Notes for LPCO Course

1 Intro

This course is about **optimization problems**. What do we mean by this? Consider the following problem: Given positive reals x, y, maximize xy subject to the constraint that $x + y = 2$. The answer is $x = y = 1$, and the geometric interpretation is that the square has the largest area amongst all rectangles with a given perimeter. This was first shown by Euclid, and is a classic example of an optimization problem. This example is a special case of the following more general problem: given a polynomial $p(x)$ and a bounded interval $I \subseteq \mathbb{R}$ find the maximum or minimum of p over I. It is clear that the optimum is attained either at the points where the derivative vanishes, or at the boundaries of the interval.

Loosely speaking, an optimization problem consists of a domain $\mathcal{D}$ and an **objective function** (or cost function) $f : \mathcal{D} \to \mathbb{R}$, and the aim is to find a point in the domain that attains the minimum value of the objective function over $\mathcal{D}$ (a minimization problem) or attains the maximum value of the objective function (a maximization problem). In general, the problems may be unsolvable, since f may fail to have an optimum value over $\mathcal{D}$. However, if the domain is bounded and f is nice (continuous, let's say) then there will be both a minimum and a maximum. It is clear that by flipping the sign of the objective function a minimization problem becomes a maximization problem and vice versa, hence it is sufficient to focus on one of the two formulations. We now look at some more examples of optimization problem.

Another classical optimization problem is called Dido's problem. Dido (also called Elissa) was a princess of Tyre in Lebanon during the Phoenician times. She had to flee her brother Pygmalion due to a dynasty struggle. In her flight with her trusted companions she came to a place in North Africa, where she requested a piece of land for herself and her followers from a Berber king. The king allowed them to take as much land that can be enclosed by an oxhide. What should Dido do? She cut the bull hide into thin rectangular strips and formed an enclosed area in the shape of a circle. Amazingly, this is the optimal solution. The account is given in Virgil's epic poem *Aeneid*. The city she founded was Carthage, the great enemy-city of Rome. Unfortunately for us, we won't be covering optimization problems of this nature. [1]

A more specific class of optimization problems are of the following nature: We are not only given an objective function $f : \mathbb{R}^n \to \mathbb{R}$ as before, but some more functions $g_1, \ldots, g_m : \mathbb{R}^n \to \mathbb{R}$ and a vector $\mathbf{b} \in \mathbb{R}^m$. The aim is to find a $\mathbf{x} \in \mathbb{R}^n$ which not only optimizes f but also satisfies the constraints $g_i(\mathbf{x}) \leq \mathbf{b}_i$, for $i = 1, \ldots, m$; note the direction of the inequality does not matter since by flipping the signs we can always ensure that it is of the desired form. Such an optimization problem is called a **mathematical optimization problem** or **mathematical programming**; the m functions $g_1, \ldots, g_m$ are called the **constraint functions**. An example is the following projection problem: Suppose we are given a non-empty set S in $\mathbb{R}^n$ defined by certain constraint functions, and a point $\mathbf{p} \in \mathbb{R}^n$; our aim is to find the "projection of p onto S", i.e., a point in S nearest to p. What is the objective function here? It depends on how distances are measured in $\mathbb{R}^n$. Let's pick the euclidean norm for measuring distances. Then the problem is to find an $\mathbf{x} \in S$ such that $\|\mathbf{x} - \mathbf{p}\|_2$ is minimized. In fact, let's be more precise in defining the set S by a system of linear equations $Ax = \mathbf{b}$, where $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, where $m \leq n$. This particular class of optimization problems has more structure to it than what we have described above, namely within the domain of mathematical programming it comes from a special class of optimization problems called **convex optimization**, because

[1] This problem is called the isoperimetry problem and its solution is an extreme case of the isoperimetric inequality. Given a closed curve of length L if A is the area covered by the curve then the inequality states that $4\pi A \leq L^2$. Equality is achieved iff the curve is a circle; in other words, amongst all closed curves of fix length L, the circle encloses the maximum area. A similar inequality holds in three dimensions, where amongst all bodies of a fixed volume the sphere attains the smallest surface area, this is the reason why a rain drop is spherical in shape.

both the objective function and the constraint functions are convex. What does it mean that a function is convex?

A set $S \subseteq \mathbb{R}^n$ is called **convex** if for any two points in $\mathbf{x}, \mathbf{y} \in S$, the line segment joining the two points is contained in S; algebraically, for all $\lambda \in [0,1]$ the set of points $\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in S$. An equivalent definition involves the notion of **convex combination** of a set of points $\mathbf{p}_1, \ldots, \mathbf{p}_n$, namely the set of all points of the form $\sum_{i=1}^n \lambda_i \mathbf{p}_i$, where $\lambda_i \in [0,1]$ and $\sum_i \lambda_i = 1$.

Claim: A set S is convex iff S is closed under taking convex combinations.

Let $S \subseteq \mathbb{R}^n$ be a convex set. A function $f : S \to \mathbb{R}$ is called a **convex function** if for every $\mathbf{x}, \mathbf{y} \in S$ and $\lambda \in [0,1]$,

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}). \tag{1}$$

In terms of convex sets, a function is convex iff its **epigraph**, i.e., the set $\{(\mathbf{x}, y) \in \mathbb{R}^{n+1} : y \geq f(\mathbf{x})\}$, is a convex set. The following are examples of convex functions:

1. All **linear functions**, i.e., functions which satisfy the following property

$$f(c\mathbf{x} + d\mathbf{y}) = cf(\mathbf{x}) + df(\mathbf{y}), \tag{2}$$

   for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, and for all $c, d \in \mathbb{R}$, are convex; this is easy to verify from (1).

2. All norms are convex functions, in particular, the euclidean norm used in our example above is a convex function.

To see the claim for norms, recall that a norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ is a function that satisfies the following properties:

1. for all $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\| \geq 0$, with equality iff $\mathbf{x} = 0$;
2. for all $c \in \mathbb{R}$ and $\mathbf{x}$, $\|c\mathbf{x}\| = |c|\|\mathbf{x}\|$; and
3. the triangular inequality holds: for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\|\mathbf{x} + \mathbf{y}\| \leq \|bfx\| + \|\mathbf{y}\|$.

Clearly,

$$\|\lambda\mathbf{x} + (1-\lambda)\mathbf{y}\| \leq \|\lambda\mathbf{x}\| + \|(1-\lambda)\mathbf{y}\| = \lambda\|\mathbf{x}\| + (1-\lambda)\|\mathbf{y}\|.$$

Example of a non-convex function is $\sqrt{x}$, for positive x, and $\sin(x)$. Therefore, the objective function for the projection problem above is convex. But why is the domain convex? Recall that the domain wsa given by the solution to a system of inequalities $g_i(\mathbf{x}) \leq \mathbf{b}_i$, where g_i are convex functions. To see this we need two results.

LEMMA 1. *Intersections of convex sets is a convex set.*

Proof. For every $\mathbf{x}, \mathbf{y}$ in the common intersection we know that the line segment joining them belongs to all the sets, and hence to their intersection. **Q.E.D.**

LEMMA 2. *Let $g : \mathbb{R}^n \to \mathbb{R}$ be a convex function and $b \in \mathbb{R}$. Then the set $\{\mathbf{x} \in \mathbb{R}^n : g(\mathbf{x}) \leq b\}$ is a convex set.*

Proof. Follows from the definition of convexity and the property of convex combinations. **Q.E.D.**

But perhaps the most important aspect of convex programming is the uniqueness of the optimum. Recall that a **local minima x** for an objective function f with domain $\mathcal{D}$ is a point such that there is an ϵ-neighborhood of $\mathbf{x}$ such that for all points $\mathbf{y}$ in this neighborhood $f(\mathbf{x}) \leq f(\mathbf{y})$. In contrary to this, a **global minima x** is one such that for all points $\mathbf{y} \in \mathcal{D}$, $f(\mathbf{x}) \leq f(\mathbf{y})$. One reason why general optimization problems are difficult to solve is because there may be many local minimma that are situated far away from a global minima, which means that locally optimal decisions may not lead to a global optimum. In the case of convex programming, however, this situation does not arise:

THEOREM 3. *For a convex function $f : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D}$ is a convex set, every local minima is also a global minima.*

Proof. Let $\mathbf{x}$ be a local optimum and $\mathbf{z} \in \mathcal{D}$ be an arbitrary point. Then we know that there exists an ϵ-neighborhood of $\mathbf{x}$ on which $\mathbf{x}$ is the local minima. Consider the line segment joining $\mathbf{x}$ and $\mathbf{z}$ and let $\mathbf{y}$ be a point on this segment that is contained in the ϵ-neighborhood of $\mathbf{x}$. Then we have $f(\mathbf{x}) \leq f(\mathbf{y})$, but by definition $\mathbf{y} = \lambda\mathbf{x} + (1-\lambda)\mathbf{z}$ and it is in $\mathcal{D}$ since $\mathcal{D}$ is convex. Therefore, from the convexity of f we have

$$f(\mathbf{x}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{z}).$$

Since $(1-\lambda) \geq 0$, this implies that $f(\mathbf{x}) \leq f(\mathbf{z})$ for all $\mathbf{z} \in \mathcal{D}$. **Q.E.D.**

Remark: Note that we do not have a converse to this result, i.e., if a function has a unique optimum then it is convex. The functions $\sqrt{x}$ is a counterexample. ∎

Most of the course will be around a rather special class of convex programming, namely **linear programming**, i.e., an optimization problem where both the objective function and the constraint functions are linear functions. More precisely: Given a linear objective function $f : \mathcal{D} \to \mathbb{R}$ and m linear functions $g_i : \mathcal{D} \to \mathbb{R}$ a linear program is to

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \text{ subject to} \\ &\quad g_i(\mathbf{x}) \leq b_i \text{ for } i = 1, \ldots, m. \end{aligned} \tag{3}$$

The minimization part can be suitably replaced with maximization, by flipping the sign.

Dantzig was the first to realize the importance of this set of problems and gave the simplex algorithm to solve such problems. We will see this method. The simplex gives us more than one algorithms to solve LP depending upon certain choice of "pivot rules". For most of the known choices of these rules, LP takes exponential time in the worst case. Whether there is a choice of rules which gives a poly-time algorithm is still unknown. There are randomized algorithms whose expected running time is polynomial, we will see some of them. But we will see the first two deterministic polynomial time algorithms – Khachiyan's ellipsoid method, and Karmarkar's interior point method.

In CS we come across a different kind of optimization problem, namely where the domain is discrete. For instance, MST on a undirected weighted graph, or the TSP. Such optimization problems are called Discrete/Combinatorial Optimization problems. We will see how sometimes the LP solution to the problem can be used to obtain good approximations to the optimal solution, or occasionally even the optimal solution. The problem of optimizing over a discrete set is related to the integer optimization problem.

Other topics are the notion of duality in LP, which gives a unified proof to many theorems of the form: max-cut is equal to min-flow, or Menger's theorem (max-vertex disjoint paths = min-cut between two vertices).

Generalizations to SDP and vector programming.

1.1 Some interesting examples of LPs

Realizing that an optimization problem is an LP and formulating it in the form (3) is some times not straightforward, as illustrated by the following examples.

1. Flows in a graph. A **flow network** is a directed graph $G = (V, E)$ with a capacity function $c : E \to \mathbb{R}_{\geq 0}$ and two distinguished vertices, a source vertex s and a target vertex t. For every vertex v, let $I(v)$ denote the set of edges directed to v and similarly define $O(v)$. A flow in G is a function $f : E \to \mathbb{R}$ satisfying the following conditions:

   (a) Non-negativity: for all $e \in E$, $f(e) \geq 0$,

   (b) Capacity constraints: for all $e \in E$, $f(e) \leq c(e)$, and

   (c) Flow conservation: for all $v \in V \setminus \{s, t\}$, $\sum_{e \in I(v)} f(e) = \sum_{e \in O(v)} f(e)$.

   The size of the flow is defined as $\sum_{e \in O(s)} f(e) - \sum_{e \in I(s)} f(e)$, which measures the gradient of flow across the source node. The optimization problem is to obtain the largest flow for G. How do we model this as an LP? For every edge e introduce a variable x_e. Then the set of constraints is of the form: for all e, $0 \leq x_e \leq c(e)$, and similarly for flow conservation. The optimization function is simply $\sum_{e \in O(s)} x_e - \sum_{e \in I(s)} x_e$, which is clearly linear.

2. In Nature in year 2009 some scientists showed that there is a linear relationship between the total cumulative emissions and global temperature change for the last two centuries. How can they show such a claim? Or suppose we have some oil wells in the plane, and you want to lay a single pipeline that is optimal wrt its distance from all the oil wells, how will we do it? This is the problem of linear data-fitting. Basically, given a set of points $(x_i, y_i) \in \mathbb{R}^2$, $i = 1, \ldots, n$, we have to find a line $y = ax + b$ that best fits the data. One notion of best fitting could be to minimize $\sum_i (ax_i + b - y_i)^2$, but this optimization function is not linear. How about taking $\sum_i |ax_i + b - y_i|$ as our optimization function? Can we write an LP for this problem? This is not immediately evident because of the absolute value, however, we know that this function is still piecewise linear. The basic problem is the following: given a number $a \in \mathbb{R}$, we want to compute its absolute value. Can we write an LP to do that? Yes, and here is how: minimize x, subject to $x \geq a, -a$. From this observation, the LP for the problem follows: One way to do this is to introduce variables $e_1, \ldots, e_n$ for the absolute values, and a, b for the line, then the LP is minimize $\sum e_i$ subject to $e_i \geq (ax_i + b - y_i), -(ax_i + b - y_i)$.

3. Suppose we are given a convex polygon P by the equation of the lines corresponding to its segments, and we want to find the largest disc that can be fitted inside it P. On the face of it, the problem again doesn't appear to be linear because we are trying to fit a non-linear object, a disc, inside a piecewise linear object. The only variables that seem to be required are the coordinates (p, q) of the center of the disc and its radius r. We want to maximize r, subject to the constraint that p, q is inside the polygon and its distance to all the lines is at least r. The distance of (p, q) from a line $y = ax + b$ is

$$\left| \frac{q - ap - b}{\sqrt{1 + a^2}} \right|.$$

Let's suppose that we know the order of the lines around the polygon. Then the lines can be partitioned into a lower hull (where the slopes are increasing) and an upper hull where the slopes are decreasing. The value above is positive for the lines in the lower hull and negative for the lines in the upper. A proper choice of sign for each gives us the desired set of n inequalities.

2 Two Forms of LP – Canonical and Equational

In (3) we have seen one way to formulate an LP. There are many neat ways to express the same. The crucial point is that a linear functional has a very special structure:

LEMMA 4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a linear functional then f corresponds to the inner product operator corresponding to some vector $\mathbf{v}$, i.e., for all $\mathbf{x} \in \mathbb{R}^n$, $f(\mathbf{x}) = \langle \mathbf{v}, \mathbf{x} \rangle$.*

Proof. Let $\mathbf{e}_1, \ldots, \mathbf{e}_n$ be the standard basis vectors in $\mathbb{R}^n$ and $\mathbf{x} = \sum_i \alpha_i \mathbf{e}_i$. Then by linearity of f we have

$$f(\mathbf{x}) = \sum_i \alpha_i f(\mathbf{e}_i).$$

Let $\mathbf{v}_i := f(\mathbf{e}_i)$, then the claim follows. **Q.E.D.**

From this result it follows that for f and each g_i there is a corresponding vector whose inner-product operation gives us the function. Let $\mathbf{c}$ be the vector corresponding to the objective function, and $\mathbf{a}_i$ be the vectors corresponding to the constraint function. Then the LP in (3) can be expressed as

$$\begin{aligned} &\text{maximize } \langle \mathbf{c}, \mathbf{x} \rangle \text{ subject to} \\ &\quad \langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i \text{ for } i = 1, \ldots, m. \end{aligned} \tag{4}$$

Let A be the $m \times n$ matrix whose rows are the vectors $\mathbf{a}_i$'s then we can express the problem above even more succinctly:

$$\text{maximize } \langle \mathbf{c}, \mathbf{x} \rangle \text{ subject to } A\mathbf{x} \leq \mathbf{b} \tag{5}$$

This will be called the **canonical form** of LP. We will usually assume in this form that $m \geq n$.

The canonical form gives us a neat way to formalize LPs. However, given our comparatively better understanding of solving linear equations, we would ideally like to replace the complicated inequalities $A\mathbf{x} \leq \mathbf{b}$ by equality constraints $A\mathbf{x} = \mathbf{b}$ and very simple inequalities on the variables, namely that all the variables are positive; the transformation is also necessary for the simplex method. More formally, the **standard form** or **equational form** of an LP is the following:

$$\text{maximize } \langle \mathbf{c}, \mathbf{x} \rangle \text{ subject to } A\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq 0. \tag{6}$$

It is easy to see that this form is a special case of the canonical form. We next show the other way round, namely, obtaining the ef from cf.

First, how do we convert inequalities to equalities? Given an inequality, $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i$, we introduce a **slack variable** $y_i \geq 0$, and solve for $\langle \mathbf{a}_i, \mathbf{x} \rangle + y_i = b_i$. This would seem to address our concerns. However, there may be variables that are unconstrained, whereas the equational form emphasizes that *all variables* are positive. To address this issue, we introduce two positive variables w, z for each unconstrained variable x and express x as their difference, i.e., $x = w - z$, and substitute x by $w - z$ in all the equations. Thus in total we could have introduced at most $m + 2n$ new variables, m for each inequality and 2 for each of the n variables. Thus, wlog, we will always *assume in the equational form that the number of variables exceeds the number of inequalities.* Consider the following LP:

$$\begin{aligned} \text{maximize } & 2x_1 - 3x_2 \\ \text{s.t. } & x_1 - x_2 \leq 4 \\ & 3x_1 + x_2 \geq 2 \\ & x_2 \geq 0. \end{aligned}$$

The canonical form is obtained by flipping the sign of the second constraint:

$$\begin{aligned} \text{maximize } & 2x_1 - 3x_2 \\ \text{s.t. } & x_1 - x_2 \leq 4 \\ & -3x_1 - x_2 \leq -2 \\ & x_2 \geq 0. \end{aligned}$$

The equational form is obtained by adding slack variables x_3, x_4 to the first two constraints and as x_1 is unconstrained by expressing it as a difference of two positive numbers $x_5 - x_6$. The resulting ef is the following:

$$\begin{aligned} \text{maximize } & 2x_5 - 2x_6 - 3x_2 \\ \text{s.t. } & x_5 - x_6 - x_2 + x_3 = 4 \\ & -3x_5 + 3x_6 - x_2 + x_4 = -2 \\ & x_2, x_3, x_4, x_5, x_6 \geq 0. \end{aligned}$$

In the next few sections we will be mostly concerned with an lp in ef. We cannot overemphasize the point that statements valid in equational form do not carry over to the canonical form.

3 Basic Feasible Solutions and A First Algorithm for LP

We next try to understand the solution set to the system $A\mathbf{x} = \mathbf{b}$ subject to $\mathbf{x} \geq 0$, where A is an $m \times n$ matrix, $m \leq n$. Geometrically, the set of solutions to $A\mathbf{x} = b$ is an affine space (i.e., a translated version of a vector space). The set of **feasible solutions** are the solutions to this system that are in the positive orthant. For any feasible solution $\mathbf{x}$, let $\pi(\mathbf{x}) \subseteq [n]$ denote the set of indices j for which $x_j > 0$. We will show that amongst the feasible solutions, there are certain interesting solutions that will attain the optimum value, if there is an optimum value. We first cleanup our system of equations.

How can we check if an LP is feasible or not? We know that under elementary row and column operations the set of solutions to $A\mathbf{x} = \mathbf{b}$ remains unchanged. We claim that we can do these operations on A to obtain another system of equations with full row rank having the same set of solutions as the original system. The idea is to perform Gaussian elimination on the augmented matrix $[A|\mathbf{b}]$. If at any point we get a row whose first n entries are zero and the last entry is non-zero, we know that the original system of equations has no solution; if this does not happen, this means that the equation corresponding to that row is linearly dependent on other rows and hence is redundant in our system of equations and can be discarded. The system of equations $A'\mathbf{x} = \mathbf{b}'$ obtained at the end of this process has the following two properties:

1. the system has a solution and
2. the matrix A' is full row-rank.

Thus from now on we can assume that the original system $A\mathbf{x} = \mathbf{b}$ satisfies these two assumptions, and hence $\texttt{rank}(A) = m$. The definition of an equational form will always include these two assumptions.

If A is square, i.e., $m = n$, then $A\mathbf{x} = \mathbf{b}$ has a unique solution $A^{-1}\mathbf{b}$, and if this solution is feasible then clearly this is the optimal solution. In general, when $m < n$, can we find similar interesting solutions? Let $B \subseteq [n]$ be a set of m-indices of columns of A, and A_B be the $m \times m$ matrix obtained from A after removing all the columns from A except those indexed by B. Suppose the columns of A_B are linearly independent; we know that such a B exists because $\texttt{rank}(A) = m$. Then A_B is a non-singular matrix and we can solve the equation $A_B\mathbf{x}_B = \mathbf{b}$, and use this solution to construct a solution for $A\mathbf{x} = \mathbf{b}$ by amending $\mathbf{x}_B$ by introducing zeros at all the indices in $\mathbf{x}$ that are not in B. A **basic feasible solution** (bfs) of $A\mathbf{x} = \mathbf{b}$ is a feasible solution with the following properties:

1. there exists a set $B \subseteq [n]$ of size m such that A_B is non-singular,
2. $x_j = 0$ for all $j \notin B$, and
3. $\mathbf{x} \geq 0$.

The set B above is called a **feasible basis**. The set of variables indexed by B are **basic variables**, and those not in B are **non-basic variables**. Note that a bfs is uniquely determined by the set B, but two different feasible bases can yield the same bfs. For example, suppose

$$A = \begin{bmatrix} 1 & 5 & -3 & 1 \\ 0 & 3 & -2 & 1 \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

Then $B = \{1, 4\}$ is a feasible basis and $A_B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ and the corresponding bfs is $(1, 0, 0, 1)^t$. Consider the set $B = \{2, 3\}$, which is also feasible as $A_B = \begin{bmatrix} 5 & -3 \\ 3 & -2 \end{bmatrix}$ is non-singular and the corresponding bfs is $(0, 1, 1, 0)^t$. Why are bfs interesting?

THEOREM 5. *If an optimum solution to the ef of an LP exists, then there is a bfs that is optimum.*

Algorithm: Choose all subsets of m columns of A; check if A_B is non-singular; if so, compute the basic solution; if the basic solution is non-negative, then pick the one that maximizes the objective function. The algorithm takes $\binom{n}{m}$ steps; if $n = 2m$, then this is roughly 2^{2m}, which is exponential. What about the sizes of the numbers involved? We can show using Cramer's rule that the bit-size of all bfs's is polynomially bounded.

We first show the following result:

LEMMA 6. *A feasible solution* $\mathbf{x}$ *is a bfs iff the columns in* A *indexed by the set* $\pi(\mathbf{x})$ *are lid.*

Proof. One direction is easy: If $\mathbf{x}$ is a bfs corresponding to a feasible basis B then we know that any subset of columns of A_B is linearly independent, in particular, the columns corresponding to $\pi(\mathbf{x})$ are lid.

For the converse, suppose $\mathbf{x}$ is a feasible solution and the columns of $A_{\pi(\mathbf{x})}$, are lid. Then we know from basic linear algebra that there are $m - |\pi(\mathbf{x})|$ columns of A that are lid of the columns in $A_{\pi(\mathbf{x})}$ and together form a basis for the column space of A. Therefore, $\pi(\mathbf{x})$ along with the $m - |J|$ indices for these additional columns gives us a feasible basis B, and as $A\mathbf{x} = \mathbf{b}$ it follows that $A_B\mathbf{x}_B = \mathbf{b}$, and hence $\mathbf{x}$ is a bfs. **Q.E.D.**

So, for instance, in our example LP above, $\mathbf{x} = (1/3, 1/3, 0, 0)^t$ is a feasible solution and clearly the first two columns of A are lid, therefore, $\mathbf{x}$ is also a bfs. The proof above demonstrates how the same bfs can correspond to more than one feasible basis. For example, consider the earlier example but with $\mathbf{b} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$. Then $\mathbf{x} = (0, 0, 0, 2)^t$ is a solution but it corresponds to all the bases $\{1, 4\}$, $\{2, 4\}$ and $\{3, 4\}$, as we can combine the last column with any of the remaining columns to get a feasible basis (the corresponding $\mathbf{x}_B$ differs for each choice of B, namely it has 2 for the fourth column and zero everywhere else). We now prove Theorem 5.

Proof. Let $\mathbf{x}^*$ be an optimal solution to the LP. The basic idea is to give an iterative process that transforms $\mathbf{x}^*$ to a bfs by making its positive entries zero. For convenience, let $J := \pi(\mathbf{x}^*)$.

If $\mathbf{x}^*$ is not a bfs, then from the lemma above we know that the columns of A_J are ld, i.e., there is a vector $\mathbf{v} \in \mathbb{R}^{|J|}$ such that $A_J\mathbf{v} = \mathbf{0}$. Note that $\mathbf{v}$ can have negative entries. Define $\mathbf{w}$ as the vector $w_j := v_j$, for $j \in J$ and $w_j = 0$, for $j \notin J$. Then $A\mathbf{w} = A_J\mathbf{v} + A_{\overline{J}}\mathbf{0} = \mathbf{0}$. Thus $\mathbf{x}^* + \lambda\mathbf{w}$ is always a solution to the equation $A\mathbf{x} = \mathbf{b}$; in fact, we can add $\lambda\mathbf{w}$ to any solution of $A\mathbf{x} = \mathbf{b}$ to obtain another solution.

We next give an iterative procedure to obtain a bfs from $\mathbf{x}^*$.

1. Let $\mathbf{x}^0 \leftarrow \mathbf{x}^*$, $i \leftarrow 0$.
2. while columns in $\pi(\mathbf{x}^i)$ are ld and $\mathbf{x}^i \neq 0$
2.a. Let $\mathbf{w}^i$ be the vector constructed as above corresponding to $\mathbf{x}^i$.
$\mathbf{x}^{i+1} := \mathbf{x}^i + \lambda\mathbf{w}^i$, where λ is choosen as the smaller
in absolute value of the following two quantities
$\max -(x^i_j/w^i_j)$, for $j \in \pi(\mathbf{x}^i)$ and $w^i_j > 0$
and
$\min(x^i_j/|w^i_j|)$, for $j \in \pi(\mathbf{x}^i)$ and $w^i_j < 0$.
◁ *The choice of* λ *ensures that* $\mathbf{x}^{i+1} \geq 0$.
◁ *Moreover, if* k *is an index that defines* λ *then* $x^{i+1}_k = 0$.
$i \leftarrow i + 1$.
3. Return $\mathbf{x}^i$ as a bfs.

When the while-loop terminates, we will either reach a bfs, or the zero vector; the latter case is very special as in this case $A\mathbf{x} = \mathbf{0}$ has only one bfs, namely the origin, if the LP is bounded.

The crucial claim is that the iterative approach above does not change the value of the objective function, i.e., $\langle \mathbf{c}, \mathbf{x}^* \rangle = \langle \mathbf{c}, \mathbf{x}^i \rangle$, for all $i \geq 0$. We only prove this for $i = 1$. By construction $\langle \mathbf{c}, \mathbf{x}^1 \rangle$ is $\langle \mathbf{c}, \mathbf{x}^* \rangle + \lambda \langle \mathbf{c}, \mathbf{w}^1 \rangle$. We claim that $\langle \mathbf{c}, \mathbf{w}^1 \rangle = 0$. For sake of contradiction, suppose $\langle \mathbf{c}, \mathbf{w}^1 \rangle > 0$. Then there exists a $\beta > 0$ sufficiently small such that $\mathbf{x}^* + \beta \mathbf{w}^1$ is a feasible solution and

$$\langle \mathbf{c}, \mathbf{x}^* + \beta \mathbf{w}^1 \rangle = \langle \mathbf{c}, \mathbf{x}^* \rangle + \beta \langle \mathbf{c}, \mathbf{w}^1 \rangle > \langle \mathbf{c}, \mathbf{x}^* \rangle.$$

But this is a contradiction as $\mathbf{x}^*$ was an optimum. We can similarly argue that if $\langle \mathbf{c}, \mathbf{w} \rangle < 0$, then there exists a $\beta > 0$ sufficiently small such that $\mathbf{x}^* - \beta \mathbf{w}$ has a larger objective value, which again gives a contradiction. Note that if the objective function is unbounded on the set of feasible solutions then we cannot derive this contradiction.

Q.E.D.

A point $\mathbf{x}$ in a convex set S is called **extreme** if it cannot be expressed as a convex combination of any two other points $\mathbf{y}, \mathbf{z} \in S \setminus \{\mathbf{x}\}$, i.e., $\mathbf{x}$ is not contained in the interior of any line segment in S. There are two other properties of bfs that make them interesting: (i) they are "extreme" and (ii) given a bfs, there is an objective function for which only it attains the optimum value. The next two lemmas give the proof.

LEMMA 7. *A feasible solution is basic iff it is extreme.*

Proof. The easy part first: a bfs is extreme. If a bfs $\mathbf{x}$ is not extreme, then there are feasible solutions $\mathbf{y}, \mathbf{z}$ distinct from $\mathbf{x}$ and $\lambda \in [0, 1]$ such that

$$\mathbf{x} = \lambda \mathbf{y} + (1 - \lambda)\mathbf{z}.$$

Then considering this equation coordinate-wise we have for all $j \in [n] \setminus \pi(\mathbf{x})$, $0 = \lambda y_j + (1 - \lambda) z_j$, which is possible iff $y_j = z_j = 0$. Suppose B is an extension of $\pi(\mathbf{x})$ such that B is a feasible set corresponding to $\mathbf{x}$, i.e., $A_B \mathbf{x}_B = \mathbf{b}$. Since the entries of $\mathbf{y}$ and $\mathbf{z}$ are zero outside $\pi(\mathbf{x})$, we have $A\mathbf{y} = A_B \mathbf{y}_B = \mathbf{b}$, which implies that $\mathbf{y}_B = A_B^{-1}\mathbf{b} = \mathbf{x}_B$, i.e., $\mathbf{y}$ and $\mathbf{x}$ agree on the indices in B and hence on $\pi(\mathbf{x})$; similarly for $\mathbf{z}$. Thus $\mathbf{x} = \mathbf{y} = \mathbf{z}$, which gives us a contradiction. Therefore, $\mathbf{x}$ must be extreme.

Assume that $\mathbf{x}$ is a feasible solution that is extreme. If $\mathbf{x}$ is not a bfs then the columns indexed by the positive entries of $\mathbf{x}$ are ld, i.e., there is a vector $\mathbf{v}$ such that $A\mathbf{v} = 0$. For $\lambda > 0$ small enough we know that $\mathbf{x} \pm \lambda \mathbf{v}$ are also feasible solutions. Moreover

$$\mathbf{x} = \frac{1}{2}(\mathbf{x} + \lambda \mathbf{v}) + \frac{1}{2}(\mathbf{x} - \lambda \mathbf{v}).$$

i.e., $\mathbf{x}$ is not extreme which is a contradiction. Thus an extreme feasible solution is also a bfs. **Q.E.D.**

We will need the following lemma later:

LEMMA 8. *Given a bfs $\mathbf{x}^*$ to the sytem of inequalities $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, there is a vector $\mathbf{c}$ such that $\mathbf{x}^*$ attains the maximum value wrt this vector over all the feasible set, i.e., for all other feasible solutions $\mathbf{y}$, $\langle \mathbf{c}, \mathbf{x}^* \rangle > \langle \mathbf{c}, \mathbf{y} \rangle$.*

Proof. Let B be a basis corresponding to $\mathbf{x}^*$. Then define $\mathbf{c}$ as $c_j = 0$, for $j \in B$, and $c_j = -1$, for $j \notin B$. Clearly, $\langle \mathbf{c}, \mathbf{x} \rangle = 0$, and for any other feasible solution $\mathbf{y}$ we know that there is a $j \notin B$ such that $y_j > 0$, and hence $\langle \mathbf{c}, \mathbf{y} \rangle < 0$. **Q.E.D.**

Note that in the proofs above we have crucially used the fact that $\mathbf{x} \geq 0$. Dropping this assumption and just looking at the set $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}\}$ will not give us extreme points, as is obvious from the example $x + y = 1$ (however, $x + y = 1$ *and* $x, y \geq 0$ has extreme points); intuitively, the solutions to the system $A\mathbf{x} = \mathbf{b}$ define an affine space of dimension $n - m$, and hence cannot contain zero-dimensional points such as vertices. In general, we can define the notion of a bfs for any lp as a point where some n constraints hold as equalities. However, it may not be true anymore that a bfs attains opt, even if an opt exists; e.g., the following lp in canonical form maximize $x + y$ s.t. $x + y \leq 1$ has an opt, but no bfs. As we said earlier, this distinction between equational form and other forms of lp cannot be overemphasized. In the following section, we give a geometric interpretation of the results above.

4 Some Geometry Behind Linear Programming

We start with some basic definitions. We know that $\mathbb{R}^n$ is a vector space over the field of reals. That means we can define linear combinations of vectors, the linear span of a set of vectors, and the linear space spanned by a set of vectors. A **linear subspace** of $\mathbb{R}^n$ is a set closed under addition of vectors and scalar multiplication, i.e., it is a subset which itself forms a smaller vector space in $\mathbb{R}^n$. Examples of linear subspaces are lines and planes containing the origin in $\mathbb{R}^3$. The dimension of a subspace is the maximum number of linearly independent vectors in it. Any $(n-1)$ dimensional subspace is the set of all points that are orthogonal to a vector $\mathbf{a}$; such a space is called a **linear hyperplane**, i.e., a hyperplane containing the origin. A neat way to capture any k-dimensional linear subspace L is as the intersection of $(n-k)$ linear hyperplanes, or as the solution set to the system of equations $A\mathbf{x} = 0$, where A is an $(n-k) \times n$ matrix. Thus the $(n-k)$ rows of A correspond to a set of linearly independent vectors in the space orthogonal to L, i.e., the set $\{\mathbf{y} | \langle \mathbf{y}, \mathbf{x} \rangle = 0, \text{ for all } \mathbf{x} \in L\}$.

A line, in general, is not a linear subspace since it may not contain the origin. However, it essentially has the structure of a linear subspace, in the sense that if we translate it to the origin then we can work with this translated structure as a linear subspace and translate the results back. The notion that captures this geometry is that of an **affine subspace** that is a set of the form $\mathbf{a} + L$, where $\mathbf{a} \in \mathbb{R}^n$ and L is a linear subspace. Thus an affine subspace is merely a translation of a linear subspace. Examples would be any point, line, or plane in $\mathbb{R}^3$. Other notions carry as well. In particular, there is a notion corresponding to linear combination. Given $(k+1)$ points $\mathbf{p}_0, \ldots, \mathbf{p}_k$ in an affine space $\mathcal{A}$, we translate $\mathbf{p}_0$ to the origin, take a linear combination, and translate back by $+\mathbf{p}_0$ to get a point in $\mathcal{A}$. The point so obtained has the form

$$\sum_{i>0} \alpha_i (\mathbf{p}_i - \mathbf{p}_0) + \mathbf{p} = \sum_{i>0} \alpha_i \mathbf{p}_i + (1 - \sum_{i>0} \alpha_i) \mathbf{p}_0.$$

Thus an **affine combination** of a set of points $\mathbf{p}_0, \ldots, \mathbf{p}_k$ is a combination of the form

$$\sum_{i=0}^{k} \alpha_i \mathbf{p}_i, \text{ where } \sum_{i=0}^{k} \alpha_i = 1.$$

An **affine hull** of a set of points or **affine space** spanned by of a set of points is the set of all affine combinations of those points. It is clear that convex combinations and convex hulls are special type of affine combinations. An affine hull of a set $S \subseteq \mathbb{R}^n$ is the set of all affine combinations of the points in S, or the smallest affine space containing S. There is a natural notion corresponding to linear dependence: The points $\mathbf{p}_0, \ldots, \mathbf{p}_k$ are **affinely dependent** if the points the points $\mathbf{p}_i - \mathbf{p}_0$ are linearly dependent, i.e., there exists α_i's not all zero such that

$$\sum_{i>0} \alpha_i (\mathbf{p}_i - \mathbf{p}_1) = 0$$

which is the same as saying

$$\sum_{i=0}^{k} \alpha_i \mathbf{p}_i = 0 \text{ where } \sum_{i} \alpha_i = 0.$$

Thus in $\mathbb{R}^n$ there are $(n+1)$ affinely independent points (e.g., the n standard vectors e_i and the origin). The **dimension** of an affine subspace is one less than the maximum number of affinely independent points in it, or the dimension of the underlying linear subspace. Certain special affine subspaces are interesting: Affine subspaces in $\mathbb{R}^n$ of dimension 0, 1, 2 and $(n-1)$ are called points, lines, planes, and **hyperplanes**, respectively. In other words, a hyperplane is the solution set to the equation $\langle \mathbf{a}, \mathbf{x} \rangle = c$, for some vector $\mathbf{a} \in \mathbb{R}^n \setminus \mathbf{0}$ and $c \in \mathbb{R}$; a **linear hyperplane** is one where $c = 0$. Any k-dimensional affine subspace can be expressed as the intersection of $(n-k)$ hyperplanes. The **dimension of a set** in $\mathbb{R}^n$ is the dimension of the smallest affine space containing it.

To understand lps geometrically, we need to understand the geometry of linear inequalities. Given a hyperplane h, a **halfspace** is the set of points on one side of it. More precisely, suppose h is given by the equation $\langle \mathbf{a}, \mathbf{x} \rangle = b$ then we can associate two halfspaces with h

$$h^+ := \{\mathbf{x} : \langle \mathbf{a}, \mathbf{x} \rangle \geq b\}$$

and

$$h^- := \{\mathbf{x} : \langle \mathbf{a}, \mathbf{x} \rangle \leq b\}.$$

Clearly, $h = h^+ \cap h^-$. These are the closed halfspaces associated with h; if we replace $\geq$ and $\leq$ by $>$ and $<$ respectively then we obtain open halfspaces. A **polyhedron** is the intersection of finitely many halfspaces in $\mathbb{R}^n$. Since intersection of convex sets is convex, a polyhedron is always a convex set. They can be unbounded, as in the case of a halfspace.[2] A **polytope** is a bounded polyhedron. Now it is clear that the set of feasible solutions of an LP, whether in equational or canonical form, is given as the common intersection of closed halfspaces, and hence is a polyhedron. In fact, any convex set can be represented as the intersection of hyperplanes (not necessarily finite); e.g., the unit disc in the plane can be represented as the intersections of the hyperplanes parametrized . Some polytopes are specially interesting: A **simplex** is the convex hull of a set of affinely independent points. A d**-simplex** is a simple whose dimension is d. Thus a 1-simplex is a line segment (two affinely independent points), a 2-simplex is a triangle (three affinely independent points), and a 3-simplex is a tetrahedron (four affinely independent points). A hyperplane h is called a **supporting hyperplane** for a polyhedron P if $P \cap h \neq \emptyset$ and $P \subset h^+$ or $P \subset h^-$, i.e., P intersects h and is strictly contained in one of the halfspaces defined by h. The intersection of a supporting hyperplane with a polyhedron is called its **face**. A vertex of P is a face of dimension 0, an edge is a face of dimension 1, and a **facet** is a face of dimension one less than the dimension of P. A face can have more than one supporting hyperplane, as is clear in the case of a vertex. The faces of P form a poset under subset inclusion.

The following theorem shows the geometrical interpretation of bfs:

THEOREM 9. *Let P be the set of all feasible solutions of an LP in equational form. Then $\mathbf{x}$ is a vertex of P iff it is a bfs.*

Proof. We shows that a bfs is a vertex. We know that for every bfs $\mathbf{y}$ there is an objective function $\mathbf{c}$ such that $\langle \mathbf{y}, \mathbf{c} \rangle = 0$ and for all other feasible solutions this value is negative. Thus there is a supporting hyperplane through the origin that meets P exactly at $\mathbf{y}$.

To show that a bfs has to be a vertex, we show that vertices are extreme points of P. Suppose a vertex $\mathbf{v}$ is not an extreme point and is on the segment joining $\mathbf{u}, \mathbf{w} \in P$. Since $\mathbf{v}$ is a vertex we know that there is a supporting hyperplane h that meets this segment exactly in $\mathbf{v}$. This is possible iff $\mathbf{u}$ and $\mathbf{w}$ are on opposite sides of h, which gives us a contradiction since h is a supporting hyperplane.

Q.E.D.

4.1 Some Results from Convex Geometry

LEMMA 10 (Radon's Lemma). *Any set of $d+2$ points in $\mathbb{R}^d$ can be partitioned into two non-empty subsets such that the convex hull of these two sets intersect.*

Proof. Suppose $\mathbf{p}_0, \ldots, \mathbf{p}_{d+1}$ are our points. Then we know that any $d+2$ points are affinely dependent. Hence there exists α_i's, not all zero, such that

$$\sum_i \alpha_i \mathbf{p}_i = 0 \text{ and } \sum_i \alpha_i = 0.$$

Note that some of the α_i's must be positive and some negative; let P and N be the corresponding index sets. Then we have

$$S := \sum_{i \in P} \alpha_i = \sum_{i \in N} |\alpha_i|$$

and

$$\sum_{i \in P} \alpha_i \mathbf{p}_i = \sum_{i \in N} |\alpha_i| \mathbf{p}_i.$$

Dividing both sides by S we get the desired convex combination and the partition of the point set. **Q.E.D.**

[2] There are two ways to define unboundedness of polyhedra: one, if the supremum of the norms of all points in the set is unbounded, and second, if the set contains a *ray* in it. The first definition uses the notion of a norm, the second is purely algebraic and so is independent of any norm. However, the first applies to any set, but the second does not.

THEOREM 11 (Caratheodory's theorem). *If a point in $\mathbb{R}^d$ belongs to the convex hull of a set of n points P, then it can be expressed as a convex combination of at most $d+1$ points from P.*

Proof. Suppose $\mathbf{b}$ is a convex combination of $\mathbf{p}_1, \ldots, \mathbf{p}_n$. Then we know that there exists a solution to the following LP

$$P\mathbf{x} = \mathbf{b}, \langle \mathbf{x}, \mathbf{1} \rangle = 1, \mathbf{x} \geq 0.$$

This can be reduced to the equational form by moving the constraint $\langle \mathbf{x}, \mathbf{1} \rangle = 1$ into the system of equations by adding a new row to P of all ones; let P' be the resulting $(d+1) \times n$ matrix. Then we know that the following system has a feasible solution

$$P'\mathbf{x} = \begin{pmatrix} \mathbf{b} \\ 1 \end{pmatrix}, \mathbf{x} \geq 0.$$

That means there is a bfs $\mathbf{x}$ with the corresponding index set B of size $(d+1)$, which implies that $\mathbf{b}$ can be expressed as a convex combination of at most $(d+1)$ points, corresponding to the columns indexed by B. **Q.E.D.**

Consider the following problem: If two unit line segments $[a, b]$ and $[c, d]$ in the plane are such that $\text{diam}(a, b, c, d) \leq 1$, then show that the line segments intersect. With some geometric manipulations it is possible to show this claim. But note that line segments are 1-simplices in $\mathbb{R}^2$, therefore, we can generalize the problem to higher dimension as follows: Given two regular $(d-1)$ simplices Δ_1, Δ_2 in $\mathbb{R}^d$ suppose the diameter, i.e., the maximum distance any pair of points, of the union of their vertices is at most one then the two simplices intersect, and in fact, share at least $(d-2)$ vertices. For $d = 2$ the claim is trivial, though for line segments we asked for something non-trivial. The problem is solved in three dimensions, and remains open for higher dimensions.

5 Dantzig and the Simplex Method

5.1 Brief Biography and Buildup to LP

His full name was George Bernard Dantzig, son of a mathematician Tobias Dantzig with his wife Anja Ourisson; they had met while Tobias was studying mathematics at the Sorbonne. Dantzig was named after George Bernard Shaw, hoping that he would become a writer like his namesake. Incidentally, his younger brother was named after Henri Poincaré and later on became an applied mathematician. [3]

Dantzig completed his undergraduate in mathematics and physics from the university of maryland in 1936. Subsequently he joined university if michigan, ann arbor, to do a ma in mathematics, which he completed in 1938. The only course he enjoyed there was a statistics course by HC Carver, and found the rest of the curriculum excessively abstract. After finishing his ma he got a job as a statistical clerk at the Bureau of Labour Statistics. This turned out to be a fateful decision for him. While working at the Bureau he was asked to review a paper by Jerzy Neyman, a leading statistician of his day. Dantzig was so impressed by the logically based approach to statistics in the paper, and not just a bag of tricks that he decided to do a doctorate under Neyman, who had by now moved to UCB. In 1939 Dantzig enrolled in the phd program of UCB. He took only two courses from Neyman, and had a remarkable experience in one of them, so remarkable that it is legend now.

Dantzig arrived lated in one of the lectures of Neyman, and saw two problems mentioned on the blackboard. He mistook these to be homework problems and worked hard on them. He found them to be more challenging than the usual ones, but still managed to solve them, and submitted the solution to Neyman. As it turned out, these problems were actually two open questions in the theory of mathematical statistics. One of these would later become part of his theses and was published in 1940; the other, for appeared only a decade later in 1951 as a joint paper. His work in the theses turns out to be fundamental later in the development of simplex method and linear programming. Here is the event described in his own words:

> During my first year at Berkeley I arrived late one day to one of Neyman's classes. On the blackboard were two problems which I assumed had been assigned for homework. I copied them down. A few days later I apologized to Neyman for taking so long to do the homework - the problems seemed to be a little harder to do than usual. I asked him if he still wanted the work. He told me to throw it on his desk. I did so reluctantly because his desk was covered with such a heap of papers that I feared my homework would be lost there forever.
>
> About six weeks later, one Sunday morning about eight o'clock, Anne and I were awakened by someone banging on our front door. It was Neyman. He rushed in with papers in hand, all excited: "I've just written an introduction to one of your papers. Read it so I can send it out right away for publication." For a minute I had no idea what he was talking about. To make a long story short, the problems on the blackboard which I had solved thinking they were homework were in fact two famous unsolved problems in statistics. That was the first inkling I had that there was anything special about them.

Without completing the formal requirement for getting a phd, Dantzig joined the is air force office of statistical control in pentagon in order to contribute to the war effort in 1941. Some of the work there involved planning. He returned to UCB in 1946 to complete his phd requirements. Though he was later offered a position at Berkeley (though at a very low salary) he decided to become a mathematical advisor to the is air force comptroller's office. The second fateful decision set him on a path to the discover of linear programming and the simplex method. This is what he had to say:

> My own contributions grew out of my World War II experience in the Pentagon. During the war period (1941– 45), I had become an expert on programming-planning methods using desk calculators. In 1946, I was Mathematical Advisor to the US Air Force Comptroller in the Pentagon. I had just received my PhD (for research I had done mostly before the war) and was looking for an academic position that would pay better than a low offer I had received from Berkeley.[4] In order to entice me to not take another job, my Pentagon colleagues, D. Hitchcock and M. Wood,

[3] Tobias was very impressed by Poincaré and wrote a book on the bequest of Poincaré's scientific philosophy.

[4] "A grand salary of $1400 per annum."

challenged me to see what I could do to mechanize the planning process. I was asked to find a way to more rapidly compute a time-staged deployment, training and logistical supply program. In those days mechanizing planning meant using analog devices or punch-card equipment. There were no electronic computers.

5.2 The Literature on Linear Inequalities before 1947

Here we only mention the significant literature that was know till 1947. Though all of them was unknown to Dantzig when he started to work on lps. He was instead influenced by a work of the economist Leontief from 1932 who had given a modeling of what he called the "Interindustry Input-Output Model of the American Economy", for which he obtained the Noble prize in 1976. Dantzig remarks: "I greatly admired Leontief for having taken the three steps necessary to achieve a successful application: 1. Formulating the inter-industry model. 2. Collecting the input data during the Great Depression. 3. Convincing policy makers to use the output."

1. Even though there was a wealth of literature on solving linear equations, until 1947 there was very little for linear inequalities; perhaps one reason being that the applicability of such solutions was not clear. The earliest known work on linear inequalities dates back to Jean-Baptiste Joseph Fourier in 1826 (Solutions d'une question particulère du calcul des inégalités), who gave an elimination style approach to solving linear inequalities and has some overlap with the simplex method. There was a subsequent paper in 1911 by de la Vallée Poussin (Sur la Methode de l'approximation minimum).

2. One of the most important contribution to the field before 1947 was by the russian mathematician Leonid V. Kantorovich, who was the head of dept. of mathematics at the institute of mathematics and mechanics of leningrad state university. He was consulted by some engineers for from the laboratory if the veneer trust. The outcome of this interaction was a report on linear programming published in 1939, which only becomes available to the west in the 1950s. There were some later followup papers that were known in the west.

3. The other key principal contributor was the mathematician and economist TJ Koopmans, who later got a nobel prize in economics for his contributions. His earliest work on lp was during the war in 1942, and therefore remained classified until 1970. In 1947, he formulated what is called the transportation problem. However, the transportation problem was already earlier published in 1941 by an algebraist from MIT, named FL Hitchcock, but was unknown to him (just as Knatorovich's work was unknown to the rest). Koopmans was one the earliest people that Dantzig met in June 1947, from where he got to know about the earlier work of Koopman. Though they had a slow start, Koopman soon realized the significance of Dantzig's work and was the first person to organize a conference (Conference on Activity Analysis of Production and Allocation) in 1949 on lp giving a much needed impetus to the field.

4. The fourth significant contributor was J. von Neumann. His work on two players game in 1928 and book with Oscar Mrogenstern published in 1944 are intimately connected to lp. Dantzig also visited him in 1947, and received a lecture on duality.

The four most pre-1947 publications considered significant by Dantzig are Fourier, de la Vallee Poussin, Kantorovich and Hitchcock. He mentions that all except Kantorovich proposed a solution method similar to the way we describe the simplex method today. Given these results, Dantzig, in retrospect, terms lp as an anachronism. According to him his significant contribution was realizing that the objective function can be modelled as a linear function, which is made clear in the following quote:

> What seems to characterize the pre-1947 era was lack of any interest in trying to optimize. T. Motzkin in his scholarly thesis written in 1936 cites only 42 papers on linear inequality systems, none of which mentioned an objective function.

5.3 The Simplex in the Simplex Method

If we follow the idea of gradient descent then we know that in order to maximize a function over a polyhedra we have to walk along the edges of the polyhedra from one vertex to a neighboring vertex. Even though this is how we conceptually think of the simplex method now, Dantzig considered this very inefficient, and instead wated a method that went directly through the interior. We retrace Dantzig's original setting and motivation, which also reveals the terminology of the method.

The problem that Dantzig addressed in his thesis was "proving the existence of optimal Lagrange multipliers for a semi-infinite linear program with bounded variables." Let's see what that means. Let Ω be a probability space with probability distribution $dP(u)$, where $u \in \Omega$. The problem was to prove the existence of a subset $\omega \in \Omega$ that satisfies the conditions of the so called Neymon-Pearson Lemma: if $\alpha \in (0, 1)$ then

1. the size of ω is α
$$\int_\omega dP(u) = \alpha.$$
2. For a given vector function $f : \Omega \to \mathbb{R}^{m-1}$ and $\mathbf{b} \in \mathbb{R}^{m-1}$
$$\int_\omega f(u)dP(u) = \mathbf{b}\alpha,$$
i.e., the expected value of f over ω is $\mathbf{b}$.
3. For a given scalar function $f : \Omega \to \mathbb{R}$
$$\int_\omega g(u)dP(u) = z\alpha,$$
i.e., the unknown expected value z of g over ω is to be minizmed.

Instead of finding ω, we can find a characteristic function $\phi : \Omega \to \{0, 1\}$ for the set. With this notation the three conditions above are modified to

$$\int_\Omega \phi(u)dP(u) = \alpha,$$
$$\int_\Omega \phi(u)f(u)dP(u) = \mathbf{b}\alpha,$$
$$\int_\Omega \phi(u)g(u)dP(u) = z\alpha.$$

Now consider the discrete version of the problem where we pick n sample points $u_1, \ldots, u_n \in \Omega$. Let $\Delta_1, \ldots, \Delta_n$ be the corresponding discrete point probabilities $dP(u_j)$; note that n may be finite or infinite. For each j, introduce a variable x'_j to denote the unknown $\phi(u_j)$. Then the equations above reduce to the following:

$$\alpha^{-1} \sum_{j=1}^{n} x'_j \Delta_j = 1,$$
$$\alpha^{-1} \sum_{j=1}^{n} x'_j f(u_j) \Delta_j = \mathbf{b},$$
$$\alpha^{-1} \sum_{j=1}^{n} x'_j g(u_j) \Delta_j = z.$$

For the sake of convenience, define $x_j := x'_j \Delta_j / \alpha$, then we have the simplified integer program

$$\sum_{j=1}^{n} x_j = 1, \ \sum_{j=1}^{n} x_j A_j = \mathbf{b}, \ \sum_{j=1}^{n} x_j c_j = z,$$

where $A_j := f(u_j)$, $c_j := g(u_j)$ and our aim is to minimize z. Dantzig considered the lp relaxation of the above program, i.e., $x_j \in [0, 1]$. Since n could be infinite, Dantzig considered it natural to look at the "column geometry". The geometric interpretation is as follows. Consider the m-dimensional points (A_j, c_j) in $\mathbb{R}^m$. Consider the "solution line" $(\mathbf{b}, z)$ in $\mathbb{R}^m$. Our aim is to take find a convex combination of the points (A_j, c_j) that is on the line $(\mathbf{b}, z)$ and attains the smallest value for the z-coordinate. In order to achieve that, the algorithm consider the simplicies corresponding to the "triangulation" of the faces of the polyhedra defined by the points (A_j, c_J). Clearly, we are interested in those simplicies that intersect the line $(\mathbf{b}, z)$. It is this interpretation that is the origin of the name simplex method.

With this geometric interpretation, the algorithm can be described geometrically as follows. Suppose we have an $(m-1)$-dimensional simplex Δ of m points that intersects the line $(\mathbf{b}, z)$. Then we find a point (A_j, c_j) farthest from the plane containing the simplex and below it. The m dimensional simplex defined by (A_j, c_j) and Δ intersects the solution line at some face. Replace Δ with the simplex corresponding to this face; note that the value of z has decreased. If we reach a simplex Δ, such that all the points (A_j, c_j) are on one side of the containing hyperplane then we can stop.

In this geometric description, it seems we are going through the interior of the polyhedra containing the points (A_j, c_j). However, Dantzig realized later, using duality, that this is the same as moving along the edges of the polyhedron in the "row geometry", an idea he had dispensed earlier as too slow. To quote him, "It is my opinion that any well trained mathematician viewing the linear programming problem in the row geometry of the variables would have immediately come up with the idea of solving it by a vertex descending algorithm as did Fourier, de la Vallee Poussin, and Hitchcock before me – each of us proposing it independently of the other. I believe, however, that if anyone had to consider it as a practical method, as I had to, he would have quickly rejected it on intuitive grounds as a very stupid idea without merit. My own contributions towards the discovery of the simplex method were (1) independently proposing the algorithm, (2) initiating the development of the software necessary for its practical use, and (3) observing by viewing the problem in the geometry of the columns rather than the rows that, contrary to geometric intuition, following a path on the outside of the convex polyhedron, might be a very efficient procedure."

6 The Simplex Method

The input to an algorithm solving lp is A, $\mathbf{b}$, $\mathbf{c}$, which define the lp in equational form. The output of the algorithm should be one of the following: (i) the lp is infeasible, or (ii) the lp is feasible but unbounded (i.e., opt does not exist), or (iii) the lp is feasible and bounded, in which case we output an optimum solution. We next describe a method, starting with concrete examples, that allows us to detect all these cases. Start with the examples of Chapter 5 in Matousek-Gartner. These examples clearly demonstrate the functioning of the method, and the possible problems that occur.

The simplex method is an approach for solving LPs in *equational form*; the difference between various formulations of the simplex method arise because of the various options of performing a certain step, called the "pivot step". The method is famously attributed to Dantzig, but some earlier work of Fourier and Kantorovich also contains related approaches.

The general strategy is to start from a bfs and obtain another bfs from it while increasing the value of the objective function. While doing so, the algorithm also figures out whether the **LP is feasible**, i.e., the set of feasible solutions is not empty and the objective function is bounded on the feasible set. We assume that we have a bfs in hand to begin with; note that checking for feasibility of an LP is as hard as getting the optimal solution, so this is not very straightforward; however, the examples reveal situations in which a starting bfs is evident; the idea otherwise is to construct an auxillary lp whose optimum gives us a bfs for the original lp.

Consider an LP in ef. The simplex method, as in the examples above, constructs a sequence of tableaus. Each tableaux corresponds to a feasible basis B, and gives us the bfs and the value of the objective function at the bfs. Let $\mathbf{x}_B$, $\mathbf{x}_N$ represent the basic and nonbasic variables corresponding to a feasible basis B. A **simplex tableaux**, or Tucker's condensed schemata, $T(B)$ corresponding to B is a set of $m+1$ equations in $n+1$ variables of the form

$$\begin{aligned} \mathbf{x}_B &= \mathbf{p} + Q\mathbf{x}_N \\ z &= z_B + \langle \mathbf{r}, \mathbf{x}_N \rangle, \end{aligned} \tag{7}$$

where $\mathbf{p} \in \mathbb{R}^m_{\geq 0}$, Q is an $m \times (n-m)$ matrix, $z_B \in \mathbb{R}$, and $\mathbf{r} \in \mathbb{R}^{n-m}$. Moreover, the set of solutions of the system above is the same as that of the system of equations $A\mathbf{x} = \mathbf{b}$, and $z = z_B + \langle \mathbf{r}, \mathbf{x}_N \rangle = \langle \mathbf{c}, \mathbf{x} \rangle$ is the value of the objective function. It is clear that $\mathbf{x}_B = \mathbf{p}$ and $\mathbf{x}_N = \mathbf{0}$ is a solution to this system, and since $\mathbf{p} \geq 0$, it is clearly a bfs. Moreover, z_B is the value of the objective function at this bfs. Why should such a tableau exist for a feasible basis B? The following lemma shows that a tableau exists and is unique for B.

LEMMA 12. *The simplex tableau corresponding to a feasible basis B is uniquely determined and is the following:*

$$\mathbf{p} = A_B^{-1}\mathbf{b},\ Q = -A_B^{-1}A_N,\ z_B = \langle \mathbf{c}_B, A_B^{-1}\mathbf{b} \rangle,\ \textit{and}\ \mathbf{r} = \mathbf{c}_N - (A_B^{-1}A_N)^t\mathbf{c}_B.$$

Proof. Consider the system $A\mathbf{x} = \mathbf{b}$. The LHS of this equation is equal to $A_B\mathbf{x}_B + A_N\mathbf{x}_N$. Therefore, $A_B\mathbf{x}_B = \mathbf{b} - A_N\mathbf{x}_N$, which implies that $\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N$. Therefore, $\mathbf{p} = A_B^{-1}\mathbf{b}$ and $Q = -A_B^{-1}A_N$; note the dimension of the LHS is $m \times (n-m)$.

The value of the objective function is captured by the variable z, and is $\langle \mathbf{c}, \mathbf{x} \rangle = \langle \mathbf{c}_B, \mathbf{x}_B \rangle + \langle \mathbf{c}_N, \mathbf{x}_N \rangle$. Substituting the value of $\mathbf{x}_B$ from above we obtain that

$$\langle \mathbf{c}, \mathbf{x} \rangle = \langle \mathbf{c}_B, A_B^{-1}\mathbf{b} \rangle - \langle \mathbf{c}_B, A_B^{-1}A_N\mathbf{x}_N \rangle + \langle \mathbf{c}_N, \mathbf{x}_N \rangle.$$

Rewriting the RHS we obtain that $\mathbf{r} = \mathbf{c}_N - (A_B^{-1}A_N)^t\mathbf{c}_B$.

For the uniqueness, suppose we have two system of equations of the form (7). We know that the solution set for the two systems is the same. In particular, substituting $\mathbf{x}_N = \mathbf{0}$, we get that the vector $\mathbf{p}$ and the value z_B must be the same for the two systems. Substituting the standard vectors e_i, $i = 1, \ldots, n-m$ for $\mathbf{x}_N$, we get that the ith columns of the matrix Q and the ith entry of the vector $\mathbf{r}$ are the same in both the systems. Thus the simplex tableau is uniquely defined wrt B. **Q.E.D.**

Remark: : In practice, only the $m \times m$ matrix A_B in its LU decomposition, and the vector $\mathbf{p}$ is maintained, since this reduces the cost from mn to m^2. For instance, only a column of Q will be needed in the pivot step, but this can be obtained by taking $A_B^{-1}A_u$, where A_u is the column corresponding to the incoming variable.

The next claim is easy to see: If in a simplex tableau $T(B)$, the vector $\mathbf{r} \leq \mathbf{0}$ then the corresponding bfs is optimal. If $\mathbf{x}$ is any other feasible solution then it is clear that $\langle \mathbf{r}, \mathbf{x} \rangle \leq 0$ and hence $\langle \mathbf{c}, \mathbf{x} \rangle \leq z_B$, where z_B is the value of the objective function at the bfs.

Given $T(B)$, the simplex method constructs another st $T(B')$ by constructing a feasible basis B' from B, where B' substitutes a basic variable by a nonbasic variable in B. This is done in the **pivot step** where we first choose a nonbasic variable x_v and then choose a basic variable x_u to be dropped. The variable x_v is chosen to increase the value of the objective function. Thus *a nonbasic variable x_v can come in the new tableau $T(B')$ iff its coefficient in $\mathbf{r}$ is positive.* We know that such an x_v exists if $\mathbf{r} < 0$. How to choose the basic variable x_u that leaves B? This is determined by the entering variable x_v. Look at the column of Q corresponding to x_v. We will suppose that the column has a negative entry; otherwise, we will show that the lp is unbounded. Then corresponding to this negative entry we have an equation with a basic variable on the LHS. The variable x_u is chosen such that its non-negativity gives us the strongest upper bound on x_v. To be more precise, suppose $B = \{1, \ldots, m\}$, $N = \{m+1, \ldots, n\}$, and $v = n$. The set of first m equations in $T(B)$ are of the form: for $i = 1, \ldots, m$,

$$x_i = p_i + \sum_{j=1}^{n-m} q_{ij} \cdot x_{m+j}.$$

We plan to make x_n positive, and this has to be done while ensuring that the x_i's remain non-negative. For each i such that $q_{in} < 0$, this gives us a constraint that $x_n \leq -p_i/q_{in}$. The variable x_n can take $\min -p_i/q_{in}$, over all i's such that $q_{in} < 0$. Let k be an index for which the minimum is attained. Then x_n replaces x_k in B to get B'. We now have to show two things:

1. B' is a feasible basis, and
2. if the column corresponding to x_n in Q has no negative entries then the LP is unbounded.

We start with the latter.

Suppose the column Q_{n-m} corresponding to x_n is non-negative and has one non-zero entry. Then let $x_n := t$ and consider the following set: for $i = 1, \ldots, m$

$$x_i = p_i + q_{i,n-m}t$$

and $x_i = 0$ for all the other non-basic variables. Then this line is a feasible solution to the system. Moreover, the objective function is of the form $z_0 + r_{n-m}t$, which increases as t increases. Therefore, the objective function is unbounded on the feasible set.

To show the former, suppose that x_n replaces x_k in the set B to get B'; thus $q_{kn} \neq 0$. We have to show that $A_{B'}$ is also non-singular. Recall that $A_N = A_B Q$, which implies that the last column of A_N (which is the same as the last column A_n of A) can be expressed as a linear combination of the columns of A_B. In particular, since $q_{kn} \neq 0$, the kth column of A_B (which is A_k in A) must occur in this linear combination, i.e.,

$$A_n = q_{kn}A_k + \sum_{i=1; i\neq k}^{m} q_{in}A_i.$$

Thus A_k can be expressed as a linear combination of A_n and all the remaining columns of A_B. That means the columns of A can be expressed as a linear combination of the columns in $A_{B'}$, and hence $A_{B'}$ must be non-singular, which implies that B' must be a feasible basis.

6.1 Infeasibility and finding a starting bfs

The previous sections shows that if we are given a bfs for an lp, which means that we can detect whether the feasible solution set of an lp is empty or not, then we can detect whether the opt exists, and if yes, then find an optimum. In this section, we want to remove the first assumption. The assumption was unnecessary in the examples that we started with, because they were given to us in canonical form, rather than equational, and after introducing slack variables for all inequalities it was clear that the system of equations has a bfs corresponding to setting the non-slack variables to zero and for an appropriate choice of slack variables.

However, the input to the simplex method is an lp in ef. How can we detect whether an lp in ef has a feasible solution and get a starting bfs to invoke the simplex method? The idea is similar to "adding slcack variables" as in the canonical form.

We want to substitute zero for the variables in the equation $A\mathbf{x} = \mathbf{b}$. However, doing that does not give us a solution. How "far away" we are from a solution can be captured by introducing m new variables $\mathbf{y}$, and looking for the solutions to the system

$$A\mathbf{x} + \mathbf{y} = \mathbf{b}. \tag{8}$$

In this case the vector $(\mathbf{0}, \mathbf{b})$ is a solution to the system; it need not be feasible since $\mathbf{b}$ can have negative entries; to overcome this we flip the sign of the equations corresponding to negative entries in $\mathbf{b}$ in the original system, which guarantees that $\mathbf{b} \geq 0$ and we have a basic feasible solution. Thus it is clear that if $A\mathbf{x} = \mathbf{b}$ has a solution then so does $A\mathbf{x} + \mathbf{y} = \mathbf{b}$, where $\mathbf{y} = 0$. What about the converse? It is clear that any solution of (8) of the form $(\mathbf{x}, \mathbf{0})$ is a solution to the original system. But not every solution of (8) has this form. How do we obtain a solution of (8) where the entries corresponding to $\mathbf{y}$ are zero? We formulate it as an LP:

$$\max -(\sum_i y_i) \text{ s.t. } A\mathbf{x} + \mathbf{y} = \mathbf{b}, \text{ and } \mathbf{x},\ \mathbf{y} \geq 0. \tag{9}$$

We know that the LP above is feasible, since we have a starting bfs. But why is the lp bounded? Note that the value of the objective function cannot exceed zero; since there is a bfs, it is bounded from below as well; therefore, the lp is feasible and bounded. We can now invoke the simplex method to obtain an opt solution $(\mathbf{x}^*, \mathbf{y}^*)$. We claim the following: The original LP is feasible iff the LP above has an optimal solution where $\mathbf{y}^* = 0$. If $\mathbf{y}^* = 0$ then we have a bfs for the original LP, and so it is feasible. Conversely, any feasible solution $\mathbf{x}$ of the original LP gives us an optimum solution $(\mathbf{x}, \mathbf{0})$ for the new LP as the value of the objective function is zero. The solution to the new lp returns a feasible basis B'. If this feasible basis has none of the new variables, then we are done. Otherwise, we know that the columns indexed by B' corresponding to the variables $\mathbf{x}$ are lid. It is straightforward linear algebra to extend this set from the columns of A to form a feasible basis B of size m.

6.2 Termination and Cycling

Does the simplex method terminate? The method goes from one tableau to another. Each tableau $T(B)$ is uniquely determined by a feasible basis B. But there are only finitely many feasible basis ($\binom{n}{m}$ to be precise). So the method will not terminate iff a feasible basis, and hence the corresponding tableau, repeats itself in the method. This behavior is called **cycling**. When can this happen? Suppose the incoming nonbasic variable x_v occurs in the equation $x_u = p_u - q_{uv}x_v$, then replacing x_v by $p_u/q_{uv} - x_u$ in the objective function increases the value of the objective function by $r_v p_u/q_{uv}$. Well, it increases iff $p_u \neq 0$. Thus degeneracy can occur iff $p_u = 0$, i.e., a basic variable takes the value zero in the bfs. Such a bfs is called **degenerate bfs**, because it has more than one feasible basis associated with it. This was precisely what we observed in the examples in the starting of this section. But recall that in the examples considered, we were able to escape a degenerate vertex after one pivot step. This shows that even though you may have degenerate bfs's, it's not necessary that you end in a cycle – the method can **stall** at a bfs and come out of it after some pivot steps. In practice, stalling is a more common phenomenon than cycling. Nevertheless, it is desirable to have guarantees on the algorithm that for certain choice of pivot rule, we will never cycle (we may stall, but that will not prevent from termination). How to prevent cycling? This is done by choosing an appropriate incoming variable and an exiting variable, or a pivoting rule. From a theoretical perspective, one can argue that degeneracies are rare since the probability of more that n hyperplanes passing through a point is fairly small. However, this is a specious argument, as observed by Dantzig:

> He (Koopmans) wanted me to try to prove that the algorithm would converge without a non-degeneracy assumption, an assumption which I felt initially was reasonable. After all, what was the probability of four planes in three space meeting in a point (for example)? But then something unexpected happened. It turned out that although the probability of a L.P. being degenerate was zero, every practical problem tested by my branch in the Air Force turned out to be so. Degeneracy couldn't happen but it did. It was the rule not the exception!

6.3 Pivoting Rules

It is clear from the description of the simplex method given above that there are certain choices that have to be made in the pivoting step. Two most important choices are the nonbasic variable that enters in the pivot and the corresponding basic variable that should leave. Of these, the first one is of primary importance. Amongst the nonbasic variables with positive coefficient in $\langle \mathbf{r}, \mathbf{x}_N \rangle$, let's call these variables as the **improving variables**, which one should we choose in the pivot step? Pivoting rules give us a uniform way to do make these choices. Here we mention some of the pivoting rules. In this section, we will assume that $\mathbf{r} \in \mathbb{R}^n$, with zero entries corresponding to the basic variables; therefore, the value of the objective function at a point $\mathbf{x}$ is $\langle \mathbf{c}, \mathbf{x} \rangle = z_B + \langle \mathbf{r}, \mathbf{x} \rangle = z_B + \langle \mathbf{r}_N, \mathbf{x}_N \rangle$. This convention is mostly for convenience later.

1. Largest coefficient: Amongst the improving variables, pick one that has the largest positive coefficient in the objective function in the tableau. The rational is that since we are trying to increase the objective function, we should pick a variable that maximizes the improvement of the objective function for unit increase of the improving variable. This was the rule originally proposed by Dantzig.

2. Largest increase: Every improving variable gives us a new feasible basis B, a corresponding bfs, and a value z_B of the objective function at that bfs. Therefore, it makes sense to choose that improving variable for which this value is maximized. That is, make a locally optimum decision. However, this rule is computationally more expensive. If $\mathbf{x}_n$ and $\mathbf{x}_o$ are the new and old bfs respectively, then this rule picks an n that maximizes $\langle \mathbf{c}, \mathbf{x}_n - \mathbf{x}_o \rangle = \max r_v p_u / |q_{uv}|$, where v is incoming and u is such that $q_{uv} < 0$.

3. Steepest Edge: Pick the new variable such that the unit vector from the old bfs to the new bfs is the closest to the optimizing vector $\mathbf{c}$, i.e., the projection of the unit vector onto $\mathbf{c}$ is the largest, or an edge that attains the maximum increase in the objective function for a unit movement along the edge. More precisely, maximize $\langle \mathbf{c}, \mathbf{x}_n - \mathbf{x}_o \rangle / \|\mathbf{x}_n - \mathbf{x}_o\|$, where $\mathbf{x}_n$ is the new bfs and $\mathbf{x}_o$ is the old bfs. In practice, this rule is the most preferred. However, as described the rule is a bit expensive to compute. There are more practical rules that compute approximations to the steepest edge and are usually preferred in practice; one such rule is given by PMJ Harris called Devex (from *devexus* latin for steepest).

4. Bland's rule: The rules presented above do not prevent cycling. R. Bland proposed a rule where it can be shown that cycling does not occur. He suggested that amongst the improving variable we should choose the one with the smallest index, and similarly for the leaving variable. We will show that this prevents cycling.

5. If randomness is allowed, then it is most natural to pick an improving variable randomly. This strategy works, and gives an algorithm whose expected running time can be theoretically bounded (reference?).

An aspect common to all approaches to avoid cycling is to introduce some additional ordering information. Since while cycling the bfs and the value of the objective function does not change, this ordering usually comes from some combinatorial information, such as the indices of the non-basic variables, or some lexicographic ordering on the columns of Q. We see two such rules that avoid cycling.

6.3.1 Bland's rule to avoid cycling

Bland stated the following pivot rule: Amongst the possible entering variables, choose the one with the smallest index; similarly, for the exiting variables. In this section we show that this prevents cycling. Though in practice it is not preferred, and hence is only of theoretical significance.

Suppose the simplex method runs into a cycle of tableaux. We will derive a contradiction from this supposition. While going from one tableaux to another we replace a basic variable by a non-basic variable. However, since we are in a cycle, any variable that enters has to also exit at some point in the cycle (may be this happens more than once); the entry and exit point may not be in that sequence, e.g., an index which is present in the starting of the cycle exits first and then enters; for our proof, this ordering does not matter. We call all such variables as **fickle variables** and denote their set by F. As already observed earlier, we stay at the same bfs $\mathbf{x}^*$ in a cycle. Our first claim is that all fickle variables are set to zero in $\mathbf{x}^*$. To see this suppose x_v is an fickle variable, and consider a basis when it is incoming; the value of the objective

function is some constant plus $r_v x_v$; if x_v takes a positive value then the objective function increases, which is a contradiction since in a cycle the objective function remains the same. The crucial idea of the proof is to pick an extreme index from F and see what happens to the lp when this variable enters and leaves the lp. From these observations we will derive another LP that has an optimum solution and is also unbounded, which would yield us the desired contradiction.

Two choices for a variable from F are the one with the smallest index and the one with the largest index. For reasons that will become clear later, we choose the latter of the two, and denote it by x_v. Let B be the basis when x_v enters and B' be the basis when it leaves, and is replaced by some nonbasic variable x_u; note that B and B' may not be unique. We now understand the role of x_v when it is incoming and exiting variable.

Observation 1: Consider the tableau for a feasible basis B and let x_n be the incoming variable. Then we know that r_v in $\langle \mathbf{r}, \mathbf{x} \rangle$ must be positive. Moreover, since Bland's rule always picks the nonbasic variable with the smallest index it must be the case that all the other coefficients r_i, $i < v$, must be negative. Specifically, the coefficient in $\mathbf{r}$ corresponding to all the fickle variables except v must be negative, as v is the index of the largest fickle variable.

Observation 2: Now consider the case when x_v is leaving the basis B' and is being replaced by x_u. We know that the coefficient of x_u in $\mathbf{r}$ is positive. Also the coefficient in $\mathbf{p}'$ corresponding to x_v is zero, and the entry $q_{v,u}$ is negative. Again from Bland's rule we know that the coefficient of $q_{v',u}$ for all other basic fickle variables must be positive, otherwise $x_{v'}$ would be exiting instead of x_v.

We use the two observations above to devise an LP that has an optimum solution and is unbounded at the same time, which gives us the desired contradiction. Our new LP should have the property that the bfs $\mathbf{x}^*$ corresponding to B (and hence also to B') must be feasible, and better still optimal for it. Note that the coefficient r_v is positive, so that decreasing x_v decreases the value of the objective function; similarly, for all the other fickle variables the coefficient in $\mathbf{r}$ is negative, so increasing them decrease the value of the objective function. Thus if $\mathbf{x}^*$ has to be the optimal solution in the new LP then it makes sense to assume that $x_{v'} \geq 0$, for $v' \in F \setminus \{v\}$ and $x_v \leq 0$; the rest of the LP almost remains the same. More precisely, consider the following LP: maximize $\langle \mathbf{c}, \mathbf{x} \rangle$ such that

$$A\mathbf{x} = \mathbf{b}, \mathbf{x}_{F \setminus \{v\}} \geq 0, x_v \leq 0, \mathbf{x}_{N \setminus F} = 0.$$

Clearly, $\mathbf{x}^*$ satisfies all the constraints, and hence is feasible; moreover, by the argument above it is also optimal, since for any other feasible solution to this system the value of the objective function can only decrease. Note that there is no constraint on the basic variables that are not fickle in this LP; also note that the set of nonbasic non-fickle variables is the same whether we consider B or B'.

Now consider the case when we are at the tableau corresponding to B'. We will show that the LP above is unbounded; this proof is similar to the unboundedness that we had seen earlier. Let t denote the value of the incoming variable x_u; keep all the other nonbasic variables as zero. Since the coefficient of x_u is positive in $\mathbf{r}$, the value of the objective function is $z_{B'} + tr_u$, which clearly increases as t increases. We know that the solutions of $A\mathbf{x} = \mathbf{b}$ are the same as the solutions of $\mathbf{x}_{B'} = \mathbf{p}' + Q'\mathbf{x}_{N'}$. From observation 2 we know that for all basic fickle variables v' their value is $q_{v',u}t$, which is positive since $q_{v',u} > 0$; for x_v its value is $q_{v,u}t$, which is negative since $q_{v,u} < 0$; the rest of the basic variables vary with t, but we are not concerned about them. It is clear that the line parametrized by t satisfies the constraints in the LP above, while the objective function grows unbounded on this line. But why can we increase x_u? This is because x_u is also fickle variable distinct from x_v and the constraints in the LP above allow us to increase x_u. Thus we have a contradiction to the existence of an optimal solution that we had shown earlier.

6.4 Worst Case Running Time

In 1972, Klee and Minty showed that the simplex method with Dantzig's pivot rule of largest coefficient takes exponential time. Their basic idea was to construct an LP for which the simplex method visits all the vertices of the polytope. The polytope itself, as we will see later, looks like a deformation of the hypercube, and hence has 2^n vertices. This shows that in the worst case the algorithm is not better than the naive algorithm that enumerated all the bfs.

Let us try to construct examples where the simplex method takes exponentially many tableaux. We first do this for smaller values of n, which will help us construct the examples in general. Consider the following

most straightforward LP for the case $n = 1$: maximize x subject to $x \leq 1$. It is not hard to verify that we need exactly two tableaux for this example, as desired. We next try the following sequence of LPs in two variables; our goal is to come with an lp that constructs 4 tableaux.

1. maximize x s.t. $x \leq 1$. This involves two tableaux.
2. maximize $x_1 + x_2$ s.t. $x_1 \leq 1$, $x_1 + x_2 \leq 1$. There is one problem with this lp, namely the choice of the incoming variable is not clear in the first step. However, it is clear that if x_1 enters then we need more tableau; so let's ensure that x_1 is the first incoming variable by increasing its coefficient in the objective function to two to obtain our next lp.
3. maximize $2x_1 + x_2$ s.t. $x_1 \leq 1$, $x_1 + x_2 \leq 1$. The issue with this lp is that it has a degenerate bfs. So to avoid that we increase the rhs of the inequality to two and get the next lp.
4. maximize $2x_1 + x_2$ s.t. $x_1 \leq 1$, $x_1 + x_2 \leq 2$. This lp is almost there. We need three tableaux to reach the opt. In order to get one more tableau, we ought to make the coefficient of y_1 (the slack variable such that $x_1 + y_1 = 1$) positive. If we observe carefully, then this is achieved by increasing the coefficient of x_1 in the second inequality, so let us increase it by 4 to get the next lp.
5. maximize $2x_1 + x_2$ s.t. $x_1 \leq 1$, $4x_1 + x_2 \leq 2$. This lp is unbounded since in the second tableau x_2 can grow unbounded as their is no positivity constraint bounding it. The least such constraint is attained by increasing the rhs of the second inequality to something greater than 4, let's choose 5. The lp so obtained is the next one.
6. maximize $2x_1 + x_2$ s.t. $x_1 \leq 1$, $4x_1 + x_2 \leq 5$. It can be verified now that this lp takes exactly four tableaux, giving us our example in two variables. Note how x_1 enters in the first tableau and leaves in the last one. How about three variables? Consider the following lp. Can we argue inductively that it requires 8 tableaux?
7. maximize $4x_1 + 2x_2 + x_3$ s.t. $x_1 \leq 1$, $4x_1 + x_2 \leq 5$, $8x_1 + 4x_2 + x_3 \leq 25$.

Consider the following LP $L_n(x_1, \ldots, x_n)$:

$$\text{maximize } \sum_{i=1}^{n} 2^{n-i} x_i$$

subject to

$$\begin{aligned}
x_1 &\leq 1 \\
4x_1 + x_2 &\leq 5 \\
8x_1 + 4x_2 + x_3 &\leq 25 \\
&\vdots \\
&\vdots \\
2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + x_{n-1} &\leq 5^{n-2} \\
2^n x_1 + 2^{n-1}x_2 + \cdots + 4x_{n-1} + x_n &\leq 5^{n-1} \\
x_1, \ldots, x_n &\geq 0.
\end{aligned}$$

We first convert the LP to the equational form by introducing n slack variables

$$\begin{aligned}
x_1 + y_1 &= 1 \\
4x_1 + x_2 + y_2 &= 5 \\
8x_1 + 4x_2 + x_3 + y_3 &= 25 \\
&\vdots \\
&\vdots \\
2^n x_1 + 2^{n-1}x_2 + \cdots + 4x_{n-1} + x_n + y_n &= 5^{n-1} \\
x_1, \ldots, x_n, y_1, \ldots, y_n &\geq 0.
\end{aligned}$$

Let this LP be denoted by L_n. We will show that the simplex method constructs 2^n tableaux to solve this LP using the largest coefficient rule. The proof is by induction, but first we prove certain properties of this LP. Why is this LP feasible, and even if it is why should there be an optimal solution?

Prop1. There exists a *unique optimal solution* to the LP, namely $x_i = 0$ and $y_i = 5^{i-1}$ for all i except $x_n = 5^{n-1}$ and $y_n = 0$. Note that the objective function has coefficients smaller than the coefficients in the last constraint, so that the maximum value it can take is 5^{n-1}. But we can make this relation more precise, namely the last contraint can be expressed as $2\langle \mathbf{c}, \mathbf{x}\rangle - x_n \leq 5^{n-1}$. Thus the value of the objective function is upper bounded by $(5^{n-1} + x_n)/2$. Thus the objective function will be maximized if we choose x_n as large as possible. The maximum value that x_n can take is 5^{n-1}, therefore the objective function is upper bounded by 5^{n-1} (as argued earlier), but more importantly attains it for the values of x_i's and y_i's mentioned above. Note that for any other feasible solution if any other x_i is non-zero then from the last inequality we know that $x_n < 5^n$ and hence this feasible solution cannot be an optimum solution.

Prop2. Exactly one of x_i, y_i is a basic variable. We have n basic and n nonbasic variables in any given tableau. Note that at least one of x_i, y_i is non-zero in any feasible solution of the equational form. This is because the ith constraint is twice the $(i-1)$th constraint plus $x_i + y_i$; now the largest value that the $(i-1)$th constraint can take is 5^{i-1}, and twice of that can never be equal to 5^i unless one of x_i, y_i is non-zero. This means that in any bfs at least one of them is there, and since we can have only n basic variables in a bfs, therefore, both x_i, y_i cannot simultaneously included.

We now claim that the simplex method with the largest coefficient rule computes 2^n tableaux for the LP above (we are of course considering the equational form). The proof is via induction. The starting tableau is evident: there are i equations for $i = 1, \ldots, n$

$$y_i = 5^{i-1} - 2^i x_1 - 2^{i-1} x_2 - \cdots - 4x_{i-1} - x_i,$$

and

$$z = \left(2^{n-1} x_1 + 2^{n-2} x_2 + \cdots + 2x_{n-1} + x_n\right);$$

the corresponding bfs has $x_i = 0$ and $y_i = 5^{i-1}$; the value of the objective function is 0. What about the final tableau? We know the set of feasible variables for the optimum solution, namely $x_n, y_1, \ldots, y_{n-1}$, and the bfs. What about the matrix Q and the vector $\mathbf{r}$? It is easy to construct Q in the final tableau by expressing $y_1, \ldots, y_{n-1}, x_n$ in terms of the non-basic variables. Thus the final tableau has the following form:

$$\begin{aligned}
y_1 &= 1 - x_1 \\
y_2 &= 5 - 4x_1 - x_2 \\
y_3 &= 25 - 8x_1 - 4x_2 - x_3 \\
&\vdots \\
&\vdots \\
x_n &= 5^{n-1} - 2^n x_1 - 2^{n-1} x_2 - \cdots - 4x_{n-1} - y_n \\
z &= 5^{n-1} - 2^{n-1} x_1 - 2^{n-2} x_2 - \cdots - 2x_{n-1} - y_n.
\end{aligned}$$

Consider the initial tableau. Observe that the objective function is of the form $2(2^{n-2}x_1 + \cdots + x_{n-1}) + x_n$, i.e., it is twice the objective function for $(n-1)$ variables plus x_n. Therefore, to maximize this objective we can try to maximize the objective function for the $(n-1)$ variables, along with the first $n-1$ constraints, which is exactly the lp L_{n-1}; we do this inductively, and further observe that the coefficients, if positive, of $x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1}$ in the objective function will alway be twice that of x_n, i.e., x_n will not be an incoming variable (and hence y_n will not be outgoing from Prop2 above) unless the coefficients of the remaining variables in the objective function is negative. This happens precisely when we reach the opt of

$L_{n-1}(x_1, \ldots, x_{n-1})$. Using induction we know that the tableau after 2^{n-1} pivot steps has the following form:

$$\begin{aligned}
y_1 &= 1 - x_1 \\
y_2 &= 5 - 4x_1 - x_2 \\
y_3 &= 25 - 8x_1 - 4x_2 - x_3 \\
&\vdots \\
&\vdots \\
x_{n-1} &= 5^{n-2} - 2^{n-1}x_1 - 2^{n-2}x_2 - \cdots - 4x_{n-2} - y_{n-1} \\
y_n &= 5^{n-1} - 2^n x_1 - 2^{n-1}x_2 - \cdots - 4x_{n-1} - x_n \\
&= 5^{n-2} + 2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + 4y_{n-1} - x_n \\
z &= 2(5^{n-2} - 2^{n-2}x_1 - 2^{n-3}x_2 - \cdots - y_{n-1}) + x_n.
\end{aligned}$$

At this instance, x_n is the incoming variable and y_n is the outgoing variable. The resulting tableau is

$$\begin{aligned}
y_1 &= 1 - x_1 \\
y_2 &= 5 - 4x_1 - x_2 \\
y_3 &= 25 - 8x_1 - 4x_2 - x_3 \\
&\vdots \\
&\vdots \\
x_{n-1} &= 5^{n-2} - 2^{n-1}x_1 - 2^{n-2}x_2 - \cdots - 4x_{n-2} - y_{n-1} \\
x_n &= 5^{n-2} + 2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + 4y_{n-1} - y_n \\
z &= 3 \cdot 5^{n-2} + 2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + 2y_{n-1}) - y_n.
\end{aligned}$$

Again, the subsequent steps will solve $L_{n-1}(x_1, \ldots, x_{n-2}, y_{n-1})$ inductively (note the change from x_{n-1} earlier to y_{n-1}) to obtain the final tableau given above; note that the coefficient of y_n remains negative and it never enters the feasible basis, which means by Prop2 that x_n never leaves the feasible basis. It is instructive to try these steps on $L_3(x_1, x_2, x_3)$ using the solution for L_2 that we derived earlier.

Similar lower bounds have been derived for other pivot rules as well. Is it possible to argue that no matter what choice of pivot rule, simplex method will always take exponential time. One such approach could be based on the "clairvoyant pivot rule", namely, an oracle that always gives us the shortest path between two vertices $\mathbf{v}, \mathbf{w}$ of a polyhedron P. Let $\rho_P(\mathbf{v}, \mathbf{w})$ be one less than the number of vertices on the shortest path from $\mathbf{v}$, $\mathbf{w}$ while moving from neighbor to neighbor. The diameter of P, Δ_P is defined as the maximum such distance over all pairs of vertices. It is clear then that the clairvoyant rule will take Δ_P steps for some pair of vertices. Further define, $\Delta(m, n)$ as the maximum over all diameters over all polyhedra defined by $m \geq n$ facets in $\mathbb{R}^n$. If we can show an exponential lower bound on $\Delta(m, n)$ then we will have a correspondng lower bound on the simplex method. However, this is far from the bounds we have. How about tight upper bounds? The following section gives the best known upper bound on $\Delta(m, n)$.

6.5 Hirsch's conjecture and the Kalai-Kleitman bound

In practice, Dantzig observed that the number of iterations of the simplex method hardly exceeded $1.5m$. To justify this behavior, Hirsch conjectured that $\Delta(m, n) \leq m - n$. The first counterexample to the conjecture was given by Klee and Walkup who showed that there is a polyhedron in four dimensions defined by 8 halfspaces such that its diameter is at least five, i.e., $\Delta(8, 4) > 4$. However, theor polyhedron was unbounded. The conjecture remained open for polytopes. This was recently proved to be false by Francisco Santos in 2010, who showed that $\Delta(82, 41) > 41$; the counterexample has been simplified further. One can, nevertheless, ask whether the diameter is bounded by a polynomial in m and n.

In 1992, Kalai gave a subexponential upper bound on the diameter. The proof was simplified and presented in a paper jointly with Kleitman. Their main claim is the following:

LEMMA 13. $\Delta(m,n) \leq \Delta(m-1,n-1) + 2\Delta(\lfloor m/2 \rfloor, n) + 2$.

Proof. Let P be a polyhedron and $\mathbf{v}, \mathbf{w}$ be two vertices such that $\rho_P(\mathbf{v},\mathbf{w}) = \Delta(m,n)$, i.e., the pair of vertices and the polyhedron witness the worst case diameter for (m,n)-polyhedra. Let F be a subset of $\lfloor m/2 \rfloor$ facets from the m facets defining P. Define $k_{\mathbf{v},F}$ as the furthest point from $\mathbf{v}$ in F, i.e., the furthest one can go while taking the facets in F. Further define $k_{\mathbf{v}}$ as the maximum of $k_{\mathbf{v},F}$ over all subsets F of size $\lfloor m/2 \rfloor$. Let $F_{\mathbf{v}}$ be a subset of facets that attain the distance $k_{\mathbf{v}}$. Now consider the set $G_{\mathbf{v}}$ of facets that we can reach from $\mathbf{v}$ in $k_{\mathbf{v}}+1$ steps; from the definition of $F_{\mathbf{v}}$, it follows that $F_{\mathbf{v}} \subset G_{\mathbf{v}}$, that is $|G_{\mathbf{v}}| \geq \lfloor m/2 \rfloor + 1$. Similarly define $k_{\mathbf{w}}$, $F_{\mathbf{w}}$ and $G_{\mathbf{w}}$. Since $|G_{\mathbf{v}}|, |G_{\mathbf{v}}| \geq \lfloor m/2 \rfloor + 1$, it follows that the two sets share a common facet f. Let $\mathbf{t} \in f$ be the nearest point in $G_{\mathbf{v}}$ from $\mathbf{v}$ and $\mathbf{u} \in f$ be the nearest from $\mathbf{w}$ in $G_{\mathbf{w}}$. Then $\rho_P(\mathbf{v},\mathbf{w})$ is bounded by the distance of $\mathbf{v}$ to $\mathbf{t}$ plus the distance of $\mathbf{w}$ to $\mathbf{u}$ plus the distance of $\mathbf{t}$ to $\mathbf{u}$. We claim that the first two distances are at most $\Delta(\lfloor m/2 \rfloor, n) + 1$ and the last is $\Delta(m-1, n-1)$, which will complete the proof.

The last claim is easy to see: since $\mathbf{t}$, $\mathbf{u}$ are on the same facet, P restricted to the facet is affinely isomorphic to a polyhedron in $(n-1)$ dimensions defined by at most $(m-1)$ facets. For the first claim, we show that $k_{\mathbf{v}} \leq \Delta(\lfloor m/2 \rfloor, n)$. observe that the shortest distance from $\mathbf{v}$ to $\mathbf{t}$ in P is

Q.E.D.

Todd, 2014, used a careful inductive argument to show that $\Delta(m,n) \leq (m-n)^{\log n}$.

7 Randomized Algorithms

As we have seen above, the running time of simplex algorithm is exponential in the worst case. For a long time it was an open problem whether LP problems can be solved in polynomial time. This was settled by Khachiyan in 1980 (using the ellipsoid method), and later on by Karmarkar in 1984 (using interior points method). The running time of both methods depends on the size of the coefficients of the constraint matrix and the objective function. Such algorithms are not considered strongly polynomial time algorithms, i.e., algorithms whose running time in the Real RAM model is polynomially bounded. E.g., Euclid's algorithm for computing gcds of two integer polynomials is a strongly polynomial time algorithm. Or, in other words, the algebraic complexity(as compared to the bit-complexity) of the algorithm does not depend on the size of the numbers given as input; the algorithm can assume that each operation takes unit cost. One reason why the complexity of the ellipsoid method and the interior point methods depend on the bit-length of the input integers is because they exploit the geometry of the polyhedra underlying the LP. This is unlike the simplex algorithm which only works with the combinatorial structure of the polyhedra, namely by moving from one vertex to a neighboring vertex. Therefore, simplex algorithm is really a combinatorial algorithm and its worst case complexity is upper bounded by $\binom{m}{n} = O(m^n)$, where m is the number of constraints and n is the number of variables. In this section, we will focus only on combinatorial algorithms.

The first significant improvement on the running time of the simplex algorithm was by Meggido, who gave an $O(2^{2^n}m)$ algorithm, i.e., assuming the dimension is a constant, a linear algorithm in the number of constraints. Dyer and Clarkson independently improved it to $O(3^{n^2}m)$. The best deterministic bound is by Chazelle and Matousek, who give an $(n^{O(n)}m)$ running time algorithm. In the case of randomized algorithms, better bounds are known. Already, Dyer and Frieze gave a $O(n^{3n}m)$ runnnig time algorithm, which was improved to $O(n^{n/2}\log n + n^2m + n^4\sqrt{m}\log m)$ by Clarkson. This was improved to $O(n!m)$ by Seidel, who gave a very elegant algorithm, and to $O(2^n n^3 m)$ by Sharir and Welzl. The first subexponential bound is by Kalai and Matousek-Sharir-Welzl. The current best bound of $O(n^2m + e^{O(\sqrt{n\log n})})$ is a combination of the subexponential algorithm with those of Clarkson. Such randomized algorithms are useful otherwise also, as Chazelle and Matousek's algorithm is actually a derandomization of such algorithms. In this section, we study some of these randomized algorithms. There is another type of randomized algorithms that are variants of the simplex method with a random pivot rule. Two such rules are either to pick a random facet or a random edge over all the possible incoming variable. Subexponential upper bounds have been derived for both these variants (this is the algorithm of Matousek-Sharir-Welzl); however, at the same time subexponential lower bounds are also known (this is the work of Friedmann-Hansen-Zwick using lower bounds from game theory). We start by considering simple randomized algorithms. Whether there is a strongly polynomial time algorithm, either randomized or deterministic, is still open.

Consider the canonical form of lp (5). Suppose the underlying polyhedra is actually bounded, i.e., it is a polytope. Clearly, in this case the optimum solution is well-defined and is actually attained at some vertex of the polytope; further suppose that this vertex is unique. Now from hw1 we know that the vertices of this polytope correspond to some n constraints satisfied as equations. The key idea behind all the algorithms is to get hold of a small enough subset of the m constraints that contain the n constraints defining the optimum solution. Once we have this small subset of constraints then we can go ahead and use either simplex method or brute force enumeration to find the optimum solution. Thus the key is to prune the set of constraints and figure out the constraints that are critical to the lp.

7.1 Notation

We first consider a slightly modified version of the canonical lp:

$$\text{minimize } \langle \mathbf{c}, \mathbf{x}\rangle \text{ subject to } A\mathbf{x} \leq \mathbf{b}.$$

We will further make certain assumptions that will help us reduce the technicalities, starting with the assumption that the lp is feasible (this can be checked as was done earlier). We will use the convenient notation $w(\mathbf{x}) := \langle \mathbf{c}, \mathbf{x}\rangle$.

Let H be the set of m halfspaces denoted by $A\mathbf{x} \leq \mathbf{b}$. For $G \subseteq H$, consider the set of points attaining the least value of the objective function over all the vertices of the polyhedron P_G defined by G; amongst all such points, define $\mathbf{v}_G$ as the *lexicographically smallest* point; this ensures the uniqueness of $\mathbf{v}_G$, which may

now be thought of as the optimum wrt G, or the value of G. But we have to be slightly careful in defining $\mathbf{v}_G$, since there may be no lexicographically smallest point attaining the minimum, in which case $\mathbf{v}_G$ is defined as $-\infty$; note this can be caused by three factors: either the objective function is unbounded but the lexically smallest point is well defined (e.g., minimize y, s.t. $x \geq |y|$), or because there is no lexicographically minimum point in the set of points attaining the minimum (e.g., minimize y such that $y \geq 1$), or both the objective function and the lexically smallest point are unbounded. Clearly, according to our assumptions on the canonical form, the point $\mathbf{v}_H$ is the unique optimum for the lp. For two sets F, G, we say $\mathbf{v}_F \leq \mathbf{v}_G$ if either $w(\mathbf{v}_F) < w(\mathbf{v}_G)$, or if the objective function takes the same value then $\mathbf{v}_F \underset{\text{LEX}}{\leq} \mathbf{v}_G$. Thus, '$\leq$' is a total ordering on the opts.

A **basis of** a set $G \subseteq H$ is a *minimal* subset F of G such that $\mathbf{v}_F = \mathbf{v}_G$; in words, the optimum wrt F is the same as the optimum wrt G; in Figure 1 for $G = \{1, 2, 3, 6\}$ the sets $\{1, 2\}$, $\{2, 6\}$ are a basis, whereas $\{1, 6\}$ is not. We say that a constraint (or halfspace) $h \in H$ is **violated by** G iff $\mathbf{v}_G$ does not satisfy h. Note that if $\mathbf{v}_G$ satisfies h then $\mathbf{v}_G = \mathbf{v}_{G \cup h}$. Finally, a **constraint** h **is extreme in** G if h is violated by $G - h$, i.e., the optimum wrt $G - h$ is different than the optimum wrt G (or dropping of h changes the optimum); e.g., if $G = \{1, 2, 3, 6\}$ in Figure 1 then the constraint 2 is extreme, whereas 1 and 6 are not. Note that removing an extreme constraint does not necessarily mean that the value of the objective function changes; e.g., if $G := \{1, 2, h\}$, then h is extreme, however, $w(\mathbf{v}_G) = w(\mathbf{v}_{G \setminus h})$. Also it is possible that G has no extreme constraints: for instance, if $G = \{1, 2, 6, 7\}$, then removing a single constraint does not change $\mathbf{v}_G$. See Figure 1 for an illustration of these definitions.

We now study some properties of these definitions.

LEMMA 14. *Let $F, G \subseteq H$ be such that $\mathbf{v}_F$ and $\mathbf{v}_G$ are finite.*

(i) If $F \subseteq G$, then $w(\mathbf{v}_F) \leq w(\mathbf{v}_G)$ and $\mathbf{v}_F \underset{\text{LEX}}{\leq} \mathbf{v}_G$.

(ii) If $\mathbf{v}_F = \mathbf{v}_G$ then h is violated by F iff h is violated by G.

(iii) A basis of G has exactly n constraints.

(iv) h is violated by G iff h is extreme in $G \cup h$.

(v) Extreme constraints in G belong to all basis of G, or in other words, constraints in $G \setminus F$, for a basis F of G, are not extreme.

(vi) G has at most n extreme constraints.

(vii) Let F be such that $\mathbf{v}_F$ violates a constraint in G. Then for all basis B of G, $\mathbf{v}_F$ violates a constraint in B.

Proof.

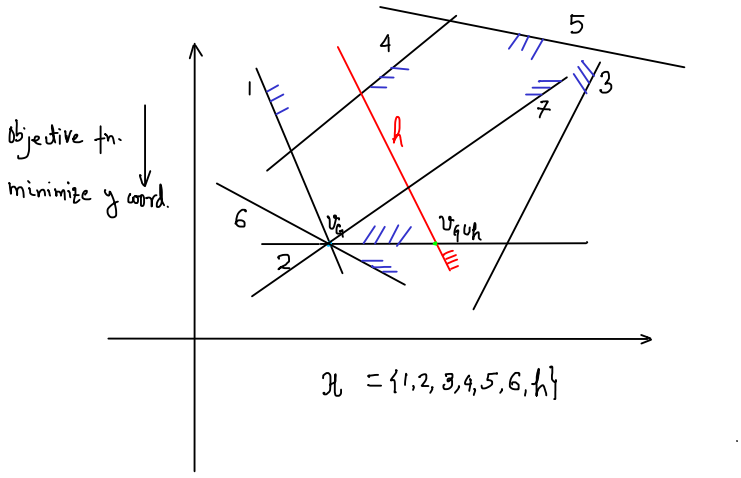
(i) In going from F to G we are adding new constraints, therefore, the set $P_G \subseteq P_F$. Clearly, the minimum (either lexicographically speaking or of the objective function) over P_F cannot exceed the minimum over P_G.

(ii) Since the two opts are the same, if the opt violates h then it is violated for both the sets F, G.

(iii) This is because a vertex of the polyhedron P_G is uniquely determined by some n constraints holding as equalities. Therefore, by picking exactly these set F of constraints we ensure that attaining the optimum on P_F is the same as attaining the optimum on P_G. Note that there can be other constraints that capture the same vertex of P_G (e.g., 1, 6 for $G = \{1, 2, 6\}$ in Figure 1), however, they do not define the same polyhedron as P_G on that vertex.

(iv) This follows from the definition of extreme constraint.

(v) Suppose h does not belong to a basis F. We will show that h is not extreme in G, i.e., $\mathbf{v}_{G \setminus h} = \mathbf{v}_G$; this suffices because we know that $\mathbf{v}_G$ satisfies h, so h is not violated by G, and hence h is not an extreme constraint. Since $F \subseteq G \setminus h \subset G$, from (i) we know $\mathbf{v}_F \underset{\text{LEX}}{\leq} \mathbf{v}_{G \setminus h} \underset{\text{LEX}}{\leq} \mathbf{v}_G$. But as F is a basis, $\mathbf{v}_F = \mathbf{v}_G$, and hence $\mathbf{v}_{G \setminus h} = \mathbf{v}_G$.



Let $G = \{1,2,3,6\}$

- $F = \{1,2\}$ or $\{1,6\}$ is a basis of G since in both cases $v_F = v_G$. and both are minimal subsets.
- G violates h.
- the constraints labelled 1 and 2 are extreme for G.

If $G = \{1,2,6,7\}$ then no constraint in G is extreme.

Figure 1: Illustrating various definitions

(vi) Since all extreme constraints appear in a basis, and a basis has exactly n constraints, it follows that there are not more than n extreme constraints in G.

(vii) Proof is by contradiction. Suppose $\mathbf{v}_F$ does not violate any constraint in B, i.e., satisfies all the constraints in B, then $\mathbf{v}_F = \mathbf{v}_{B\cup F}$. But recall that $\mathbf{v}_B$ satisfies all the constraints in G, and so also the constraints in $B \cup F$. Hence $\mathbf{v}_{B\cup F} = \mathbf{v}_B = \mathbf{v}_G$, which means $\mathbf{v}_F$ does not violate any constraint in G, giving us a contradiction.

Q.E.D.

7.2 Two Algorithms of Clarkson

The aim of the algorithm is to find a basis of H. The idea is to construct a set G containing a basis of H iteratively; initially, $G := \emptyset$. To construct G, we will pick a set R of size r from H uniformly at random, and if the set of constraints V in H violated by R is small then we reassign $G := G \cup V$; we again pick a random set R from H (i.e., pick with replacement), look at the constraints V violated by $G \cup R$, and again if V is small we add it to G; we keep on doing this until G does not violate any constraint in H, which means that

G contains a basis of H, and hence $\mathbf{v}_G = \mathbf{v}_H$. Hopefully, the size of G is much smaller than m at any given step.

```
Clarkson 1
INPUT: The set H of constraints and the objective function c.
1.   If m is small (roughly n^2) then
1.a      Do a brute force enumeration to find a basis G of H.
         Return v_G.
     Let r be small compared to m and G ← ∅.
2.   repeat
2.a      choose R uniformly at random from (H choose r).
2.b      Compute the optimum v_{G∪R}.
2.c      Let V ⊆ H be the set of constraints violated by v_{G∪R}.
2.d      If |V| is small compared to m, then G ← G ∪ V.
     until V = ∅.
3.   Return v and G ∪ R.
```

To give precise values for r, and the bound on $|V|$, we need to bound the expected size of V at any given iteration of the repeat-until loop.

LEMMA 15. *Let H be a multiset of m constraints, G be a subset of constraints of H, and $1 \leq r \leq m$. Given a subset R of H of size r, picked uniformly at random, the expected number of constraints in H violated by $G \cup R$ is bounded by $n(m-r)/(r+1)$.*

Proof. The expectation is given as

$$E(|V|) = \sum_{R \in \binom{H}{r}} \frac{1}{\binom{m}{r}} \sum_{h \in H} [h \text{ is violated by } G \cup R],$$

where the notation $[\cdot]$ returns one if h is violated by $G \cup R$, and zero otherwise; note that for $h \in G \cup R$ this function is always zero. Also observe that h is violated by $G \cup R$ iff h is extreme in $G \cup R \cup h$. Therefore,

$$E(|V|) = \binom{m}{r}^{-1} \sum_{R \in \binom{H}{r}} \sum_{h \in H} [h \text{ is extreme in } G \cup R \cup h],$$

If we define $Q := R \cup h$, then we can rewrite the equation above as

$$E(|V|) = \binom{m}{r}^{-1} \sum_{Q \in \binom{H}{r+1}} \sum_{h \in H} [h \text{ is extreme in } G \cup Q].$$

But we know that any set of constraints contains at most n extreme constraints; note that Q can be a multiset, and that means only fewer extreme constraints. Thus

$$E(|V|) \leq \binom{m}{r}^{-1} \sum_{Q \in \binom{H}{r+1}} n = n \binom{m}{r}^{-1} \binom{m}{r+1} = \frac{n(m-r)}{r+1}.$$

Q.E.D.

What value of r should we choose to minimize the expectation? Clearly $r = m$ will minimize it, but then we haven't made any progress since if we choose all the constraints then in step 2.b we are solving the original lp. So our aim is to minimize r and also the expectation. So it makes sense to minimize the sum

$$\frac{n(m-r)}{r+1} + r.$$

Differentiating it wrt r and equating to zero we get that r must be roughly $\sqrt{mn}$ and then the expectation is also roughly $\sqrt{mn}$. This is still slightly large for us. Our aim is to have around $\sqrt{m}$ expected violations. To obtain this we increase r to $n\sqrt{m}$. Note that for this choice of r the expected value is positive if $m > n\sqrt{m}$, i.e., $m > n^2$. If m is smaller than this value then we do the brute force search given in step 1.a.

How many repeat-until loops do we need to ensure that V is small compared to m? From Markov's inequality, we know that $\Pr(|V| > a) < E(|V|)/a$; for $a = 2\sqrt{m}$, we get that the probability the number of violated constraints exceeds $2\sqrt{m}$ is at most half. Thus the expected number of repeat-until loops to ensure that $V \leq 2\sqrt{m}$ is at most 2.

How many repeat-until loops do we need to ensure that V is empty? Since at each step $|V|$ is on the average $\sqrt{m}$, it would appear that we would need at most $\sqrt{m}$ many iterations. However, from Lemma 15(vii), we know that V must contain a constraint from all the bases of H. Since a basis has exactly n constraints, the number of times we increase G is at most n. Thus the expected number of iterations of the repeat-until loop to ensure that $|V| < 2\sqrt{m}$ and that it will become empty is bounded by $2n$.

The arithmetic cost of step 2.c to find the number of constraints violated by $\mathbf{v}$ is $O(mn)$. Therefore, the expected number of arithmetic operations is $O(n^2m)$, and in Step 2.b the optimum is computed for an lp with at most $O(n\sqrt{m})$ constraints. Of course, all of this is if $m > n^2$. To summarize we have the following concrete algorithm:

```
Clarkson 1
INPUT: The set H of constraints and the objective function c.
1.    If m < n^2 then
1.a       Do a brute force enumeration to find a basis G of H.
          Return v_G.
      Let r := n√m and G ← ∅.
2.    repeat
2.a       choose R uniformly at random from (H choose r).
2.b       Compute the optimum v for G ∪ R.
2.c       Let V ⊆ H be the set of constraints violated by v.
2.d       If |V| ≤ 2√m, then G ← G ∪ V.
      until V = ∅.
3.    Return v and G ∪ R.
```

The aforementioned arguments are summarized in the following result:

LEMMA 16. *If $m > n^2$ then the procedure above computes the optimum with an expected number of $O(n^2m)$ arithmetic operations and $O(n)$ calls to a subroutine to solve lps with at most $O(n\sqrt{m})$ constraints.*

If we do a brute-force search or use simplex algorithm for the smaller lps then the running time of the algorithm is $O(n^{n+1}m^{n/2})$, which is slightly better than $O(m^n)$. The crucial step here is to use a different algorithm to solve these smaller lps.

The second algorithm of Clarkson is based on the following observation: we have seen that if the iteration does not terminate then every basis must contain a violated constraint; if we increase the probability of picking these constraints in subsequent iterations then we will hopefully pick a basis in our sampling. However, this will come at the cost of increasing the number of iterations. Thus in the new algorithm after picking R and identifying the violated constraints V, instead of enforcing these constraints the algorithm will increase the probability that they are picked in the next iteration. This is effected by assigning weights to every constraint and doubling the weight of all the violated constraints at every iteration. Thus over time the constraints in the bases have such high weights that they are picked with very high probability. The weights are implemented as multiplicity of a constraint, i.e., with every h we associate its multiplicity μ_h and treat H as a multiset. For any set $V \subseteq H$ define $\mu(V) := \sum_{h \in V} \mu_h$; for H, $\mu(H) = |H|$. The algorithm is as follows:

```
Clarkson 2
INPUT: The set H of constraints and the objective function c.
OUTPUT: The optimum solution v_H.
1.   If m < n^2 then
1.a      Do a brute force enumeration to find a basis G of H.
         Return v_G.
     Let r := 4n^2.
2.   repeat
2.a      choose R uniformly at random from (H choose r).   ◁ R is a multiset .
2.b      Compute the optimum v_R for R.
2.c      Let V ⊆ H be the set of constraints violated by v.   ◁ Note V is not a multiset
2.d      If μ(V) ≤ μ(H)/(2n) then
             For every h ∈ V, let μ_h ← 2μ_h.
     until V = ∅.
3.   Return v_R and R.
```

From Lemma 15, it follows that the expected value of $\mu(V)$ is $\mu(H)/n$, where $r := 4n^2$. Therefore, it again follows from Markov's inequality that the expected number of trials to ensure that $\mu(V) < \mu(H)/2n$ is bounded by 2. We now want to bound the expected number of iterations to ensure $V = \emptyset$. The claim is that for all bases B, $\mu(B)$ increases exponentially, but it cannot go beyond $\mu(H)$, which is roughly m, so that we would need around $\log m$ iterations. A successful iteration is one where the condition in step 2.d holds. The following lemma makes it more precise:

LEMMA 17. *Let B be a basis of H and k be some positive number. After kn successful iterations we have $2^k \leq \mu(B) \leq me^{k/2}$.*

Proof. Let H_i be the set of total constraints and V_i be the set of violations at the ith iteration,. After the ith successful iteration $\mu(H_{i+1}) - \mu(H_i)$ is at least $\mu(V_i) \leq \mu(H_i)/(2n)$, i.e., $\mu(H_{i+1}) \leq \mu(H_i)(1 + 1/(2n))$. Thus after kn iterations the size is bounded by $m(1 + 1/(2n))^{kn} \leq me^{k/2}$, where $m := \mu(H_0)$.

For the lower bound, we again use the observation that at every iteration all bases must contain a violating constraint. Since a basis contains exactly n constraints, by pigeonhole principle we know that after kn successful iterations there must be a constraint in every basis whose weight has been doubled at least k times, which implies that $\mu(B) \geq 2^k$.

Q.E.D.

Note that the lower bound is increasing at a faster rate than the upper bound, and for k sufficiently large will overtake it. In particular, after $O(n \log m)$ iterations this is bound to happen, i.e., we would have found a basis for H. Again, step 2.c takes $O(nm)$ arithmetic operations. Then we have the following:

LEMMA 18. *If $m > n^2$ then the procedure above computes the optimum with an expected number of $O(n^2 m \log n)$ arithmetic operations and $O(n \log m)$ calls to a subroutine to solve lps with at most $O(n^2)$ constraints.*

So we have reduced the number of constraints at the expense of making more iterations. Solving the lps with n^2 constraints using brute force or simplex takes $n^{O(n)}$ pivot operations. Therefore, the expected running time of the algorithm has been reduced to $n^{O(n)}mn^{O(1)}$ arithmetic operations.

7.3 Matousek-Sharir-Welzl Algorithm

Sharir-Welzl developed an algorithm based on Seidel's algorithm. Their idea was to not throw the information of $\mathbf{v}$ if it violates h and start from scratch as Seidel did. In fact, the constraints defining $\mathbf{v}$ are likely to contribute to the optimal vertex. Therefore, they introduce a new parameter, besides the set of constraints H, a candidate basis C of constraints is also maintained. This set develops into a basis of H, just as in the simplex method an initial feasible basis turns into a feasible basis for the optimum; in fact, the initial candidate basis is a feasible basis in the dual. Their original analysis didn't yield a subexponential time algorithm. However, along with Matousek they gave an analysis that yielded a subexponential bound on the running time of the same algorithm.

In this section, we will assume that the canonical form of the lp is the following: minimize $\langle \mathbf{c}, \mathbf{x} \rangle$, where $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$, i.e., we add the non-positivity constraints H_+ along with the m constraints H given by rows of A. This set of constraints will be our initial candidate basis. We also introduce the notion of a **basis** in a set of constraints G: A subset $C \subseteq G$ of constraints such that $-\infty < \mathbf{v}_C$ and for all $C' \subset C$, $\mathbf{v}_{C'} < \mathbf{v}_C$. Thus, B is a *basis of* G iff it is a basis in G and $\mathbf{v}_B = \mathbf{v}_G$.

There may be many basis in G but not all are a basis of G; e.g., H_+ is a basis in $H \cup H_+$ though it may not be a basis of that set. The procedure `Subexp` takes as input a set of constraints G, a candidate basis C in G and outputs a basis of G. We assume that $\mathbf{v}_C > -\infty$, that is, the lp is bounded. To solve the original lp call `Subexp`$(H \cup H_+, H_+)$. We now give the details of the procedure:

```
Subexp(G, C)
INPUT: A set of m constraints G and a basis C ⊆ G.
OUTPUT: The optimum v_G and a basis of G.
1.  If G = C then return C
    else
        Choose an h uniformly at random from G \ C.
        Let B' ← Subexp(G − h, C).
        If v_{B'} does not satisfy h then
            C' ← basis(B' ∪ h)).
            Return Subexp(G, C').
        else
            Return B'.
```

Here `basis`$(B' \cup h)$ computes a basis of the set $B' \cup h$. Since the set has $(n+1)$ constraints, this can be done by brute force by considering all possible subsets of size n and finding the opt for each and choosing the smallest amongst them; there are exactly n such sets, and finding the opt of each is solving a system of n equations in n unknowns; so this takes $O(n^4)$ time.

We can inductively prove the correctness of the algorithm: if $\mathbf{v}_{B'}$ satisfies h, i.e., h is not violated by B', then from Lemma 14(ii) we know that h is not violated by $G - h$, and hence $\mathbf{v}_{G-h} = \mathbf{v}_G$; as B' is a basis of $G - h$ and h is not extreme, we obtain that B' is a basis of G. If, however, $\mathbf{v}_{B'}$ does not satisfy h then the correctness follows by applying induction to $\mathbf{v}_{B'}$: we claim $\mathbf{v}_{B'} < \mathbf{v}_{B' \cup h}$; since $B \subseteq G - h$, we know $\mathbf{v}_B \leq \mathbf{v}_{G-h} = \mathbf{v}_{B'} < \mathbf{v}_{B' \cup h}$; the last is a strict inequality because $\mathbf{v}_{B'}$ did not satsify h, i.e., h is extreme in $B' \cup h$. This argument also shows that there are only fiintely many levels of recursion, since in the first call we decrease the number of constraints and in the second we increase the value $\mathbf{v}_B$ of the candidate basis; since there are only finitely many basis, this increase is bounded.

We now bound the expected running time of the algorithm. What is the probability that we enter the second recursive call? Recall that h is violated by $G - h$ iff h is extreme in G. Since there are at most n extreme constraints in G, the probability of picking one in $G \setminus C$ is bounded by $n/(m-n)$. But this is too weak for us, since there can be extreme constraints in C, which will not be violated. Suppose there are at most j extreme constraints outside C (note C is *not a basis of* G), then the probability is bounded by $j/(m-n)$ and trivially by $\min\{m-n, j\}/(m-n)$. The key idea is that this probability decreases when we make the second recursive call.

To understand this probability, we need to understand the deficit that a candidate basis C has from becoming a basis B of G. What is the relation between the opt of two such basis? Since $C \subseteq G$, we know that $\mathbf{v}_C \leq \mathbf{v}_G = \mathbf{v}_B$, where the last inequality follows since B is a basis of G. Thus the opt of a candidate basis does not exceed the opt of a basis of G. So in some sense, we ought to increase this optimum of the candidate basis, which is precisely what the second recursion does. But when can we say that some constraints in C must occur in a basis of G? This is captured by the following lemma:

LEMMA 19. *A constraint $h \in G$ satisfies $\mathbf{v}_{G-h} < \mathbf{v}_C$ iff h appears in all bases B in G for which $\mathbf{v}_B \geq \mathbf{v}_C$. In particular, h appears in C and is an extreme constraint of G.*

Proof. Let us prove left to right by contradiction. Suppose $\mathbf{v}_{G-h} < \mathbf{v}_C$ and B is a basis in G such that $\mathbf{v}_C \leq \mathbf{v}_B$, which implies $\mathbf{v}_{G-h} < \mathbf{v}_B$. We claim that $h \in B$, if not then $B \subseteq G - h$, and consequently $\mathbf{v}_B \leq \mathbf{v}_{G-h}$ giving us a contradiction. Moreover, h is an extreme constraint, since $C \subseteq G$, we have

$\mathbf{v}_{G-h} < \mathbf{v}_C \leq \mathbf{v}_G$. However, not all extreme constraints in C satisfy the property that $\mathbf{v}_{G-h} < \mathbf{v}_C$; e.g., in Figure 1, if $G := \{1, 2, 3, h\}$, and $C := \{3, h\}$ then C is a basis in G, h is extreme in G, but $\mathbf{v}_{G-h} > \mathbf{v}_C$.

For the converse, we again prove by contradiction: Suppose h appears in all bases B in G for which $\mathbf{v}_B \geq \mathbf{v}_C$ but $\mathbf{v}_{G-h} \geq \mathbf{v}_C$. Now let B' be a basis of $G - h$, then we have $\mathbf{v}_{B'} = \mathbf{v}_{G-h} \geq \mathbf{v}_C$. Note, however, that B' is also a basis in G, which does not contain h, giving us a contradiction. **Q.E.D.**

Let $e_{G,C}$ be the number of constraints common to all basis B in G such that $\mathbf{v}_B \geq \mathbf{v}_C$. From the lemma above we know that $e_{G,C} = |\{h \in G | \mathbf{v}_{G-h} < \mathbf{v}_C\}|$, i.e., $e_{G,C}$ is a lower bound on the number of extreme constraints of G in C. The **hidden dimension** of a basis C wrt to a set G is defined as $k_{G,C} := n - e_{G,C}$, which is an *upper bound* on the deficit that we need to complete to make C a basis of G; we say upper bound, because C may contain extreme constraints not accounted by $e_{G,C}$. The definition implies that there are some $h_1, \ldots, h_{n-k_{G,C}}$ extreme constraints in C that must be present in all bases of G. These $n - k_{G,C}$ constraints define a $k_{G,C}$-dimensional flat, and the overall opt is contained in it. If $k_{G,C} = 0$, then C is a basis of G because all the constraints in C are present in a basis of G. However, the converse is not true, i.e., if C is a basis of G then $k_{G,C}$ may not be zero, but is exactly n minus the number of extreme constraints in G; e.g., Figure 1, the two bases of $G := \{1, 2, 7\}$ are $\{1, 2\}$ and $\{1, 7\}$, the constraint 1 is extreme, and the hidden dimension of both bases is exactly one.

We next claim that the hidden dimension does not increase in the two recursive calls in the algorithm. To see this let us enumerate the constraints h_i in G in such a way that

$$\mathbf{v}_{G-h_1} \leq \mathbf{v}_{G-h_2} \leq \cdots \mathbf{v}_{G-h_m},$$

where $m := |G|$. Note that the ordering is not unique. Where does the opt of a basis C sit in this ordering? We know that there are $n - k_{G,C}$ constraints h in C for which $\mathbf{v}_{G-h} < \mathbf{v}_C$. If we let $h_1, \ldots, h_{n-k_{G,C}} \in C$ be these constraints in the ordering above then we have

$$\mathbf{v}_{G-h_1} \leq \mathbf{v}_{G-h_2} \leq \cdots \mathbf{v}_{G-h_{n-k_{G,C}}} < \mathbf{v}_C \leq \mathbf{v}_{G-h_{n-k_{G,C}+1}} \leq \cdots \mathbf{v}_{G-h_{|G|}}.$$

The next two claims show that the hidden dimension does not increase in the recursive calls.

Claim 1. Let $C \subseteq F \subseteq G$, then $k_{F,C} \leq k_{G,C}$. This implies that the hidden dimension in the first recursive call does not increase. Note that $h_1, \ldots, h_{n-k_{G,C}} \in C$ and hence in F. Moreover, as $F \subseteq G$, for any constraint h, $\mathbf{v}_{F-h} \leq \mathbf{v}_{G-h}$. Therefore, $k_{F,C}$ cannot exceed $k_{G,C}$

Claim 2. If $h \in G - C$, B' and C' are as in the algorithm then $k_{G,C'} \leq k_{G,C}$. Since $h \in G - C$, $\mathbf{v}_{G-h}$ will not violate any of the constraints in C. The possiblity of a violation can only come from the constraints $h_{n-k_{G,C}+1}, \ldots, h_m$. But we claim that constraints beyond h_n are not extreme in G, that is, for $i > n$, $\mathbf{v}_{G-h_i} = \mathbf{v}_G$. Since $G - h_i \subseteq G$, we know $\mathbf{v}_{G-h_i} \leq \mathbf{v}_G$. So suppose for some $i > n$, $\mathbf{v}_{G-h_i} < \mathbf{v}_G = \mathbf{v}_B$, where B is a basis of G. Then from Lemma 19 we know that $h_1, \ldots, h_i$ belong to B, but a basis cannot have more then n constraints, which gives us a contradiction. Hence, for $i > n$, $\mathbf{v}_{G-h_i} = \mathbf{v}_G$. Let's say h was one of $h_{n-k_{G,C}+i}$, $i \in \{1, \ldots, k_{G,C}\}$, and a violation occured by $\mathbf{v}_{G-h}$. How large is $k_{G,C'}$? We know $\mathbf{v}_{B'} = \mathbf{v}_{G-h_{n-k_{G,C}+i}}$. Since $\mathbf{v}_{B'}$ violated h, we know that $\mathbf{v}_{B'} < \mathbf{v}_{B' \cup h}$. Therefore, $\mathbf{v}_{G-h_{n-k_{G,C}+i}} < \mathbf{v}_{C'}$ and hence $k_{G,C'} \leq k_{G,C} - i$, i.e., it goes down in the second recursive call. This drop in the hidden dimension is what helps us set up the recursive bound on the number of call to violation test and basis-computation procedure.

Let $T(m, k)$ be the maximum, over all possible inputs, the expected size of the recursion tree entailed by a call of `Subexp`(G, B), where G has m constraints and hidden dimension is at most k. Based on the two claims above, we know that k also bounds the hidden dimension over all recursive calls. More precisely, $k - i$ is an upper bound in the second scenario, which occurs with probability $1/(m - n)$. Therefore, we have the following expected bound on $T(m, k)$ as a recursion:

$$T(m, k) \leq 1 + T(m - 1, k) + \frac{1}{m - n} \sum_{i=1}^{\min\{m-n,k\}} T(m, k - i). \tag{10}$$

The upper bound in the summation is selected since k may be larger than $m-n$; e.g., $k = n$ and $n < m < 2n$. Also note that $T(n, k) = 0$, for $k = 0, \ldots, n$, since if we have exactly n constraints then they form a basis of our set, i.e., there will be no recursive calls.[5]. Also, k is between 0 and n (the latter can occur for a basis that has no constraint from the basis of G, e.g., it can happen for H_+ initially). A simple induction shows that $T(m, k)$ is bounded by $2^k(m - n)$, which gives the initial exponential bound of Sharir-Welzl. The key insight in the improvement is that when m is not very large compared to k, then a subexponential bound is possible. More precsisely, if $m \leq e^{k/4}\sqrt{k}$ then

$$T(m, k) \leq (m - n) \exp\left(2\left(k \ln \frac{m}{\sqrt{k}}\right)^{1/2} + O(\sqrt{k} + \ln m)\right).$$

The proof uses interesting tools from generatingfunctionology of Wilf. For $m = O(n^2)$, as ensured by Clarkson 2, and $k = n$, we get the desired subexponential bound. They also show that the bound above is essentially tight for the recursion in (10).

7.4 LP-type problems – An abstract framework

The algorithms that we have seen above require very few properties. With every subset G of constraints we associated a unique optimum. Moreover, these opts were totally ordered under '$\leq$'. The two key properties were the value of the opt increases when adding more constraint, and if two sets had the same value then a constraint violated by one is violated by other. If we can axiomatize these properties, then it appears that the algorithms above apply to a larger class of problems.

We consider following type of optimization problems. Given a pair (H, v), where H is a finite set and $v : 2^H \to \mathcal{W}$, such that $(\mathcal{W}, \leq)$ is a totally odered set that has an element $-\infty$ smaller than every other element. H is called the set of constraints, as before, and for $G \subseteq H$, $v(G)$ is the value of (or optimum wrt) G. The optimization problem is to find a minimal subset B if H with the same value as H. Moreover, the two key properties correspond to the following:

Monotonicity: If $F \subseteq G$, then $v(F) \leq v(G)$, i.e., adding more constraints increases the value.

Locality: If $F \subseteq G$ such that $-\infty < v(F) = v(G)$ then for all $h \in H$ if $v(G) < v(G \cup h)$ then $v(F) < v(F \cup h)$, i.e., if F and G take the same value and h is violated by G then h is violated by F as well.

Such problems are called **LP-type** problems. It is clear that linear programming satisfies the two axioms above (see Lemma 14(i, ii)); recall that the two axioms were required to prove the correctness of the algorithm. We can now define the notion of basis B as a set of constraints such that for all $B' \subset B$, $v(B') < v(B)$; note that the definition implies minimality of B. Again, a basis of G is a basis in B such that $v(B) = v(G)$. A constraint h is violated by G if $v(G) < v(G \cup h)$. A constraint h is extreme in G, if h is violated in $G - h$. What is the analogue of n in this setting? Recall that n was the size of any basis in the lp. Therefore, it is natural to define the **combinatorial dimension** $\dim(H, w)$ of the problem corresponding to the pair (H, w) as the maximum cardinality of any basis in H. As was the case in Lemma 15 where n played a crucial role, this parameters is crucial in measuring the running time of the algorithms. The algorithms also need the following three sub-routines:

Violation test: Given a constraint h and a basis B, test whether h is violated by B.

Basis Computation: Given a constraint h and a basis B, compute a basis for $B \cup h$.

Initial Basis: Compute an initial basis of (H, w).

Given these sub-routines, we can now apply the algorithms above to solve the optimization problem corresponding to (H, w). But can we claim the subexponential running time as we did earlier? Not quite. Implicit in the analysis of the subexp algorithm is that we do not make any recursive calls when we had exactly n constraints since then the candidate basis is also a basis of G (this showed in the analysis of the recursion (10) where $T(n, k) = 0$, for $k = 0, \ldots, n$). However, now this may not be true, as bases have different sizes; it is possible that the size of the candidate basis is $\dim(H, w)$, whereas the size of a basis of H is a constant. If, however, all the basis have the same size, namely the combinatorial dimension, then we have a subexp time algorithm. This property is called the *basis-regularity* property. If this property is not true, then the

[5]This point will be crucial later when we study LP-type problems

single exponential running time analysis still holds, giving us an algorithm that makes $2^{\dim(H,w)}$ expected calls to the two tests, violation and basis computation.

7.4.1 Examples of LP-type problems

An example of an LP-type problem is to find the smallest ball containing a given set of m points H in $\mathbb{R}^n$. Let us verify that this is indeed an LP-type problem. What is our v? For any subset $G \subseteq H$, let $v(G)$ be the smallest radius amongst all balls containing G. Our first claim is that $v : G \to \mathbb{R}_{\geq 0}$ is a function, i.e., the smallest radius is uniquely defined, which follows if we show that the smallest ball containing the points is unique.

8 Fundamental Theorem of Linear Inequalities

When does a system of linear equations $A\mathbf{x} = \mathbf{b}$ has a solution? It is clear that there is a solution iff $\mathbf{b}$ belongs to the linear subspace L generated by the columns of A. Consider a vector $\mathbf{y}$ that is orthogonal to the columns of A. If $\mathbf{b} \in L$ then it follows that $\langle \mathbf{y}, \mathbf{b} \rangle = 0$. Is the converse true, i.e., for all $\mathbf{y}$ orthogonal to the columns of A if $\langle \mathbf{y}, \mathbf{b} \rangle = 0$ then $A\mathbf{x} = \mathbf{b}$ has a solution? Let C be a set of lid vectors such that $CA = 0$; thus the rows of C span the space orthogonal to L. By assumption, it follows that $C\mathbf{b} = 0$, and hence $\mathbf{b}$ belongs to L. Thus we have the following theorem:

THEOREM 20 (Fundamental Theorem of Linear Algebra). *The system $A\mathbf{x} = \mathbf{b}$ has a solution iff $\langle \mathbf{y}, \mathbf{b} \rangle = 0$ for all $\mathbf{y}$ such that $\mathbf{y}^t A = \mathbf{0}$.*

There are many ways to state the theorem. However, this form has the advantage that it states that either $\mathbf{b}$ belongs to the columns-space of A or gives a vector $\mathbf{y}$ orthogonal to the column space that witnesses $\mathbf{b}$ not being in the column space. So we could have stated the theorem as either $A\mathbf{x} = \mathbf{b}$ has a solution or there exists a $\mathbf{y}$ orthogonal to the columns of A such that $\langle \mathbf{y}, \mathbf{b} \rangle \neq 0$. Is there a corresponding theorem for linear inequalities? For instance, consider the standard form for lps. When can we say that the feasible set is not empty? The fundamental theorem of linear inequalities, gives us the analogous characterization:

THEOREM 21 (Fundamental Theorem of Linear Inequalities). *Let A be a matrix in $\mathbb{R}^{m \times n}$, $\mathbf{a}_i$ be its n columns, and $\mathbf{b} \in \mathbb{R}^m$. Then either*

(I) $A\mathbf{x} = \mathbf{b}$ has a non-negative solution

or

(II) there exists a hyperplane h through the origin such that $\mathbf{a}_i$'s are on one side of h and $\mathbf{b}$ is strictly on the other side, i.e., there exists a $\mathbf{y} \in \mathbb{R}^m$ such that $\langle \mathbf{y}, \mathbf{a}_i \rangle \geq 0$, for $i = 1, \ldots, n$, and $\langle \mathbf{y}, \mathbf{b} \rangle < 0$.

To understand the geometric import of this theorem, we introduce the following concept: the **cone generated by vectors** $\mathbf{a}_1, \ldots, \mathbf{a}_n$ is the set of all non-negative linear combinations of these vectors. This is clearly a convex set; moreover, it is the convex hull of the n rays $\{ta_i, t \geq 0\}$, $i = 1, \ldots, n$. Note that the origin always belongs to the cone. Thus the theorem above states that either $\mathbf{b}$ belongs to the cone $\mathcal{C}$ generated by $\mathbf{a}_i$'s, or there is a hyperplane h through the origin that separates $\mathbf{b}$ from $\mathcal{C}$. Before we proceed with the proof of this theorem and its various equivalent formulations, let us observe that the two conditions (I) and (II) cannot hold simultaneously: if $\mathbf{y}^t A \geq \mathbf{0}$ then $\mathbf{y}^t A\mathbf{x} \geq 0$ (as $\mathbf{x}$ is non-negative), whereas $\mathbf{y}^t \mathbf{b} < 0$, which is a contradiction.

We see various proofs of Theorem 21. The first one is constructive, and similar to something that we have seen earlier.

8.1 Proof 1 – Constructive

Recall that in the standard form we have $\texttt{rank}(A) = m$; thus $\mathbf{a}_i$'s span $\mathbb{R}^m$.

INPUT: The set $\mathbf{a}_1, \ldots, \mathbf{a}_n$, and $\mathbf{b}$.
OUTPUT: Either (I) of (II).

Let $B \subseteq [n]$ be a set of indices corresponding to lid set of vectors from $\mathbf{a}_1, \ldots, \mathbf{a}_n$.
do
  Find λ_i's such that $\mathbf{b} = \sum_{i \in B} \lambda_i \mathbf{a}_i$; thus $A_B \lambda_B = \mathbf{b}$.
  If λ_i's are non-negative return (I). ◁ *Obtain a solution from λ_i's by introducing zeros*
  Let j be the smallest index such that $\lambda_j < 0$.
  Consider the linear space spanned by the $(m-1)$ lid vectors in $B - \{j\}$.
  Let $\mathbf{y}$ be a vector orthogonal to this hyperplane, normalized such that $\langle \mathbf{y}, \mathbf{a}_j \rangle = 1$.
    ◁ *Thus $\langle \mathbf{y}, \mathbf{b} \rangle = \lambda_j < 0$.*
  If $\langle \mathbf{y}, \mathbf{a}_i \rangle \geq 0$, for $i = 1, \ldots, n$, then return (II). ◁ *$\mathbf{y}$ is the witness vector in Theorem 21*
  Else find the smallest index $k \in [n]$ such that $\langle \mathbf{y}, \mathbf{a}_k \rangle < 0$.
    $B \leftarrow (B \setminus \{j\}) \cup \{k\}$, i.e., replace j by k in B.
    ◁ *Note that B still contains m lid vectors, as a_k is outside the hyperplane defined by $\mathbf{y}$.*

The procedure above is similar to simplex method with Bland's rule. We argue that it terminates. Let B_i denote the set B at the ith iteration. If the procedure does not terminate then there is a sequence of iterations such that $B_p = B_q$, where $p < q$. Again consider all the indices that leave in this sequence. Let ℓ be the index of the largest such hyperplane, and suppose that it leaves at some iteration s and enters (or entered) at some iteration t; note that there is no ordering between t and s; e.g., t will be smaller than s, if ℓ was not present in the initial basis B_p, but it will be larger otherwise. Thus indices larger than ℓ that are present in B_p are present throughout the cycle, in particular

$$B_s \cap \{\mathbf{a}_{\ell+1}, \ldots, \mathbf{a}_n\} = B_t \cap \{\mathbf{a}_{\ell+1}, \ldots, \mathbf{a}_n\}. \tag{11}$$

Let $\mathbf{y}_i$ be the vector in step 6 at the ith iteration. Let $\lambda_{j,s}$ be the value of λ_js in the sth iteration. We have the following claims:

C1. $\lambda_{j,s} \geq 0$, for all $j < \ell$; since ℓ is the smallest index picked in step 5 in the sth iteration, this is obvious for $j \in B_s$ that are smaller than ℓ; for the indices outside B_s, we have $\lambda_j = 0$.

C2. $\lambda_{\ell,s} < 0$; again by step 5.

C3. $\langle \mathbf{y}_t, \mathbf{a}_j \rangle \geq 0$, for $j < \ell$, and $\langle \mathbf{y}_t, \mathbf{a}_\ell \rangle < 0$; because, in step 8 in the tth iteration ℓ is the smallest index.

C4. $\langle \mathbf{y}_t, \mathbf{a}_j \rangle = 0$, for $j \in B_s$ and $j > \ell$; from (11), we know that these indices are also contained in the set $B_t - \{\ell\}$; the claim then follows from the construction of $\mathbf{y}_t$ in step 6.

Now at the tth iteration we know that $\langle \mathbf{y}_t, \mathbf{b} \rangle < 0$. But at the sth iteration we have

$$\langle \mathbf{y}_t, \mathbf{b} \rangle = \sum_{j \in B_s} \lambda_{j,s} \langle \mathbf{y}_t, \mathbf{a}_j \rangle = \sum_{j \in B_s, j < \ell} \lambda_{j,s} \langle \mathbf{y}_t, \mathbf{a}_j \rangle + \lambda_{\ell,s} \langle \mathbf{y}_t, \mathbf{a}_\ell \rangle + \sum_{j \in B_s, j > \ell} \lambda_j \langle \mathbf{y}_t, \mathbf{a}_j \rangle.$$

From C1 and C3 it follows that the first summation is non-negative; from C2 and C3, that the second term is positive; from C4 that the last summation is zero. Thus $\langle \mathbf{y}_t, \mathbf{b} \rangle > 0$, which gives us a contradiction. So the procedure has to terminate. Note that just picking B_s or B_t is not sufficient, since C3 does not apply to the $\mathbf{y}_s$ in the sth iteration, and the λ's in the tth iteration do not satisfy C1 and C2.

Remark:

1. Note that if $\mathbf{b}$ belongs to the cone generated by $\mathbf{a}_1, \ldots, \mathbf{a}_n$ then the algorithm actually finds a set of m lid vectors whose cone contains $\mathbf{b}$.

2. Similarly the separating hyperplane through origin contains $(m-1)$ lid vectors, or more precisely, one less than the rank of $\mathbf{a}_1, \ldots, \mathbf{a}_n, \mathbf{b}$.

∎

8.2 Applications

A polytope can be defined in two ways: either as the bounded intersection of finitely many halfspaces, or as the convex hull of a set of points. Let the first definition be called an H-polytope and the second a V-polytope. Both definitions have their advantages and disadvantages; e.g., to show that intersection of a polytope with a polyhedra is a polytope the first definition is useful; whereas, to show that a projection of a polytope is also a polytope, the second definition is handy. Why are the two definitions equivalent? The fact that they are is the fundamental theorem of polytope theory. To prove it we will show an analogous result for polyhedra;since polytopes are bounded polyhedra, the theorem for polytopes follows easily. We already know that polyhedra are intersection of finitely many halfspaces, let such sets be called H-polyhedra. How can we define V-polyhedra? For that we need the following concepts.

A (convex) **cone** is a set $\mathcal{C}$ of points in euclidean space such that if $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ then so is $\alpha\mathbf{x} + \beta\mathbf{y}$, for all $\alpha, \beta \geq 0$. Note that the origin always belongs to the cone. Analogous to the definition of convex hull, we can define a cone as a set that is closed under non-negative linear combinations. For an arbitrary set $S \subseteq \mathbb{R}^n$ its **conical hull** $\mathcal{C}(S)$ is the intersection of all the cones containing S, or the smallest cone containing S. Egs, conical hulls of polygons and disc with origin on its boundary. A cone is **polyhedral** if it can be represented

as the intersection of finitely many linear halfspaces (that is, those containing the origin), or equivalently as the set $\{\mathbf{y} : \mathbf{y}^t A \geq 0\}$. A cone $\mathcal{C}$ is said to be **finitely generated** if there are vectors $\mathbf{p}_1, \ldots, \mathbf{p}_\ell$ such that $\mathcal{C} = \mathcal{C}(\mathbf{p}_1, \ldots, \mathbf{p}_\ell)$.

Note the definition of cone, and conical hull is analogous to convex and convex hull or linear space and linear hull. The role of convex combinations (or linear combinations) is replaced by non-negative linear combinations. Therefore, it lies somewhere in between a convex set and a linear subspace.

Are all cones finitely generated? Not really – the standard ice-cream cone in 3-d is not. How do finitely generated cones look like? Can it be that a cone is polyhedral but not finitely generated? See Figure 3 illustrating these definitions and a cone in 2-d that is not finitely generated. There are two ways of generating a cone, either as a set generated by non-negative linear combinations of *finitely* many points, or as a *finite* set of inequalities (similar to the definition of V-polytopes and H-polytopes). The next theorem states that the two notions are equivalent.

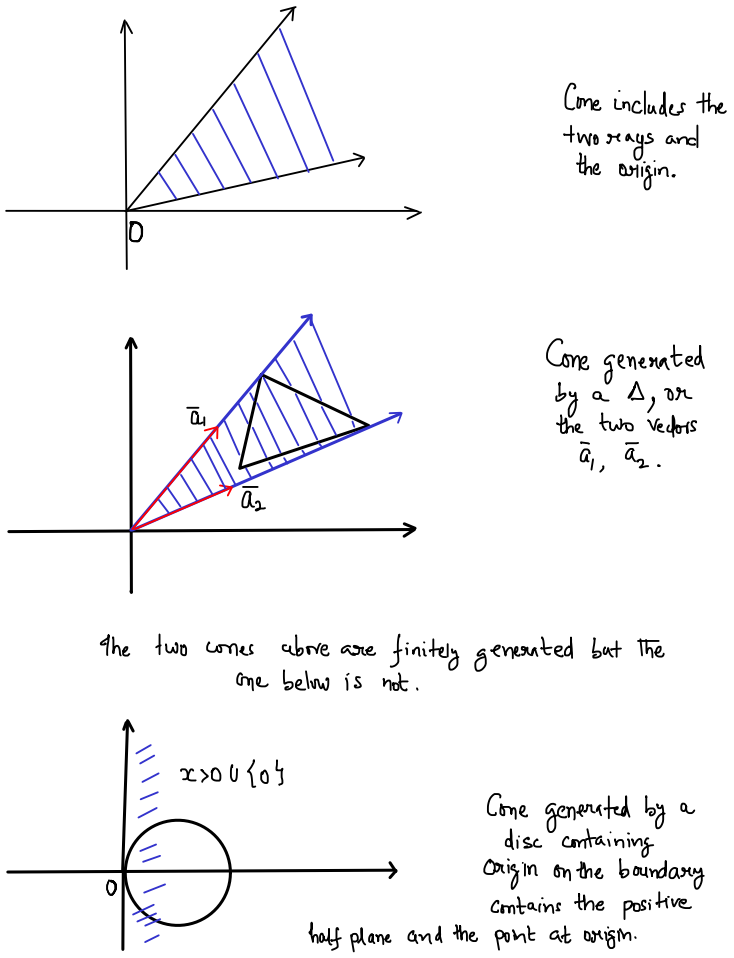


Figure 2: Cones – Examples

THEOREM 22 (Farkas-Minkowski-Weyl). *A convex cone is polyhedral iff it is finitely generated.*

Proof. Suppose the column vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^m$ generate $\mathcal{C}$. Further suppose that they span the euclidean space $\mathbb{R}^m$; otherwise apply the theorem in their linear span, and lift the half spaces in the linear span to half spaces in $\mathbb{R}^m$ (this lift is not unique); express the linear span as the intersection of finitely many

halfspaces in $\mathbb{R}^m$; these halfspaces along with the lifted halfspaces give us the polyhedral representation of $\mathcal{C}$; see Figure 3 for an example in the plane. What are our candidate halfspaces in the polyhedral representation? Let H be the set of all the linear halfspaces that contain all $\mathbf{a}_i$'s and the corresponding hyperplane is spanned by exactly $m-1$ of the a_i's; note that the hyperplane contains the origin. We claim that $\mathcal{C}$ is equal to the intersection of these finitely many halfspaces. This follows from Theorem 21: clearly, the cone is contained in these halfspaces; moreover, any $\mathbf{b}$ that is not in $\mathcal{C}$ is outside a halfspace in H. Thus $\mathcal{C}$ is equal to the intersection of the halfspaces in H.

The proof idea is demonstrated in Figure 3. Suppose $\mathcal{C}$ is a polyhedral cone, i.e., the intersection of all the halfspaces $\{\mathbf{x} : \langle \mathbf{a}_i, \mathbf{x}\rangle \geq 0\}$, for $i = 1, \ldots, n$, where $\mathbf{a}_i \in \mathbb{R}^m$ and we assume that they span the whole space (otherwise, work in their linear span). Consider the cone generated by $\mathbf{a}_1, \ldots, \mathbf{a}_n$. By the first part we know that every finitely generated cone is polyhedral, i.e, there exists $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$ such that

$$\mathcal{C}(\mathbf{a}_1, \ldots, \mathbf{a}_m) = \{\mathbf{x} : \langle \mathbf{b}_i, \mathbf{x}\rangle \geq 0, i = 1, \ldots, \ell\}\,. \tag{12}$$

We claim that $\mathcal{C} = \mathcal{C}(\mathbf{b}_1, \ldots, \mathbf{b}_\ell)$.

1. $\mathbf{b}_i \in \mathcal{C}$, for $i = 1, \ldots, \ell$. Since $\mathcal{C}(\mathbf{a}_1, \ldots, \mathbf{a}_m) = \{\mathbf{y} : \mathbf{y}^t B \geq \mathbf{0}\}$, and clearly $\mathbf{a}_i \in \mathcal{C}(\mathbf{a}_1, \ldots, \mathbf{a}_m)$, it follows that $\langle \mathbf{b}_i, \mathbf{a}_j\rangle \geq 0$, for $i = 1, \ldots, \ell$ and $j = 1, \ldots, m$. Since inner product is commutative, it follows that $\mathbf{b} \in \mathcal{C}$.

2. Thus for any non-negative linear combination $\mathbf{b}$ of the $\mathbf{b}_i$'s we have $\langle \mathbf{a}_j, \mathbf{b}\rangle \geq 0$, for $j = 1, \ldots, m$, and hence $\mathcal{C}(\mathbf{b}_1, \ldots, \mathbf{b}_\ell) \subseteq \mathcal{C}$.

3. To show that inclusion the other way consider a $\mathbf{y} \in \mathcal{C}$ but not in $\mathcal{C}(\mathbf{b}_1, \ldots, \mathbf{b}_\ell)$. Then by Theorem 21(II) we know that there is a vector $\mathbf{w}$ such that $\langle \mathbf{b}_i, \mathbf{w}\rangle \geq 0$, for $i = 1, \ldots, \ell$, and $\langle \mathbf{w}, \mathbf{y}\rangle < 0$; but the condition on $\langle \mathbf{b}_i, \mathbf{w}\rangle$ and (12) imply that $\mathbf{w} \in \mathcal{C}(\mathbf{a}_1, \ldots, \mathbf{a}_m)$, i.e., $\mathbf{w} = \sum_i \lambda_i \mathbf{a}_i$, where $\lambda_i \geq 0$. Thus

$$\langle \mathbf{w}, \mathbf{y}\rangle = \sum_{i=1}^{n} \lambda_i \langle \mathbf{a}_i, \mathbf{y}\rangle \geq 0,$$

where in the last inequality we use the fact that $\mathbf{y} \in \mathcal{C}$ and hence $\langle \mathbf{y}, \mathbf{a}_i\rangle \geq 0$, for $i = 1, \ldots, n$. But this is a contradiction since $\langle \mathbf{w}, \mathbf{y}\rangle < 0$.

Q.E.D.

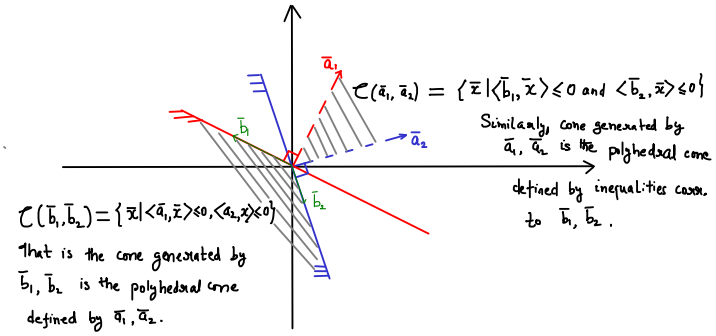


Figure 3: Demonstrating the proof of the sufficiency part of Theorem 22

To define the notion of V-polyhedra we also need the following concept: The **Minkowski sum** (or vector sum) of two sets A, B is the set

$$A + B := \{\mathbf{v} + \mathbf{w} : \mathbf{v} \in A \text{ and } \mathbf{w} \in B\}\,. \tag{13}$$

See Figure 4 for illustration; we assume that the circle and the two rays are hinged at origin. We already know that an H-polyhedra is the intersection of finitely many halfspaces. Given two finite point sets Q, R in euclidean space, a V-polyhedra is defined as the Minkowski sum of the convex hull of Q and the cone generated by R. The next theorem is the fundamental theorem of polyhedral theory, namely the concept of H-polyhedra is the same as V-polyhedra. The proof will reduce it to the case of cones.

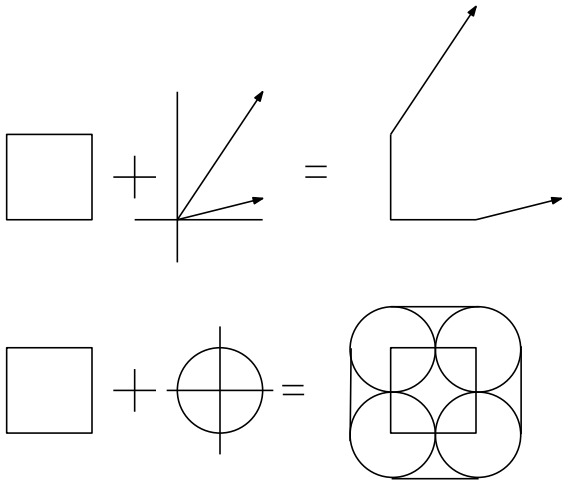


Figure 4: Minkowski Sum

THEOREM 23 (Decomposition Theorem of Polyhedra). *A set P of euclidean space is an H-polyhedron iff it is a V-polyhedron.*

Proof. Let $P = \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}\}$ be the H-polyhedra. The idea to get the decomposition is to homogenize/projectivize P; this expresses P as a system of homogeneous inequalities, i.e., as a polyhedral cone, which we know is finitely generated; the generators of this cone come in two sets – those that are at infinity and those that are finite; the former points give us the cone part of P and the latter the polytope part, and hence the V-polyhedra formulation of P.

Homogenizing the system of inequalities $A\mathbf{x} \leq \mathbf{b}$, we get a system $A'(\mathbf{x}, y)^t \leq 0$, where A' is obtained from A by adding an extra column $-\mathbf{b}$. From Theorem 22 we know that the polyhedral cone in $\mathbb{R}^{n+1}$ defined by $\mathcal{C} := \{(\mathbf{x}, y)^t : A'(\mathbf{x}, y)^t \leq 0 \text{ and } y \geq 0\}$ is finitely generated by say the vectors $(\mathbf{x}_i, y_i)^t$, for $i = 1, \ldots, k$; the non-negativity constraint on y is required to ensure that $A\mathbf{x} \leq \mathbf{b}$ holds and we do not flip the sign of $\mathbf{b}$. Moreover, as scaling the vectors $(\mathbf{x}_i, y_i)^t$ by a positive quantity does not change their satisfying the inequalities, we may assume that for those vectors with y_i non-zero (hence positive) is actually 1. Thus $\mathcal{C}$ is generated by vectors of the form $(\mathbf{x}_i, 1)^t$, $i = 1, \ldots, j$, and $(\mathbf{x}_i, 0)^t$, for $i = j+1, \ldots, k$. The former are the finite part of the homogenized version of P, so let Q be the convex hull of $\mathbf{x}_1, \ldots, \mathbf{x}_j$; the latter are the points at infinity so let $\mathcal{C} := \mathcal{C}(\mathbf{x}_{j+1}, \ldots, \mathbf{x}_k)$. Now $\mathbf{x} \in P$ iff $(\mathbf{x}, 1)^t$, which is equivalent to

$$\begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} = \sum_{i=1}^{j} t_i \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} + \sum_{i=j+1}^{k} t_i \begin{pmatrix} \mathbf{x}_i \\ 0 \end{pmatrix},$$

where $t_i \geq 0$. This instead is equivalent to

$$\mathbf{x} = \sum_{i=1}^{j} t_i \mathbf{x}_i + \sum_{i=j+1}^{k} t_i \mathbf{x}_i, \quad \text{and} \quad \sum_{i=1}^{j} t_i = 1.$$

The first sum on the RHS clearly belongs to Q and the second sum to $\mathcal{C}$.

For the converse, suppose $Q = \{\mathbf{x}_1, \ldots, \mathbf{x}_j\}$ and $R = \{\mathbf{x}_{j+1}, \ldots, \mathbf{x}_k\}$. Then $\mathbf{v} \in \mathrm{CH}(Q) + \mathcal{C}(R)$ iff there exists non-negative $\mathbf{t}_1, \ldots, \mathbf{t}_k$ such that

$$\mathbf{v} = \sum_{i=1}^{k} \mathbf{t}_i \mathbf{x}_i, \ \sum_{i=1}^{j} \mathbf{t}_i = 1.$$

This is equivalent to stating that

$$\begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix} = \sum_{i=1}^{j} \mathbf{t}_i \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} + \sum_{i=j+1}^{k} \mathbf{t}_i \begin{pmatrix} \mathbf{x}_i \\ 0 \end{pmatrix},$$

which is the same as saying that $(\mathbf{v}, 1)$ belongs to the cone generated by $(\mathbf{x}_i, 1)$, for $i = 1, \ldots, j$, and $(\mathbf{x}_i, 0)$, $i = j+1, \ldots, k$. Since any finitely generated cone is also polyhedral it follows that there is a matrix A' such that $(\mathbf{v}, 1) \in \{\mathbf{x}' : A'\mathbf{x}' \leq 0\}$. If $A' = (A\ \mathbf{b})$ then this implies that $A\mathbf{v} \leq -\mathbf{b}$, i.e., $\mathbf{v}$ is in the polyhedra $\{\mathbf{x} : A\mathbf{x} \leq -\mathbf{b}\}$.

Q.E.D.

As a consequence we have the finite basis theorem of polytope theory: A set P is a polytope iff it is a bounded polyhedron.

Our second application is the following theorem of Caratheodory, which follows as a consequence of the first remark following the proof of Theorem 21:

THEOREM 24 (Caratheodory's Theorem). *If $S \subseteq \mathbb{R}^m$ is a finite point set and $\mathbf{p} \in \mathcal{C}(S)$, then $\mathbf{p}$ is in the cone of at most m lid vectors from S.*

Using this we can easily prove Theorem 11: If $\mathbf{p}$ belongs to the convex hull of a set S then it belongs to the convex hull of some $m+1$ affinely independent points in S. To see this consider the set $T := \{(\mathbf{x}, 1)^t : \mathbf{x} \in S\} \subseteq \mathbb{R}^{m+1}$. Now $\mathbf{p} \in \mathrm{CH}(S)$ iff $(\mathbf{p}, 1)^t \in \mathcal{C}(T)$. From the theorem above we know that there are $(m+1)$ lid vectors $(\mathbf{x}_0, 1)^t, \ldots, (\mathbf{x}_m, 1)^t$ in T such that $(\mathbf{p}, 1)^t$ is in their cone. Again, this implies that $\mathbf{p}$ is in the convex hull of $\mathbf{x}_0, \ldots, \mathbf{x}_m \in S$. Moreover, these points are affinely independent as $(\mathbf{x}_0, 1)^t, \ldots, (\mathbf{x}_m, 1)^t$ are lid.

8.3 Variants – Farkas's Lemma

The theorem Theorem 21 is actually about a system of equations have a non-negative solution.

Corollary 25 (Farkas's Lemma)**.** *Let A be a matrix and $\mathbf{b}$ a vector. Then there exists a vector $\mathbf{x} \geq 0$ with $A\mathbf{x} = \mathbf{b}$ iff $\langle \mathbf{y}, \mathbf{b} \rangle \geq 0$ for all vectors $\mathbf{y}$ such that $\mathbf{y}^t A \geq \mathbf{0}$.*

Proof. The necessary part is easy: if there exists a non-negative solution to $A\mathbf{x} = \mathbf{b}$ then for all vectors $\mathbf{y}$ such that $\mathbf{y}^t A \geq \mathbf{0}$, we have

$$\langle \mathbf{y}, \mathbf{b} \rangle = \mathbf{y}^t A\mathbf{x} \geq 0$$

since $\mathbf{x} \geq 0$. For the sufficiency part, suppose there is no $\mathbf{x} \geq 0$ such that $A\mathbf{x} = \mathbf{b}$, i.e., $\mathbf{b}$ is not in the cone generated by the columns of A, then from II we know that there is vector $\mathbf{y}$ such that $\mathbf{y}^t\mathbf{b} < 0$ and $\mathbf{y}^t A \geq 0$, which would be a contradiction. **Q.E.D.**

Geometrically, if $\mathbf{b}$ is in the cone generated by the columns of A then we know that for any halfspace that contains all the columns of A also contains its cone, and hence $\mathbf{b}$.

The variants above all talk about equations having a non-negative solution. Is it possible to say something about inequalities?

Corollary 26 (Variant 2)**.** *The system $A\mathbf{x} \leq \mathbf{b}$ has a non-negative solution iff $\langle \mathbf{y}, \mathbf{b} \rangle \geq 0$ for each* non-negative *vector $\mathbf{y}$ such that $\mathbf{y}^t A \geq 0$.*

Proof. After adding slack variables $\mathbf{z}$, we know that $A\mathbf{x} \leq \mathbf{b}$ has a non-negative solution iff $[A\ I](\mathbf{x}, \mathbf{z})^t = A\mathbf{x} + \mathbf{z} = \mathbf{b}$ has a non-negative solution in $(\mathbf{x}, \mathbf{z})$. Applying Corollary 25 to this system we obtain that it has a non-negative solution iff for all vectors $\mathbf{y}$ such that $\mathbf{y}^t[A\ I] \geq 0$ implies $\langle \mathbf{y}, \mathbf{b} \rangle \geq 0$. But $\mathbf{y}^t[A\ I] \geq 0$ iff $\mathbf{y}^t A \geq 0$ and $\mathbf{y} \geq 0$. **Q.E.D.**

Note the strengthening of the condition on $\mathbf{y}$.

Corollary 27 (Variant 3)**.** *The system $A\mathbf{x} \leq \mathbf{b}$ has a solution iff $\langle \mathbf{y}, \mathbf{b} \rangle \geq 0$ for each* non-negative *vector $\mathbf{y}$ such that $\mathbf{y}^t A = 0$.*

Proof. The proof is applying Farkas lemma to the system obtained by adding slack variables and representing each variable as a difference of two non-negative variables. The result is obtained by applying Farkas's lemma to the system $[I\ A\ -A]\mathbf{x}' = \mathbf{b}$; note that the condition $\mathbf{y}^t[I\ A\ -A] \geq 0$ is equivalent to $\mathbf{y} \geq 0$ and $\mathbf{y}^T A = 0$. **Q.E.D.**

9 Duality in Linear Programming

One of the most fundamental concepts of LP is the notion of duality. This notion already occurs in linear algebra where a point in $\mathbb{R}^n$ is the dual of a hyperplane through the origin; we have seen glimpses of this while going through the proof of equivalence of polyhedral cones and finitely generated cones, Theorem 22. In this section, we extend this concept to LP.[6]

On October 30, 1947, the year Dantzig proposed the simplex method, he decided to meet John Von Neumann, considered the leading mathematician of that time, to gain insight into the problem. This is how Dantzig decribes the meeting:

> I remember trying to describe to von Neumann (as I would to an ordinary mortal) the Air Force problem. I began with the formulation of the linear programming model in terms of activities and items, etc. He did something which I believe was uncharacteristic of him. "Get to the point," he snapped at me impatiently. Having at times a somewhat low kindling point, I said to myself, "OK, if he wants a quickie, then thats what hell get." In under one minute I slapped on the blackboard a geometric and algebraic version of the problem. Von Neumann stood up and said, "Oh that!" Then, for the next hour and a half, he proceeded to give me a lecture on the mathematical theory of linear programs. At one point, seeing me sitting there with my eyes pop- ping and my mouth open (after all I had searched the literature and found nothing), von Neumann said: "I dont want you to think I am pulling all this out of my sleeve on the spur of the moment like a magician. I have recently completed a book with Oscar Morgenstern on the theory of games. What I am doing is conjecturing that the two problems are equivalent. The theory that I am outlining is an analogue to the one we have developed for games." Thus I learned about Farkas' Lemma, and about Duality for the first time.

The term Primal was introduced around 1954. This is how Dantzig describes it:

> It came about this way: W. Orchard-Hays, who is responsible for the first commercial grade L.P. software, said to me at RAND one day around 1954: "We need a word that stands for 'the original problem of which this is the dual'." I, in turn, asked my father, Tobias Dantzig, mathematician and author, well known for his books popularizing the history of mathematics. He knew his Greek and Latin. Whenever I tried to bring up the subject of linear programming, Toby (as he was affec- tionately known) became bored and yawned. But on this occasion he did give the matter some thought and several days later suggested Primal as the natural antonym since both primal and dual derive from the Latin. It was Toby's one and only contribution to linear programming...

9.1 Duality – Algebraic Viewpoint

The example in Figure 5 and Figure 6 show us the idea behind the algebraic approach. The input LP is called the **primal** LP and the LP obtained after doing the reduction suggested in the example is called the **dual** LP. Thus given a primal LP in the canonical form

$$\text{maximize } \langle \mathbf{c}, \mathbf{x} \rangle \text{ s.t. } A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$$

we have the corresponding dual form:

$$\text{minimize } \langle \mathbf{b}, \mathbf{y} \rangle \text{ s.t. } A^t\mathbf{y} \geq \mathbf{c},\ \mathbf{y} \geq \mathbf{0}.$$

The construction illustrated in the examples only handle the case when the constraints are of the form "$\leq$". In general lp we can have "$\geq$" and equality constraints. What should we do then? It is clear that if a constraint is an equality then the multiplying variable y can take any value (positive/negative/zero); similarly if a constraint is "$\geq$" then the multiplying variable is taken to be non-positive. The constraints in the dual are governed by the sign of the corresponding variable in the objective function. Suppose the linear combination $\mathbf{y}^t A$ is expressed as $\sum_{i=1}^{n} d_i x_i$, where d_i depends on $\mathbf{y}$. Then we want that $\langle \mathbf{c}, \mathbf{x} \rangle \leq \sum_i d_i x_i$, i.e.,

[6] Dantzig meets Von Neumann.

$\sum_i (d_i - c_i)x_i \geq 0$. This can be ensured by making each term positive. Thus if $x_i \geq 0$ in then primal then the corresponding constraint in the dual is "$d_i \geq c_i$"; similarly, if $x_i \leq 0$ in the primal then the corresponding constraint in the dual is "$d_i \leq c_i$"; and, if $x_i \in \mathbb{R}$ in the primal then the corresponding constraint in the dual is $d_i = c_i$. Also note the change in the number of variables in the dual, where it equals the dimension of $\mathbf{b}$ (i.e., number of rows in A), and the number of inequalities, which now equals the number of variables; this observation can be used to choose the appropriate form for an algorithm that performs faster depending on the smaller of the two quantities. It can be verified that the dual of the dual is the primal itself.

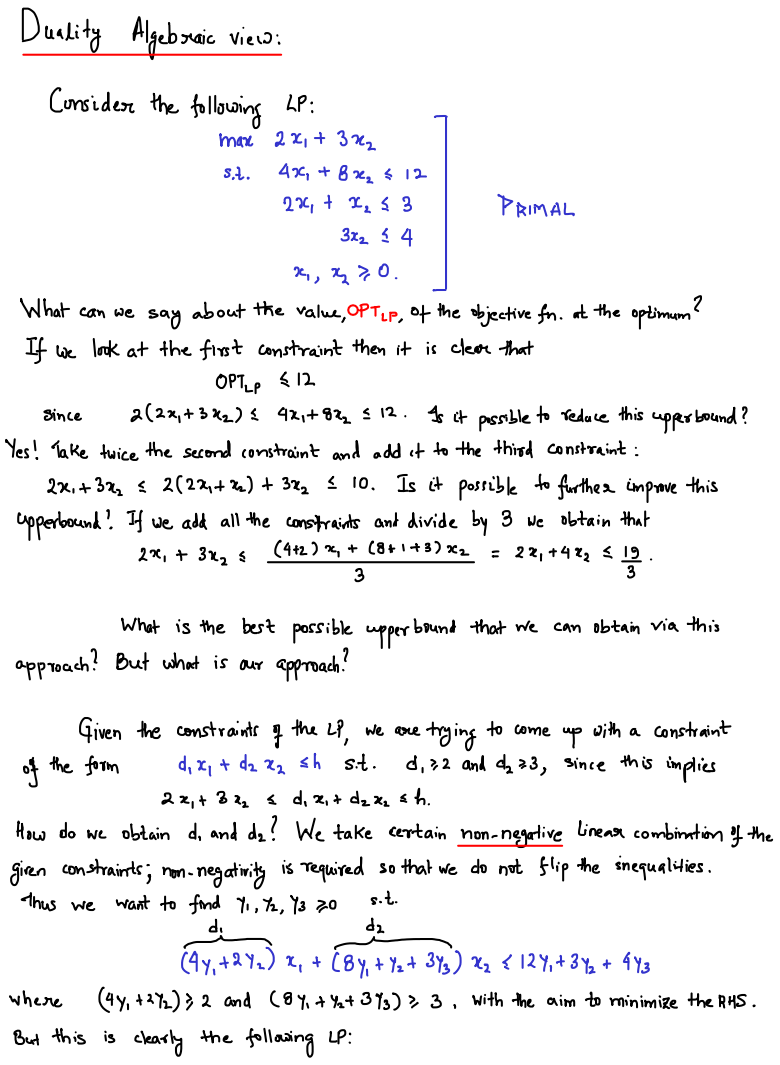


Figure 5

By the construction we have the following:

LEMMA 28 (Weak Duality). *For all feasible solutions* $\mathbf{y}$ *of the dual and for all feasible solutions* $\mathbf{x}$ *of the primal we have*

$$\langle \mathbf{c}, \mathbf{x} \rangle \leq \langle \mathbf{b}, \mathbf{y} \rangle.$$

In particular, if primal is unbounded then the dual is infeasible, and if the dual is unbounded then the primal is infeasible.

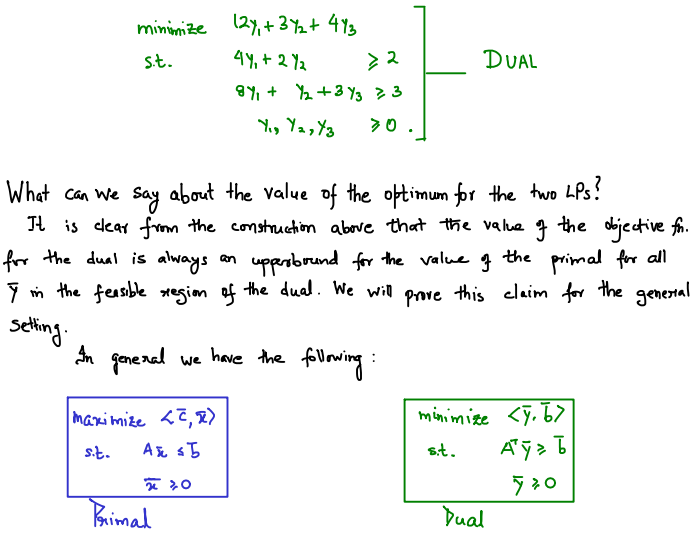


Figure 6

		Dual		
		Infeasible	Unbounded	Opt-exists
Primal	Infeasible	?	?	?
	Unbounded	✓	NA	NA
	Opt-exists (feasible and bounded)	?	?	?

Table 1: Dependence of the Dual on the Primal implied by Weak Duality

Proof. Suppose $\mathbf{x}$ is feasible solution for primal and $\mathbf{y}$ for dual; then we know that $A^t\mathbf{y} \geq \mathbf{c}$. Therefore,

$$\langle \mathbf{c}, \mathbf{x} \rangle = \mathbf{c}^t\mathbf{x} \leq \mathbf{y}^t A\mathbf{x} \leq \mathbf{y}^t\mathbf{b} = \langle \mathbf{y}, \mathbf{b} \rangle.$$

Thus $\text{OPT}_{\text{Primal}} \leq \text{OPT}_{\text{Dual}}$. **Q.E.D.**

An LP can be exactly one of infeasible, unbounded and optimum-exists. The corresponding relations implied by the weak duality between the primal and dual are given in Table 1; basically, we only know the relation when primal is unbounded. The entries marked with questions need to be resolved. The following examples resolve the first row.

1. maximize $x_1 + x_2$ such that $x_1 + x_2 \leq -1$, $x_1, x_2 \geq 0$; the corresponding dual is minimize $-y$, such that $y \geq 1$; it is clear that the primal is infeasible and the dual is unbounded.

2. maximize $x_1 + x_2$ such that $x_2 \geq 1$ and $x_1 \leq -1$, $\mathbf{x} \geq 0$. The dual is minimize $-y_1 - y_2$ such that $y_1 \leq -1$ and $y_2 \geq 1$. But observe that this is the primal lp. This example is interesting because it is self-dual. This happens exactly when A is skew-symmetric, i.e., $A^t = -A$, and $\mathbf{b} = -\mathbf{c}$.

The remaining entries in Table 1 will be resolved by the strong duality theorem. The stronger version of the duality theorem state that in the last row only the last column has to be check marked, i.e., if the primal has an optimum then so does the dual and vice versa. We give two proofs – one based on the Simplex method and another on Farkas's Lemma.

The statement of the strong duality is the following:

		Dual		
		Infeasible	Unbounded	Opt-exists
Primal	Infeasible	✓	✓	NA
	Unbounded	✓	NA	NA
	Opt-exists (feasible and bounded)	NA	NA	✓

Table 2: Dependence of the Dual on the Primal implied by Strong Duality. Exactly one of ✓occurs.

THEOREM 29 (Strong Duality). *Given a primal LP and its dual exactly one of the following situation holds:*

1. *Both primal and dual are infeasible.*
2. *Primal is infeasible and dual is unbounded.*
3. *If primal is unbounded then dual is infeasible (by weak duality).*
4. *If primal is feasible and bounded then dual is also feasible and bounded. Moreover, $OPT_{Primal} = OPT_{Dual}$.*

The relations are shown in Table 2. The theorem states that exactly one of the checkmarks takes place. The first two conditions imply the entries in the first row, and the last condition implies the entries in the last row.

¶1. Strong Duality via Simplex Method We will show the following: if the simplex method returns an optimum solution for the primal then from the associated simplex tableau we will be able to construct an optimum solution for the dual. First observe that the primal is not in the equational form. We bring it into the equational form by adding slack variables. Let the resulting form be maximize $\langle \overline{\mathbf{c}}, \overline{\mathbf{x}} \rangle$ s.t. $\overline{A}\overline{\mathbf{x}} = \mathbf{b}$ and $\overline{\mathbf{x}} \geq \mathbf{0}$, where $\overline{\mathbf{x}}$ is an $(m+n)$-dimensional vector, $\overline{\mathbf{c}}$ is $\mathbf{c}$ followed by m zeros, and $\overline{A} = (A\ I_m)$. Assume that we have an optimum solution $\overline{\mathbf{x}}^*$ to this equational form. Let $B \subseteq [m+n]$ be the feasible basis associated with the final tableaux. Then from Lemma 12 we know that $\overline{\mathbf{x}}_B = \overline{A}_B^{-1}\mathbf{b}$ and

$$\text{OPT}_{\text{Primal}} = \langle \overline{\mathbf{c}}, \overline{\mathbf{x}} \rangle = \langle \overline{\mathbf{c}}_B, \overline{A}_B^{-1}\mathbf{b} \rangle = \overline{\mathbf{c}}_B^t(\overline{A}_B^{-1}\mathbf{b}) = (\overline{\mathbf{c}}_B^t\overline{A}_B^{-1}) \cdot \mathbf{b}.$$

Therefore, it makes sense to define $\mathbf{y}^* := (\overline{\mathbf{c}}_B^t\overline{A}_B^{-1})^t$. Then it is clear from above that $\text{OPT}_{\text{Primal}} = \langle \mathbf{y}^*, \mathbf{b} \rangle$, and since for any other dual-feasible solution the value of the objective function is at least $\text{OPT}_{\text{Primal}}$, to show dual-optimality of $\mathbf{y}^*$ it suffices to show that $\mathbf{y}^*$ is indeed dual-feasible, i.e, $A^t\mathbf{y}^* \geq \mathbf{c}$ and $\mathbf{y}^* \geq 0$. Note that $\mathbf{y}^*$ is an m-dimensional vector, and hence these two constraints are implied if we show that $\overline{A}^t\mathbf{y}^* \geq \overline{\mathbf{c}}$. From the definition of $\mathbf{y}^*$ we have

$$\mathbf{w} := \overline{A}^t\mathbf{y}^* = \overline{A}^t(\overline{\mathbf{c}}_B^t\overline{A}_B^{-1})^t = (\overline{\mathbf{c}}_B^t\overline{A}_B^{-1}\overline{A})^t$$

The vector $\mathbf{w}$ is an $(m+n)$ dimensional vector. We consider its entries corresponding to the feasible basis B, represented by $\mathbf{w}_B$ and its entries corresponding to the nonbasic indices, represented by $\mathbf{w}_N$. From the preceding equation we have that the basic entries of $\mathbf{w}$ correspond to picking the basic columns in $\overline{A}$, i.e.,

$$\mathbf{w}_B = (\overline{\mathbf{c}}_B^t\overline{A}_B^{-1}\overline{A}_B)^t = \overline{\mathbf{c}}_B.$$

Doing the same for the non-basic entries we have

$$\mathbf{w}_N = (\overline{\mathbf{c}}_B^t\overline{A}_B^{-1}\overline{A}_N)^t = (\overline{A}_B^{-1}\overline{A}_N)^t\overline{\mathbf{c}}_B,$$

which from Lemma 12 we know is equal to $\overline{\mathbf{c}}_N - \mathbf{r}$. But as $\mathbf{x}^*$ was the optimum we know that $\mathbf{r} \leq \mathbf{0}$, and hence $\mathbf{w}_N \geq \overline{\mathbf{c}}_N$. Thus we have shown that $\mathbf{w} = \overline{A}^t\mathbf{y}^* \geq \overline{\mathbf{c}}$ as desired.

¶2. Strong Duality via Farkas's Lemma Again assume that the primal LP has an optimum and that γ is $\text{OPT}_{\text{Primal}}$. We will show that the dual has an optimum and the objective value is the same as γ. Since the primal is feasible we know that the following system of inequalities has a non-negative solution:

$$A\mathbf{x} \leq \mathbf{b} \text{ and } \langle \mathbf{c}, \mathbf{x} \rangle \geq \gamma. \tag{14}$$

Moreover, as $\gamma = \text{OPT}_{\text{Primal}}$, for any $\epsilon > 0$ the following system has *no non-negative* solution:

$$A\mathbf{x} \leq \mathbf{b} \text{ and } \langle \mathbf{c}, \mathbf{x} \rangle \geq \gamma + \epsilon. \tag{15}$$

Adding slack variables to (14), we define

$$A' := \begin{bmatrix} A & I_m \\ -\mathbf{c}^t & \mathbf{0} \end{bmatrix} \text{ and } \mathbf{b}'_\epsilon := \begin{pmatrix} \mathbf{b} \\ -\gamma - \epsilon \end{pmatrix}.$$

Then (14) implies that $\mathbf{b}'_0$ belongs to the cone generated by the columns of A' and (15) implies that $\mathbf{b}'_\epsilon$ does not belong to the cone, for all $\epsilon > 0$. This is possible iff $\mathbf{b}'_0$ is on the boundary of the cone. Therefore, there exists a non-negative $\mathbf{y} = (\mathbf{u}\ w)^t$, namely the normal corresponding to a face of the cone containing $\mathbf{b}'_0$, such that

1. $\mathbf{y}^t A' \geq 0$,
2. $\langle \mathbf{y}, \mathbf{b}'_\epsilon \rangle < 0$, for all $\epsilon > 0$, and
3. $\langle \mathbf{y}, \mathbf{b}'_0 \rangle = 0$.

The conditions above are equivalent to

$$A^t\mathbf{u} \geq w\mathbf{c},\ w\gamma = \langle \mathbf{u}, \mathbf{b} \rangle < w(\gamma + \epsilon).$$

The bounds on $\langle \mathbf{u}, \mathbf{b} \rangle$ imply that $w\epsilon > 0$, and since $\epsilon > 0$, it follows that $w > 0$. Therefore, if we consider the vector $\mathbf{v} := \mathbf{u}/w$, then we have

$$A^t\mathbf{v} \geq \mathbf{c} \text{ and } \langle \mathbf{b}, \mathbf{v} \rangle = \gamma.$$

That is, $\mathbf{v}$ is a feasible solution for the dual, and its objective value is $\text{OPT}_{\text{Primal}}$ as desired. Moreover, $\mathbf{v} \geq \mathbf{0}$, as $\mathbf{u} \geq \mathbf{0}$ and w was positive. Thus $\mathbf{v}$ is an optimum for the dual, since from the weak duality we know that the objective value of any dual feasible solution is at least γ.

9.2 Duality – Geometric Viewpoint

Suppose the polyhedron $P := \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}\}$ is not empty. Consider moving the hyperplane orthogonal to $\mathbf{c}$ across P in the direction of $\mathbf{c}$. If P is bounded in that direction, then we must reach a scenario as illustrated in Figure 7. That is, there must be at least n constraints $\mathbf{a}_i$ indexed by I, such that the solution $\mathbf{x}^*$ to the equation $A_I\mathbf{x} = \mathbf{b}_I$ attains the maximum for the objective function. For this to happen, the vector $\mathbf{c}$ must be in the cone generated by $\mathbf{a}_i$'s, i.e., there exists non-negative $\mathbf{y}^*$ such that $\mathbf{c}^t = \sum_{i \in I} \mathbf{y}^*_i \mathbf{a}_i$, i.e., $\mathbf{c}^t = (\mathbf{y}^*_I)^t A_I$, or equivalently $(\mathbf{y}^*_I)^t = \mathbf{c}^t A_I^{-1}$; note $\mathbf{c}$ is a column vector whereas $\mathbf{a}_i$'s are row vectors, and $\mathbf{y}^*$ is zero outside I. Therefore,

$$\langle \mathbf{c}, \mathbf{x}^* \rangle = \langle \mathbf{c}, A_I^{-1}\mathbf{b}_I \rangle = (\mathbf{c}^t A_I^{-1})\mathbf{b}_I = \langle \mathbf{y}^*_I, \mathbf{b}_I \rangle = \langle \mathbf{y}^*, \mathbf{b} \rangle.$$

Thus $\mathbf{y}^*$ is the solution to the system $\{\mathbf{y} : A^t\mathbf{y} = \mathbf{c},\ \mathbf{y} \geq 0\}$, which minimizes $\langle \mathbf{y}, \mathbf{b} \rangle$ (the minimization follows from weak duality); note the equality in $A^t\mathbf{y} = \mathbf{c}$, this is because we started with the weaker condition $A\mathbf{x} \leq \mathbf{b}$ (since non-negativity of $\mathbf{x}$ is missing, the optimum $\mathbf{x}^*$ is a vertex of the polyhedron defined by $A\mathbf{x} \leq \mathbf{b}$). The geometric insight also shows us that if $\mathbf{y}^*_i$ is positive, then the ith constraint $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i$ is actually satisfied as an equality by $\mathbf{x}^*$. Algebraically, this can be stated as follows:

$$(\langle \mathbf{a}_i, \mathbf{x}^* \rangle - b_i) \cdot y^*_i = 0, \text{ for } i = 1, \ldots, m,$$

i.e., either the constraint $\langle \mathbf{a}_i, \mathbf{x}^* \rangle = b_i$ or $y^*_i = 0$ is satisfied as equality (no slack) by $\mathbf{x}^*$ and $\mathbf{y}^*$. Since $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i$, and $\mathbf{y}^* \geq 0$, the above set of equations is equivalent to the following succinct form:

$$\langle \mathbf{y}^*, A\mathbf{x}^* - b \rangle = 0. \tag{16}$$

This condition is called **complementary slackness** and is an equivalent formulation of strong duality:

LEMMA 30 (Complementary Slackness iff strong duality). *If* $\mathbf{x}^*$ *is an optimum for the primal and* $\mathbf{y}^*$ *for the dual then* $\langle \mathbf{y}^*, A\mathbf{x}^* - \mathbf{b} \rangle = 0$ *iff* $\langle \mathbf{c}, \mathbf{x}^* \rangle = \langle \mathbf{b}, \mathbf{y}^* \rangle$.

Proof.
Since $\mathbf{y}^*$ is the dual optimum we know that $A^t\mathbf{y}^* = \mathbf{c}$. Therefore,

$$\langle \mathbf{c}, \mathbf{x}^* \rangle = \langle A^t\mathbf{y}^*, \mathbf{x}^* \rangle = (\mathbf{y}^*)^t A\mathbf{x}^* = \langle \mathbf{y}^*, A\mathbf{x}^* \rangle \leq \langle \mathbf{y}^*, \mathbf{b} \rangle,$$

since $A\mathbf{x}^* \leq \mathbf{b}$. It is clear that $\langle \mathbf{c}, \mathbf{x}^* \rangle = \langle \mathbf{y}^*, \mathbf{b} \rangle$ iff the last inequality holds as an equality, i.e., iff $\langle \mathbf{y}^*, A\mathbf{x}^* \rangle = \langle \mathbf{y}^*, \mathbf{b} \rangle$. **Q.E.D.**

Remark: Note that complementary slackness does not claim that the *only* constraints satisfied as equalities by $\mathbf{x}^*$ are those corresponding to positive entries of $\mathbf{y}^*$ (see (16)). There can be constraints with corresponding entry in $\mathbf{y}^*$ as zero that are satisfied as equality by $\mathbf{x}^*$. This is illustrated, e.g., when $\mathbf{c}$ coincides with $\mathbf{a}_i$, in which case a complete facet can be optimum. ∎

In fact, we can say something more: The constraints corresponding to the positive entries of $\mathbf{y}^*$ are linearly independent. This is because if $\mathbf{c} \in \mathcal{C}(\mathbf{a}_1, \ldots, \mathbf{a}_m)$ then we know from the algorithm from Theorem 21 that $\mathbf{c}$ is in the cone generated by some lid vectors $\mathbf{a}_i$'s. Clearly, the constraints corresponding to positive entries of this linear combination are lid.

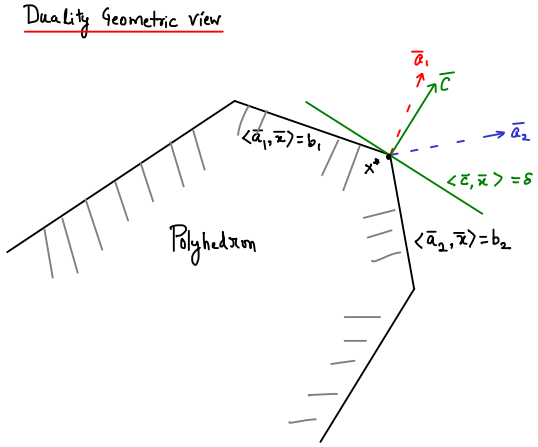


Figure 7

9.3 Applications

As applications we will show that various minimax theorems follow from strong duality. We show this for two theorems, namely max-flow min-cut and König's theorem.

9.4 Max-flow min-cut theorem

Recall the LP for maximizing flow in a network. Let $G = (V, E)$ be a directed graph with a capacity function $c : E \to \mathbb{R}_{\geq 0}$ and two distinguished vertices, a source vertex s and a target vertex t. For every vertex v, let $I(v)$ denote the set of edges directed to v and similarly define $O(v)$. A flow in G is a function $f : E \to \mathbb{R}$ satisfying the following conditions:

1. Non-negativity: for all $e \in E$, $f(e) \geq 0$,
2. Capacity constraints: for all $e \in E$, $f(e) \leq c(e)$, and
3. Flow conservation: for all $v \in V \setminus \{s, t\}$, $\sum_{e \in I(v)} f(e) = \sum_{e \in O(v)} f(e)$.

The size of the flow is defined as $\sum_{e \in O(s)} f(e) - \sum_{e \in I(s)} f(e)$, which measures the gradient of flow across the source node. The optimization problem is to obtain an f that maximizes the flow for G.

An **s-t cut** is a subset W of vertices such that $s \in W$, $t \notin W$. Let $I(W)$ denote the set of edges from $\overline{W}$ to W; similarly, define $O(W)$ as the set of edges going from W to $\overline{W}$. The capacity of an $s - t$ cut is $\sum_{e \in O(W)} c_e$. The **minimum capacity** of G is the minimum capacity over all $s - t$ cuts in G.

The max-flow min-cut theorem of Ford-Fulkerson states that the maximum flow in a directed graph is *equal to* its minimum capacity. We will prove this as a consequence of strong duality.

To get the LP corresponding to the flow problem, we introduce a variable x_e for every edge e. Then the set of constraints is of the form: for all $e \in E$, $0 \leq x_e \leq c(e)$, for all v except s and t we have $\sum_{e \in I(v)} x_e = \sum_{e \in O(v)} x_e$. The optimization function is simply $\sum_{e \in O(s)} x_e - \sum_{e \in I(s)} x_e$, which is clearly linear. Thus there is a constraint for every vertex, except s and t, and every edge. Let us write the primal in a more structured manner:

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in O(s)} x_e - \sum_{e \in I(s)} x_e \\ \text{s.t. for all } v \in V \setminus \{s,t\} \quad & \sum_{e \in I(v)} x_e - \sum_{e \in O(v)} x_e = 0, \\ & 0 \leq x_e \leq c_e. \end{aligned} \tag{17}$$

For every vertex $u \in V$ except s, t we introduce a variable p_u in the dual; since the constraints for vertices are equalities $p_u \in \mathbb{R}$. For every edge $e \in E$, we introduce a dual variable d_e; since the constraints for edges are $x_e \leq c_e$, these variables take non-negative values, i.e., $d_e \geq 0$. We now take a linear combination of the constraints in (17) wrt these variables and construct the dual. For every edge $(uv) \in E$ except those in $I(s) \cup O(s)$, we have the following constraint:

$$p_v - p_u + d_{uv} \geq 0.$$

For an edge $sv \in O(s)$ we have the constraint $p_v + d_{sv} - 1 \geq 0$, and for an edge $us \in I(s)$, we have the constraint $-p_u + d_{us} \leq 1$, i.e., $1 + p_u - d_{us} \geq 0$. The objective function clearly is $\sum_{e \in E} d_e c_e$. To summarize, the dual is the following:

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} d_e c_e \\ \text{s.t. for all } uv \in E \setminus (I(s) \cup O(s)), \quad & d_{uv} \geq p_u - p_v, \\ \text{for } sv \in O(s), \quad & p_v + d_{sv} - 1 \geq 0, \\ \text{for } us \in I(s), \quad & 1 + p_u - d_{us} \geq 0, \\ & d_e \geq 0, p_v \in \mathbb{R}. \end{aligned} \tag{18}$$

We claim that all $s-t$ cuts are feasible for this dual. We will then show an $s-t$ cut whose minimum capacity is equal to the maximum flow. The interpretation is that p_u is the potential stored at a vertex and d_{uv} is an upper bound on the drop in the potential as we move from u to v.

Consider an $s-t$ cut W. Let $p_u = 1$, for all $u \in W$ and zero otherwise. Similarly, let $d_{uv} = 1$, if $u \in W$ and $v \in \overline{W}$; otherwise, if both $u, v \in W$ or both are in $\overline{W}$ we set it to zero. It is easy to verify that these choices satisfy the constraints in (18). The value of the objective function in the dual is clearly the capacity of the cut. Thus by weak duality we have shown that maximum flow in a graph is smaller than its minimum capacity. Why are we not done? Suppose $\mathbf{x}^*$ is an optimum solution to the primal and $(\mathbf{p}^*, \mathbf{d}^*)$ is an optimal solution to the dual. Then as p_v are arbitrary reals, it is not immediately clear what is the cut defined by this optimum. Let define our $s-t$ cut $W^* := \{u \in V : p_u^* \geq 1\} \cup \{s\}$; clearly it cannot be empty. Even then, as d_e^* may take values greater than one, it is not clear why $\sum_{e \in E} d_e^* c_e$ should be the capacity of W^*. That is precisely what we show next. Then we will be done, because we would have shown that OPT_{Dual} is the capacity of W^*, and by strong duality we know $\text{OPT}_{\text{Dual}} = \text{OPT}_{\text{Primal}}$, i.e., the capacity of W^* is the maximum flow in the graph. By weak duality, it then follows that the capacity of W^* is also the minimum capacity of G.

LEMMA 31. *We show that* $OPT_{Dual} = \sum_{e \in E} d_e^* c_e = \sum_{e \in O(W^*)} c_e$.

Proof. We know by strong duality that

$$\sum_{e \in E} d_e^* c_e = \sum_{e \in O(s)} x_e^* - \sum_{e \in I(s)} x_e^*.$$

We will in fact show that the quantity on the RHS equals the capacity of W^*.

If $e = uv \in O(W^*)$ then we know that $p_u \geq 1 > p_v$. Therefore, $d_{uv} \geq p_u - p_v > 0$, and it follows from complementary slackness that the corresponding constraint in the primal is satisfied as equality by $\mathbf{x}^*$, i.e., $x_e^* = c_e$. Now consider an edge $e = uv \in I(W^*)$; again, we know from the choice of W^* that $p_v \geq 1 > p_u$. We claim that $x_e^* = 0$; if not, then from complementary slackness we know that $0 \leq d_{uv} = p_u - p_v < 0$, which is a contradiction. For the edges in $O(s) \cup I(s)$ a similar argument shows that $x_e^* = c_e$, for $e \in O(s)$, and $x_e^* = 0$, for $e \in I(s)$. To summarize, we have shown that $x_e^* = c_e$, for $e \in O(W^*)$, and $x_e^* = 0$, for $e \in I(W^*)$.

Consider the value of the flow given by $\mathbf{x}^*$:

$$\text{OPT}_{\text{Primal}} = \sum_{e \in O(s)} x_e - \sum_{e \in I(s)} x_e.$$

Since for all other vertices $v \in W^*$ this difference vanishes, the quantity above is equal to

$$\sum_{v \in W^*} \left(\sum_{e \in O(v)} x_e - \sum_{e \in I(v)} x_e \right).$$

For any edge e with both its vertices in W^* the term x_e is counted once as an outgoing edge and once as an incoming edge, and hence disappears from the sum above. Thus, the sum above is equal to

$$\sum_{e \in O(W^*)} x_e - \sum_{e \in I(W^*)} x_e.$$

Since $x_e^* = c_e$, for $e \in O(W^*)$, and $x_e^* = 0$, for $e \in I(W^*)$, it follows that the sum above is equal to $\sum_{e \in O(W^*)} c_e$, which is the capacity of W^*. Thus we have shown that the capacity of W^* is $\text{OPT}_{\text{Primal}}$, and by weak duality it follows that W^* is also optimum. **Q.E.D.**

9.5 König's Theorem

Given a bipartite graph $G = (X \cup Y, E)$, König's theorem states that the size of the maximum matching is equal to the minimum vertex cover. How can we formulate matching as an LP? Let's associate a variable x_e

with each edge e, which can take value in $\{0,1\}$. For every vertex v, exactly one of the x_e is equal to one amongst the edges e incident on v; this can be captured as $\sum_{e\in I(v)} x_e = 1$. Thus our LP is:

$$\begin{aligned} &\text{maximize } \sum_{e\in E} x_e \\ \text{s.t. for all } v \in X\cup Y,\ &\sum_{e\in I(v)} x_e \le 1, \\ &\text{and for all } e\in E,\ x_e \in \{0,1\}\,. \end{aligned} \tag{19}$$

A more succinct way to express this is to use the vertex-edge incidence matrix $A = [a_{ij}]$, where $a_{ij} = 1$ iff v_i is an endpoint of e_j. Thus every column of A has exactly two entries as 1. Using the incidence matrix, the LP above can be written as

$$\text{maximize } \sum_{e\in E} x_e \text{ s.t. } A\mathbf{x} \le \mathbf{1},\ \text{and for all } e\in E,\ x_e \in \{0,1\}\,.$$

Instead of restricting x_e to $\{0,1\}$ we allow it to take values as positive integers, since the constraint corresponding to a vertex will imply that x_e is either 0 or 1. Thus we finally get the following LP for finding the maximum matching:

$$\text{maximize } \sum_{e\in E} x_e \text{ s.t. } A\mathbf{x} \le \mathbf{1},\ \mathbf{x} \in \mathbb{Z}^m_{\ge 0}. \tag{20}$$

The LP above is called an **integer linear program** since the variable takes integral values. Note that so far we haven't used the fact that G is a bipartite graph crucially; the derivation above actually applies to any graph. However, the incidence matrix of a bipartite graph has a nice property that allows us to work with the LP above by relaxing the integrality condition by allowing the x_e's to vary over $\mathbb{R}$, or what is called the **LP-relaxation** of the ilp. Overlooking the integrality part, the corresponding dual would be

$$\text{minimize } \sum_{v\in X\cup Y} y_v \text{ s.t. } A^t\mathbf{y} \ge \mathbf{1},\ \mathbf{y} \in \mathbb{Z}^n_{\ge 0}. \tag{21}$$

This not entirely correct, since strictly speaking $\mathbf{y} \in \mathbb{R}^n_{\ge 0}$. The rows of the constraint $A^t\mathbf{y} \ge 1$ are of the form $y_u + y_v \ge 1$, for an edge $uv \in E$, and from the *integrality* of $\mathbf{y}$ it follows that at least one of the endpoints of an edge is picked by the optimum, i.e., one y_u of y_v is greater than 1. Note that since the y_u's are positive integers and we are minimizing their sum, we can assume that actually $y_u \in \{0,1\}$, since any larger value could be reduced while satisfying the constraints and decreasing the objective value. Thus all vertex covers are feasible for this lp, and the value of the objective function is the size of the vertex cover.

It is clear that the LP-relaxations of the two lps above are dual of each other. Suppose $\mathbf{x}^*$ and $\mathbf{y}^*$ are the optimum for the relaxed primal and dual lps. Can we recover Konig's theorem by applying strong duality? But now it is not clear what the value of the objective function means for two lps as we x^*_e's and y^*_u's can take fractional values. For the dual we can take the set $V := \{v : y_v \ge 1/2\}$ as our vertex cover, but the value of the objective function is not the size of V. This idea of using LP-relaxation apparently seems to fail. But so far we have not used bipartitness anywhere. We will now show that the vertices of the polyhedra defined by the constraints in (53) and (54) have integer coordinates, which means that the optimum for the LP-relaxations are integer vectors and are in fact solutions to the corresponding integral lps.

The special property that the incidence matrix A of a bipartite graph has is that it is **totally unimodular**, i.e., the determinant of *any submatrix* of A is in $\{-1,0,1\}$. In particular, the 1×1 determinants, namely the entries of A, are 0, 1, or -1. Such matrices have the following property:

LEMMA 32. *Given a $m\times n$ tum A and $\mathbf{b}\in\mathbb{Z}^n$, the polyhedra $P := \{\mathbf{x} : A\mathbf{x}\le \mathbf{b}, \mathbf{x}\ge 0\}$ is integral, i.e., the vertices of the polyhedra are in $\mathbb{Z}^n$.*

Proof. Assume $m > n$ (otherwise?). We know that the vertices of P are obtained as some n lid constraints being satisfied as equalities. Suppose $I \subseteq [m]$, $|I| = n$, is such a set of lid constraints and A_I the corresponding submatrix of A. Then the corresponding vertex is the solution to the equation $A_I\mathbf{x}_I = \mathbf{b}$. From Cramer's rule we know that the solution to this equation is of the form $\det(A'_I)/\det(A_I)$, where A'_I is obtained by

replacing some column of A_I by $\mathbf{b}$. Since A is tum, and A_I is a non-singular submatrix of A we know that $\det(A_I) = \pm 1$. **Q.E.D.**

We next prove by induction that the incidence matrix of A is tum; the base case is obvious as all entries are 0, 1. Let Q be an $k \times k$ submatrix of A. There are three cases to consider:

1. Q has a column that is all zero.
2. Q has a column that has exactly one 1.
3. All columns of Q have exactly two 1's. Partition the rows of Q into two sets: those corresponding to vertices in X and those in Y. Consider the k-dimensional vector $\mathbf{v}$ that has all 1's at the indices corresponding to vertices of X and -1's elsewhere. Then since every column has exactly two ones, it is easy to verify that the product $\mathbf{v}^t Q = \mathbf{0}$, which implies that the rows of Q are linearly dependent, i.e., $\det(Q) = 0$.

10 The Ellipsoid Method

We have seen that optimization version of LP, or LP-opt for short, can be solved in exponential time using simplex method. The decision version of LP-opt is defined as

$$\text{LP-Decision} := \{(A, \mathbf{b}, \mathbf{c}, \lambda) : \text{ there exists a } \mathbf{x} \text{ s.t. } A\mathbf{x} \leq \mathbf{b} \text{ and } \langle \mathbf{c}, \mathbf{x} \rangle \geq \lambda\}.$$

Then it is clear that LP-decision is in NP, since given $(A, \mathbf{b}, \mathbf{c}, \lambda)$ and a witness $\mathbf{x}$ we can easily verify whether the constraints hold in polynomial time and hence if $(A, \mathbf{b}, \mathbf{c}, \lambda) \in$ LP-Decision. How about deciding if $(A, \mathbf{b}, \mathbf{c}, \lambda) \notin$ LP-Decision? If an instance $(A, \mathbf{b}, \mathbf{c}, \lambda)$ not in LP-Decision then it means that $\max\langle \mathbf{c}, \mathbf{x} \rangle < \lambda$ for all $\mathbf{x}$ satisfying $A\mathbf{x} \leq \mathbf{b}$. But then from duality we know that there must exists a $\mathbf{y}$ such that $\langle \mathbf{b}, \mathbf{y} \rangle < \lambda$, where $\mathbf{y}$ satisfies $A^t\mathbf{y} \geq \mathbf{c}$. Thus if an instance does not belong to LP-Decision, then this is witnessed by the dual. Hence LP-Decision is in co-NP.

We have seen that LP can be solved in exponential time using simplex method. For a long time people tried various modifications of simplex-type algorithms to obtain a polynomial time algorithm for solving lps. However, it was Khachiyan in 1979, who gave an extension of an earlier algorithm of Shor-Yudin-Nemirovskii for non-linear programming, and obtained the first poly-time algorithm for solving lps. A fascinating account of Khachiyan's discovery in the west and its portrayal in the media is given in Lawler's article, "The Great Mathematical Sputnik of 1979."

The algorithm proposed by Khachiyan is actually an algorithm to check if a system of linear inequalities is feasible or not. We know that using such a feasibility-test we can solve for the optimum. Let us see how to do that.

¶3. Using Feasibility to Solve Optimality: Suppose we want to solve the LP in equational form maximize $\langle \mathbf{c}, \mathbf{x} \rangle$ over the set $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$. given a solver for testing feasibility or non-emptiness of a polyhedron $\{\mathbf{x}' : A'\mathbf{x}' \leq \mathbf{b}'\}$. Moreover, assume that the input of the LP (i.e., the entries of $A, \mathbf{b}, \mathbf{c}$) are integers of bit-length L. Then we know from Theorem 5 that if there is an optimum then a bfs attains that optimum. From Cramer's rule we can derive upper bounds on the abolute value of the rationals involved in a bfs in terms of the dimensions of A and L; suppose R is such an upper bound (basically an upper bound on subdeterminants of A with $\mathbf{b}$ using Hadamard's bound). Then we do the following:

1. To test if the original LP is feasible is straightforward using the feasibility-test.

2. To check if the original LP is bounded, we can call the feasibility-test on $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq R \cdot \mathbf{1}\}$. However, if this set is non-empty then it only means that the polyhedron in the original LP is unbounded, the LP itself may be bounded. To test boundedness, we need to know how large $\langle \mathbf{c}, \mathbf{x} \rangle$ can be over all bfs's. But as we know upper and lower bounds on the size of a bfs, we know an upper bound on the absolute value of the objective function over all bfs's; let R' be this upper bound (something like $n2^L R$), i.e., if $\mathbf{v}$ is a bfs of $A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$, then $|\langle \mathbf{c}, \mathbf{v} \rangle| \leq R'$. Thus to test boundedness we call the feasibility-test on $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0, \langle \mathbf{c}, \mathbf{x} \rangle > R'\}$ and $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0, \langle \mathbf{c}, \mathbf{x} \rangle < -R'\}$; if the test returns feasible, we know that the original lp is unbounded; otherwise, the lp is bounded and hence has an optimum.

3. Now we show how to find the optimum using feasibility-test. The basic idea is to do a binary search on the interval $[-R', R']$. We first call the feasibility-test on $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0, 0 \leq \langle \mathbf{c}, \mathbf{x} \rangle \leq R'\}$. If it returns non-empty then we call the feasibility-test on the interval $[R'/2, R']$; otherwise, we call the feasibility-test on $[-R', 0]$. In this way, we can narrow down the maximum value of the objective function on the polyhedron. When do we stop? Suppose δ is the maximum value of the objective function on the feasible region; from the bound R on the rationals in a bfs and the bound on the entries of $\mathbf{c}$, we can derive upper bound on the size of the numerator and denominator of δ, say this bound is R'' (basically, n times an upper bound on the lcm of the denominators, so something like nR^n). The idea is that we are trying to minimize $|\langle \mathbf{c}, \mathbf{x} \rangle - \delta|$, where $\mathbf{x}$ has rational coordinates bounded by R in absolute value. If this quantity is not zero then it is at least some quantity dependent on R'' (more precisely, $1/(R'')^2$), say it is greater than $2^{-\tau}$, where $\tau > 0$. After i iterations, we have narrowed our estimate of δ to an interval of width $2^{-i+1}R'$, that is we have an $\mathbf{x}$ such that $|\langle \mathbf{x}, \mathbf{c} \rangle - \delta| \leq 2R'/2^i$. Thus, if i is sufficiently large, say $i > 2 + \tau + \log R'$, then we are sure that the unique rational number, with numerator and denominarotr bounded by R'', closest to $\langle \mathbf{c}, \mathbf{x} \rangle$ is the value δ.

4. Though the binary search yields the value of the objective function, we still have to find the optimum. Since the optimum is a bfs, we only have to figure out which of the m columns of A give us a bfs such that the value of the objective function at $\mathbf{v}$ is δ. The idea is to drop one variable, say starting from x_n, at a time and solve the lower dimensional lp. If the value of the objective function remains the same, then the corresponding column n can be dropped from A; otherwise, the index n forms part of a feasible basis, so we keep it and proceed recursively on the remaining set of columns; we stop when no more columns can be dropped from A; the resulting set of indices forms a feasible basis and the corresponding bfs attains the value δ.

The approach above is a bit ugly, since it depends on the bit-lengths of the numbers. A better approach is to use duality. From strong duality we know that if the primal has an optimum then the dual also has an optimum; moreover, the objective function for the two take the same value. That is, if there exists an $\mathbf{x}^* \geq 0$ such that $A\mathbf{x}^* \leq \mathbf{b}$, and a $\mathbf{y}^* \geq 0$ such that $A^t\mathbf{y}^* \geq \mathbf{c}$, and $\langle \mathbf{c}, \mathbf{x}^* \rangle = \langle \mathbf{b}, \mathbf{y}^* \rangle$ then $\mathbf{x}^*$ and $\mathbf{y}^*$ are the optimum for the respective primal and dual lp. Thus given a feasibility-test we can give it all these constraints:

$$A\mathbf{x} \leq \mathbf{b},\ A^t\mathbf{y} \geq \mathbf{c},\ \mathbf{x} \geq 0, \mathbf{y} \geq 0,\ \langle \mathbf{c}, \mathbf{x} \rangle = \langle \mathbf{b}, \mathbf{y} \rangle.$$

If we have a feasible solution $(\mathbf{x}^*, \mathbf{y}^*)$ to this system, then we clearly have an optimum solution to the original LP. So the question of finding an optimal solution to an lp reduces to checking the feasibility of a system of inequalities $P := \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}\} \subseteq \mathbb{R}^n$. How do we do that?

The main idea behind Khachiyan's algorithm is a geometric form of binary search. Suppose we know that P is bounded and is contained in a hypercube $\mathcal{B}$. Our aim is to either find a point inside P or declare that P is empty. Suppose we know an ϵ such that if $P \neq \emptyset$ then $\mathrm{Vol}(P) \geq 2^{-\epsilon}$. The algorithm proceeds as follows: we first check if the center of $\mathcal{B}$ is contained in P; if it is then we know that $P \neq \emptyset$; otherwise, we know that the center violates a hyperplane h of P; then $P \subseteq \mathcal{B} \cap h$ and we construct a hyper-rectangle $\mathcal{B}'$ containing this intersection and proceed recursively on it. We terminate when we either find a point inside P or the volume of the set containing P is smaller than $2^{-\epsilon}$, in which case we declare that $P = \emptyset$. Well this is the outline of the algorithm. The actual algorithm differs in many ways, one of which is that instead of hyper-rectangles the algorithm works with ellipsoids, and hence is also called the **ellipsoid algorithm**. The overview also has a drawback: If the affine dimension of P is smaller than n the dimension of the ambient space then $\mathrm{Vol}(P) = 0$, even though $P \neq \emptyset$. Thus from now on we will make the following two assumptions:

A1. P is bounded, i.e., it is contained in $B(\mathbf{0}, R)$ the hypersphere centered at origin with radius $R < \infty$, and

A2. P is full-dimensional, i.e., its affine dimension is n.

We will later see how to get rid of these assumptions. Even with these assumptions, we still need to find a lower bound on $\mathrm{Vol}(P)$ if it is not empty. We first start with the requisite background for working with ellipsoids. What are ellipsoids?

10.1 Ellipsoids – some fundamentals

An **ellipsoid** is an affine transformation of the unit ball $B(\mathbf{0}, 1)$ in $\mathbb{R}^n$, i.e.,

$$\mathcal{E} := \{A\mathbf{x} + \mathbf{t} : \langle \mathbf{x}, \mathbf{x} \rangle \leq 1, \mathbf{t} \in \mathbb{R}^n, \text{ and } A \text{ is non-singular}\}. \tag{22}$$

The point $\mathbf{t}$ is called the center of $\mathcal{E}$. There are many ways to define $\mathcal{E}$. If $\mathbf{y} \in \mathbb{R}^n$, then we know that there is a unique $\mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{y} = A\mathbf{x} + \mathbf{t}$, which is equivalent to saying that $\mathbf{x} = A^{-1}(\mathbf{y} - \mathbf{t})$. Now, this $\mathbf{x}$ is in the unit ball iff the norm of $A^{-1}(\mathbf{y} - \mathbf{t})$ does not exceed one. Thus an equivalent description of an ellipsoid is

$$\mathcal{E} := \left\{\mathbf{y} \in \mathbb{R}^n : \langle A^{-1}(\mathbf{y} - \mathbf{t}), A^{-1}(\mathbf{y} - \mathbf{t}) \rangle \leq 1, \mathbf{t} \in \mathbb{R}^n, \text{ and } A \text{ is non-singular}\right\}. \tag{23}$$

Opening the inner product notation, we obtain that

$$\mathcal{E} := \left\{\mathbf{y} \in \mathbb{R}^n : (\mathbf{y} - \mathbf{t})^t (A^{-1})^t A^{-1} (\mathbf{y} - \mathbf{t}) \leq 1, \mathbf{t} \in \mathbb{R}^n, \text{ and } A \text{ is non-singular}\right\}. \tag{24}$$

Note that the matrix $Q := A^tA$ is a real symmetric matrix. Moreover, it is **positive definite**, i.e., all its eigenvalues are positive. There are other equivalent ways to define a positive definite matrix:

THEOREM 33. *The following three statements are equivalent for a real symmetric matrix Q.*

1. *Q is psd.*
2. *$Q = B^tB$ for some matrix B.*
3. *For all $\mathbf{x} \in \mathbb{R}^n$, the quadratic form $\mathbf{x}^tQ\mathbf{x} \geq 0$.*

If Q is positive definite then B in part (ii) is non-singular and the quadratic form in (iii) vanishes only at origin.

Proof. Since Q is real symmetric, we know that its set of eigenvectors form an orthonormal set U, and that $Q = UDU^t$, where D is a diagonal matrix containing eigenvalues of Q.

1. ($1 \implies 2$) Since D has non-negative entries, we can define the diagonal matrix $\sqrt{D} = [\sqrt{\lambda_i}]_i$ uniquely. Therefore, $Q = U\sqrt{D}(\sqrt{D})^tU^t = U\sqrt{D}(U\sqrt{D})^t$. Note $U\sqrt{D}$ is non-singular iff no eigenvalue is zero, i.e., Q is positive definite.
2. ($2 \implies 3$) If $Q = B^tB$ then for all $\mathbf{x}$, the quadratic form
$$\mathbf{x}^tQ\mathbf{x} = \mathbf{x}^tB^tB\mathbf{x} = \|B\mathbf{x}\|^2 \geq 0.$$
3. ($3 \implies 1$) Express $\mathbf{x}$ in terms of the eigenvectors of U; i.e., let $\mathbf{x} = U\mathbf{y}$. Then we get
$$\mathbf{x}^tQ\mathbf{x} = \mathbf{y}^tU^tQU\mathbf{y} = \mathbf{y}^tD\mathbf{y} = \sum_{i=1}^{n} \lambda_i y_i^2.$$
In particular, if $\mathbf{x}$ is an eigenvector corresponding to λ_i then $\mathbf{y}$ is the vector with all zeros except a 1 at the ith place. In which case, $\mathbf{x}^tQ\mathbf{x} = \lambda_i \geq 0$.

The last proof actually reveals the principal axes of the ellipsoid, namely the columns of U. Representing $\mathbf{x}$ as $U\mathbf{y}$ is the same as transforming the standard axis by U, i.e., performing a sort of rotation that makes the ellipsoid axis-aligned and then we have the standard quadratic form equation of the ellipsoid with no mixed terms. Note that the length of the principal axes is the inverse of the square-root of the eigenvalues of Q. **Q.E.D.**

In particular, (24) can be modified to the following definition:

$$\mathcal{E} := \left\{\mathbf{y} \in \mathbb{R}^n : (\mathbf{y} - \mathbf{t})^tQ^{-1}(\mathbf{y} - \mathbf{t}) \leq 1, \mathbf{t} \in \mathbb{R}^n, \text{ and } Q \text{ is positive definite}\right\}. \tag{25}$$

According to this definition an ellipsoid is uniquely determined by its center $\mathbf{t}$ and the associated positive definite matrix Q and will be represented as $\mathcal{E}(\mathbf{t}, Q)$. What is the volume of this ellipsoid? We claim the following:

$$\text{Vol}(\mathcal{E}(\mathbf{t}, Q)) = \sqrt{\det(Q)}\text{Vol}(B(\mathbf{0}, 1)) = |\det(A)|\text{Vol}(B(\mathbf{0}, 1)),$$

where A is the non-singular map that takes the unit ball to the ellipsoid (note that translation does not affect the volume), or equivalently the square-root of Q. This follows from the observation that A takes the unit hypercube to the parallelepiped formed by the columns of A; hence the unit volume element is scaled by $|\det(A)|$ under the linear transformation by A. Note that this claim generalizes the observation in two and three dimensions: the area of the ellipse $x^2/a^2 + y^2/b^2 \leq 1$ is πab and the volume of the ellipsoid $x^2/a^2 + y^2/b^2 + z^2/c^2 \leq 1$ is $(4/3)\pi abc$. In general, if two sets $X, Y \subseteq \mathbb{R}^2$ are such that $Y = AX$, for some non-singular transformation A, then their volumes (if they are measurable) are related by $\text{Vol}(Y) = |\det(A)|\text{Vol}(X)$. Modulo some technical details, we are now in a position to describe the ellipsoid algorithm in some technical detail.

The algorithm maintains a sequence of ellipsoids $\mathcal{E}_i := \mathcal{E}(\mathbf{t}_i, Q_i)$, with the invariant that $P \subseteq \mathcal{E}_i$; initially, since P is bounded, $\mathcal{E}_0 = B(\mathbf{0}, R)$. At the ith step we do the following: If $\mathbf{t}_i \in P$ then we output feasible; otherwise, we know that there must be a constraint $\langle \mathbf{a}_j, \mathbf{x}\rangle \leq b_j$ that is violated by $\mathbf{t}_i$; $\mathcal{E}_{i+1}$ is the ellipsoid with the smallest volume containing $\mathcal{E}_i \cap \{\mathbf{x} : \langle \mathbf{a}_j, \mathbf{x}\rangle \leq \langle \mathbf{a}_j, \mathbf{t}_i\rangle\}$. We claim that the volume of $\mathcal{E}_{i+1}$ is a certain fraction smaller than $\mathcal{E}_i$. Hence after i-steps the volume has gone down geometrically. If we know a lower bound on $\text{Vol}(P)$, assuming it is non-empty, and if at any iteration volume of $\mathcal{E}_i$ is smaller than this bound then we know that $P = \emptyset$. Thus the algorithm either finds a point inside P or detects that P is empty. We need some more technical results to give the full details. How do we construct the next ellipsoid? What is a lower bound on $\text{Vol}(P)$? Let us start with the latter.

¶4. Lower bound on volume By assumption A2, we know that P is full-dimensional, i.e., its dimension is n. This means that there must be $n+1$ affinely independent points in P; moreover, we can takes these points to be vertices of P. Suppose $\mathbf{p}_0, \ldots, \mathbf{p}_n$ are $n+1$ such points. Then we claim the following:

LEMMA 34. *Given $n+1$ affinely independent points $\mathbf{p}_0, \ldots, \mathbf{p}_n$ the volume of their convex hull is*

$$\mathit{Vol}(\mathit{CH}(\mathbf{p}_0, \ldots, \mathbf{p}_n)) = \frac{1}{n!} \det \begin{bmatrix} \mathbf{p}_1 - \mathbf{p}_0 & \mathbf{p}_2 - \mathbf{p}_0 & \cdots & \mathbf{p}_n - \mathbf{p}_0 \end{bmatrix} = \frac{1}{n!} \det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix}.$$

Proof. The proof is actually a combinatorial argument at heart. We know that the convex hull of the $n+1$ points is a simplex. Moreover, this simplex is obtained by the linear transformation T that maps the identity matrix to the vectors $\mathbf{p}_i - \mathbf{p}_0$, $i = 1, \ldots, n$, followed by a translation by $\mathbf{p}_0$. Since translations do not affect the volume, and the n vectors $\mathbf{p}_i - \mathbf{p}_0$ are linearly independent, we obtain that

$$\mathrm{Vol}(\mathrm{CH}(\mathbf{p}_0, \ldots, \mathbf{p}_n)) = \mathrm{Vol}(\Delta) \cdot \det \begin{bmatrix} \mathbf{p}_1 - \mathbf{p}_0 & \mathbf{p}_2 - \mathbf{p}_0 & \cdots & \mathbf{p}_n - \mathbf{p}_0 \end{bmatrix}$$

where Δ is the simplex with the origin and e_i's as its vertices. What is the volume of Δ? Consider the $n+1$ affinely independent points

$$(\mathbf{0}, (1,0,0,\ldots,0), (1,1,0,\ldots,0), (1,1,1,\ldots,0), \ldots, (1,1,1,\ldots,1)).$$

Then it is clear that the simplex Δ' corresponding to them is obtained from Δ by the non-singular linear transformation:

$$\Delta' = \Delta \cdot \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Therefore, $\mathrm{Vol}(\Delta) = \mathrm{Vol}(\Delta')$. We now claim that the latter volume is $1/n!$. A point $\mathbf{x} \in \Delta'$ is a convex combination of the $n+1$ points, say the combination is given by $\alpha_0, \ldots, \alpha_n$. Then it follows that

$$x_1 = \sum_{i=1}^{n} \alpha_i, x_2 = \sum_{i=2}^{n} \alpha_i, , \ldots, x_n = \alpha_n.$$

This implies that

$$1 \geq x_1 \geq x_2 \geq x_3 \geq \cdots \geq x_n \geq 0,$$

i.e., the coordinates of $\mathbf{x}$ have a certain ordering on them. The converse is also true, namely, any $\mathbf{x}$ that has the ordering above belongs to Δ'; take $\alpha_n := x_n$, $\alpha_{n-1} := x_{n-1} - x_n$ and so on $\alpha_1 = x_1 - x_2$, and $\alpha_0 := 1 - \sum_{i=1}^{n} \alpha_i$. Thus Δ' is uniquely associated with an ordering on the coordinates of $\mathbf{x}$. Clearly, Δ' belongs to the unit hypercube $\{\mathbf{x} : 0 \leq x_i \leq 1\}$. Moreover, every point of the unit hypercube has some ordering on its coordinates and hence belongs to a simplex congruent to Δ'. Since there are $n!$ orderings on the coordinates, there are $n!$ many copies of Δ' in the unit hypercube, which implies that $\mathrm{Vol}(\Delta') = 1/n!$. **Q.E.D.**

We now come to the most important ingredient of the algorithm: constructing the $(i+1)$th ellipsoid at the i-th iteration. Recall that the problem is the following: given $\mathcal{E}(\mathbf{t}, Q)$ and a vector $\mathbf{a}$, we wanted to find an ellipse that contains the intersection of $\mathcal{E}(\mathbf{t}, Q)$ with the halfspace $\{\mathbf{x} : \langle \mathbf{a}, \mathbf{x} - \mathbf{t} \rangle \leq 0\}$. The hyperplane passes through the center $\mathbf{t}$ and bisects the ellipsoid into two halves. We are interested in constructing an ellipsoid containing one such half. We will start with a simpler setting where we work with the unit ball and the half plane $x_n \geq 0$, and then reduce the general case to this simpler case.

Suppose we are given the unit ball $\mathbb{B} := B(\mathbf{0}, 1)$ and we want to find an ellipsoid $\mathcal{E} := \mathcal{E}(\mathbf{t}, Q)$ with the least volume that contains the half-ball $\mathcal{H} := \mathbb{B} \cap \{\mathbf{x} : x_n \geq 0\}$. Notice the shape of $\mathcal{H}$ – it is symmetric about the x_n-axis, so the center should be some point on the x_n axis, and compared to the ball it is squeezed along this axis, and relatively wide orthogonal to the axis. So the idea is to lift the unit ball to some point on the

x_n-axis and try to stretch or squeeze it to contain $\mathcal{H}$. More precisely, suppose $\mathbf{t} := (0, 0, 0, \ldots, \tau)$ and Q^{-1} is the diagonal matrix that scales x_n by $1/\beta$ and all the remaining coordinates by $1/\alpha$. Our aim is to find appropriate values of τ, α, β such that $\det(Q) = \alpha^{n-1}\beta$ is minimized, while ensuring that $\mathcal{H} \in \mathcal{E}(\mathbf{t}, Q)$. To ensure this last constraint, we must ensure that the vector $e_n = (0, 0, \ldots, 1)$ must be in $\mathcal{E}$; moreover, it must be on the boundary of $\mathcal{E}$, i.e., it should satisfy the following

$$(e_n - \mathbf{t})^t Q^{-1} (e_n - \mathbf{t}) = 1,$$

which is equivalent to $(1 - \tau)^2 = \beta$. Similarly, the points on the intersection of the hyperplane $x_n = 0$ and $\mathbb{B}$ must also be on the boundary of $\mathcal{E}$. If $(\mathbf{x}, 0)$ is such a point (i.e., $\|\mathbf{x}\| = 1$) then we must have

$$((\mathbf{x}, 0) - \mathbf{t})^t Q^{-1} ((\mathbf{x}, 0) - \mathbf{t}) = 1,$$

which is equivalent to

$$\begin{bmatrix} \mathbf{x} & -\tau \end{bmatrix} \begin{bmatrix} 1/\alpha & 0 & 0 & \cdots & 0 \\ 0 & 1/\alpha & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1/\alpha & 0 \\ 0 & 0 & 0 & \cdots & 1/\beta \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -\tau \end{bmatrix} = 1.$$

But as $\|\mathbf{x}\| = 1$ this means that

$$\frac{1}{\alpha} + \frac{\tau^2}{\beta} = 1.$$

Substituting $\beta = (1 - \tau)^2$, we obtain that $\alpha = (1 - \tau)^2/(1 - 2\tau)$. Now our aim is to minimize

$$F(\tau) := \alpha^{n-1}\beta = \left(\frac{(1-\tau)^2}{(1-2\tau)}\right)^{n-1} (1 - \tau)^2 = \frac{(1-\tau)^{2n}}{(1-2\tau)^{n-1}}.$$

Differentiating wrt τ we obtain that

$$F'(\tau) = (-2n(1 - 2\tau) + 2(1 - \tau)(n - 1)) \frac{(1-\tau)^{2n-1}}{(1-2\tau)^n}.$$

Solving for $F'(\tau) = 0$ we get that $\tau = 1/(n + 1)$, which gives us that $\alpha = n^2/(n^2 - 1)$ and $\beta = n^2/(n + 1)^2$. Thus

$$\det(Q) = \alpha^{n-1}\beta = \left(1 + \frac{1}{n^2 - 1}\right)^{n-1} \left(1 - \frac{1}{(n+1)}\right)^2.$$

Using the inequality that $1 + x \leq e^x$, for all $x \in \mathbb{R}$, it follows that

$$\det(Q) \leq e^{\frac{1}{n+1}} e^{\frac{-2}{n+1}} = e^{\frac{-1}{n+1}}.$$

Therefore,

$$\mathrm{Vol}(\mathcal{E}) = \sqrt{\det(Q)}\mathrm{Vol}(\mathbb{B}) \leq e^{\frac{-1}{2(n+1)}} \mathrm{Vol}(\mathbb{B}). \tag{26}$$

To summarize, we have shown that the half-ball $\mathcal{H}$ obtained by intersecting $\mathbb{B}$ with the halfspace $\{\mathbf{x} : x_n \geq 0\}$ is contained in the ellipsoid $\mathcal{E}(\mathbf{e}_n/(n + 1), Q)$, where Q is a diagonal matrix with the first $(n - 1)$ diagonal entries as $n^2/(n^2 - 1)$, and the last diagonal entry as $(n/(n + 1))^2$.

As mentioned earlier, the general case can be reduced to the special setting above. Let $\mathcal{E} = \mathcal{E}(\mathbf{t}, Q)$ and a vector $\mathbf{a}$, we wanted to find an ellipse that contains the intersection of $\mathcal{E}$ with the halfspace $\{\mathbf{x} : \langle -\mathbf{a}, \mathbf{y} - \mathbf{t}\rangle \geq 0\}$. Let this convex set be denoted by $\mathcal{E}^-$. We know that $\mathcal{E}$ is obtained from $\mathbb{B}$ by the affine transformation $A\mathbf{x} + \mathbf{t}$, where $Q = A^t A$. So the idea is to take the inverse of this transformation, go back to the unit ball, use the results derived above, and come back to the ellipsoid. When we take the inverse, $\mathcal{E}^-$ is mapped to some half-ball obtained from $\mathbb{B}$ by cutting it with a hyperplane h through the origin. This hyperplane corresponds to $\{\mathbf{y} : \langle -\mathbf{a}, \mathbf{y} - \mathbf{t}\rangle = 0\}$ in the original setting. We have to figure the vector defining h. Note

that $\mathbf{y} - \mathbf{t} = A\mathbf{x}$, for some vector $\mathbf{x}$ in the domain of the affine transformation (we use $\mathbf{x}$'s for the domain and $\mathbf{y}$'s for the range). Thus

$$0 = \langle -\mathbf{a}, \mathbf{y} - \mathbf{t} \rangle = \langle -\mathbf{a}, A\mathbf{x} \rangle = \langle -A^t\mathbf{a}, \mathbf{x} \rangle,$$

Thus the vector defining the hyperplane h is the unit vector in the direction $-A^t\mathbf{a}$, i.e., $\mathbf{z} := -A^t\mathbf{a}/\|A^t\mathbf{a}\|$ and the halfspace corresponding to $x_n \geq 0$ is $\{\mathbf{x} : \langle -A^t\mathbf{a}, \mathbf{x} \rangle \geq 0\}$. So the center has to move a fraction $1/(n+1)$ in the direction of this unit vector, and then we apply the affine transformation to get the center for $\mathcal{E}'$, namely

$$\mathbf{t}' := \mathbf{t} + \frac{1}{n+1} A\mathbf{z} = \mathbf{t} - \frac{1}{n+1} \frac{Q^t\mathbf{a}}{|\mathbf{a}^t Q \mathbf{a}|}.$$

The corresponding matrix Q' can be obtained similarly. What is crucial is the following:

$$\text{Vol}(\mathcal{E}') = \sqrt{\det(Q)}\text{Vol}(\mathcal{E}'')$$

where $\mathcal{E}''$ is the ellipsoid containing the half ball. From (26) we know that $\text{Vol}(\mathcal{E}'') \leq e^{-1/(2n+2)}\text{Vol}(\mathbb{B})$. Therefore,

$$\text{Vol}(\mathcal{E}') \leq \sqrt{\det(Q)} e^{-1/(2n+2)} \text{Vol}(\mathbb{B}).$$

But $\text{Vol}(\mathcal{E}) = \sqrt{\det(Q)}\text{Vol}(\mathbb{B})$. Therefore,

$$\frac{\text{Vol}(\mathcal{E}')}{\text{Vol}(\mathcal{E})} \leq e^{-1/(2n+2)}. \tag{27}$$

To summarize, we have shown the following:

LEMMA 35. *Given $\mathcal{E}(\mathbf{t}, Q)$ and a halfspace $\{\mathbf{x} : \langle \mathbf{a}, \mathbf{y} - \mathbf{t} \rangle \leq 0\}$ their intersection is contained in the ellipsoid $\mathcal{E}(\mathbf{t}', Q')$ where*

$$\mathbf{t}' := \mathbf{t} - \frac{1}{n+1}\mathbf{v},$$

$$Q' := \frac{n^2}{n^2-1}\left(Q - \frac{2}{n+1}(\mathbf{v} \cdot \mathbf{v}^t)\right),$$

and $\mathbf{v} := Q\mathbf{a}/\|\mathbf{a}^t Q \mathbf{a}\|$.

We can now describe the algorithm in some more detail, and analyze its running time.

```
EM(E, P)
INPUT: An ellipsoid E(Q, t) containing the polyhedron P given by its inequalities.
OUTPUT: Either a point in P or that P is empty.
1. If Vol(E) is smaller than the lower bound in Lemma 34 then return P = ∅.
2. Else if t ∈ P return t.
3. Else
       Let ⟨a, x⟩ = b be a hyperplane violated by t.
       Construct the ellipsoid E' containing E(Q, t) ∩ {x : ⟨a, x⟩ ≤ ⟨a, t⟩},
       i.e., the ellipsoid corresponding to the central-cut, as given in Lemma 35.
       Return EM(E').
```

Testing whether the center is in P takes $O(mn)$ time. The depth of the recursion tree is bounded as follows: After i iterations the volume of $\mathcal{E}_i$ is smaller than e^{-2in} times the volume of the initial ball, which is bounded by the volume of the cube of edge length $2R$ containing it, namely $(2R)^n$. If the volume falls below the bound given in Lemma 34 then we can terminate. It can be shown that the number of iterations are bounded by roughly $O(n^2 \log R)$. The construction of the ellipsoid takes $O(n^2)$ time. Therefore, the overall complexity is $O(mn^3 \log R)$ in the real ram model. Note the dependency of the running time on the size of the input. This dependence makes the ellipsoid method weakly polynomial time. However, there is one more problem with our description of the algorithm: the computation involves working with irrational numbers. In the Turing machine model this is not possible – we have to determine the exact precision requirements from the intermediate computations. This means that the center of the ellispsoids are approximate by rational

vectors and to account for this perturbation the resulting ellipsoids are enlarged to ensure that the the containement of the smaller regions is still maintained. The resulting algorithm is called the **central-cut ellipsoid method** (central because the cut passes through the center of the ellipsoid, unlike the deep-cut or shallow cut). In the framework of Turing machine model, or finite precision arithmetic, the output of the algorithm is as follows. We describe it more generally in terms of a separating oracle for a compact convex set $\mathcal{K}$:

THEOREM 36. *The central-cut ellipsoid method solves the following problem. The input to the problem is a rational number $\epsilon > 0$ and a set convex set $\mathcal{K}$ contained in $B(0, R)$ given by a separation oracle $\texttt{Sep}_K$ that takes as input a $\mathbf{y} \in \mathbb{Q}^n$ and a positive rational number δ and either asserts that $\mathbf{y} \in \mathcal{K} + \delta$ or finds a vector $\mathbf{c}$ of unit norm such that $\langle \mathbf{c}, \mathbf{x} \rangle \leq \langle \mathbf{c}, \mathbf{y} \rangle + \delta$, for all $x \in \mathcal{K}$. The output of the algorithm is one of the following:*

1. *either a vector $\mathbf{a} \in \mathcal{K} + \epsilon$, or*
2. *An ellispod $\mathcal{E}(Q, \mathbf{t})$ such that $\mathcal{K} \subseteq \mathcal{E}(Q, \mathbf{t})$ and $Vol(\mathcal{E}(Q, \mathbf{t})) < \epsilon$.*

10.2 Removing the assumptions

Let $A, \mathbf{b}$ have integral entries. Consider the augmented matrix $[A\ \mathbf{b}]$ and let Δ be the maximum absolute value of the determinant over all submatrices of this matrix. To achieve boundedness, we know that all bfs of $A\mathbf{x} \leq \mathbf{b}$ have infinity norm smaller than Δ. Thus to achieve boundedness we can consider the hypercube $\{\mathbf{x} : |x_i| \leq \Delta, i = 1, \ldots, n\}$.

The following lemma tells us how to ensure full-dimensionality.

LEMMA 37. *Given $A \in \mathbb{Q}^{m \times n}$ and $\mathbf{b} \in \mathbb{Q}^m$, let R be an upper bound on the size of the corresponding bfs's. Let $P := \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}\}$ and $P_\epsilon := \{\mathbf{x} : A\mathbf{x} < \mathbf{b} + \epsilon \cdot \mathbf{1}\}$. Then there exists an $\epsilon > 0$ such that $P = \emptyset$ iff $P_\epsilon = \emptyset$.*

Proof. Since $P \subset P_\epsilon$, for all $\epsilon > 0$, one direction is easy, namely if $P_\epsilon = \emptyset$ then $P = \emptyset$.

For the converse, we will use Corollary 27: If $P = \emptyset$ then we know that there is a $\mathbf{y} \geq 0$ such that $\mathbf{y}^t A = 0$ but $\langle \mathbf{y}, \mathbf{b} \rangle < 0$; let's us scale $\mathbf{y}$ such that $\langle \mathbf{y}, \mathbf{b} \rangle = -1$. Intuitively, we know that there is a small enough ϵ perturbation of $\mathbf{b}$ such that $\langle \mathbf{y}, \mathbf{b} + \epsilon \cdot \mathbf{1} \rangle < 0$. **Q.E.D.**

11 Weak problems and their reductions

The introdcution of errors in our computation forces us to revise the definition of our optimization problems. In this section, we would like to optimize a linear function over a closed compace convex set $\mathcal{K}$. There is another reason to do so – the ellipsoid method revelas that if we have an oracle that answers a separation query for $\mathcal{K}$, i.e., given $\mathbf{y}$ either says $\mathbf{y} \in \mathcal{K}$ or provides a separating hyperplane (of poly bit-size in the input) then we can solve the linear optimization problem over $\mathcal{K}$. In this section, we consider similar reductions. However, as $\mathcal{K}$ is an arbitarry convex set we have to relax our assumptions on the oracles representing it (basically our queries are never exact near the boundary of $\mathcal{K}$ and there is a certain margin of error). We denote this by blowing the boundary of $\mathcal{K}$: given $\epsilon > 0$, a point $\mathbf{y} \in \mathcal{K} + \epsilon$ if it is within ϵ distance of a point in $\mathcal{K}$, and a point $\mathbf{y} \in \mathcal{K} - \epsilon$ if it is in the interior of $\mathcal{K}$ and there is no boundary point at a distance ϵ from $\mathbf{y}$.

Consider the following three problems, which are weak versions of corresponding strong problems:

WEAK OPTIMIZATION PROBLEM (WOPT)
INPUT: A vectpr $\mathbf{c} \in \mathbb{Q}^n$ and $\epsilon \in \mathbb{Q}_{>0}$.
OUTPUT:
1. Either find a vector $\mathbf{y} \in \mathbb{Q}^n$ such that $\mathbf{y} \in \mathcal{K} + \epsilon$ and for all $\mathbf{x} \in \mathcal{K} - \epsilon$, $\langle \mathbf{c}, \mathbf{x} \rangle \leq \langle \mathbf{c}, \mathbf{y} \rangle + \epsilon$, or
2. assert that $\mathcal{K} - \epsilon$ is empty.

WEAK VIOLATION PROBLEM (WVIOL)
INPUT: A vector $\mathbf{c} \in \mathbb{Q}^n$, $\gamma \in \mathbb{Q}$ and $\epsilon \in \mathbb{Q}_{>0}$.
OUTPUT:
1. Either assert that for all $\mathbf{x} \in \mathcal{K} - \epsilon$, $\langle \mathbf{c}, \mathbf{x} \rangle \leq \gamma + \epsilon$, or
2. find a vector $\mathbf{y} \in \mathbb{Q}^n$ such that $\mathbf{y} \in \mathcal{K} + \epsilon$ and $\langle \mathbf{c}, \mathbf{y} \rangle \geq \gamma - \epsilon$.

WEAK SEPARATION PROBLEM (WSEP)
INPUT: A vector $\mathbf{y} \in \mathbb{Q}^n$, and $\delta \in \mathbb{Q}_{>0}$.
OUTPUT:
1. Either assert that $\mathbf{y} \in \mathcal{K} + \delta$, or
2. find a vector $\mathbf{c} \in \mathbb{Q}^n$ such that $\|\mathbf{c}\| = 1$ and for all $\mathbf{x} \in \mathcal{K} - \delta$, $\langle \mathbf{c}, \mathbf{x} \rangle \leq \langle \mathbf{c}, \mathbf{y} \rangle + \delta$.

WEAK MEMBERSHIP PROBLEM (WMEM)
INPUT: A vector $\mathbf{y} \in \mathbb{Q}^n$ and $\epsilon \in \mathbb{Q}_{>0}$.
OUTPUT:
1. Either assert that $\mathbf{y} \in \mathcal{K} + \epsilon$ or
2. assert that $\mathbf{y} \notin \mathcal{K} - \epsilon$.

One of the most surprising results is that WOPT can be solved in poly time using an oracle for WMEM, i.e., just by querying polynomially many points for membership it is possible to find the opt. However, we start witht another intersting result, namely, that WVIOL can be solved in poly time using an oracle for WSEP (along with an upper bound R on a ball containing $\mathcal{K}$).

11.1 Solving Weak Violation given Weak Separation

12 Interior Point Methods (IPM)

So far we have seen two algorithms for solving LPs. The simplex method walks on the boundary of the feasible region until it reaches the optimum. The ellipsoid algorithm starts with an upper and lower bound on the feasible region, and up until the very end the algorithm is checking points outside the feasible region. As the name suggests, the interior point methods take a different approach – they always work with points inside the feasible region, and try to approach the optimum solution from within the polyhedra, and only at the very end do they move from the interior to the boundary. The history of interior point methods is very interesting. Although interior-point techniques, primarily in the form of barrier methods, were widely used during the 1960s for problems with nonlinear constraints, their use for the fundamental problem of linear programming was unthinkable because of the total dominance of the simplex method. During the 1970s, barrier methods were superseded, nearly to the point of oblivion, by newly emerging and seemingly more efficient alternatives such as augmented Lagrangian and sequential quadratic programming methods. By the early 1980s, barrier methods were almost universally regarded as a closed chapter in the history of optimization. This picture changed dramatically in 1984, when Narendra Karmarkar announced a fast polynomial-time interior method for linear programming; in 1985, a formal connection was established between his method and classical barrier methods. Since then, interior methods have continued to transform both the theory and practice of constrained optimization. At least, they give a uniform lens to view both linear and non-linear programs, overcoming a bias that was long entrenched in the optimization community, where linear programming was invariably dominated by the simplex method. IPMs is a vast family of algorithms, and we will only see one of these in the context of linear programs. The following quote of M. Wright sums the perspective on ipm: "The interior-point revolution, like many other revolutions, includes old ideas that are rediscovered or seen in a different light, along with genuinely new ideas."

There are three key ingredients that we need for ipm:

1. a way to ensure that we always stay inside the feasible region,
2. a way to capture the objective function, and
3. a way to move inside the interior of the feasible region while trying to minimize the objective function.

We will figure out these details one at a time. Let P be the polyhedron given by $A\mathbf{x} \leq \mathbf{b}$, and suppose it is not empty. Let $f(\mathbf{x}) := \langle \mathbf{c}, \mathbf{x} \rangle$ be the objective function that we want to maximize. In order to ensure that we stay inside the feasible region, we will construct a barrier along the boundary of P that prevents us from leaving the feasible region. How do we do this? If $\mathbf{a}_i$ is the ith row of A, then the i-th constraint is $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i$. Consider the function $\ln(b_i - \langle \mathbf{a}_i, \mathbf{x} \rangle)$. This function tends to $-\infty$ as $\mathbf{x}$ approaches the hyperplane corresponding to the ith constraint. If along with $f(x)$, we are trying to maximize this function as well then it is clear that the maximum will not be near the hyperplane. Doing this for all the constraints, we get the following **barrier function** corresponding to P:

$$\sum_{i=1}^{m} \ln(b_i - \langle \mathbf{a}_i, \mathbf{x} \rangle). \tag{28}$$

If we try to maximize this function over P, then it is clear that the optimum is far away from the boundary of P and somewhere in its interior. Thus the first ingredient is attained by introducing maximizing the barrier function. However, we are interested in maximizing the original objective function $f(\mathbf{x})$. To combine both these optimizations into one, we introduce a new objective function that combines a good measure of the barrier function and the objective function:

$$f_\mu(\mathbf{x}) := f(\mathbf{x}) + \mu \sum_{i=1}^{m} \ln(b_i - \langle \mathbf{a}_i, \mathbf{x} \rangle), \tag{29}$$

where $\mu \in \mathbb{R}_{\geq 0}$. Note that if $\mu = 0$ then this is the original objective function; otherwise, there is a component of the barrier function. If P is bounded then we can claim the following:

LEMMA 38. *If P is a polytope then the function f_μ attains a unique optimum in the interior of P.*

Proof. Intuitively this is because f_μ is a strictly concave function. Since P by assumption is bounded and f_μ is continuous, we know that it must have a maximum on the set. To see uniqueness, we first show that f_μ is concave. This is not hard to see, since $f(\mathbf{x})$ is linear and ln is a strictly concave function. More formally, from (29) we have

$$f_\mu(t\mathbf{x} + (1-t)\mathbf{y}) = f(t\mathbf{x} + (1-t)\mathbf{y}) + \mu \sum_{i=1}^{m} \ln(b_i - (t\langle \mathbf{a}_i, \mathbf{x}\rangle + (1-t)\langle \mathbf{a}_i, \mathbf{y}\rangle)).$$

From the linearity of f, we obtain that

$$f_\mu(t\mathbf{x} + (1-t)\mathbf{y}) = tf(\mathbf{x}) + (1-t)f(\mathbf{y}) + \mu \sum_{i=1}^{m} \ln(b_i - (t\langle \mathbf{a}_i, \mathbf{x}\rangle + (1-t)\langle \mathbf{a}_i, \mathbf{y}\rangle)).$$

Writing $b_i = tb_i + (1-t)b_i$, we further obtain that

$$f_\mu(t\mathbf{x} + (1-t)\mathbf{y}) = tf(\mathbf{x}) + (1-t)f(\mathbf{y}) + \mu \sum_{i=1}^{m} \ln((t(b_i - \langle \mathbf{a}_i, \mathbf{x}\rangle) + (1-t)(b_i - \langle \mathbf{a}_i, \mathbf{y}\rangle)).$$

The concavity of ln implies the concavity of f_μ. This instead implies that if there are two distinct maxima $\mathbf{x}, \mathbf{y} \in P$, then all points on the line segment joining $\mathbf{x}$ and $\mathbf{y}$ are also maxima. As ln is a strictly concave function, this possible iff $b_i - \langle \mathbf{a}_i, \mathbf{x}\rangle = b_i - \langle \mathbf{a}_i, \mathbf{y}\rangle$, i.e., iff $A\mathbf{x} = A\mathbf{y}$. Thus $\mathbf{x} - \mathbf{y}$ belongs to the null space of A, and hence P contains the line $\{\mathbf{x} + t \cdot (\mathbf{x} - \mathbf{y})\}$, which is a contradiction to the boundedness of P. We have basically shown that a strictly concave function has a unique maxima on a convex set. **Q.E.D.**

The unique maxima $\mathbf{x}^*_\mu$ is called the analytic center of P. If μ is large the analytic center is away from the boundary, but as μ decreases it starts moving in the direction of the objective function f. The idea of interior point methods is to approximate the trajectory of $\mathbf{x}^*_\mu$, also called the central path, towards an optimum for the LP; and when it has reached sufficiently close to the boundary, it rounds the solution to the optimum. We now describe in some more detail a slightly different approach. Instead of working with the LP in the canonical form, we will consider the equational form, just as we did for the simplex method.

12.1 Constrained Optimization Problems

A general constrained optimization problem is to maximize an objective function $f : \mathbb{R}^n \to \mathbb{R}$ subject to some constraints. The constraints may contains inequalities of the form $g_i(\mathbf{x}) \geq c_i$, where $g_i : \mathbb{R}^n \to \mathbb{R}$, but using slack variables we can always convert them to equalities and non-negativity constraints of the form $\mathbf{x} \geq 0$. Therefore, we will assume wlog that a general constraint optimization problem is of the following form: maximize $f(\mathbf{x})$ subject to $g_i(\mathbf{x}) = 0$, $\mathbf{x} \geq 0$. If f and g_i's are all linear functions, then this is just linear programming. However, we will be interested in solving the more general problem of nonlinear programming, with the aim of specializing this approach to linear programing. Our goal is to show that all these problems, in one way or another, are related to solving a system of equations.

Consider the special case of unconstrained optimization to maximize $f(\mathbf{x})$, where $\mathbf{x}$ is unbounded. For instance, maximize $f(x) := 2 - x^2$; it is clear that the maximum is attained at the origin; but to get the maximum formally, we take the derivative $f'(x) = -2x$ and the solution of the equation $f'(x) = 0$; such points are stationary points, or extreme points; it is clear, in this particular case, that the derivative vanishes at the origin. Let us consider the bivariate case $f(x, y) := 2 - x^2 - y^2$, where $(x, y) \in \mathbb{R}^2$. Again, it is clear that the maximum is attained at the origin, which in this case is the solution to gradient vanishing, i.e., $\nabla f(x, y) = 0$. Assuming that the maximum is well defined, why should the gradient vanish at the maximum? Intuitively, the gradient gives the first order approximation to $f(\mathbf{x})$ in the neighborhood of a point, i.e., $f(\mathbf{y}) \sim f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{y})$, for $\mathbf{y}$ close enough to $\mathbf{x}$. So, if the gradient is non-zero, there is always a direction from $\mathbf{x}$ in which $f(\mathbf{y})$ increases and a direction in which it decreases; in fact, the maximum increase is obtained along the unit vector from $\mathbf{x}$ in the direction of the gradient, and the maximum decrease in the opposite direction. Thus the gradient at a point gives us the direction of maximum increase from that point. Now it is clear that for a (local) maximum or a minimum the gradient has to vanish. It is clear that if $f(\mathbf{x})$ is a concave (convex) function, then the vanishing of the gradient is a necessary and sufficient condition for

a point to be a maximum (resp., minimum). Thus finding the maximum can be reduced to solving a system of n equations in n unknowns.

Now let us add some constraints, starting from the univariate case. How about if maximizing $2 - x^2$ where we also have a constraint, say $x = 1$? In the univariate case, this can be solved easily. Suppose we want to maximize $2 - (x^2 + y^2)$, s.t. $x + y = 1$. The aim is to reduce this problem to an unconstrained optimization one. The idea is the following: consider the level sets or contours of $x^2 + y^2$ in increasing order (or imagine concentric ripples from the origin). The first time such a contour meets the line is when the maximum is attained. Since for any large contour, the circle intersects the line at two places, and at both the places we can slide along the line to increase the value of the contour. Notice that the first time the contour meets the line, the latter is a tangent to the former. Thus the point where the maximum is attained is such that it is on the line, and the normal to the circle at that point is proportional to the normal of the line, i.e., a point (x, y) such that $x + y = 1$ and $2(x, y) = \lambda(1, 1)$. Clearly, the solution is $x = y = 1/2$. If we define $f(x, y) := 2 - (x^2 + y^2)$ and $g(x, y) := x + y - 1$, then these conditions are equivalent to

$$g(x, y) = 0 \text{ and } \nabla f = \lambda \nabla g$$

where ∇ is the gradient operator. Both these conditions can be succinctly captured by the **Lagrangian**:

$$\Lambda(x, y, \lambda) := f(x, y) + \lambda g(x, y).$$

Maximizing the Lagrangian is an unconstrained optimization problem and the stationary points are the solution to the system $\nabla\Lambda = 0$. But the vanishing of the partial derivatives of Λ wrt x and y is the system $\nabla f = \lambda \nabla g$, and the partial derivative wrt λ vanishing is equivalent to $g(x, y) = 0$. Clearly, the approach works in higher dimensions as well. The scalar λ is called a **Lagrange multiplier**.

What if we had more than one constraint? Let us first reinterpret the derivation above. For any contour, the gradient is normal to it and at any point gives us the direction of increase. For the curve $g = 0$, the direction orthogonal to the gradient are the possible directions along with we can move slightly while still satisfying the constraint. Thus if ∇f has a component in the direction orthogonal to ∇g then at that point it is possible to increase the value of f by moving along the constraint. This interpretation generalizes to the setting of more than one constraint, namely, if ∇f has a component in the space orthogonal to the space spanned by ∇g_i, for the m constraints, then it is possible to increase f while satisfying the constraints. Thus at the point which attains the maximum ∇f must be in the span of ∇g_i, i.e., there exists λ_i, not all zero, such that

$$\nabla f(\mathbf{x}) = \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x})$$

and $g_i(\mathbf{x}) = 0$, for $i = 1, \ldots, m$. Again, all these conditions can be succinctly captured by maximizing the Lagrangian

$$\Lambda(\mathbf{x}, \lambda) := f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}).$$

Taking the partial derivative wrt the $n + m$ variables, we get the following system of $n + m$ equations:

$$g_i(\mathbf{x}) = 0, \text{ for } i = 1, \ldots, m, \text{ and } \nabla f(\mathbf{x}) = \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}).$$

This is called the method of Lagrangian multipliers.

Now let us add the non-negativity constraints $\mathbf{x} \geq 0$ to our optimization problem to consider the general setting of constrained optimization problems. In this setting, we will not be able to reduce the problem to a single unconstrained optimization problem, but a family of them parametrized by a parameter $\mu \geq 0$. The idea is to introduce a function along with the objective function such that maximizing that function ensures that the inequalities are always satisfied strictly, or in other words, that we are always in the interior of the feasible region. One such choice is the **logarithmic barrier function** $\sum_{i=1}^{n} \log x_i$. Since this function goes to $-\infty$ if any x_i approaches zero, it is clear that maximizing this function means we always stay inside the positive orthant. However, our aim is to maximize the objective function $f(\mathbf{x})$, which may actually achieve

an optimum on the boundary of the feasible region. Therefore, we take a combination of the barrier function with the objective function parameterized by $\mu \in \mathbb{R}_{\geq 0}$:

$$f_\mu(\mathbf{x}) := f(\mathbf{x}) + \mu \sum_{i=1}^{n} \log x_i.$$

So our constrained optimization problem maximize $f(\mathbf{x})$ subject to $g_i(\mathbf{x}) = 0$, $i = 1, \ldots, m$, and $\mathbf{x} \geq 0$ reduces to the following: given a μ, maximize $f_\mu(\mathbf{x})$ subject to $g_i(\mathbf{x}) = 0$. For every choice of μ, we use the Lagrangian multiplier approach to solve this constrained optimization problem. Under some restrictions on f_μ it can be shown that the optimum is unique $\mathbf{x}^*_\mu$, and is continuously dependent on μ. So as μ approaches zero, the optimum $\mathbf{x}^*_\mu$ traces a path, called the **central path**, converging to the optimum of the original constrained optimization problem.

So far we have seen that a constrained optimization problem with inequalities reduces to a family of constrained optimization problem, which can be solved using the Lagrangian multiplier approach. The latter approach reduces to solving a system of some n nonlinear equations $f_i(\mathbf{w})$, $i = 1, \ldots, n$, in n unknowns. How do we solve such a system? We use Newton iteration starting from a sufficiently good approximation to a solution, to ensure quadratic convergence. How is a Newton step done? Given an iterate $\mathbf{w}_k$, we take a first order approximation to $F(\mathbf{w}) := (f_1(\mathbf{w}), \ldots, f_n(\mathbf{w}))$ and equate it to zero, i.e., $F(\mathbf{w}) \sim F(\mathbf{w}_k) + J(\mathbf{w}_k) \cdot (\mathbf{w}_{k+1} - \mathbf{w}_k) = 0$, where $J(\mathbf{w}_k)$ is the jacobian matrix evaluated at $\mathbf{w}_k$. Thus

$$\mathbf{w}_{k+1} = \mathbf{w}_k - J(\mathbf{w}_k)^{-1} F(\mathbf{w}_k), \tag{30}$$

assuming that the jacobian is non-singular. We will now apply this approach to solving constrained optimization problems with non-negativity constraints to our setting of linear programs.

12.2 Primal-Dual Central Path

Consider the equational form of LP:

$$\text{maximize } \langle \mathbf{c}, \mathbf{x} \rangle \text{ subject to } A\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq 0,$$

where A is an $m \times n$ matrix of rank m. Since the barrier function is meant to prevent us from violating inequalities, in this setting it has the form $\sum_{i=1}^{n} \ln x_i$ and the corresponding modification of the objective function is

$$f_\mu(\mathbf{x}) := f(\mathbf{x}) + \mu \sum_{i=1}^{n} \ln x_i. \tag{31}$$

We will now consider the following optimization problem: Given a $\mu > 0$,

$$\text{maximize } f_\mu(\mathbf{x}) \text{ subject to } A\mathbf{x} = \mathbf{b}, \tag{32}$$

i.e., in the interior of the positive orthant. Applying the Lagrangian multiplier approach to (32), we get that the optimum satisfies

$$A\mathbf{x} = \mathbf{b} \text{ and } \nabla f_\mu(\mathbf{x}) = \sum_{i=1}^{m} y_i \nabla \langle \mathbf{a}_i, \mathbf{x} \rangle,$$

where $f_\mu(\mathbf{x})$ is given by (31). Thus

$$\nabla f_\mu(\mathbf{x}) = \mathbf{c} + \mu \left(\frac{1}{x_1}, \ldots, \frac{1}{x_n} \right).$$

Moreover, $\nabla \langle \mathbf{a}_i, \mathbf{x} \rangle = \mathbf{a}_i$. Therefore, we obtain that the optimum satisfies

$$A\mathbf{x} = \mathbf{b} \text{ and } \mathbf{c} + \mu \left(\frac{1}{x_1}, \ldots, \frac{1}{x_n} \right) = \sum_{i=1}^{m} y_i \mathbf{a}_i = A^t \mathbf{y}.$$

We simplify this further by defining

$$\mathbf{s} := \mu \left(\frac{1}{x_1}, \ldots, \frac{1}{x_n} \right).$$

This is equivalent to saying that $(s_1 x_1, s_2 x_2, \ldots, s_n x_n) = \mu \mathbf{1}$. Since $\mathbf{x} > 0$ and $\mu > 0$, it follows that $\mathbf{s} > 0$. Let X and S be the diagonal matrices with entries x_i and s_i respectively. Then this condition can be succinctly stated as $XS\mathbf{1} = \mu \mathbf{1}$. Thus an optimum of (32) must satisfy the following constraints: for $\mu > 0$

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ A^t \mathbf{y} - \mathbf{s} &= \mathbf{c} \\ XS\mathbf{1} &= \mu \mathbf{1} \\ \mathbf{x} &> 0. \end{aligned} \tag{33}$$

This system of conditions is called the Karush-Kuhn-Tucker system. Note that we have reintroduced $\mathbf{x} > 0$; since $\mu > 0$, from the third equation we could equivalently have stated that $\mathbf{s} > 0$, but we need at least one of these inequalities, as our claim would be to show that any solution of this system is an optimum for (32). Note that apart from the third constraint, this is a system of linear equations. The third constraint will only be satisfied partially (basically its linear part), and the error term will play a crucial role. Our problem is now reduced to solving the system of equations above, with the additional non-negativity constraints. This will be done using Newton's method, but before we do that let us understand what this system of equations means.

We want that when $\mu = 0$ the $\mathbf{x}$-coordinate of the solution of this system of equations must be the optimum for the lp. But if $\mu = 0$, then $s_i x_i = 0$, for $i = 1, \ldots, n$, and since both are non-negative vectors this is equivalent to the statement that $\langle \mathbf{s}, \mathbf{x} \rangle = 0$. Now

$$0 = \langle \mathbf{s}, \mathbf{x} \rangle = \langle A^t \mathbf{y} - \mathbf{c}, \mathbf{x} \rangle = \langle \mathbf{y}, A\mathbf{x} \rangle - \langle \mathbf{c}, \mathbf{x} \rangle = \langle \mathbf{y}, \mathbf{b} \rangle - \langle \mathbf{c}, \mathbf{x} \rangle,$$

i.e., $\langle \mathbf{c}, \mathbf{x} \rangle = \langle \mathbf{b}, \mathbf{y} \rangle$. But note that this is the equality of the objective functions for the primal and dual linear programs, or equivalently, the complementary slackness condition. Indeed the dual of maximizing $\langle \mathbf{c}, \mathbf{x} \rangle$ s.t. $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, is minimizing $\langle \mathbf{b}, \mathbf{y} \rangle$ s.t. $A^t \mathbf{y} \geq \mathbf{c}$, $\mathbf{y} \in \mathbb{R}^m$. And what does $\mathbf{s}$ stand for? They are the set of slack variables for the dual. So for $\mu = 0$, the solution of (33) actually contains the optimum for the dual and the primal. In fact, we can prove the strong duality from properties of the solution (we will show that there is a unique solution) $\mathbf{w}^*_\mu := (\mathbf{x}^*_\mu, \mathbf{y}^*_\mu, \mathbf{s}^*_\mu)$ of (33). The **duality gap** of a point $\mathbf{w}^*_\mu$ is defined as

$$\gamma(\mathbf{w}^*_\mu) := \langle \mathbf{c}, \mathbf{x}^*_\mu \rangle - \langle \mathbf{b}, \mathbf{y}^*_\mu \rangle.$$

Substituting the equation $\mathbf{c} = A^t \mathbf{y}^*_\mu - \mathbf{s}^*_\mu$ above, we obtain that $\gamma(\mathbf{w}^*_\mu) = \langle \mathbf{s}^*_\mu, \mathbf{x}^*_\mu \rangle = n\mu$. Thus as μ tends to zero the duality gap tends to zero and hence $\mathbf{w}^*_\mu$ approaches the primal-dual optimum, i.e., $\mathbf{x}^*_\mu$ approaches a primal optimum and $(\mathbf{y}^*_\mu, \mathbf{s}^*_\mu)$ approaches the dual optimum. The algorithm will, therefore, try to trace the path $\mathbf{w}^*_\mu$ as μ approaches zero. However, before we do that, we need to show that this path actually exists, i.e., for a given μ, the solution of (33) is well-defined and actually unique (well, under certain assumptions). We will often need the following set

$$\mathcal{W} := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{x} \text{ is feasible in the primal, and } (\mathbf{y}, \mathbf{s}) \text{ is feasible in the dual}\}. \tag{34}$$

LEMMA 39. *Suppose the primal and dual of an lp are both feasible, i.e., $\mathcal{W} \neq \emptyset$ and the matrix A is full row-rank. Then for every $\mu > 0$ the system of equations (33) has a unique solution $(\mathbf{x}^*_\mu, \mathbf{y}^*_\mu, \mathbf{s}^*_\mu) \in \mathbb{R}^{n+m+n}$ and the solution attains the maximum for the lp in (32).*

Proof. We will first show that (33) has a solution, and then that this solution is unique.

To show the existence, we will show that there is a unique maximum of (32), and as a maximum of the lp is a solution of the system of equations, the existence follows. Let $\overline{\mathbf{x}}$ be a feasible point in the primal and $(\overline{\mathbf{y}}, \overline{\mathbf{s}})$ be a feasible point in the dual (we are not assuming that $\overline{\mathbf{x}}$ and $\overline{\mathbf{s}}$ are related). We claim that the set

$$Q := \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0, f_\mu(\mathbf{x}) \geq f_\mu(\overline{\mathbf{x}})\}$$

is a closed and bounded set. Since f_μ is continuous on this set, it attains a maximum as well. Therefore, the lp (32) has an optimum, and hence this optimum is a solution of(33). This will gives us the existence of a solution.

For $\mathbf{x} \in Q$, consider

$$f_\mu(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle + \mu \sum_{j=1}^{n} \ln x_j.$$

As given above, the objective looks unbounded. However, we know that the linear part in the function cannot be unbounded because the dual is feasible, i.e., $\langle \mathbf{c}, \overline{\mathbf{x}} \rangle < \langle \mathbf{b}, \overline{\mathbf{y}} \rangle$. Therefore, we express the objective function as a deficit from the dual value $\langle \mathbf{b}, \overline{\mathbf{y}} \rangle$. To do this substitute $\mathbf{c} = A^t\overline{\mathbf{y}} - \overline{\mathbf{s}}$ on the rhs to obtain

$$f_\mu(\mathbf{x}) = \langle \overline{\mathbf{y}}A^t, \mathbf{x} \rangle - \langle \overline{\mathbf{s}}, \mathbf{x} \rangle + \mu \sum_{j=1}^{n} \ln x_j.$$

Using the definition of inner product and the fact that $\mathbf{x}$ is feasible we get $\langle \overline{\mathbf{y}}A^t, \mathbf{x} \rangle = \langle \overline{\mathbf{y}}, A\mathbf{x} \rangle = \langle \overline{\mathbf{y}}, \mathbf{b} \rangle$, and hence

$$f_\mu(\mathbf{x}) = \langle \overline{\mathbf{y}}, \mathbf{b} \rangle - \langle \overline{\mathbf{s}}, \mathbf{x} \rangle + \mu \sum_{j=1}^{n} \ln x_j,$$

which is equivalent to

$$f_\mu(\mathbf{x}) = \langle \overline{\mathbf{y}}, \mathbf{b} \rangle + \sum_{j=1}^{n} (\mu \ln x_j - \overline{s}_j x_j). \tag{35}$$

Thus for an $\mathbf{x}$ in the primal feasible region, $f_\mu(\mathbf{x})$ can be expressed as a constant term plus terms that are obtained by evaluating univariate functions of the form $h_\alpha(x) := \mu \ln x - \alpha x$ (for $x > 0$ these functions take negative value). The function $h_\alpha(x)$ is strictly concave and attains its unique maximum at μ/α. Moreover, from the graph of $h_\alpha(x)$ it is easy to see that the set $\{x > 0 : h_\alpha(x) > \text{const.}\}$ is a bounded interval.

Now from (35) it follows that

$$Q = \left\{ \mathbf{x} > 0 : A\mathbf{x} = \mathbf{b}, \sum_{j=1}^{n} h_{\overline{s}_j}(x_j) \geq f_\mu(\overline{\mathbf{x}}) - \langle \mathbf{b}, \overline{\mathbf{y}} \rangle \right\},$$

which is contained in the set

$$\left\{ \mathbf{x} > 0 : \sum_{j=1}^{n} h_{\overline{s}_j}(x_j) \geq f_\mu(\overline{\mathbf{x}}) - \langle \overline{\mathbf{y}}, \mathbf{b} \rangle \right\},$$

where the RHS of the inequality above is a constant. But this set is contained in the following set

$$\prod_{j=1}^{n} \left\{ x : h_{\overline{s}_j}(x) \geq \text{const.} - \sum_{i \neq j} \max h_{\overline{s}_i}(x) \right\}.$$

Again the RHS of the inequality is a constant. But from our observation earlier, we know that each set in the cartesian product above is a bounded interval. Therefore, Q is a bounded set.

We now show the uniqueness of the solution. Suppose $(\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\mathbf{s}})$ is a solution to (33). As $f_\mu(\mathbf{x})$ is strictly concave, its maximum is unique on the set Q. We will show that $f_\mu(\mathbf{x})$ attains its maximum at $\overline{\mathbf{x}}$, which establishes the uniqueness of $\overline{\mathbf{x}}$. Once we have the uniqueness of $\overline{\mathbf{x}}$, then $\overline{\mathbf{s}}$ is uniquely defined as the solution to $XS\mathbf{1} = \mu\mathbf{1}$, where μ is fixed. The uniqueness of $\overline{\mathbf{y}}$ then follows, since it is a solution to the system $A^t\mathbf{y} = \mathbf{c} - \mathbf{s}$, and A is full-rank. Why does $f_\mu(\mathbf{x})$ attain its maximum at $\overline{\mathbf{x}}$? From (35) it is clear that the maximum value of $f_\mu(\mathbf{x})$ on Q is attained if $x_j = \mu/\overline{s}_j$, which is precisely $\overline{\mathbf{x}}$. Therefore, $\overline{\mathbf{x}}$ is the unique solution of (32).

Basically, we have shown that the opt of (32) is a solution of (33), and conversely the solution of the latter is an opt of the former. Therefore, getting a good approximation to the solution of the system suffices.

Q.E.D.

The set of solutions to (33)

$$\Pi := \left\{ \left(\mathbf{x}^*_\mu, \mathbf{y}^*_\mu, \mathbf{s}^*_\mu \right) \in \mathbb{R}^{n+m+n} : \mu > 0 \right\}. \tag{36}$$

is called the **primal-dual central path** of the linear program. The algorithm will actually approximate this path by using Newton-Raphson method. Let us begin by understanding one step of Newton's method. Clearly, $\Pi \subset \mathcal{W}$.

12.3 The Algorithm

The algorithm will generate a sequence of points $\mathbf{w}_k = (\mathbf{x}_k, \mathbf{y}_k, \mathbf{s}_k)$, starting from an initial point $\mathbf{w}_0$. Each point $\mathbf{w}_k$ corresponds to a given $\mu_k > 0$. The initial point $\mathbf{w}_0$ is chosen sufficiently close to the path Π; how we measure proximity will be discussed later. Given a current iterate $\mathbf{w}$ we will apply Newton's method to (33) to get the next iterate $\mathbf{w}'$. Let $F(\mathbf{w})$ be the system of equations in (33), and $J(\mathbf{w})$ be its jacobian. Then we know that the newton step $\Delta\mathbf{w} := (\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s})$ is the solution to the equation

$$J(\mathbf{w}) \cdot \Delta\mathbf{w} = F(\mathbf{w})$$

and $\mathbf{w}' := \mathbf{w} - \Delta\mathbf{w}$ (see (30)). But what is the jacobian of $F(\mathbf{w})$? It is not hard to see that

$$J(\mathbf{w}) = \begin{bmatrix} A & 0 & 0 \\ 0 & A^t & -I \\ S & 0 & X \end{bmatrix}.$$

What is $F(\mathbf{w})$? Note that the first two equations are actually linear equations, so $\mathbf{w}$ satisfies them exactly, i.e., $A\mathbf{w} = \mathbf{b}$ and $A^t\mathbf{y} - \mathbf{s} = \mathbf{c}$. So the first two coordinates of $F(\mathbf{w})$ are actually zero. It is the last coordinate that is interesting, and its value is $XS\mathbf{1} - \mu'\mathbf{1}$, where $\mu' > 0$. Therefore, the newton step $\Delta\mathbf{w} = (\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s})$ satisfies the following equations:

$$\begin{aligned} A\Delta\mathbf{x} &= 0 \\ A^t\Delta\mathbf{y} &= \Delta\mathbf{s} \\ S\Delta\mathbf{x} + X\Delta\mathbf{s} &= XS\mathbf{1} - \mu'\mathbf{1}. \end{aligned} \tag{37}$$

Note that $\Delta\mathbf{w}$ depends only on the $\mathbf{x}, \mathbf{s}$ from the previous iteration and μ', which is an update of the μ from the previous iteration. Multiplying the last equation by S^{-1} and substituting the second eqaution for $\Delta\mathbf{s}$, we obtain that

$$\Delta\mathbf{y} = \left(\left(AS^{-1}XA^t \right)^{-1} AS^{-1} \right) (XS\mathbf{1} - \mu'\mathbf{1}). \tag{38}$$

Substituting this in the second equation of (37) gives us

$$\Delta\mathbf{s} = \left(A^t \left(AS^{-1}XA^t \right)^{-1} AS^{-1} \right) (XS\mathbf{1} - \mu'\mathbf{1}), \tag{39}$$

and substituting this in the last equation of (37) we obtain

$$\Delta\mathbf{x} = \left(S^{-1} - S^{-1}XA^t \left(AS^{-1}XA^t \right)^{-1} AS^{-1} \right) (XS\mathbf{1} - \mu'\mathbf{1}). \tag{40}$$

It is clear from the equations above that the most expensive step is the computation of the inverse of $(AS^{-1}XA^t)$. We can now describe the algorithm in some detail.

The input to the algorithm is an initial starting point $\mathbf{w}_0 = (\mathbf{x}_0, \mathbf{y}_0, \mathbf{s}_0)$, such that $\mathbf{x}_0$ is feasible for the primal and $(\mathbf{y}_0, \mathbf{s}_0)$ for the dual. We are also given three constants, θ, δ and ϵ in $(0, 1]$. The constant δ is the factor that controls the decrement of μ in each iteration, and ϵ is the tolerance for the duality gap, i.e., the algorithm stops when the duality gap falls below ϵ. The role of θ is to measure how close the iterates are to the path Π. In particular, the starting point $\mathbf{w}_0$ satisfies the following property

$$\text{dist}(\mathbf{w}_0, \Pi) := \frac{\|X_0S_0\mathbf{1} - \mu_0\mathbf{1}\|_2}{\mu_0} \leq \theta, \tag{41}$$

where $\mu_0 := \langle \mathbf{x}_0, \mathbf{s}_0 \rangle / n$. We will later explain why the LHS is a good measure of distance. The algorithm is as follows:

INPUT: $A, \mathbf{b}, \mathbf{c}$, $\mathbf{w}_0$, θ, δ, ϵ.
OUTPUT: $\mathbf{w}^*$ such that $\gamma(\mathbf{w}^*) \leq \epsilon$.
1. Set $k \leftarrow 0$.
2. If $\gamma(\mathbf{w}_k) = \langle \mathbf{x}_k, \mathbf{s}_k \rangle \leq \epsilon$ then return $\mathbf{w}_k$.
3. Let $\mu_{k+1} \leftarrow \mu_k (1 - \delta)$.
4. Calculate $\Delta \mathbf{w}_k$ as in (38), (58) and (40).
5. Set $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \Delta \mathbf{w}_k$, $k \leftarrow k + 1$ and go to step 2.

We will show that the number of iterations are bounded by $O(\delta^{-1} \log(n\mu_0/\epsilon))$. We will also show that the max is upper bounded by L, the input size of the lp:

$$L := \log \max(\text{the largest absolute value over all the sub-determinants of } A, \|\mathbf{b}\|_\infty, \|\mathbf{c}\|_\infty, m + n). \quad (42)$$

Moreover, δ would be chosen such that $1/\delta = O(\sqrt{n})$. Therefore, along with the observation that step 4 requires $O(n^3)$ time, we get the desired bound of $O(n^{3.5}L)$ for the algebraic cost of the algorithm.

12.4 Convergence

Here we prove the main result about the correctness of the algorithm: If an iterate $\mathbf{w} \in \mathcal{W}$ is sufficiently close to Π, a newton step, as given above, gives us a new iterate $\mathbf{w}' \in \mathcal{W}$ such that it is close to Π, and the duality gap decreases. More formally, we will show the following claim:

THEOREM 40. *Let $\delta, \theta < 1$ be the parameters in the algorithm and $\mathbf{w} = (\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{W}$ be such that*

$$dist(\mathbf{w}, \Pi) := \frac{\|XS\mathbf{1} - \mu\mathbf{1}\|}{\mu} \leq \theta, \quad (43)$$

where $\mu := \langle \mathbf{x}, \mathbf{s} \rangle / n$. If $\mu' := \mu(1 - \delta)$ and $\mathbf{w}' := \mathbf{w} - \Delta\mathbf{w}$, where $\Delta\mathbf{w}$ is the newton step given by (38), (58) and (40), then $\mathbf{w}'$ satisfies the following:

1. *$\mathbf{w}' \in \mathcal{W}$,*
2. *$dist(\mathbf{w}', \Pi) \leq \theta$, and*
3. *$\gamma(\mathbf{w}') = \langle \mathbf{x}', \mathbf{s}' \rangle = n\mu'$.*

Let us first understand the purport of this theorem, starting from why $\text{dist}(\mathbf{w}, \Pi)$ makes sense as a measure of distance of a point $\mathbf{w}$ from Π. Consider the map

$$h(\mathbf{w}) := (x_1 s_1, \ldots, x_n s_n) = XS\mathbf{1} \quad (44)$$

that maps $\mathbb{R}^{n+m+n}$ to $\mathbb{R}^n$. Then from (33) we know that the path Π is mapped to the "diagonal" $\mu\mathbf{1}$ in $\mathbb{R}^n$. The $\mathbf{w}$ corresponding to the primal and dual optimum solution for the lp is mapped to the origin. For any other vector $\mathbf{w} \in \mathcal{W}$, the map $h(\mathbf{w})$ takes $\mathbf{w}$ to some vector in $\mathbb{R}^n$. If the vector $h(\mathbf{w})$ is close to the diagonal, then in some sense we can say that $\mathbf{w}$ is close to Π. Thus we are not measuring closeness of $\mathbf{w}$ and Π directly, but by measuring the distance between their images in the "complementary slackness space". What is the point closest to $h(\mathbf{w})$ on the diagonal? It is the projection of $h(\mathbf{w})$ along the unit vector in the direction of $\mathbf{1}$, i.e., the point

$$\langle h(\mathbf{w}), \frac{\mathbf{1}}{\sqrt{n}} \rangle \cdot \frac{\mathbf{1}}{\sqrt{n}} = \frac{\langle \mathbf{x}, \mathbf{s} \rangle}{n} \mathbf{1} = \mu\mathbf{1}.$$

Therefore, the distance function $\text{dist}(\mathbf{w}, \Pi)$ measures the relative closeness of $h(\mathbf{w})$ to the diagonal (relative, because the distance is measured wrt μ). The points that are "θ-close" to the diagonal form a cone centered at origin and symmetric around the diagonal; the angle of the cone is governed by the error parameter θ.

As a consequence of Theorem 40 we have the following claim:

Corollary 41. *The sequence of points $\mathbf{w}_k$, $k > 0$, generated by the algorithm satisfies the following:*

1. $\mathbf{w}_k \in \mathcal{W}$,

2. $dist(\mathbf{w}_k, \Pi) \leq \theta$, *and*

3. $\gamma(\mathbf{w}_k) = n\mu_k$, *where* $\mu_k := \mu_0(1-\delta)^k$.

From this corollary, we can now upper bound the iterations required by the algorithm. The iteration stops at k when the duality gap $\gamma(\mathbf{w}_k)$ falls below ϵ, the tolerance. But from the corollary above $\gamma(\mathbf{w}_k) = n\mu_0(1-\delta)^k$. Therefore, if

$$(1-\delta)^{-k} \geq \frac{n\mu_0}{\epsilon}$$

then the iteration stops. Taking natural-log on both sides, along with the observation that $-\ln(1-x) > x$, for $x < 1$, we have the following:

LEMMA 42. *The algorithm stops when*

$$k > \delta^{-1} \ln \frac{n\mu_0}{\epsilon}.$$

We will now prove Theorem 40 starting with the following:

LEMMA 43. *Let* $\mathbf{w} \in \mathcal{W}$, $\mu' > 0$ *and* $\Delta\mathbf{w}$ *be the solution to* (37) *corresponding to* $\mathbf{w}$ *and* μ'. *If* $\mathbf{w}' := \mathbf{w} - \Delta\mathbf{w}$ *then we have*

1. *The* i*th component of* $h(\mathbf{w}')$ *is* $\mu' + \Delta\mathbf{x}_i\Delta\mathbf{s}_i$,

2. *the vector* $\Delta\mathbf{x}$ *is orthogonal to* $\Delta\mathbf{s}$, *and*

3. *the duality gap* $\gamma(\mathbf{w}') = n\mu'$.

Proof.

1. The first result captures the fact that $\mathbf{w}'$ is close to the diagonal (which is the first part of Theorem 40). The proof is straightforward: the i-th component of $h(\mathbf{w}')$ is $\mathbf{x}'_i\mathbf{s}'_i = (x_i - \Delta\mathbf{x}_i)(s_i - \Delta\mathbf{s}_i)$. Opening the product we obtain that the ith component is $[x_is_i - (x_i\Delta\mathbf{s}_i + s_i\Delta\mathbf{x}_i)] + \Delta\mathbf{x}_i\Delta\mathbf{s}_i$. But from the last equation of (37), the first part is equal to μ'.

2. From second equation in (37) we get that $\langle\Delta\mathbf{x}, \Delta\mathbf{s}\rangle = \langle\Delta\mathbf{x}, A^t\Delta\mathbf{y}\rangle$, which from the definition of inner product is equal to $\langle A\Delta\mathbf{x}, \Delta\mathbf{y}\rangle$. But this is clearly zero from the first eqaution in (37).

3. From the definition of duality gap we know that $\gamma(\mathbf{w}') = \sum_{i=1}^n x'_is'_i = \sum_i [h(\mathbf{w}')]_i$. Substituting the first result of the lemma, we obtain that

$$\gamma(\mathbf{w}') = n\mu' + \sum_{i=1}^n \Delta\mathbf{x}_i\Delta\mathbf{s}_i = n\mu' + \langle\Delta\mathbf{x}, \Delta\mathbf{s}\rangle = n\mu',$$

where the last step follows from the second result. This already proves the third claim of Theorem 40.

Q.E.D.

We will now prove the first and the second part of Theorem 40. The crucial part is the second part; the proof of first part, namely $\mathbf{w}' \in \mathcal{W}$, will follow from its proof. So let us start with the second part, i.e., $\text{dist}(\mathbf{w}', \Pi) \leq \theta$, given $\text{dist}(\mathbf{w}, \Pi) \leq \theta$ and $\mathbf{w}'$ is obtained from $\mathbf{w}$ by a newton step (37) applied at $\mathbf{w}$ and μ'. From the definition of the distance function we have

$$\text{dist}(\mathbf{w}', \Pi) = \frac{\|h(\mathbf{w}') - \mu'\mathbf{1}\|}{\mu'},$$

where $\mu' := \mu(1-\delta)$. From the lemma above we know that $\|h(\mathbf{w}') - \mu'\mathbf{1}| = \|(\Delta\mathbf{x}_i\Delta\mathbf{s}_i)\|$. Our aim will be to bound this error in terms of the error at the previous iteration, i.e., $\|h(\mathbf{w}) - \mu\mathbf{1}\|$. First step is to bound $\|(\Delta\mathbf{x}_i\Delta\mathbf{s}_i)\|$ in terms of something more manageable. From the last equation in (37) we know that

$X\Delta\mathbf{s} + S\Delta\mathbf{x} = h(\mathbf{w}) - \mu'\mathbf{1}$. Multiplying both sides by $(SX)^{-1/2}$, and using the commutativity of diagonal matrices, we obtain that

$$\left(S^{-1}X\right)^{1/2}\Delta\mathbf{s} + \left(SX^{-1}\right)^{-1/2}\Delta\mathbf{x} = (SX)^{-1/2}(h(\mathbf{w}) - \mu'\mathbf{1}).$$

Let $T := \left(S^{-1}X\right)^{1/2}$, then the equation above can be succinctly written as

$$T\Delta\mathbf{s} + T^{-1}\Delta\mathbf{x} = (SX)^{-1/2}(h(\mathbf{w}) - \mu'\mathbf{1}).$$

Note that T is invertible as $\mathbf{s}, \mathbf{x} > 0$. Moreover, the two vectors $T\Delta\mathbf{s}$ and $T^{-1}\Delta\mathbf{x}$ are orthogonal as $\langle T\Delta\mathbf{s}, T^{-1}\Delta\mathbf{x}\rangle = \langle\Delta\mathbf{s}, \Delta\mathbf{x}\rangle = 0$. Also, the coordinate-wise product of the two vectors is the vector $(\Delta\mathbf{x}_i\Delta\mathbf{s}_i)$. Stepping back, we have two orthogonal vectors $\mathbf{a}$, $\mathbf{b}$ such that $\mathbf{a}+\mathbf{b}$ is related to $h(\mathbf{w})-\mu'$ and their coordinate-wise product is the vector $(\Delta\mathbf{x}_i\Delta\mathbf{s}_i)$. If we can relate the norm of $\|(a_ib_i)\|$ with $\|\mathbf{a}+\mathbf{b}\|$ then we will have some relation. Since $\mathbf{a}$ and $\mathbf{b}$ are orthogonal, it follows from Pythagoras theorem that $\|\mathbf{a}\|^2+\|\mathbf{b}\|^2 = \|\mathbf{a}+\mathbf{b}\|^2$. Moreover, $(\|\mathbf{a}\| - \|\mathbf{b}\|)^2 \geq 0$, which is equivalent to saying that

$$2\|\mathbf{a}\|\|\mathbf{b}\| \leq \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 = \|\mathbf{a}+\mathbf{b}\|^2.$$

But it is easy to verify that $(\|\mathbf{a}\|\|\mathbf{b}\|)^2 \geq \sum_{i=1}^n (a_ib_i)^2$. Therefore, we have shown the claim that

$$\|(a_ib_i)\| \leq \frac{\|\mathbf{a}+\mathbf{b}\|^2}{2}.$$

Substituting $\mathbf{a} := T\Delta\mathbf{s}$ and $\mathbf{b} := T^{-1}\Delta\mathbf{x}$, we obtain that

$$\|(\Delta\mathbf{x}_i\Delta\mathbf{y}_i)\| \leq \frac{1}{2}\|T\Delta\mathbf{s} + T^{-1}\Delta\mathbf{x}\|^2 = \frac{1}{2}\|(SX)^{-1}\left(h(\mathbf{w}) - \mu'\mathbf{1}\right)\|^2 \leq \frac{1}{2}\frac{\|h(\mathbf{w}) - \mu'\mathbf{1}\|^2}{\|h(\mathbf{w})\|_{-\infty}},$$

where $\|h(\mathbf{w})\|_{-\infty} := \min_i(s_ix_i)$. To summarize, we have shown the following:

LEMMA 44. *Let $\mathbf{w} \in \mathcal{W}$, $\mu' > 0$ and $\Delta\mathbf{w}$ be the solution to (37) corresponding to $\mathbf{w}$ and μ'. If $\mathbf{w}' := \mathbf{w} - \Delta\mathbf{w}$ then we have*

$$\|h(\mathbf{w}') - \mu'\mathbf{1}\| \leq \frac{1}{2}\frac{\|h(\mathbf{w}) - \mu'\mathbf{1}\|^2}{\|h(\mathbf{w})\|_{-\infty}}.$$

Note that so far we haven't used the fact that $\mu' = \mu(1-\delta)$. We now show that for appropriate choice of θ and δ, $\|h(\mathbf{w}') - \mu'\mathbf{1}\| \leq \theta\mu'$. This will follow if RHS of the inequality above is smaller than $\theta\mu'$. From the definition of μ', we know that

$$\|h(\mathbf{w}) - \mu'\mathbf{1}\|^2 \leq \|h(\mathbf{w}) - \mu\mathbf{1}\|^2 + (\mu\delta)^2\|\mathbf{1}\|^2 = \|h(\mathbf{w}) - \mu\mathbf{1}\|^2 + (\mu\delta)^2 n.$$

But recall from (43) that $\text{dist}(\mathbf{w}, \Pi) \leq \theta$, i.e, $\|h(\mathbf{w}) - \mu\mathbf{1}\| \leq \theta\mu$. Therefore,

$$\|h(\mathbf{w}) - \mu'\mathbf{1}\|^2 \leq (\theta\mu)^2 + (\mu\delta)^2 n.$$

But (43) also implies that $|\mu - s_ix_i| \leq \theta\mu$, or $s_ix_i \geq (1-\theta)\mu$, for all i. Therefore,

$$\frac{1}{2}\frac{\|h(\mathbf{w}) - \mu'\mathbf{1}\|^2}{\|h(\mathbf{w})\|_{-\infty}} \leq \frac{\theta^2 + \delta^2 n}{2(1-\theta)}\mu.$$

Now, if θ and δ are such that the RHS above is smaller than $\theta\mu' = \theta(1-\delta)\mu$ then we will be done, but this follows if

$$\frac{\theta^2 + \delta^2 n}{2(1-\theta)} \leq \theta(1-\delta).$$

It is clear that for the LHS to be smaller than one, $\delta \leq 1/\sqrt{n}$, so $\tau := \delta\cdot\sqrt{n}$. Then the above inequality follows if

$$\frac{\theta^2 + \tau^2}{2(1-\theta)} \leq \theta(1 - \frac{\tau}{\sqrt{n}}). \tag{45}$$

If we choose θ and τ to satisfy this inequality then we have that $\text{dist}(\mathbf{w}', \Pi) \leq \theta$.

We now show that $\mathbf{w}' \in \mathcal{W}$. Note that $\mathbf{x}'$ and $(\mathbf{y}', \mathbf{s}')$ are primal feasible and dual feasible respectively, since

$$A\mathbf{x}' = A\mathbf{x} + A\Delta\mathbf{x} = A\mathbf{x} = \mathbf{b}$$

and similarly $A^t\mathbf{y}' + \mathbf{s}' = \mathbf{c}$ (this follows directly from (37)). What remains to show is that $\mathbf{x}', \mathbf{s}' > 0$. We will prove it by contradiction. Suppose there is an index i such that $x_i < \Delta\mathbf{x}_i$ or $s_i < \Delta\mathbf{s}_i$. Since $\|h(\mathbf{w}') - \mu'\mathbf{1}\| \leq \theta\mu'$, we know that $s_i'x_i' \geq (1-\theta)\mu' > 0$. Thus if s_i' is negative then so is x_i' and vice versa. If both are negative then $x_i < \Delta\mathbf{x}_i$ *and* $s_i < \Delta\mathbf{s}_i$ implies $s_ix_i \leq \Delta\mathbf{x}_i\Delta\mathbf{s}_i$. But we also know that $h(\mathbf{w}') - \mu'\mathbf{1} = (\Delta\mathbf{x}_i\Delta\mathbf{s}_i)$ and hence the latter quantity is smaller than $\theta\mu'$. Therefore, we have the following inequalities

$$s_ix_i \leq \Delta\mathbf{x}_i\Delta\mathbf{s}_i \leq \theta\mu' \leq \theta\mu.$$

But as $\|h(\mathbf{w}) - \mu\mathbf{1}\| \leq \mu\theta$, we also know that $s_ix_i \geq (1-\theta)\mu$. Therefore, we have shown that $\theta\mu \geq (1-\theta)\mu$, which hold iff $\theta \geq 1/2$. So to avoid this scenario we take $\theta < 1/2$.

To summarize, the algorithm picks constants $\theta < 1/2$ and τ satisfying (45), and it does the following:

INPUT: $A, \mathbf{b}, \mathbf{c}$, $\mathbf{w}_0$.
OUTPUT: $\mathbf{w}^*$ such that $\gamma(\mathbf{w}^*) \leq \epsilon$.
1. Set $k \leftarrow 0$.
2. If $\gamma(\mathbf{w}_k) = \langle \mathbf{x}_k, \mathbf{s}_k \rangle \leq \epsilon$ then return $\mathbf{w}_k$.
3. Let $\mu_{k+1} \leftarrow \mu_k\,(1 - \tau/\sqrt{n})$.
4. Calculate $\Delta\mathbf{w}_k$ as in (38), (58) and (40).
5. Set $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \Delta\mathbf{w}_k$, $k \leftarrow k+1$ and go to step 2.

The parameter ϵ can be chosen as 2^{-2L} as in the feasibility reduction. We will next show that a starting point $\mathbf{w}_0$ exists with $\mu_0 \leq 2^L$. Substituting these bounds in Lemma 42, we get that $O(\sqrt{n}L)$ iterations suffice for the duality gap to fall below ϵ. As mentioned earlier, the most expensive step is Step 4, which involves inverting a matrix and hence takes $O(n^\omega)$. The presence of bit-size in the algebraic cost of the algorithm means that interior point methods are weakly polynomial time. In the next section, we focus on getting an initial starting point. The approach is similar to simplex method: construct an auxiliary lp, and apply ipm to it to get a feasible point (if any), or detect infeasibility or unboundedness.

12.5 Getting the Initial Point

We will set up an auxiliary LP that will have the following properties:

P(1)1. It is always feasible and bounded so that the ipm approach works for it.

P(2)2. The optimum attained by the ipm method will either tell us something about the primal-dual optimum of the original lp, or show that it is infeasible or unbounded.

Consider the canonical form of lp.

$$\text{maximize } \langle \mathbf{c}, \mathbf{x} \rangle \text{ subject to } A\mathbf{x} \leq \mathbf{b} \tag{46}$$

We have seen that weak duality implies that this lp has an optimum iff the following system has a feasible solution:

$$A\mathbf{x} \leq \mathbf{b}, A^t\mathbf{y} \geq \mathbf{c}, \langle \mathbf{c}, \mathbf{x} \rangle \geq \langle \mathbf{b}, \mathbf{y} \rangle, \mathbf{x}, \mathbf{y} \geq 0.$$

Homogenizing this system of equations by a positive variable $\tau > 0$ gives us the **Goldman-Tucker system**:

$$\begin{aligned} A\mathbf{x} \leq \mathbf{b}\tau,\ \ A^t\mathbf{y} \geq \mathbf{c}\tau, \\ \langle \mathbf{b}, \mathbf{y} \rangle - \langle \mathbf{c}, \mathbf{x} \rangle + \rho = 0, \\ \mathbf{x}, \mathbf{y} \geq 0,\, \tau, \rho \geq 0. \end{aligned} \tag{47}$$

This system is always feasible as origin is a trivial solution. We will be interested in non-trivial solutions. It is clear that if the system above has a solution with $\tau > 0$ then $\mathbf{x}/\tau$ and $\mathbf{y}/\tau$ are the primal and dual

optimum, respectively. It is clear that if a solution of gts has $\rho > 0$ then the original lp has no optimum. These two scenarios cannot happen at the same time because of weak duality, and that is roughly the import of the next result:

LEMMA 45. *No solution of* (47) *can have both* τ *and* ρ *non-zero, i.e., for every solution of gts* $\tau\rho = 0$. *Moreover, the following two cases take place:*

1. *If in a solution* $(\mathbf{x}, \mathbf{y}, \tau, \rho)$, $\tau > 0$ *then* $(\mathbf{x}/\tau, \mathbf{y}/tau)$ *is a primal-dual optimum pair.*
2. *If in a solution* $(\mathbf{x}, \mathbf{y}, \tau, \rho)$, $\rho > 0$ *then either the primal is either infeasible or unbounded.*

Proof. If

$$0 < \tau\rho = \tau(\langle \mathbf{c}, \mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{y}\rangle) = (\langle \tau\mathbf{c}, \mathbf{x}\rangle - \langle \tau\mathbf{b}, \mathbf{y}\rangle) \leq \langle A^t\mathbf{y}, \mathbf{x}\rangle - \langle A\mathbf{x}, \mathbf{y}\rangle = 0$$

giving us a contradiction. Therefore, exactly one of τ or ρ is non-zero.

By weak duality any solution with $\tau > 0$ must have $\rho = 0$. Conversely, any primal-dual optimum pair is a solution of gts with $\tau = 1$ and $\rho = 0$ by strong duality. Therefore, if $\rho > 0$ then $\tau = 0$, and either the primal or the dual has to be infeasible.

Now we will show that if the primal or the dual is infeasible then there is a solution of the system with $\rho > 0$. Suppose the primal is infeasible, i.e., $A\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq 0$, has no solution. Then from Farkas's lemma Corollary 26 there must be a $\overline{\mathbf{y}} \geq 0$ such that $A^t\overline{\mathbf{y}} \geq 0$ and $\langle \overline{\mathbf{y}}, \mathbf{b}\rangle < 0$. Then setting $\mathbf{x} = 0$, $\mathbf{y} = \overline{\mathbf{y}}$, $\tau = 0$ and $\rho = -\langle \mathbf{b}, \overline{\mathbf{y}}\rangle > 0$ is a solution of (47). **Q.E.D.**

Thus solutions of gts gives us a way of answering P1 and P2. Note, however, that we do not have a feasible solution in the interior (i.e., with strictly positive coordinates) as one of τ or ρ is always zero, and hence on the boundary. But to apply ipm we need feasible points in the interior. We will overcome this by forcing $\mathbf{1}$ to be a solution. Let us rewrite (47) in the form $M_0\mathbf{u} \leq 0$, where

$$M_0 := \begin{bmatrix} 0 & A & -\mathbf{b} \\ -A^t & 0 & \mathbf{c} \\ \mathbf{b}^t & -\mathbf{c}^t & 0 \end{bmatrix}$$

and $\mathbf{u} = (\mathbf{y}, \mathbf{x}, \tau)$. Note that M_0 is skew-symmetric, i.e., $M_0^t = -M_0$, and if A was $m \times n$ then M_0 is $k \times k$, where $k := m + n + 1$. But we still don't have our lp with $\mathbf{1}$ as a feasible point. To do that consider, let $\mathbf{r} := \mathbf{1} + M_0\mathbf{1}$ and define the following $(k+1) \times (k+1)$ extended matrix

$$M := \begin{bmatrix} M_0 & -\mathbf{r} \\ \mathbf{r}^t & 0 \end{bmatrix}.$$

Note that M is also skew-symmetric. Let $\mathbf{q} := (0, 0, , \ldots, , 0, k+1)$ and $\mathbf{v} := (\mathbf{u}, \sigma) = (\mathbf{y}, \mathbf{x}, \tau, \sigma)$ as the new set of variables. Then our desired lp is the following:

$$\text{maximize } -\langle \mathbf{q}, \mathbf{v}\rangle \text{ subject to } M\mathbf{v} \leq \mathbf{q},\ \mathbf{v} \geq 0. \tag{48}$$

We claim that the all ones vector of dimension $(k+1)$ is feasible for this lp, i.e., $M\mathbf{1} \leq \mathbf{q}$. This would follow if $M_0\mathbf{1} - \mathbf{r} \leq \mathbf{0}$ and $\langle \mathbf{r}, \mathbf{1}\rangle \leq k+1$; the former follows from the definition of $\mathbf{r}$; for the latter, observe that

$$\langle \mathbf{r}, \mathbf{1}\rangle = k + 1 + \langle M_0\mathbf{1}, \mathbf{1}\rangle = \langle \mathbf{1}, M_0^t\mathbf{1}\rangle = \langle \mathbf{1}, -M_0\mathbf{1}\rangle = \langle \mathbf{1}, \mathbf{1} - \mathbf{r}\rangle,$$

were the second last equation uses the skew-symmetric nature of M_0; moving $-\langle \mathbf{r}, \mathbf{1}\rangle$ to the LHS, we obtain that $2\langle \mathbf{r}, \mathbf{1}\rangle = 2(k+1)$. This lp is even more interesting than it first appears. Let us consider its dual:

$$\text{minimize } \langle \mathbf{q}, \mathbf{w}\rangle \text{ subject to } M^t\mathbf{w} \geq -\mathbf{q},\ \mathbf{w} \geq 0. \tag{49}$$

But as M is skew-symmetric $M^t = -M$, therefore, the above is equivalent to

$$\text{minimize } \langle \mathbf{q}, \mathbf{w}\rangle \text{ subject to } M\mathbf{w} \leq \mathbf{q},\ \mathbf{w} \geq 0.$$

But minimizing $\langle \mathbf{q}, \mathbf{w}\rangle$ is the same as maximizing $-\langle \mathbf{q}, \mathbf{w}\rangle$, and hence the dual above is the same as the primal, i.e., the lp (48) is **self-dual**.

Let $\mathbf{z}$ be the set of slack variables for (48), i.e., $\mathbf{z} = \mathbf{q} - M\mathbf{v}$. Consider the equational form of (48). Then a solution $(\mathbf{v}, \mathbf{z})$ of the equational form is called **strictly complementary** if for all $j = 1, \ldots, k+1$ either $v_j > 0$ or $z_j > 0$. This key notion is required because of the following:

LEMMA 46. *The lp* (48) *has the following properties:*

1. *It is feasible and bounded.*
2. *Every optimum solution* $(\mathbf{u}^*, \sigma^*)$ *has* $\sigma^* = 0$*, and hence* $\mathbf{u}^*$ *is a solution of gts.*
3. *Every strictly complementary optimum solution gives a solution of gts with either* $\tau > 0$ *or* $\rho > 0$*.*

Proof.

1. Note that $\mathbf{0}$ is always a solution of (48) and since it is self-dual, also of its dual. Therefore, the primal is feasible and bounded.
2. In fact $\mathbf{0}$ is an optimum as the value of the primal and the dual objective function is the same, namely zero. If the σ-component is not zero in an optimum solution then the primal objective function is strictly positive, so in all optimum solutions the σ-component is zero. Once the σ-component is zero, it is clear that $M_0\mathbf{u} \leq 0$ implies $\mathbf{u}$ is a solution of gts.
3. Note that the third component of $\mathbf{v}$ is τ and the corresponding slack variable is ρ. So strict complementarity implies that either $\tau > 0$ or $\rho > 0$.

Q.E.D.

Hence to solve the original lp we need a strict complementary solution of (48). The claim that ipm starting from a suitable point will converge to such a solution. We won't see the full proof. Let us look at the ef form of (48)

$$\text{maximize } -\langle \mathbf{q}, \mathbf{v} \rangle \text{ subject to } M\mathbf{v} + \mathbf{z} = \mathbf{q},\ \mathbf{w}, \mathbf{z} \geq 0.$$

We will consider the solutions to the equivalent form of (33) to this system. It is clear that $A = [M|I_{k+1}]$, $\mathbf{b} = \mathbf{q}$, $\mathbf{c} = (-\mathbf{q}, \mathbf{0})$ and $\mathbf{x} = (\mathbf{v}, \mathbf{z})$. Let $\mathbf{v}'$ stand for $\mathbf{y}$ and $(\mathbf{z}', \mathbf{z}'')$ for $\mathbf{s}$. Then the equivalent form of (33) for the lp above can be derived as follows. The system $A\mathbf{x} = \mathbf{b}$ is clearly $M\mathbf{v} + \mathbf{z} = \mathbf{q}$. The system of equations $A^t\mathbf{y} - \mathbf{s} = \mathbf{c}$ gives us two systems: $M^t\mathbf{v}' - \mathbf{z}' = -\mathbf{q}$ and $\mathbf{v}' = \mathbf{z}''$; using the skew-symmetric property, the first one is $M\mathbf{v}' + \mathbf{z}' = \mathbf{q}$, and for the second $\mathbf{z}''$ can be discarded and replaced with the constraint $\mathbf{v}' \geq 0$. The constraints $s_j x_j = \mu$ are $v_j z'_j = \mu$ and $z_j z''_j = z_j v'_j = \mu$. Thus the optimum of (48) is a solution to the following system for $\mu > 0$:

$$\begin{aligned} &M\mathbf{v} + \mathbf{z} = \mathbf{q} \\ &M\mathbf{v}' + \mathbf{z}' = \mathbf{q} \\ &v_j z'_j = v'_j z_j = \mu, \text{ for all } j = 1, \ldots, k+1 \\ &\mathbf{v}, \mathbf{v}', \mathbf{z}, \mathbf{z}' > 0. \end{aligned} \tag{50}$$

What is a solution to this system? We have already seen that $\mathbf{1}$ satisfies the first equation; the same argument works for the second; thus, $\mathbf{v} = \mathbf{v}' = \mathbf{z} = \mathbf{z}' = \mathbf{1}$ is a solution to this system for $\mu = 1$. Hence we have an initial point on the primal dual central path for the system above. The system above can be further simplified. Since it has a unique solution $(\mathbf{v}, \mathbf{z}, \mathbf{v}', \mathbf{z}')$ for a given μ, and interchanging $(\mathbf{v}, \mathbf{z})$ and $(\mathbf{v}', \mathbf{z}')$ also gives us a system, it suffices to find the solutions of the following simpler system:

$$\begin{aligned} &M\mathbf{v} + \mathbf{z} = \mathbf{q} \\ &v_j z'_j = \mu, \text{ for all } j = 1, \ldots, k+1 \\ &\mathbf{v}, \mathbf{z} > 0. \end{aligned} \tag{51}$$

To complete the proof we should show that the path converges to a strictly complementary solution of (48). Can the convergence analysis given earlier be used to show that?

13 Combinatorial Optimization

A typical combinatorial optimization problem is of the following form: maximize a linear objective function $\langle \mathbf{c}, \mathbf{x} \rangle$ over a finite set S. Egs, are minimum weight spanning trees, shortest paths in graphs, maximum weight matching. Generally, S is large so a brute force enumeration is not good enough. Our aim is to find a poly time algorithm for the optimization problem. A key idea is to reduce it to an lp. But how do we do that? Since the objective function is linear, we know that the maximum is attained on the boundary of the convex hull of CH(S). This is the classical approach to combinatorial optimization proposed by Edmonds, Ford-Fulkerson and others. How does this give us an lp? The CH(S) is bounded and hence a polytope. Therefore, if it can be represented by a system of inequalities $A\mathbf{x} \leq \mathbf{b}$ then we have that the maximizing $\langle \mathbf{c}, \mathbf{x} \rangle$ on S is equivalent to maximizing $\langle \mathbf{c}, \mathbf{x} \rangle$ on the set $\{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}\}$, which is a linear programming problem. If the size of A is polynomially bounded then any standard poly time lp solver gives us a solution poly time algorithm for combinatorial optimization. We will see some examples where this approach has proven beneficial.

13.1 Matchings

Let $G = (V, E)$ be an undirected graph. A **matching** M of G is a subset of edges such that no two edges in M share a vertex. A matching M is called **perfect** if every vertex in V is incident on some edge in M. Suppose that every edge has a weight associated with it, i.e., there is a function $w : E \to \mathbb{R}_{\geq 0}$. Then with a matching M we can associate a weight $w(M)$ defined as the sum of the weights of all the edges in M. A natural optimization problem is to find a maximum weight matching in a graph G. We will start by formulating this as an LP.

Let's associate a variable x_e with each edge e, which can take value in $\{0, 1\}$. For every vertex v, at most one of the variables x_e is equal to one amongst the edges e incident on v; this can be captured as $\sum_{e \in I(v)} x_e \leq 1$. Thus our LP is:

$$\begin{aligned} &\text{maximize } \sum_{e \in E} w_e x_e \\ \text{s.t. for all } v \in X \cup Y, &\sum_{e \in I(v)} x_e \leq 1, \\ &\text{and for all } e \in E,\ x_e \in \{0, 1\}\,. \end{aligned} \tag{52}$$

A more succinct way to express this is to use the vertex-edge incidence matrix $A = [a_{ij}]$, where $a_{ij} = 1$ iff v_i is an endpoint of e_j. Thus every column of A has exactly two entries as 1. Using the incidence matrix, the LP above can be written as

$$\text{maximize } \sum_{e \in E} w_e x_e \text{ s.t. } A\mathbf{x} \leq \mathbf{1}, \text{ and for all } e \in E,\ x_e \in \{0, 1\}\,.$$

Instead of restricting x_e to $\{0, 1\}$ we allow it to take values as positive integers, since the constraint corresponding to a vertex will imply that x_e is either 0 or 1. Thus we finally get the following LP for finding the maximum matching:

$$\text{maximize } \sum_{e \in E} w_e x_e \text{ s.t. } A\mathbf{x} \leq \mathbf{1},\ \mathbf{x} \in \mathbb{Z}^m_{\geq 0}. \tag{53}$$

The LP above is called an **integer linear program** since the variable takes integral values. Note that so far we haven't used the fact that G is a bipartite graph crucially; the derivation above actually applies to any graph. Note that every matching satisfies these inequalities. Our set S is the set of all matchings. With every matching M, we associate its characteristic vector $\chi_M \in \{0, 1\}^{|E|}$. The set S is the collection of these characteristic vectors, and its convex hull is called the **matching polytope**. What we have stated is that χ_M belong to the feasible set, i.e., $\chi_M \in \{\mathbf{x} : A\mathbf{x} \leq \mathbf{1}\}$. Can it be that this set is matching polytope? Consider a triangle $\triangle$. Then the characteristic vectors of the matchings are $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$, i.e., the three standard unit vectors. However, the vectors $(1/2) \cdot \mathbf{1}$ also satisfies the constraints $A\mathbf{x} \leq \mathbf{1}$, $\mathbf{x} \geq 0$, but it is clearly not in the matching polytope. This argument fails for an even cycle, because $(1/2) \cdot \mathbf{1}$ is a convex combination of the two matchings formed by either picking the even numbered or odd numbered

edges. But recall that a graph with only even cycles is a bipartite graph. So perhaps the polytope defined by the incidence matrix is a matching polytope for bipartite graphs. We next show that this is indeed the case.

The special property that the incidence matrix A of a bipartite graph has is that it is **totally unimodular**, i.e., the determinant of *any submatrix* of A is in $\{-1, 0, 1\}$. In particular, the 1×1 determinants, namely the entries of A, are 0, 1, or -1. Such matrices have the following property:

LEMMA 47. *Given a $m \times n$ tum A and $\mathbf{b} \in \mathbb{Z}^n$, the polyhedra $P := \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}$ is integral, i.e., the vertices of the polyhedra are in $\mathbb{Z}^n$.*

Proof. Assume $m > n$ (otherwise?). We know that the vertices of P are obtained as some n lid constraints being satisfied as equalities. Suppose $I \subseteq [m]$, $|I| = n$, is such a set of lid constraints and A_I the corresponding submatrix of A. Then the corresponding vertex is the solution to the equation $A_I\mathbf{x}_I = \mathbf{b}$. From Cramer's rule we know that the solution to this equation is of the form $\det(A'_I)/\det(A_I)$, where A'_I is obtained by replacing some column of A_I by $\mathbf{b}$. Since A is tum, and A_I is a non-singular submatrix of A we know that $\det(A_I) = \pm 1$. **Q.E.D.**

THEOREM 48. *The incidence matrix of a bipartite graph is totally unimodular.*

Proof. The proof is by induction; the base case is obvious as all entries are 0, 1. Let Q be an $k \times k$ submatrix of A. There are three cases to consider:

1. Q has a column that is all zero.
2. Q has a column that has exactly one 1.
3. All columns of Q have exactly two 1's. Partition the rows of Q into two sets: those corresponding to vertices in X and those in Y. Consider the k-dimensional vector $\mathbf{v}$ that has all 1's at the indices corresponding to vertices of X. The product $\mathbf{v}^tQ$ has all 1's, because if we only add the rows corresponding to vertices in X, every column has exactly one 1. Similarly for the rows corresponding to vertices in Y. Thus there are two distinct vectors $\mathbf{v}$ and $\mathbf{w}$ such that $\mathbf{v}^tQ = \mathbf{w}^tQ$, which implies that the rows of Q are linearly dependent, i.e., $\det(Q) = 0$.

Q.E.D.

In particular, the vertices of the polyhedra $\{\mathbf{x} : A\mathbf{x} \leq \mathbf{1}, \mathbf{x} \geq 0\}$ are all vectors of the form $\{0, 1\}^m$, and each such vector corresponds to a matching. Thus we have shown the following:

LEMMA 49. *Given a bipartite graph G, if A is its incidence matrix, then the matching polytope of G is the set $\{\mathbf{x} : A\mathbf{x} \leq \mathbf{1}, \mathbf{x} \geq 0\}$.*

Using this insight, we can actually prove König's theorem for bipartite graphs: size of maximum matching is equal to the size of the minimum vertex cover. Consider the dual of the **LP-relaxation** of the ilp (53) where the weights are all ones:

$$\text{minimize} \sum_{v \in X \cup Y} y_v \text{ s.t. } A^t\mathbf{y} \geq \mathbf{1},\ \mathbf{y} \geq 0. \tag{54}$$

Since A^t is also tum, it follows that the vertices of this polytope are also vectors in $\{0, 1\}^n$. The rows of the constraint $A^t\mathbf{y} \geq 1$ are of the form $y_u + y_v \geq 1$, for an edge $uv \in E$, and since $\mathbf{y} \in \{0, 1\}^n$ it follows that at least one of the endpoints of an edge is picked by a feasible solution, i.e., one y_u of y_v is greater than 1. Thus all vertex covers are feasible for this lp, and the value of the objective function is the size of the vertex cover. The strong duality says that the minimum value of this objective function, i.e., the minimum vertex cover, is equal to the maximum value of the primal objective function, which would be the size of the maximum matching.

What can we say about the matching polytope for non-bipartite graphs? As we have shown, the polytope defined by the incidence matrix alone is not sufficient as it contains extraneous points. But perhaps there is a way of adding some extra constraints and cutting this polytope to get the matching polytope. Edmonds showed that this is indeed possible in his celebrated and pioneering work on matchings.

13.2 Perfect Matching and Matching Polytope for Non-bipartite Graphs

In this section, we first consider perfect matchings for our graph. It is clear that G has a perfect matching iff number of vertices are even. If A is the incidence matrix of G, then we have seen earlier that every perfect matching satisfies the following system:

$$A\mathbf{x} = \mathbf{1} \text{ and } \mathbf{x} \geq 0.$$

However, as observed earlier, this set still contains points that do not belong to the matching polytope. We will impose additional constraint to get the perfect matching polytope. What kind of constraints? For a subset $W \subset V$, let $\delta(W)$ denote the edges going from W to $\overline{W} := V - W$, and $E(W)$ be the edges with both endpoints in W. Observe that if W has odd size then any perfect matching M has to match a vertex of W with a vertex in $\overline{W}$, i.e., a matching has to pick an edge in $\delta(W)$. Edmonds showed that this property along with the earlier constraints are enough to define the perfect matching polytope:

THEOREM 50 (Edmond's Perfect Matching Polytope). *The perfect matching polytope $\mathcal{P}(G)$ of a graph $G = (V, E)$ is defined by the following constraints:*

1. *Non-negativity: $x_e \geq 0$, for all $e \in E$,*
2. *Vertex constraints: $\mathbf{x}(\delta(v)) := \sum_{e \in \delta(v)} x_e = 1$, and*
3. **Odd-cut constraints**: *$\mathbf{x}(\delta(W)) := \sum_{e \in \delta(W)} x_e \geq 1$, for all odd sized sets $W \subseteq V$.*

Wlog we can assume that $|V| \geq 6$, since $|V|$ is even and for the odd-cut constraints to be different from the vertex constraints we want $|W|, |\overline{W}| \geq 3$ (note that if either one of them has size one then the constraint is not an odd-cut constraint but a vertex constraint).

Let $Q(G)$ be the polytope (why is it bounded) defined by the three constraints above and $\mathcal{P}(G)$ be the perfect matching (pm) polytope. As argued before, $\mathcal{P}(G) \subseteq Q$. Edmond's key result was to show the converse. The proof is via induction. The idea is to show that all the vertices of $Q(G)$ can be expressed as a convex combination of incidence vectors of perfect matchings. Suppose the claim is true for all graphs G' that have fewer edges or vertices that G, i.e., $Q(G') = \mathcal{P}(G')$. Let $\mathbf{x}^*$ be a vertex of $Q(G)$. The following claim tells us the interesting vertices of $Q(G)$ and some properties that G can be assumed to satisfy:

LEMMA 51. *Wlog we can assume the following about G and $\mathbf{x}^*$:*

1. *For all $e \in E$, $0 < x^*_e < 1$, and*
2. *The vertices in G have degree ≥ 2, and there exists a vertex with degree > 2.*

Proof. We show if either of the above assumptions fail then either $\mathbf{x}^* \in \mathcal{P}(G)$ or $Q(G) \subseteq \mathcal{P}(G)$.

1. If $x^*_e = 0$, for some edge e, then let $\mathbf{x}'$ be the vector obtained from $\mathbf{x}^*$ by dropping the entry corresponding to e and $G' = G - e$. We first claim that $\mathbf{x}' \in Q(G')$: the non-negativity constraint trivially holds; if v is a vertex with e incident on it then the second constraint is true because $x^*_e = 0$; similarly, the contribution of e to any set of cut edges is also zero, and hence the third constraint is true for all odd-sized subsets. Now as G' is smaller than G, by induction hypothesis $Q(G') = \mathcal{P}(G')$, and hence $\mathbf{x}'$ is a convex combination of matchings of G'. Re-introducing the component corresponding to e in the incidence vectors of these matchings, we get that $\mathbf{x}^* \in \mathcal{P}(G)$.

2. If $x^*_e = 1$, for some edge e, then let $\mathbf{x}'$ be the projection of $\mathbf{x}^*$ obtained by dropping e, the endpoints of e, and edges incident on these endpoints from G; let G' be the resulting graph. We again claim that $\mathbf{x}' \in Q(G')$. Observe that dropping the edges incident on the endpoints of $e = (v, w)$ does not matter because as $\sum_{e' \in \delta(v)} x^*_{e'} = x^*_e + \sum_{e' \in \delta(v) - e} x^*_{e'} = 1$ implies that $x^*_{e'} = 0$ for all other edges e' incident on v (similarly for w). Therefore, these edges do not contribute in the third constraint for any odd-sized set. Hence $\mathbf{x}' \in Q(G')$, but $Q(G') = \mathcal{P}(G')$, as G' is smaller than G and applying the induction hypothesis. Thus $\mathbf{x}'$ can be expressed as a convex combination of matchings in G'; add the component corresponding to e in these matchings, and we get that $\mathbf{x}^*$ is a convex combination of matchings in G, which implies $\mathbf{x}^* \in \mathcal{P}(G)$.

3. If G has a degree one vertex then for the incident edge e, $x^*_e = 1$, but from the argument above we know that in this case $\mathbf{x}^* \in \mathcal{P}(G)$.

4. If all vertices have degree exactly two, then G is a cycle. It cannot be an odd-cycle as we have an even number of vertices. Therefore, it must be an even cycle and hence a bipartite graph. Since we have only added more constraints, $Q(G) \subseteq \{\mathbf{x} | A\mathbf{x} = 1, \mathbf{x} \geq 0\} = \mathcal{P}(G)$, where A is the incidence matrix of G, and the second equality follows from earlier results.

Q.E.D.

Since $2|E| = \sum_{v \in V} \deg(v)$, and by our assumptions on G above we know that the average degree is > 2, it follows that $|E| > |V|$. Now the vertex $\mathbf{x}^*$ of $Q(G)$ is a bfs that has no zero entry, therefore, it must satisfy $|E|$ lid constraints from the vertex and odd-cut constraints as equalities. But there are only $|V|$ vertex-constraints, so there must exists an odd-sized set W such that $\sum_{e \in \delta(W)} x^*_e = 1$.

Let G' be the graph obtained by contracting W to a single vertex u', and G'' be the graph obtained by contracting $\overline{W}$ to a single vertex u''; note that contraction might introduce parallel edges, but we keep all of them. Let $\mathbf{x}'$ be the projection of $\mathbf{x}^*$ wrt G', i.e., the vector obtained by removing the entries corresponding to contracted edges; similarly, define $\mathbf{x}''$ wrt G''.

Claim 1. *The vector $\mathbf{x}' \in Q(G')$ and $\mathbf{x}'' \in Q(G'')$.*
Proof. We only show the claim for $\mathbf{x}'$; the claim for $\mathbf{x}''$ follows similarly, along with the observation that $\overline{W}$ is also odd.

The non-negativity constraint is trivially satisfied by $\mathbf{x}'$. The vertex-constraint is clearly satisfied for all vertices in $\overline{W}$; for u', it follows from the following:

$$\sum_{e \in \delta(u')} x'_e = \sum_{e \in \delta(W)} x^*_e = 1.$$

For the third constraint, consider an odd-sized subset U of G'; if U does not contain u' then $\mathbf{x}'(\delta(U)) = \mathbf{x}^*(\delta(U)) \geq 1$. Otherwise, the set $\delta(U)$ is the same as the set $\delta((U - u') \cup W)$, and since $(U - u') \cup W$ is an odd-sized subset of V, $\mathbf{x}'(U) = \mathbf{x}^*((U - u') \cup W) \geq 1$. **Q.E.D.**

As G' and G'' are smaller graphs than G, it follows that $\mathbf{x}' \in \mathcal{P}(G')$ and $\mathbf{x}'' \in \mathcal{P}(G'')$. We will use this observation to express $\mathbf{x}^*$ as a convex combination of some matchings of G. Let the convex combinations for $\mathbf{x}'$ and $\mathbf{x}''$ be given by $\mathbf{x}' = \sum_{M'} \lambda'_{M'} \chi_{M'}$ and $\mathbf{x}'' = \sum_{M''} \lambda''_{M''} \chi_{M''}$. We will massage this combination into something more amenable. Note that $\mathbf{x}^*$ has rational entries (follows from Cramer's rule); this also applies to the entries of $\mathbf{x}'$ and $\mathbf{x}''$. Since the λ's and λ''s are solutions to linear equations with rational entries, they are also rational numbers; moreover, by appropriate scaling we can assume that denominator of these rational numbers is the same number D. Thus $\lambda'_{M'} = a_{M'}/D$ and $\lambda''_{M''} = b_{M''}/D$, where $a_{M'}, b_{M''} \in \mathbb{N}$. Since $\sum_{M'} \lambda'_{M'} = \sum_{M''} \lambda''_{M''} = 1$, it follows that $\sum_{M'} a_{M'} = \sum_{M''} b_{M''} = D$. Therefore,

$$\mathbf{x}' = \frac{1}{D} \sum_{M'} a_{M'} \chi_{M'} = \frac{1}{D} \sum_{i=1}^{D} \chi'_i \tag{55}$$

where $S' := \{\chi'_i, i = 1, \ldots, D\}$ is a multiset of incidence vectors $\chi_{M'}$ such that $\chi_{M'}$ has $a_{M'}$ copies in the set. Similarly, we can express

$$\mathbf{x}'' = \frac{1}{D} \sum_{M''} b_{M''} \chi_{M'} = \frac{1}{D} \sum_{i=1}^{D} \chi''_i, \tag{56}$$

where $S'' := \{\chi''_i, i = 1, \ldots, D\}$ is a multiset of incidence vectors $\chi_{M''}$ such that $\chi_{M''}$ has $b_{M''}$ copies in the set. Note, however, that $\mathbf{x}'$ and $\mathbf{x}''$ share the entries corresponding to $\delta(W)$ in $\mathbf{x}^*$.

Consider an $e \in \delta(W)$. How many M's are there that contain e? Comparing the e-th coordinate on both sides of (55) we have

$$Dx'_e = \sum_{M'} a_{M'} [e \in M'] = |\{M' \in S' : e \in M'\}|,$$

i.e., the number of pms in S' containing e is Dx'_e. Similarly, the number of pms in S'' containing e is Dx''_e. But $x'_e = x''_e = x^*_e$, and hence the number of pms in S' containing e is the same as the number of pms in S''

containing e. Number the edges in $\delta(W)$ from $e_1, \ldots, e_k$, where $k := |\delta(W)|$. Then write the D matchings of S' and S'' in two columns such that the first $Dx^*_{e_1}$ entries all contain e_1, the next $Dx^*_{e_2}$ contain e_2 and so on; see Figure 8. Note that $D\sum_{e\in\delta(W)} x^*_e = D$, therefore, the above procedure exhausts all the matchings on both sides. If $M_i := M'_i \cup M''_i$ then clearly it is a pm in G. Now we claim the following:

CLAIM 2. *The vector $\mathbf{x}^*$ is a convex combination of M_i's, i.e., $\mathbf{x}^* = \sum_{i=1}^{D} \chi_{M_i}/D$.*

Proof. For coordinates of $\mathbf{x}^*$ not in $\delta(W)$, the claim follows directly from (55) and (56). Consider the entry $x^*_{e_1}$, where e_1 is the first edge of $\delta(W)$ in the ordering chosen earlier. We know that this entry can only come from the pms $M_1, \ldots, M_{Dx^*_{e_1}}$, and clearly

$$x^*_{e_1} = \sum_{i=1}^{Dx^*_{e_1}} \frac{[\chi_{M_i}]_{e_1}}{D} = \sum_{i=1}^{D} \frac{[\chi_{M_i}]_{e_1}}{D}.$$

This is also evident from Figure 8. **Q.E.D.**

The claim shows that all vertices $\mathbf{x}^*$ of $Q(G)$ are in $\mathcal{P}(G)$, therefore, $Q(G) \subseteq \mathcal{P}(G)$ as desired.

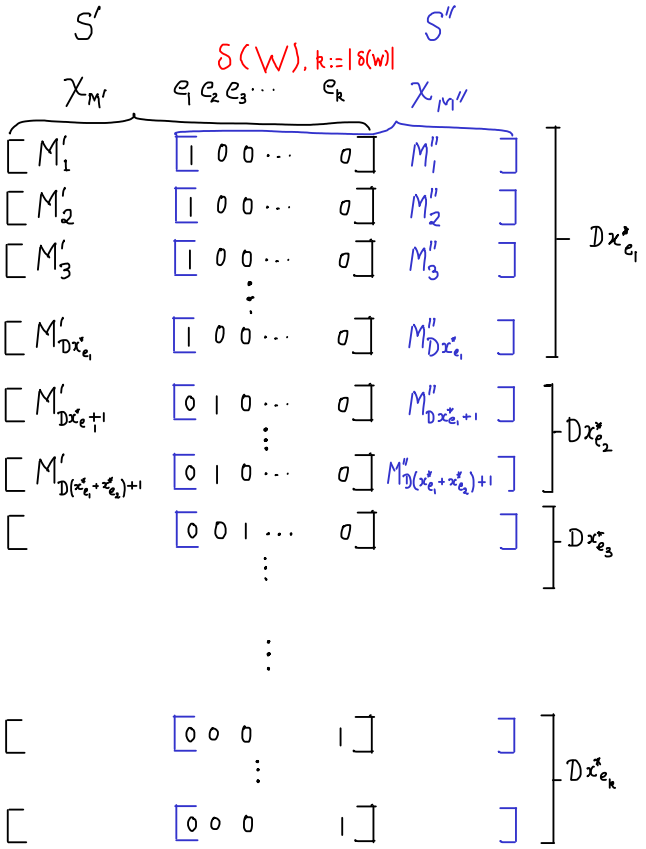


Figure 8: Matchings of G constructed from matchings of G' and G''

How about matchings in general? The **matching polytope** of G is defined as the convex hull of all the characteristic vectors of matchings of G. Consider an odd-sized subset W of V. Given a matching M of G, how many edges can we have in M with both their endpoints in W? Since each such edge will pick a distinct pair of vertices, it is clear that $2|M \cap E(W)| \leq |W|$, or in other words, $|M \cap E(W)| \leq \lfloor |W|/2 \rfloor$. As is the case, the necessary condition also turns out to be sufficient.

THEOREM 52 (Edmond's Matching Polytope). *The matching polytope of a graph $G = (V, E)$ is defined by the following constraints:*

1. $x_e \geq 0$, *for all* $e \in E$,
2. $\sum_{e \in \delta(v)} x_e \leq 1$, *and*
3. **Blossom-inequalities**: $\sum_{e \in E(W)} x_e \leq \lfloor |W|/2 \rfloor$, *for all odd-sized sets* $W \subseteq V$.

13.3 Approximation Algorithms – Vertex Cover

Sometimes lp relaxation of integer programs for combinatorial optimization problems doesn't give us the exact solution but a good enough approximation, especially for computationally hard problems, such as the one in this section. Let's consider the case of minimum vertex cover. As an ilp it can be formulated as follows:

$$\text{minimize} \sum_{v \in V} x_v \text{ s.t. } x_u + x_v \geq 1 \text{ for all edges } e = (u, v), x_v \in \{0, 1\}, \text{ for all } v \in V. \tag{57}$$

It is known that finding the minimum vertex cover is an NP-hard problem. However, using the lp-relaxation of the integer program above we can easily get a 2-approximation algorithm. What does it mean? We can get a vertex cover whose size is not more than twice the size of the smallest vertex cover. Pretty good! Let's see how. Suppose $\mathbf{x}^*$ is an optimum solution to the lp-relaxation:

$$\text{minimize} \sum_{v \in V} x_v \text{ s.t. } x_u + x_v \geq 1 \text{ for all edges } e = (u, v), 0 \leq x_v \leq 1, \text{ for all } v \in V.$$

Define $S := \{v : x^*_v \geq 1/2\}$. We claim that S forms a vertex cover. This is because for each edge uv, $x^*_u + x^*_v \geq 1$ implies that either $x^*_u \geq 1/2$ or $x^*_v \geq 1/2$ (note that both endpoints can be in S). Let $\overline{\mathbf{x}}$ be a vector that achieves the optimum for the ip given by (57), and let $S_{\text{OPT}} := \{v : \overline{x}_v = 1\}$ be the corresponding vertex cover. Our claim is that $|S| \leq 2|S_{\text{OPT}}|$. First note that as $\mathbf{x}^*$ is an optimum over a larger set we have

$$\sum_{v \in V} x^*_v \leq \sum_{v \in V} \overline{x}_v.$$

Moreover,

$$|S| \leq 2 \sum_{v \in S} x^*_v \leq 2 \sum_{v \in V} x^*_v,$$

but from the inequality above it follows that

$$|S| \leq 2 \sum_{v \in V} x^*_v \leq 2 \sum_{v \in S_{\text{OPT}}} x^*_v = 2|S_{\text{OPT}}|.$$

13.4 Approximation Algorithms – Max Cut

Given an undirected graph $G = (V, E)$, a **cut** is a pair $(W, \overline{W})$, where $W \subseteq V$. The **edge set of the cut** is the set $\delta(W)$. The size of the cut is $|\delta(W)|$. The problem of maximum cut is to find a largest sized cut in a graph G. Let $\text{OPT}_{\text{CUT}}(G)$ be the size of a maximum cut for G.

We start with a randomized algorithm, that too a most naive one: pick a set S of vertices, where each vertex is included in S with probability half and independently of all the other vertices. Our claim is that the expected size of the cut is at least half the size of the optimum value, $\text{OPT}_{\text{CUT}}(G)$. The size of the cut $|\delta(S)|$ induced by S is a random variable. We want to bound its expectation $\mathtt{E}[|\delta(S)|]$. By linearity of expectation (and using indicator variables for each edge) it follows that this is the same as $\sum_{e \in E} \mathtt{Pr}\{e \in \delta(S)\}$. The probability that $e = (u, v)$ is in $\delta(S)$ is if either $u \in S$ and $v \notin S$, which happens with probability $1/4$, and the symmetric situation that $v \in S$ and $u \notin S$, which also happens with probability $1/4$; thus the probability that $e \in \delta(S)$ is $1/4 + 1/4 = 1/2$. Therefore,

$$\mathtt{E}[|\delta(S)|] = \sum_{e \in E} \mathtt{Pr}\{e \in \delta(S)\} = \sum_{e \in E} \frac{1}{2} = \frac{|E|}{2}.$$

But, clearly, $|E| \geq \mathrm{OPT}_{\mathrm{CUT}}(G)$, giving us the desired result.

There is a derandomized version of the algorithm above The next big breakthrough in approximation algorithms for max cut was Goemans-Williamson randomized algorithm from 1994, which has an approximation factor of ~ 0.878. We next see how this was obtained using semidefinite programming (sdp). For the moment we do not define what sdp's are, but it suffices to say they are generalizations of lp having something to do with positive semidefinite matrices, and for the case of max cut we will see the required sdp.

Let's try to start with an expressing max cut as a constrained optimization problem. Label the vertices in V from $1, \ldots, n$ and associate a variable y_i with the ith vertex in V. Unlike the cases earlier, we restrict y_i to the set $\{-1, 1\}$. For some assignment of variables $y_i \in \{-1, 1\}$, define $S := \{i : y_i = 1\}$. The reason for choosing $\{-1, 1\}$ is because the term

$$\frac{1 - y_i y_j}{2}$$

gives us the contribution of an edge $\{i, j\}$ to $|\delta(S)|$: it vanishes if $i, j \in S$ or $i, j \in \overline{S}$, but is exactly equal to one when one of them is in S and the other in $\overline{S}$. Thus, if we take the sum over all pairs $\{i, j\}$ we have

$$|\delta(S)| = \sum_{\{i,j\} \in E} \frac{1 - y_i y_j}{2}. \tag{58}$$

Using this succinct representation, the max cut problem can be stated as follows:

$$\text{maximize} \sum_{\{i,j\} \in E} \frac{1 - y_i y_j}{2}, \text{ s.t. } y_i \in \{-1, 1\}, \text{ for } i = 1, \ldots, n. \tag{59}$$

Now, this is not a lp because the objective function is non-linear. We will relax this problem in a different, and rather counterintuitive, manner. Note that the set $\{-1, 1\}$ is the sphere $S^0 := \{x : x^2 = 1\}$. So, we can start generalizing the y_i's as elements of sphere in higher dimensions, i.e., as vectors of unit norm in higher dimensions. The dimension that just the job for us is n. So we replace each variable y_i with a vector $\mathbf{u}_i \in \mathbb{R}^n$ of unit norm, i.e., on the unit sphere S^{n-1} in $\mathbb{R}^n$. For instance, if $\mathbf{u}_i := (\mathbf{0}, y_i)$, then (58) can be expressed as $\sum_{\{i,j\} \in E} (1 - \langle \mathbf{u}_i, \mathbf{u}_j \rangle)/2$. So our variables are *vectors* now and the constraint is that they are unit vectors. The optimization problem (59) is subsumed by the following **vector program**:

$$\text{maximize} \sum_{\{i,j\} \in E} \frac{1 - \langle \mathbf{u}_i, \mathbf{u}_j \rangle}{2}, \text{ s.t. } \mathbf{u}_i \in S^{n-1}, \text{ for } i = 1, \ldots, n. \tag{60}$$

The program above is a vector relaxation of the program in (59), and hence has a larger feasible set, and consequently a large value for the optimum than $\mathrm{OPT}_{\mathrm{CUT}}(G)$. Why is this objective function bounded? Note that since the vectors are unit vectors their inner product is the cosine of the angle between them, which is in $[-1, 1]$; therefore, the objective function is in $[0, 1]$. The n^2 numbers $\langle \mathbf{u}_i, \mathbf{u}_j \rangle$ are the entries of the Gram matrix $U^t U$, where the columns of U are $\mathbf{u}_i$, and the diagonal entries are all ones. We know that a matrix of the form $U^t U$ is positive semidefinite. Therefore, we can think of (60) in terms of a variable which is a positive semidefinite matrix:

$$\text{maximize} \sum_{\{i,j\} \in E} \frac{1 - x_{ij}}{2}, \text{ s.t. } x_{ii} = 1, \text{ for } i = 1, \ldots, n, \text{ and } X \succeq 0, \tag{61}$$

where $X \succeq 0$ means that X is positive semidefinite. We have seen that the program (60) is a special case of the program above, but we claim that it is true the other way round. This is because any psd matrix X can be factored as $U^t U$; moreover, as the diagonal entries in X are one, the columns of U are all unit vectors. Thus the feasible set in the two programs is the same. The program in (61) is called a **semidefinite program** (sdp) – the name comes from the fact that the variable is a semidefinite matrix. Thus the optimum value of the program above $\mathrm{SDP}_{\mathrm{OPT}}(G)$ for a graph G is greater than $\mathrm{OPT}_{\mathrm{CUT}}(G)$ for the program in (59). Amazingly, the ellipsoid and interior point methods generalize to obtain a solution X^*

of an sdp in polynomial time. Moreover, we can factorize $X^* = (U^*)^t U^*$, called the Cholesky factorization, in polynomial time to obtain the vectors $\mathbf{u}_i^*$'s. Consequently, we have shown that

$$\mathrm{SDP}_{\mathrm{OPT}}(G) = \sum_{\{i,j\}\in E} \frac{1 - \langle \mathbf{u}_i^*, \mathbf{u}_j^* \rangle}{2} \geq \mathrm{OPT}_{\mathrm{CUT}}(G). \tag{62}$$

Now that we have the vectors $\mathbf{u}_i^*$'s, we want to map them to $S^0 = \{-1, 1\}$ to get our cut. Note that these vectors can be thought of as embedding the vertices of the graph on the unit sphere S^{n-1}. The vertices that will maximize $\mathrm{SDP}_{\mathrm{OPT}}$ are those that are far apart, i.e., the angle between $\mathbf{u}_i^*$ and $\mathbf{u}_j^*$ is close to π; the edges corresponding to such pair of vertices will be our cut edges. Thus to find our cut, we want to find a hyperplane that separates such pairs of vertices. Now as the angle between such pairs is large, intuitively a random hyperplane should do our job. For a $\mathbf{p} \in S^{n-1}$, we can partition the set of vectors $\mathbf{u}_i^*$ into two disjoint sets depending on which side of the hyperplane they are: a $\mathbf{u} \in S^{n-1}$ is mapped to $+1$ if $\langle \mathbf{u}, \mathbf{p} \rangle \geq 0$ and to -1 if $\langle \mathbf{u}, \mathbf{p} \rangle < 0$. How do we choose $\mathbf{p}$? We choose it uniformly at random on the unit sphere S^{n-1} (sample univariate gaussians in each coordinate). Our claim is that such a $\mathbf{p}$ will be able to distinguish the large and small angles between pairs of vectors $\mathbf{u}_i^*$, $\mathbf{u}_j^*$, i.e., it will map the pairs with larger angles to different values. The following result formalizes this intuition:

LEMMA 53. *If $\mathbf{u}, \mathbf{w} \in S^{n-1}$ then the probability that a $\mathbf{p}$ chosen uniformly at random from S^{n-1} maps $\mathbf{u}$, $\mathbf{w}$ to different values is*

$$\frac{1}{\pi} \arccos\langle \mathbf{u}, \mathbf{w} \rangle.$$

Proof. Let θ be the angle between $\mathbf{u}$ and $\mathbf{w}$. What we are claiming is that the probability that $\mathbf{p}$ maps $\mathbf{u}$ and $\mathbf{w}$ to different values is θ/π. Let us see this claim in two-dimensions. It is easy to see that the hyperplane defined by $\mathbf{p}$ separates $\mathbf{u}$ from $\mathbf{w}$ iff it lies in the two wedges of angle θ as shown in Figure 9. So the prob is θ/π, as desired. In higher dimensions, (imagine 3-d) $\mathbf{p}$ separates $\mathbf{u}$ from $\mathbf{w}$ if its image $\mathbf{r}$ in the space spanned by these two vectors is contained in the two wedges. Since $\mathbf{p}$ varies uar on S^{n-1} the direction of $\mathbf{r}$ is uniformly distributed over $[0, 2\pi]$ and hence the result from the planar case applies, giving us the desired probability. See Figure 10 for a 3d-version. **Q.E.D.**

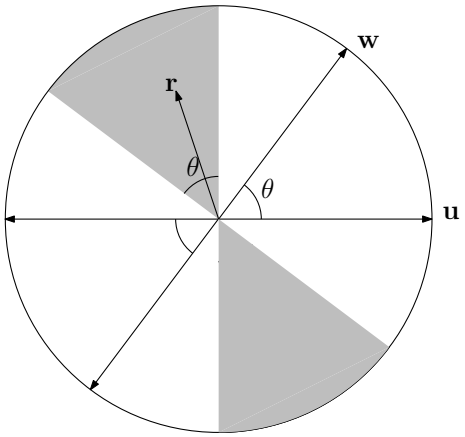


Figure 9: A a hyperplane defined by $\mathbf{p}$ separates $\mathbf{u}$ and $\mathbf{w}$ if its projection $\mathbf{r}$ is in the shaded region.

The result above shows that the expected number of edges in our cut is

$$\sum_{\{i,j\}\in E} \frac{1}{\pi} \arccos\langle \mathbf{u}_i, \mathbf{u}_j \rangle.$$

This quantity, however, does not have any apparent relation to $\mathrm{OPT}_{\mathrm{CUT}}(G)$. However, it can be shown that for $z \in [-1, 1]$,

$$\arccos(z) \geq C_{GW} \left(\frac{1 - z}{2} \right)$$

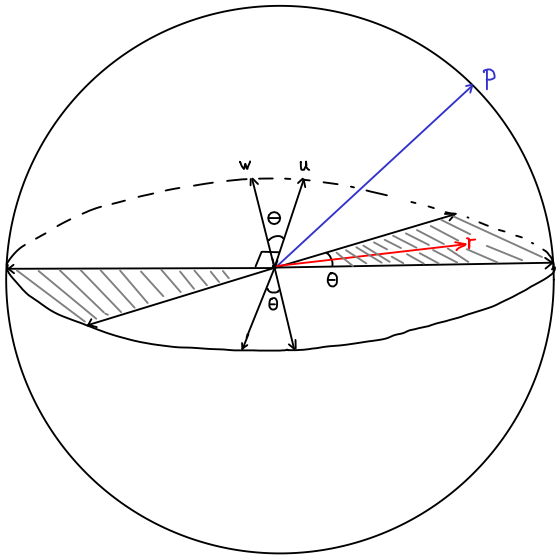


Figure 10: A a hyperplane defined by **p** separates **u** and **w** if its projection **r** is in the shaded region.

for some constant

$$C_{GW} := \min_{z\in[-1,1]} \frac{2\arccos(z)}{(1-z)} \sim 0.8785672. \tag{63}$$

Thus

$$\sum_{\{i,j\}\in E} \frac{1}{\pi}\arccos\langle \mathbf{u}_j^*, \mathbf{u}_j^*\rangle \geq \sum_{\{i,j\}\in E} C_{GW}\left(\frac{1-\langle \mathbf{u}_i^*, \mathbf{u}_j^*\rangle}{2}\right).$$

But from (62) we know that the RHS above is greater than $\mathrm{OPT}_{\mathrm{CUT}}(G)$. Therefore, we have the expected size of the cut is $\geq C \cdot \mathrm{OPT}_{\mathrm{CUT}}(G)$, where C is as defined above.

13.4.1 Tightness of the approximation ratio

There are classes of graphs for which better approximation ratios are known: dense graphs (roughly having quadratically many edges, graphs with bounded maximum degree, graphs where the max-cut is either "large" or "small". The hope of getting an approximation ratio better than C_{GW} in general seems hard: assuming the unique games conjecture and P being different from NP, there are results that show that there cannot be a poly-time algorithm obtaining a ratio better than C_{GW}; assuming only the latter condition, the claim is that there is no poly-time algorithm with a ratio better than $16/17 \sim 0.94$.

If the value computed by the algorithm is A then the result above actually shows that $\frac{A}{\mathrm{SDP}_{\mathrm{OPT}}} \geq C_{GW}$. But as $\mathrm{OPT}_{\mathrm{CUT}} \geq A$, we obtain that the integrality gap $\mathrm{SDP}_{\mathrm{OPT}}/\mathrm{OPT}_{\mathrm{CUT}}$ is bounded by $1/C_{GW}$. Can it be that there are graphs for which the approximation computed by the algorithm is near to C_{GW} even when $\mathrm{SDP}_{\mathrm{OPT}} = \mathrm{OPT}_{\mathrm{CUT}}$, i.e., instances where the sdp gives the true optimum but the algorithm fails to improve on the approximation ratio? It is not hard to come up with graphs where the sdp-opt is the opt; e.g., for bipartite graphs, the true opt and sdp opt would be $|E|$, however, for these cases the algorithm also computes the opt-cut. The following result of Karloff says that there are graphs where the integrality gap is one but the algorithm nearly attains the approximation ratio:

THEOREM 54 (Karloff'99). *For every $\epsilon > 0$, there exists a graph G such that the expected size A of the cut computed by the algorithm satisfies*

$$C_{GW} \leq \frac{A}{OPT_{CUT}} \leq C_{GW} + \epsilon.$$

Let θ_{GW} be the angle z that achieves the ratio C_{GW} in (63). The idea of the proof is to construct a graph G for which the expected size A of the cut is $(\theta_{GW}/\pi)|E|$, the size of the SDP_{OPT} cut is $(h/d)|E|$, for two parameters h, d. The size of OPT_{CUT} will be equal to SDP_{OPT}. Moreover, the ratio h/d will tend to $(1-\cos\theta_{GW})/2$, which means that the ratio $A/\text{OPT}_{\text{CUT}}$ will tend to $2\theta_{GW}/(1-\cos\theta_{GW})$, which is C_{GW}. To ensure $\text{SDP}_{\text{OPT}} = (h/d)|E|$, we ensure that the contribution $(1-\langle \mathbf{u}_i^*, \mathbf{u}_j^*\rangle)/2$ of each edge is equal to h/d. This would follow if $\langle \mathbf{u}_i^*, \mathbf{u}_j^*\rangle = (d-2h)/d$. Suppose $\mathbf{u}_i^*$ and $\mathbf{u}_j^*$ are in $\{-1,1\}^d$ and they differ in exactly h places, then their inner product is $(d-2h)$. But note that their 2-norm is $\sqrt{d}$, therefore, we make them unit vectors by dividing by d. So our graph is the **Hamming graph** with vertex set $V = \{-1,1\}^d$ and an edge between two vertices if their hamming distance is exactly h. The associated unit vector with a vertex $\mathbf{v} \in \{-1,1\}^d$ is $\mathbf{u}_v := \mathbf{v}/d$. These vectors are clearly feasible for the sdp. What is amazing is that they attain the SDP_{OPT}:

LEMMA 55. *The set of vectors $\mathbf{u}_v$ attain the SDP-opt.*

The proof of the lemma above is technical. However, assuming the lemma is true the chain of argument is as follows:

1. As $h/d \to (1-\cos\theta_{GW})/2$, the expected size of the cut of the algorithm tends to $(\theta_{GW}/\pi)|E|$.
2. The SDP-opt is attained for the vectors $\mathbf{u}_v$ and is equal to $(h/d)|E|$.
3. Consider any coordinate hyperplane $x_i = 0$. What is the number of edges crossing it? An edge cuts along h such hyperplanes. Therefore, we expect a hyperplane to cut across $(h/d)|E|$ edges. The OPT-cut is at least as large as this, but it cannot be larger than the SDP-opt. Therefore, OPT-cut is equal to SDP-opt.
4. Hence, the ratio $A/\text{OPT}_{\text{CUT}}$ tends to C_{GW}.

We only show the following result as a partial proof of the lemma above.

LEMMA 56. *Given a graph G, with adjacency matrix A*

$$SDP_{OPT}(G) \leq \frac{1}{2}|E| - \frac{\lambda n}{4}$$

where λ is the smallest eigenvalue of A.

Proof. Let $\mathbf{u}_i^*$ be the solution of sdp-opt. Then the contribution of the an edge (i,j) to the opt is $\frac{1}{2}(a_{ij}(1-\langle \mathbf{u}_i^*, \mathbf{u}_j^*\rangle)/2)$; the half is to take into account the double counting that happens because $a_{ij} = a_{ji}$. Therefore, the value of the opt is

$$\begin{aligned}
\text{SDP}_{\text{OPT}} &= \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}\frac{1-\langle \mathbf{u}_i^*, \mathbf{u}_j^*\rangle}{2} \\
&= \frac{1}{2}|E| - \frac{1}{4}\sum_{i,j}\sum_{k} a_{ij}u_{ik}^*u_{jk}^* \\
&= \frac{1}{2}|E| - \frac{1}{4}\sum_{k}\sum_{i,j} a_{ij}u_{ik}^*u_{jk}^* \\
&= \frac{1}{2}|E| - \sum_{k}\frac{\mathbf{r}_k^t A \mathbf{r}_k}{4},
\end{aligned}$$

where $\mathbf{r}_k$ are the rows of the matrix that has columns as $\mathbf{u}_i^*$. The quadratic form is related to the Rayliegh quotient $\mathbf{x}^t A\mathbf{x}/\|\mathbf{x}\|^2$ of the (symmetric) matrix A, which takes the least value at λ. Therefore, $\mathbf{r}_k^t A\mathbf{r}_k = \lambda\|\mathbf{r}_k\|^2$. But $\sum_k \|\mathbf{r}_k\|^2 = \sum_i \|\mathbf{u}_i^*|^2 = n$. This completes the proof. **Q.E.D.**

Note that equality is attained in the bound above if r_k's are the eigenvectors corresponding to the smallest eigenvalue. The challenging part is to show that the rows of the adjacency matrix of the Hamming graph are the eigenvectors corresponding to the smallest eigenvalue. Therefore, the vectors $\mathbf{u}_v$ defined above do attain the sdp-opt.

14 Semidefinite Programming

The two crucial cooncepts in defining an lp are the underlying vector space $\mathbb{R}^n$, an inner product operator $\langle \cdot, \cdot \rangle$ on this space, and a linear map $A : \mathbb{R}^n \to \mathbb{R}^m$. These three concepts are essential in defining the problem, let's say in its equational form: maximize $\langle \mathbf{c}, \mathbf{x} \rangle$ subject to $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq 0$. With this perspective, we can choose other vector spaces, inner product on these vector space, and a linear map on the vector space to obtain interesting generalizations of lp. One such choice is the following:

1. For the vector space, we take the set SYM_n of all $n \times n$ symmetric matrices with entries in $\mathbb{R}$. It is easy to verify that this is a vector space over $\mathbb{R}$.

2. The inner product of two matrices $X, Y \in \mathrm{SYM}_n$ is defined as the standard inner product treating these matrices as n^2-dimensional vectors. To be precise,

$$X \bullet Y := \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} y_{ij}. \tag{64}$$

   It is not hard to see that $X \bullet Y = \mathrm{Tr}(X^t Y)$, where $\mathrm{Tr}(\cdot)$ is the trace of a matrix, i.e., the sum of its diagonal entries. Since an inner product defines a corresponding norm, what is the norm associated with the trace? From the trace definition, it is clear that the inner product operator is symmetric, linear in its arguments, and positive-definite (i.e., $X \bullet X \geq 0$).

3. Our linear map (or operator, to be more precise) is $A : \mathrm{SYM}_n \to \mathbb{R}^m$. It can be described by m matrices, or linear maps from $\mathrm{SYM}_n to \mathbb{R}$, forming its components, i.e., $A(X) = (A_1 \bullet X, \ldots, A_m \bullet X)$.

A **semidefinite program** is a generalization of an lp using the three concepts above:

$$\begin{aligned} &\text{maximize } C \bullet X \\ &\quad \text{subject to} \\ &\qquad A_i \bullet X = b_i, \text{ for } i = 1, \ldots, m, \text{ and} \\ &\qquad X \succeq 0. \end{aligned} \tag{65}$$

The similarity with ef of lp is striking, but that is not surprising as that's how we came up with the generalization. It is easy to see that any lp can be formulated as an sdp: C is the matrix with diagonal entries of $\mathbf{c}$, similarly for A_i, and X; note that the psd of X translates to $\mathbf{x} \geq 0$. We claimed that (61) is an sdp. Why is that? It is clear that the constraints can be expressed in the form above, but what about the objective function? The part $-\sum_{i,j} x_{ij}/2$, can clearly be expressed as a trace, namely $-\mathrm{Tr}(A^t X)/4$, where A is the adjacency matrix of the graph. Then the objective function is $|E|/2 - \mathrm{Tr}(A^t X)/4$. Since we have only added a constant, we can drop it from the objective function and minimize the trace part.

Sdp's do not share many nice properties of lps. For instance, the objective function can be bounded on the feasible region, but there may not be any psd matrix achieving it. This is made clear from the following example:

$$\begin{aligned} &\text{minimize } x \\ &\text{subject to} \\ &\qquad \begin{bmatrix} x & 1 \\ 1 & y \end{bmatrix} \succeq 0. \end{aligned}$$

A matrix of the form given above is psd iff $x \geq 0$ and $xy \geq 1$. So x can get arbitrarily close to 0, and hence the infimum of the objective function on the feasible region is 0, but it cannot be attained for any matrix. The duality principle still holds, but as the example above suggests, only in the weak form. There are no extreme bfs, as was the case of lp, mainly because the boundary of the set of sdp matrices inside the set of symmetric matrices is not polyhedral. The following section elucidates on these details.

The example above can be generalized to consider matrices in PSD_2:

$$\begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

It is not hard to show that this is equivalent to the set

$$T_C := \left\{(x, y, z) : x, y \geq 0, xy \geq z^2\right\},$$

which is called the toppled ice cream cone.

14.1 Cone programming and Duality

There is a uniform and general way to treat both the lp and sdp case as a special case of something more general called **cone programming**. The observation is that as the positive orthant is a cone in $\mathbb{R}^n$ so is the set PSD_n in SYM_n:

LEMMA 57. *The subset of positive semidefinite matrices in SYM_n forms a closed convex cone K, i.e., for all $\mathbf{x} \in K$ and $\lambda \geq 0$, $\lambda\mathbf{x} \in K$, and given $\mathbf{x}, \mathbf{y} \in K$, $\mathbf{x} + \mathbf{y} \in K$.*

Proof. The proof idea is to use the quadratic form definition of psd matrix. **Q.E.D.**

Given this similarity between the two optimization problems, we work in the more general setting of cones. Let K be a closed convex cone in a vector space V, with the inner product given by $\langle \cdot, \cdot \rangle$. We will start with generalizing Farkas's lemma, but to do that we need the following notion of a **dual cone**:

$$K^* := \{\mathbf{y} \in V | \langle \mathbf{y}, \mathbf{x} \rangle \geq 0 \text{ for all } \mathbf{x} \in K\}. \tag{66}$$

It is easy to prove the convexity and conicity by using the linearity of the inner product. What is the dual of the positive orthant $\mathbb{R}^n_{\geq 0}$? It is not hard to verify that it is self-dual.

Recall Farkas's lemma for linear inequalities: Given A and $\mathbf{b}$, either $A\mathbf{x} = \mathbf{b}$ has a non-negative solution $\mathbf{x}$ or there exists a $\mathbf{y}$ such that $A^t\mathbf{y} \geq 0$ but $\langle \mathbf{b}, \mathbf{y} \rangle < 0$. An equivalent way to state it the or-condition is that there exists a $\mathbf{y}$ such that $A^t\mathbf{y} \in (\mathbb{R}^n_{\geq 0})^* = \mathbb{R}^n_{\geq 0}$ but $\langle \mathbf{b}, \mathbf{y} \rangle < 0$. By analogy, we expect the following variant of Farkas's lemma to hold for solution of linear equalities in cones:

> Given a linear operator $A : V \to W$, where W is a vector space (take W to be some euclidean space $\mathbb{R}^m$) a $\mathbf{b} \in W$ and a cone K, either $A(\mathbf{x}) = \mathbf{b}$ has a solution $\mathbf{x} \in K$ or there exists a $\mathbf{y} \in W$ such that $A^t(\mathbf{y}) \in K^*$ but $\langle \mathbf{y}, \mathbf{b} \rangle < 0$.

There is at least one problem with this formulation: what do we mean by the transpose of a linear operator $A : V \to W$ (consider $V = \text{SYM}_n$ and $W = \mathbb{R}^m$)? But there is something more fundamental that is wrong: the set $A(K)$, which is also a convex cone, is not a closed set. Consider the toppled ice cream cone T_C in $\mathbb{R}^3$. Let A be the projection operator that maps T_C to the xy-plane. Since for a fixed value of $z = c$, the projection is the hyperbola $xy = c^2$, the set T_C is projected to $\left\{\mathbb{R}^2_{>0} \cup \mathbf{0}\right\}$, which is an open set. If we take $\mathbf{b} = (1, 0)$ then both the equations $A(\mathbf{x}) = \mathbf{b}$, for $\mathbf{x} \in T_C$, and $A^t(\mathbf{y}) \in K^*$ such that $\langle \mathbf{y}, \mathbf{b} \rangle < 0$ are not feasible. Note that the projection matrix

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and $T_C^* = \left\{(x, y, z) : x, y \geq 0, 4xy \geq z^2\right\}$, i.e., an elongated version of T_C. Since $A(T_C)$ is an open set and $\mathbf{b}$ is a boundary point of its closure, there is a sequence of values $\mathbf{x}_k \in T_C$ such that in the limit $A(\mathbf{x}_k)$ tends to $\mathbf{b}$, and, therefore, there cannot be a hyperplane separating $\mathbf{b}$ from $A(T_C)$. We remedy the situation by handling this boundary case: The system $A(\mathbf{x}) = \mathbf{b}$, where $\mathbf{x} \in K$ a closed convex cone, is called **limit feasible** if there exists a sequence $(\mathbf{x}_k)_{k \in \mathbb{N}} \in K$ such that $\lim_{k \to \infty} A(\mathbf{x}_k) = \mathbf{b}$. Such a sequence $(\mathbf{x}_k)$ is called a **feasible sequence**. Note that we do not require the sequence $\mathbf{x}_k$ to converge, but only its image.

Now we handle the second technicality: What is A^t? In the theory of vector spaces, given a linear operator $A : V \to W$, its adjoint is a linear operator $A^t : W \to V$ such that for all $\mathbf{x} \in V$ and $\mathbf{y} \in W$ the followig holds:

$$\langle \mathbf{y}, A(\mathbf{x}) \rangle = \langle A^t(\mathbf{y}), \mathbf{x} \rangle.$$

This can be verified when V, W are the standard euclidean spaces and A^t is the standard transpose.

The modification of Farkas's lemma is the following:

LEMMA 58. *Let $K \subseteq V$ be a closed convex cone, $A : V \to W$ a linear operator and $\mathbf{b} \in W$. Then either the system $A(\mathbf{x}) = \mathbf{b}$ is limit feasible in K, or the system $A^t(\mathbf{y}) \in K^*$ such that $\langle \mathbf{b}, \mathbf{y} \rangle < 0$ is feasible, but not both.*

Proof. Suppose $A(\mathbf{x}) = \mathbf{b}$ is limit feasible in K, and let $(\mathbf{x}_k)$ be the sequence whose image under A converges to $\mathbf{b}$. Then we have

$$\langle \mathbf{y}, \mathbf{b} \rangle = \lim_{k \to \infty} \langle \mathbf{y}, A\mathbf{x}_k \rangle = \lim_{k \to \infty} \langle A^t(\mathbf{y}), \mathbf{x}_k \rangle,$$

where the last step uses the definition of adjoint. Now if $A^t(\mathbf{y}) \in K^*$ then $\langle A^t(\mathbf{y}), \mathbf{x}_k \rangle \geq 0$, for all $k \in \mathbb{N}$ and hence the limit is also non-negative.

If $A(\mathbf{x}) = \mathbf{b}$ is not limit feasible then $\mathbf{b}$ is not in the closure of the cone $A(K)$. Therefore, there is a hyperplane separating $\overline{A(K)}$ from $\mathbf{b}$. Suppose $\mathbf{y} \in W$ is defines the hyperplane, i.e.,

$$\langle \mathbf{y}, \mathbf{b} \rangle < 0 \text{ and for all } \mathbf{x} \in K \ \langle \mathbf{y}, A(\mathbf{x}) \rangle \geq 0.$$

Then the latter condition is equivalent to $\langle A^t(\mathbf{y}), \mathbf{x} \rangle \geq 0$, for all $\mathbf{x} \in K$, which from duality of cones is equivalent to $A^t(\mathbf{y}) \in K^*$. **Q.E.D.**

In the second part of the proof we use the following result for closed convex cones $K' \subseteq W$: If $\mathbf{b} \notin K'$ then there exists $\mathbf{y} \in W$ such that

$$\langle \mathbf{y}, \mathbf{x} \rangle \geq 0 \text{ for all } \mathbf{x} \in K' \text{ and } \langle \mathbf{y}, \mathbf{b} \rangle < 0.$$

The proof idea is to take a point $\mathbf{z} \in K'$ that is closest to $\mathbf{b}$ (since K' is closed such a point exists and is unique). Then $\mathbf{y} := \mathbf{z} - \mathbf{b}$.

We can now define the following generalization of lp's and sdp's: Given two vector spaces V, W a closed convex cone $K \in V$, $\mathbf{c} \in V$, and $\mathbf{b} \in W$ a **cone program** in equational form is the following

$$\begin{aligned} &\text{maximize } \langle \mathbf{c}, \mathbf{x} \rangle \\ &\text{subject to } A(\mathbf{x}) = \mathbf{b}, \text{ for } \mathbf{x} \in K. \end{aligned} \tag{67}$$

The **value** of the program is $\sup_{\mathbf{x} \in K} \{\langle \mathbf{c}, \mathbf{x} \rangle : A(\mathbf{x}) = \mathbf{b}\}$. Note that the program may not be feasible, in which case the value is not defined, but it may be limit feasible. The **limit value of a feasible sequence** $(\mathbf{x}_k)$ is defined as the lim-sup of $\langle \mathbf{c}, \mathbf{x}_k \rangle$. The **limit value** of a program is the supremum over all limit values. Clearly, the limit value is at least as large as the value, but can the two be different, even when the program is feasible? The follow example demonstrates the problem: the example is for the more general form of cone programs.

The situation can be saved, if we enforce the so called **Slater's constraint qualification**, i.e., the program is not only feasible over K but feasible in the interior of K. Then these exceptional scenarios disappear:

THEOREM 59. *If the cone program* (67) *has an interior feasible point in K then the value equals the limit value.*

Proof. Suppose the limit value is γ, and we are given an $\epsilon > 0$. The idea is to construct a feasible solution $\mathbf{w}^*$ such that $\langle \mathbf{c}, \mathbf{x}^* \rangle > \gamma - \epsilon$. This implies that there are feasible solutions that take value arbitrarily close to γ. Hence the value and limit value are equal.

The idea for construction of $\mathbf{w}^*$ is as follows. We take a limit feasible sequence $\mathbf{x}_k \in K$, i.e., $\lim_k A(\mathbf{x}_k) = \mathbf{b}$ and $\limsup_k \langle \mathbf{c}, \mathbf{x}_k \rangle = \gamma$. Using the presence of an interior point $\overline{\mathbf{x}}$ we will get a feasible sequence $\mathbf{w}_k \in K$ such that

$$|\langle \mathbf{c}, \mathbf{w}_k \rangle - \langle \mathbf{c}, \mathbf{x}_k \rangle| < \epsilon.$$

Therefore,

$$\limsup_k \langle \mathbf{c}, \mathbf{w}_k \rangle \geq \limsup_k \langle \mathbf{c}, \mathbf{x}_k \rangle - \epsilon = \gamma - \epsilon,$$

which implies that there is an index ℓ for which $\langle \mathbf{c}, \mathbf{w}_k \rangle > \gamma - \epsilon$, and $\mathbf{w}^* = \mathbf{w}_k$.

The idea to construct $\mathbf{w}_k$ is to use $\mathbf{x}_k$, but as $\mathbf{x}_k$'s are somewhat going towards the boundary of K we need to pull back; this is done using the interior point $\overline{\mathbf{x}}$ and taking a convex combination with $\mathbf{x}_k$. Therefore, a candidate choice is

$$\mathbf{w}_k = (1-\lambda)\mathbf{x}_k + \lambda\overline{\mathbf{x}}.$$

Clearly, it is in K, for λ small enough it is close to $\mathbf{x}_k$, but for it to be feasible $A(\mathbf{w}_k)$ should be $\mathbf{b}$. However, $A(\mathbf{w}_k) = (1-\lambda_k)A(\mathbf{x}_k) + \lambda_k\mathbf{b}$. Not $A(\mathbf{x}_k)$ is tending to $\mathbf{b}$, so we are in the right direction; the deficit is $b - A(\mathbf{x}_k)$. The interesting twist is that instead of taking a convex combination of $\mathbf{x}_k$ and $\overline{\mathbf{x}}$, we take a nearby point $\mathbf{x}_k + \delta_k$, where $\delta_k \in V$ is such that $A(\delta_k) = \mathbf{b} - A(\mathbf{x}_k)$. Why does such a point exists? Note that both $\mathbf{b} = A(\overline{\mathbf{x}})$ and $A(\mathbf{x}_k)$ are in the image of V under A; since the latter set is itself a vector space it contains the difference of the two vectors, and hence an element in V exists that is mapped to the difference $\mathbf{b} - A(\mathbf{x}_k)$.

Define

$$\mathbf{w}_k := (1-\lambda_k)(\mathbf{x}_k + \delta_k) + \lambda_k\overline{\mathbf{x}}.$$

Our claim is the following:

1. $A(\mathbf{w}_k) = \mathbf{b}$ by construction.
2. $\mathbf{w}_k$ is close enough to $\mathbf{x}_k$ to ensure that $|\langle \mathbf{c}, \mathbf{w}_k\rangle - \langle \mathbf{c}, \mathbf{x}_k\rangle| < \epsilon$. For this a sufficiently small value λ_k, and an index k large enough suffices.
3. $\mathbf{w}_k \in K$. Note that by convexity $(1-\lambda)\mathbf{x}_k + \lambda\overline{\mathbf{x}}$ is in the interior of K. Therefore, there is sufficiently small ball around this point that is also in K. We only need to ensure that $\|(1-\lambda)\delta_k\|$ is smaller than the radius of this ball. This can be ensured by picking k large enough, since $\lim_k A(\mathbf{x}_k) = \mathbf{b}$.

Q.E.D.

We are now in a position to state the various forms of duality for cone programs. But first, what is the dual of (67)? It is not hard see that the generalization of the analogy from lp is the following:

$$\begin{aligned} &\text{minimize } \langle \mathbf{b}, \mathbf{y}\rangle \\ &\text{subject to } A^t(\mathbf{y}) - \mathbf{c} \in K^*,\ \mathbf{y} \in W. \end{aligned} \tag{68}$$

We start with the almost obvious version of weak duality:

THEOREM 60 (Weak Duality for Cone Programs). *If the primal program is limit feasible and the dual program is feasible, then the limit value of primal is smaller than the value of the dual.*

Proof. Since $A^t(\mathbf{y}) - \mathbf{c} \in K^*$, it follows that for all $\mathbf{x} \in K$, we have

$$0 \leq \langle A^t(bfy) - \mathbf{c}, \mathbf{x}\rangle = A^t(bfy) \bullet \mathbf{x} - \langle \mathbf{c}, \mathbf{x}\rangle,$$

which implies that $\langle \mathbf{c}, \mathbf{x}\rangle \leq \langle A^t(\mathbf{y}), \mathbf{x}\rangle$, and from the property of the adjoint $\langle \mathbf{c}, \mathbf{x}\rangle \leq \langle \mathbf{y}, A(\mathbf{x})\rangle = \langle \mathbf{y}, \mathbf{b}\rangle$. Therefore, the limit value of the primal is always less than the value of the dual. **Q.E.D.**

The second result tells us when the primal limit value matches the dual value:

THEOREM 61 (Regular Duality for Cone programs). *The dual is feasible and has a finite value D_v iff the primal is limit feasible and has a limit value P_{lv}. Moreover, the two values are equal $D_v = P_{lv}$.*

The strong duality states that if Slater's condition is met then primal value and limit value are the same and hence the dual value and primal value coincide:

THEOREM 62 (Strong Duality for Cone Programs). *If the primal is interior feasible and has a finite value then the dual is also feasible and has the same value.*

14.1.1 Duals in SDP

What is the form of the dual when $V = \text{SYM}_n$, $K = \text{PSD}_n$, and $W = \mathbb{R}^m$? Suppose the component matrices of A are given by $(A_1, \ldots, A_m)$. Then what is the adjoint A^t of A? By definition

$$\langle \mathbf{y}, A(X) \rangle = \sum_i y_i (A_i \bullet X) = \sum_i (y_i A_i) \bullet X = (\sum_i y_i A_i) \bullet X.$$

Therefore, it makes sense to define the adjoint $A^t := \sum_i y_i A_i$. The more interesting question is what is the dual of PSD_n? The claim is the following:

LEMMA 63. *PSD_n is self-dual.*

Proof. For this purpose we use the fact that a real symmetric matrix M is unitary diagionalizable, i.e., $M = U^t \Lambda U$, where U is an orthonormal matrix (whose columns are eigenvectors of M) and Λ is the diagonal matrix containing eigenvalues. This means that M can be expressed as a sum of rank-one matrices $M = \sum_i \lambda_i u_i u_i^t$, where u_i's are columns of U. Clearly, the rank-one matrices are psd. Now if $M \succeq 0$ then it means that M belongs to the cone generated by rank-one matrices.

1. $\text{PSD}_n \subseteq \text{PSD}_n^*$, i.e., given $X, Y \succeq 0$, $X \bullet Y = \text{Tr}(X^t Y) \geq 0$. Express $X = \sum_i \lambda_i u_i u_i^t$, where $\lambda_i \geq 0$. Then we have
$$X^t Y = XY = \sum_i \lambda_i u_i u_i^t Y.$$
We now use the fact that trace is a linear map, and that $\text{Tr}(AB) = \text{Tr}(BA)$. This gives us
$$\text{Tr}(XY) = \sum_i \lambda_i \text{Tr}(u_i^t Y u_i).$$
But as Y is psd and $\lambda_i \geq 0$ the terms on the rhs are non-negative.

2. $\text{PSD}_n^* \subseteq \text{PSD}_n$, i.e., an M in the dual is always psd. Since M is in the dual we know that for any rank-one matrix uu^t we have, $M \bullet uu^t \geq 0$. But
$$M \bullet uu^t = \text{Tr}(uu^t M) = \text{Tr}(u^t M u) = u^t M u.$$
Therefore, $M \succeq 0$.

Q.E.D.

Therefore, the primal and dual sdp's are

$$\begin{aligned} &\text{maximize } C \bullet X \\ &\text{subject to } A_i \bullet X = b_i, \text{ for } i = 1, \ldots, m, \text{ where } X \succeq 0. \end{aligned} \tag{69}$$

and

$$\begin{aligned} &\text{minimize } \langle \mathbf{b}, \mathbf{y} \rangle \\ &\text{subject to } \sum_i y_i A_i - C \succeq 0, \mathbf{y} \in \mathbb{R}^m. \end{aligned} \tag{70}$$