have a set cover C such that $c(C) \leq 6\log n \sum\limits_{S\in\mathcal{S}} c(S)x_S^*$ which gives us an $O(\log n)$-approximation algorithm for Set Cover problem.

Note:-
$O(\log n)$-approximatiobn is also the best we can do for set cover. Doing better than that is NP-hard.

2.2 Makespan Minimization

Makespan

Input: $\mathcal{M}$: Set of m machines
$\mathcal{J}$: Set of n jobs
$P \in \mathbb{N}^{m\times n}$: Matrix where P_{ij} is the time taken by machine i to complete job j.

Question: Given set of machines M, set of jobs $\mathcal{J}$ and the matrix of time taken by i^{th} machine to complete j^{th} job find an assignment $\sigma : \mathcal{J} \to \mathcal{M}$ of jobs to machines to minimize the makespan $S_\sigma = \max\{l_i : i \in \mathcal{M}\}$ where $l_i = \sum\limits_{j:\sigma(j)=i} P_{ij}$ i.e. time taken by machine i to complete all jobs assigned by σ

Theorem 2.2.1

Makespan problem is weakly NP-hard by reduction from subset-sum.

Note:-

Weakly NP-hard means there exists a pseudo polynomial time algorithm i.e. if all parameters are polynomially large the algorithm can solve the problem in polynomial time.

Theorem 2.2.2

It is NP-hard to approximate within a factor of 1.5

Here we will show a 2-approximate solution of makespan optimization. First let's construct the LP for makespan optimization.

2.2.1 LP Construction

We'll use the variable x_{ij} as an indicator for j^{th} job assigned to i^{th} machine. Then here is the LP:

$$\begin{aligned}
\text{minimize} \quad & T \\
\text{subject to} \quad & \sum_{i\in\mathcal{M}} x_{ij} \geq 1 \quad && \forall\, j \in \mathcal{J}, \\
& \sum_{j\in\mathcal{J}} P_{ij} x_j \leq T && \forall\, i \in \mathcal{M}, \\
& x_{ij} \geq 0 && \forall\, i \in \mathcal{M}, j \in \mathcal{J}
\end{aligned}$$

So here the first constrain basically says that every job assigned to some machine. The second constraint says that for every machine the total time taken by the machine to complete the jobs should be at most the makespan where T denotes the makespan. But this LP is not good enought. Consider the following example where there is only one job and $P_{i1} = m$ then $\text{OPT}_{LP} = 1$ by setting $x_{i1} = \frac{1}{m}$ where as actually the optimal makespan is m. Hence this LP will not work. We have to strengthen the LP.

So now assume we already know the optimal makespan T. Then if any $P_{ij} > T$ then we know that we can't assign the j^{th} job to i^{th} machine. So now we have the new updated LP:

$$\begin{aligned}
\text{minimize} \quad & 0 \\
\text{subject to} \quad & \sum_{i\in\mathcal{M}} x_{ij} \geq 1 \quad && \forall\, j \in \mathcal{J}, \\
& \sum_{j\in\mathcal{J}} P_{ij} x_j \leq T && \forall\, i \in \mathcal{M}, \\
& x_{ij} \geq 0 && \forall\, i \in \mathcal{M}, j \in \mathcal{J}, \\
& x_{ij} = 0 && \text{If } P_{ij} > T\, \forall\, i \in \mathcal{M}\ j \in \mathcal{J}
\end{aligned}$$

This basically checks the feasibility for a specific T. Hence now we can do a binary search over T's to find the smallest feasible T.

Theorem 2.2.3

By binary search $O(\log n)$ round we can find the smallest T such that $LP(T)$ is feasible.

Chapter 3

Bibliography