**CSC473: Advanced Algorithm Design** Winter 2018

# Week 7–8: Linear Programming

*Aleksandar Nikolov*

## 1 Introduction

*You have seen the basics of linear programming in CSC373, so much of this should be review material.*

A linear program (LP for short) is an optimization problem in which the constraints are linear inequalities and equalities, and the objective function is also linear. There are many equivalent *standard forms* for LPs. We will use the following form for a maximization problem:

$$\begin{aligned}&\max c^\intercal x\\&\text{s.t.}\\&Ax \le b\\&x \ge 0\end{aligned}$$

Let us explain the notation a bit. Here $A$ is an $m \times n$ matrix (i.e. $m$ constraints and $n$ variables) and $b$ is an $m \times 1$ column vector; $c$ is an $n \times 1$ column vector which encodes the objective function and $c^\intercal$ is its transpose; $x$ is an $n \times 1$ column vector which contains the variables we are optimizing over. The inequalities between vectors mean that the inequality should hold in all coordinates simultaneously. The value of this LP is the minimum value of the objective $c^\intercal x$ achieved subject to $x$ satisfying the constraints. Any value of $x$ that satisfies the constraints $x \ge 0$ and $Ax \le b$ is called *feasible*. The set of feasible $x$ is called the *feasible set* or *feasible region* of the LP. When the feasible set is empty, the LP is called *infeasible*. The maximum value of the objective $c^\intercal x$ over feasible $x$ is the optimal value of the LP. If this maximum is infinity, i.e. for any $t \in \mathbb{R}$ there exists a feasible $x$ s.t. $c^\intercal x \ge t$, then the LP is called *unbounded*.

Analogously, the standard form we use for a minimization problem is:

$$\begin{aligned}&\min c^\intercal x\\&\text{s.t.}\\&Ax \ge b\\&x \ge 0\end{aligned}$$

Just for concreteness, let us write a tiny example of a linear program:

$$\begin{aligned}&\min x_1 + x_2 + x_3 && (1)\\&\text{s.t.}\\&x_1 + x_2 \ge 1 && (2)\\&x_2 + x_3 \ge 1 && (3)\\&x_1 + x_3 \ge 1 && (4)\\&x_1, x_2, x_3 \ge 0 && (5)\end{aligned}$$

This LP corresponds to

$$b = c = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

**Exercise 1.** *Let $G = (V, E)$ be a directed connected graph, let $c : E \to \mathbb{R}$ be the capacities, and let $s, t \in V$ be, respectively, a source and a target vertex. Use linear programming to decide whether there exists a flow $f : E \to \mathbb{R}$ from $s$ to $t$ of value 1 that* strictly *respects the capacity constraints, i.e. such that for all $e \in E$ we have $0 \leq f(e) < c(e)$. Write a linear program which is feasible and has positive value if such a flow exists, and is infeasible or has value 0 if no such flow exists.*

## 2 Geometric View

A geometric view is very useful in understanding LPs. Let us plot the the points $(x_1, x_2, x_3)$ satisfying the following system of inequalities:

$$x_1 + x_2 + x_3 \leq 1 \tag{6}$$

$$x_1, x_2, x_3 \geq 0 \tag{7}$$

Figure 1 shows the points satisfying these inequalities. Let us introduce some terminology. Recall

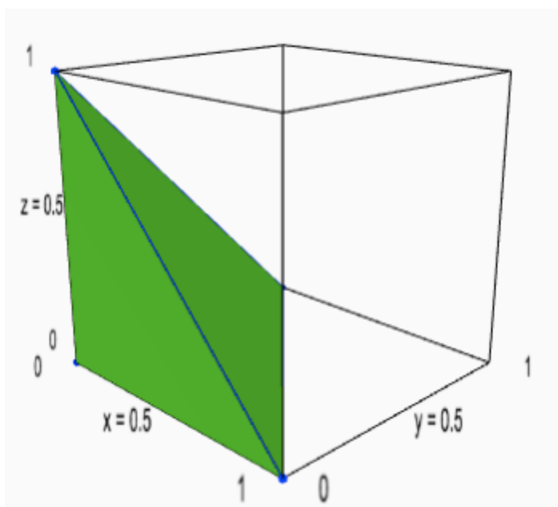


Figure 1: A polytope in 3 dimensions.

the equation of a line in two dimensions $a_1x_1 + a_2x_2 = b$, which can be written in vector notation as $a^\intercal x = b$. Similarly, in three dimensions, the equation of a plane is $a_1x_2 + a_2x_2 + a_3x_3 = b$ which in vector notation is $a^\intercal x = b$. In general the set $\{x \in \mathbb{R}^n : a^\intercal x = b\}$, where $\mathbb{R}^n$ is the set of vectors with $n$ real coordinates, is called a *hyperplane*. For some geometric intuition, let us mention that any line lying in the hyperplane $H = \{x \in \mathbb{R}^n : a^\intercal x = b\}$ which intersects the line $\ell = \{ta : t \in \mathbb{R}\}$ is perpendicular to $\ell$.

**Exercise 2.** *If $\ell$ is the line $\ell = \ell = \{ta : t \in \mathbb{R}\}$ and $H$ is the halfspace $H = \{x \in \mathbb{R}^n : a^\intercal x = b\}$, determine the point $z = \ell \cap H$. Show that any line $\ell' = \{z + tv : t \in \mathbb{R}\}$ contained in $H$ must satisfy $a^\intercal v = 0$.*

The set $\{x \in \mathbb{R}^n : a^\intercal x \leq b\}$ is called a *halfspace*, and $\{x :\in \mathbb{R}^n : a^\intercal x = b\}$ is its *supporting* or *bounding hyperplane.* To get a picture of a halfspace, notice that in 2 dimensions a halfspace is everything on one side of a line, and in 3 dimensions it's everything on one side of a plane. See Figure 2 for examples. The figure satisfying equations (6)–(7) is the intersection of the four

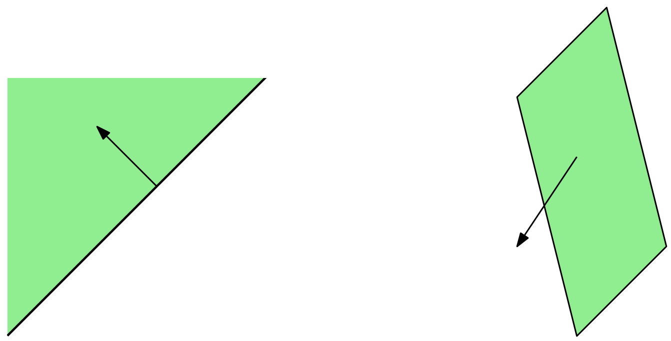

Figure 2: Halfspaces in 2 and 3 dimensions.

halfspaces $\{x \in \mathbb{R}^3 : x_1 + x_2 + x_3 \leq 1\}$, $\{x \in \mathbb{R}^3 : x_1 \geq 0\}$, $\{x \in \mathbb{R}^3 : x_2 \geq 0\}$, $\{x \in \mathbb{R}^3 : x_3 \geq 0\}$. A set which is the intersection of halfspaces is called a *polyhedron.* We can always write a polyhedron $P$ as $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ for some matrix $A$ and vector $b$.

A polyhedron $P \subseteq \mathbb{R}^n$ is *unbounded* when there exists a point $x \in P$ and a direction $v \in \mathbb{R}^n$ such that for every $t \geq 0$, $x + tv \in P$. (A set of the type $\{x + tv : t \geq 0\}$ is called a *ray.*) Intuitively, this means that there is a starting point and a direction in which we can go infinitely long. For example, the polyhedron satisfying the inequalities

$$\begin{aligned} -x_1 + 2x_2 &\geq 1 \\ 2x_1 - x_2 &\geq 1 \end{aligned}$$

is unbounded, because the ray $\{(1,1) + t(1,1) : t \geq 0\}$ is contained in it (see Figure 2). When a polyhedron is bounded (i.e. not unbounded), it is called a *polytope.* For example, the set in Figure 1 is a polytope.

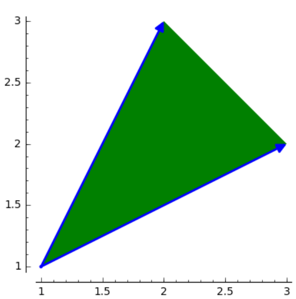


Figure 3: Unbounded polyhedron

Notice the structure of the polytope in Figure 1: its surface "consists" of 4 triangles glued to each other (we see three of them and one is hidden from view). These triangles are called the facets of the polytope. The triangles two by two along a line segment: these line segments (6 of them) are called the edges of the polytope. Finally, the triangles meet three by three at a point: these points (4 of them) are called the vertices of the polytope.

Formally, a *face* of a polyhedron $P = \{x : Ax \le b\} \subset \mathbb{R}^n$ is a set of the type $F = \{x : Ax \le b\} \cap \{x : a_i x = b_i \forall i \in S\}$, where $a_i$ is the $i$-th row of the matrix $A$, and $S$ is some subset of the rows of $A$. Usually, we also assume that $F \neq \emptyset$. Let us use the notation $S_F$ for the set of all $i$ such that $a_i x = b_i$ for all $x \in F$. When the dimension of the span of $\{a_i : i \in S_F\}$, or, equivalently, the rank of the submatrix $A_F$ of $A$ consisting of the rows of $A$ indexed by $S_F$, is $n - j$, we say that $F$ is a face of dimension $j$, or, in short, a *$j$-face*. The $(n-1)$-faces are called facets; the 1-faces are called edges, and the 0-faces are called vertices.

**Exercise 3.** *Let $v$ be a vertex of the polytope $P = \{Ax \le b\}$ and let $S$ be the set $S = \{i : a_i v = b_i\}$. Give a formula for $v$ in terms of $S$, $A$, and $b$. Give an upper bound on the number of vertices of $P$ in terms of the dimensions of the matrix $A$.*

For example, let $P$ be the polytope satisfying the constraints (6)–(7). The triangle with vertices $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$ is a facet (and also a 2-face), and can be written as $F = P \cap \{x_1 + x_2 + x_3 = 1\}$. The edge $e$ connecting $(1,0,0)$ and $(0,1,0)$ is a 1-face, can be written as $e = P \cap \{x_1 + x_2 + x_3 = 1, x_3 = 0\}$. The vertex $v = (0,0,1)$ is a 0-face, can be written as $v = P \cap \{x_1 + x_2 + x_3 = 1, x_1 = 0, x_2 = 0\}$. See Figure 4 for a 2-dimensional example.

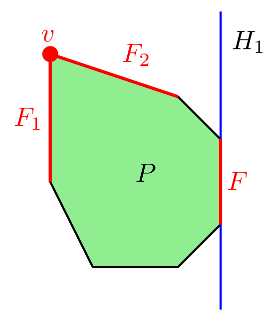


Figure 4: Faces of a 2-dimensional polytope (i.e. a polygon). The 1-face $F$ is the intersection $P \cap H$. The vertex $v$ is the intersection of the two facets $F_1$ and $F_2$.

A polytope is determined by its vertices in a strong sense. To make this precise, we define the notion of a convex hull: the *convex hull* of the points $v_1, \ldots, v_N \in \mathbb{R}^n$ is the set

$$\mathrm{conv}\{v_1, \ldots, v_N\} = \{\lambda_1 v_1 + \ldots + \lambda_N v_N : \lambda \ge 0, \lambda_1 + \ldots + \lambda_N = 1\}.$$

For some geometric intuition, we mention that the convex hull of two points $v_1, v_2$ is simply the line segment connecting them; the convex hull of three points $v_1, v_2, v_3$ is the triangle with the points as its vertices. In general, $\mathrm{conv}\{v_1, \ldots, v_N\}$ is the smallest polytope that contains $v_1, \ldots, v_N$.

**Exercise 4.** *Show that if $v_1, \ldots, v_N$ belong to the polytope $P = \{Ax \le b\}$, then $\mathrm{conv}\{v_1, \ldots, v_N\} \subseteq P$.*

We have the following basic theorem. You will not be responsible for the proof, but we include it for your interest.

**Theorem 1.** *A polytope $P$ with vertices $v_1, \ldots, v_N$ satisfies $P = \mathrm{conv}\{v_1, \ldots, v_N\}$.*

*Proof Sketch.* Let $P = \{x : Ax \le b\}$, and let $S_x = \{i : a_i x = b_i\}$. Let $A_x$ be the submatrix of $A$ consisting of the rows indexed by $S_x$. We will show that every $x \in P$ is in the convex hull of the vertices $v_1, \ldots, v_N$; i.e. we will show that there exist non-negative $\lambda_1, \ldots, \lambda_N$, which sum to 1, and give $\lambda_1 v_1 + \ldots + \lambda_N v_N = x$. This will show that $P \subseteq \text{conv}\{v_1, \ldots, v_N\}$. The other containment $\text{conv}\{v_1, \ldots, v_N\} \subseteq P$ follows from Exercise 4.

The proof is by induction on $n - \text{rank } A_x$. In the base case we have $\text{rank } A_x = n$, and then $x$ is a vertex of $P$, so there is nothing to show. Assume then that $\text{rank } A_x < n$. Then there exists some vector $y \in \mathbb{R}^n$ for which $A_x y = 0$. Let

$$\alpha = \max\{\alpha' : x + \alpha' y \in P\},$$
$$\beta = \max\{\beta' : x - \beta' y \in P\}.$$

These maxima must exist, because $P$ is bounded. Pictorially, we are finding some direction $y$ such that we can walk a positive distance in the direction of $y$ and stay inside $P$, and also we can walk a positive distance in the direction of $-y$ and also stay inside $P$. This is illustrated in Figure 5. The main point of the proof is that the furthest points we can travel in these directions lie in lower dimensional faces of $P$.

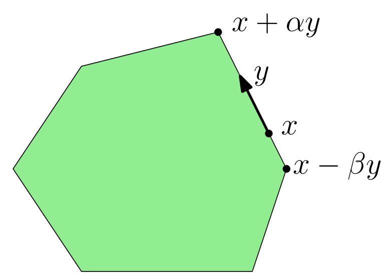


Figure 5: Illustration of the inductive step in the proof of Theorem 1

It is easy to check that

$$\alpha = \min\left\{\frac{b_i - a_i x}{a_i y} : a_i y > 0\right\},$$
$$\beta = \min\left\{\frac{b_i - a_i x}{|a_i y|} : a_i y < 0\right\}.$$

From these expressions it is clear that $\alpha, \beta > 0$. Moreover, there exists some $i_+$ (any one one achieving the minimum) such that $a_{i_+} y > 0$ and $a_{i_+} x + \alpha a_{i_+} y = b_{i_+}$. We have $S_{x+\alpha y} \supseteq S_x \cup \{i_+\}$. Since $a_{i_+} y > 0$, $a_{i_+}$ cannot be in the linear span of $\{a_i : i \in S_x\}$. Therefore, $\text{rank } A_{x+\alpha y} > \text{rank } A_x$. Analogously, we can show $\text{rank } A_{x-\beta y} > \text{rank } A_x$. By induction, we have

$$x + \alpha y = \lambda'_1 v_1 + \ldots + \lambda'_N v_N,$$
$$x - \beta y = \lambda''_1 v_1 + \ldots + \lambda''_N v_N,$$

for non-negative $\lambda', \lambda''$ such that $\lambda'_1 + \ldots + \lambda'_N = \lambda''_1 + \ldots + \lambda''_N = 1$. Then, we can write

$$x = \frac{\alpha\beta}{\alpha(\alpha+\beta)}(x + \alpha y) + \frac{\alpha\beta}{\beta(\alpha+\beta)}(x - \beta y).$$

We can then define $\lambda_i = \frac{\alpha\beta}{\alpha(\alpha+\beta)}\lambda'_i + \frac{\alpha\beta}{\beta(\alpha+\beta)}\lambda''_i$ for all $i$, and we are done. $\square$

Let us now interpret LPs geometrically. Let's take a maximization problem $\max\{c^\intercal x : Ax \leq b, x \geq 0\}$. The feasible set $P = \{Ax \leq b, x \geq 0\}$ is a polyhedron. We can view the objective $c$ as a vector pointing from the origin 0 to the point with coordinates $c$. So the LP asks us to find the point in $P$ which is the farthest out in the direction of the vector $c$. For example, consider the LP which maximizes the value of $x_3$ subject to the constraints (6)–(7). This LP corresponds to finding the point furthest along the direction of the vector pointing from the origin to $(0, 0, 1)$ in the polytope in Figure 1. I.e. we want to find the point in the polytope which is highest up. Visually, it's clear that the optimal solution of the LP is the point $(0, 0, 1)$. Notice that the optimal solution is a vertex. This is a general phenomenon, and in fact follows easily from Theorem 1.

**Corollary 2.** *If* $\max\{c^\intercal x : Ax \leq b, x \geq 0\}$ *is an LP whose feasible region* $P = \{x : Ax \leq b, x \geq 0\}$ *is a polytope, then the LP has an optimal solution which is a vertex of* $P$*.*

*Proof.* Let $x$ be any optimal solution of the LP. By Theorem 1, we can write $x = \lambda_1 v_1 + \ldots + \lambda_N v_N$, where $\lambda \geq 0$, $\lambda_1 + \ldots + \lambda_N = 1$, and $v_1, \ldots, v_N$ are the vertices of $P$. We have

$$c^\intercal x = \lambda_1 c^\intercal v_1 + \ldots + \lambda_N c^\intercal v_N \leq \lambda_1 \max_{i=1}^{N} c^\intercal v_i + \ldots + \lambda_N \max_{i=1}^{N} c^\intercal v_i = \max_{i=1}^{N} c^\intercal v_i.$$

Therefore any vertex $v_i$ achieving the maximum on the right hand side is also an optimal solution of the LP. □

Note that if the LP has many optimal solutions, then not all of them will be vertices, but there always will be at least one which is a vertex. For example, suppose we maximize the objective $x_1 + x_2 + x_3$ subject to the constraints (6)–(7). Then one optimal solution is $(1/3, 1/3, 1/3)$, which is not a vertex. Nevertheless, the vertices $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$ are all optimal solutions as well.

# 3 Simplex and Ellipsoid Algorithms

After this introduction to geometry, we will describe, on a very high level, two algorithms for solving linear programs. We will only describe these algorithms geometrically, and will not worry about the implementation details, which are far from trivial. Our goal is just to get the geometric intuition behind the algorithms, which are both quite beautiful.

## 3.1 Simplex Algorithm

The idea of the simplex algorithm is simple. Suppose we want to solve the LP $\max\{c^\intercal x : Ax \leq b, x \geq 0\}$. Let $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ be the feasible region. As we already saw, $P$ is a polyhedron; for simplicity, let's assume that it is also a polytope. The simplex algorithm starts at some vertex $x^{(0)}$ of $P$. (Getting a vertex to start from is not always easy. However, for many LPs in practice there is a clear choice.) Let $N(x^{(0)})$ be the set of neighboring vertices to $x^{(0)}$, i.e. vertices $y$ such that there is an edge of $P$ connecting $x^{(0)}$ and $y$. The algorithm picks an element $x^{(1)}$ of $N(x^{(0)})$ such that $c^\intercal x^{(1)} > c^\intercal x^{(0)}$. Then this process continues: at each step $t$, the algorithm computes a new vertex $x^{(t)}$ from $x^{(t-1)}$ by picking a vertex $x^{(t)} \in N(x^{(t-1)})$ s.t. $c^\intercal x^{(t)} > c^\intercal x^{(t-1)}$. We stop once the objective function cannot be improved anymore: in this case, if $P$ is a polytope, i.e. bounded, we have found an optimal solution to the LP, and, moreover, the solution is a vertex. In Figure 6 we show a run of the simplex algorithm on a 3-dimensional cube.

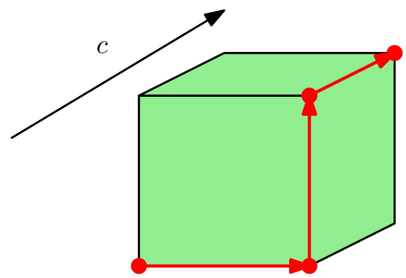


Figure 6: The simplex algorithm, run on a cube.

We have swept many things under the rug:

- How do we find a starting vertex $x^{(0)}$?
- How do we know if $P$ is bounded?
- How do we compute the neighboring vertices of $x^{(t)}$?
- If there are multiple options for $x^{(t)}$ which improve the objective value, which one do we pick?

An very interesting question is the running time of the simplex algorithm. While the algorithm seems to perform really well in practice, for essentially all known variants of it the worst-case complexity is exponential. Here by "variants" we mean the rule used to pick a neighbor $x^{(t)}$ of $x^{(t-1)}$, when there are multiple options. Such rules are known as *pivot rules*. It remains an important open problem to find a pivot rule for which the simplex algorithm runs in time polynomial in the number of variables and the number of constraints, or to show that no such pivot rule exists. One way to show that no such pivot rule exists would be to give a counterexample to the *polynomial Hirsch conjecture*: come up with a polytope $P \subset \mathbb{R}^n$, determined by $m$ constraints, and two vertices $x, y$ of $P$ such that the shortest path between $x$ and $y$ has exponentially many edges.

### 3.2 The Ellipsoid Algorithm

The ellipsoid algorithm is based on very different ideas. In order to understand the algorithm, we need to take another detour into geometry, and introduce ellipsoids. Recall that in two dimensions we can write an ellipse (with major axes parallel to the coordinate axes) as $\{x \in \mathbb{R}^2 : a^2(x_1 - y_1)^2 + b^2(x_2 - y_2)^2 \le 1\}$. See Figure 7 for an example with $y_1 = y_2 = 1$, $a = 1$, $b = 2$. In higher dimensions we define an *ellipsoid* as the set $E(y, M) = \{x \in R^n : (x - y)^\intercal M^\intercal M(x - y) \le 1\}$, where $M$ is an $n \times n$ invertible matrix, and $y \in R^n$ is the *center* of the ellipsoid. When $M = \frac{1}{r}I$ ($I$ here is the identity matrix) we write $B(y, r) = E(y, r^{-1}I)$; such en ellipsoid is called a *ball of radius* $r$. In 2 dimensions, this is a disc of radius $r$, and in 3 dimensions it is a 3-dimensional ball of radius $r$. Indeed, in 2 dimensions, $B(y, r) = \{(x_1, x_2) : \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \le r\}$, and in 3 dimensions $B(y, r) = \{(x_1, x_2, x_3) : \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} \le r\}$. In general $E(y, M) = y + M^{-1}B(0, 1)$, where $M^{-1}B(0, 1) = \{M^{-1}z : z \in B(0, 1)\}$. In other words, an

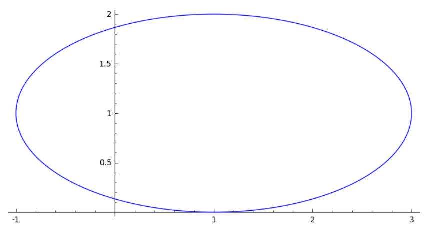


Figure 7: An ellipse in 2 dimensions

ellipsoid is the image of the unit ball under a translation and a linear map. (The combination of a translation and a linear map is known as an *affine map* or an *affine transformation*.)

Here we are going to focus on the *feasibility problem*: given a polytope, described by inequalities, decide if it is empty. There are several ways to reduce solving LPs to this feasibility problem, and possibly the simplest one is based on binary search. Suppose we have the LP

$$\begin{aligned}
&\max c^{\mathsf{T}}x \\
&\text{s.t.} \\
&Ax \leq b \\
&x \geq 0
\end{aligned}$$

and let $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ be its feasible region. It's usually easy to get two values $v, V \in \mathbb{R}$ such that the optimal solution of the LP is guaranteed to be in the interval $[v, V]$. Let $Q_t = P \cap \{x \in \mathbb{R}^n : c^T x \geq t\}$. For any value of $t$, $Q_t$ is a polytope, and solving the LP is equivalent to finding the largest $t$ such that $Q_t \neq \emptyset$. If we have an efficient procedure to solve the feasibility problem, we can get arbitrarily close to this optimal $t$ by doing binary search on $[v, V]$.

Let us then focus on the feasibility problem: given a polytope $Q = \{x \in R^n : Dx \leq e\}$, decide if it is empty or not. In fact, we need a little more information. We need a number $R$ so that $Q \subseteq B(0, R)$. We also need another number $r < R$, and we need the "promise" that either $Q = \emptyset$, or there exists a center $y \in \mathbb{R}^n$ for which $B(y, r) \subseteq Q$. $R$ can be computed from the description of $Q$. In order to satisfy the promise, we compute another polytope $\tilde{Q}$, based on $Q$ such that if $Q$ is empty, then $\tilde{Q}$ also is empty, and if $Q$ is not empty, then $\tilde{Q}$ contains a ball of radius $r$. In the description of the algorithm below, we assume that we have substituted $Q$ with $\tilde{Q}$ and the promise is satisfied.

We can finally describe the algorithm, which is actually quite intuitive. At each time step $t$, the algorithm keeps an ellipsoid $E_t = E(y_t, M_t)$ so that $Q \subseteq E_t$. Initially, $E_0 = B(0, R)$. At step $t$, the algorithm checks if the center $y_{t-1}$ of $E_{t-1}$ is in $Q$. If it is, then $Q$ is non-empty, and the algorithm terminates. Otherwise, we can find some constraint of $Q$ violated by $y_{t-1}$, i.e. some row $d_i$ of the matrix $D$ so that $d_i y_{t-1} > e_i$. Let $H_i^-$ be the halfspace $\{x \in \mathbb{R}^n : d_i x \leq e_i\}$. We know that $Q \subseteq E_{t-1} \cap H_i^-$. Moreover, $E_{t-1} \cap H_i^-$ has volume[1] at most half that of $E_{t-1}$, because it does not contain the center of $E_{t-1}$. Then we can compute a new ellipsoid $E_t$ which contains $E_{t-1} \cap H_i^-$, and, therefore, contains $Q$ as well. While $E_t$ will have volume slightly larger than that of $E_{t-1} \cap H_i^-$, we can still show that its volume is strictly less than that of $E_{t-1}$. Therefore, the volume of the

[1] There is a natural way to generalize 2-dimensional area and 3-dimensional volume to $n$-dimensional volume. The idea is that that a set in $\mathbb{R}^n$ of volume 1 has as much "space" inside of it as the side 1 cube $\{x \in \mathbb{R}^n : 0 \leq x_i \leq 1 \; \forall i \in \{1, \ldots n\}\}$.

ellipsoids $E_0, E_1, E_2, \ldots$ keeps decreasing, and after a while we know that the volume of $E_t$ is less than that of any ball of radius $r$. This means that there is no $y \in \mathbb{R}^n$ such that $B(y, r) \subseteq Q$, and, by the promise we had on $r$ and $Q$, we know that $Q = \emptyset$.

A step of the algorithm is illustrated in Figure 8.

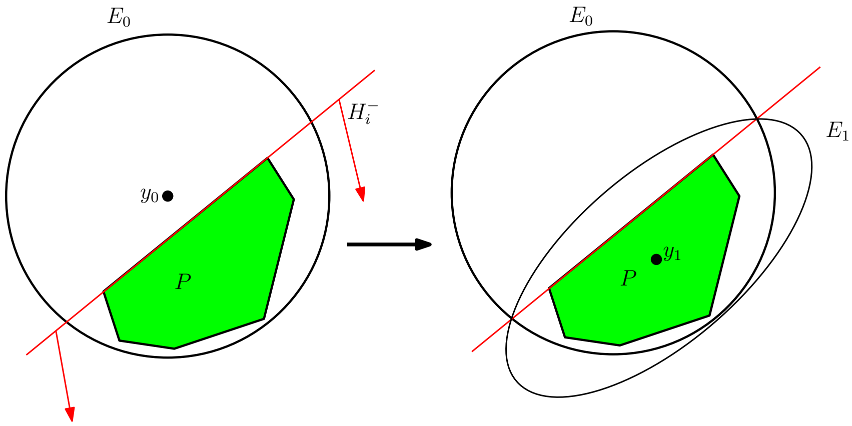


Figure 8: One step of the ellipsoid algorithm.

Unlike the simplex algorithm, the ellipsoid algorithm has worst-case running time polynomial in the number of bits needed to describe the LP. In fact, it was the first such algorithm, and of significant theoretical importance. However, in practice the simplex algorithm usually performs quite well, while the ellipsoid algorithm is not really practical. It seems that the simplex algorithm, unlike the ellipsoid algorithm, achieves its worst-case performance only on very special instances. Moreover, the ellipsoid algorithm needs to perform algebraic operations down to very high precision, which slows it down significantly.

There is another family of algorithms which we will not mention: interior point methods. These algorithms reduce solving an LP to solving an unconstrained, but non-linear optimization problem. Their name comes from the fact that, geometrically, they trace a path towards the optimal solution inside the feasible region, rather than on the boundary, as the simplex algorithm. The most efficient interior point methods give the best of both worlds: their worst-case running time is polynomial, and they also tend to perform well in practice. Interior point methods are currently a very active area of research.

Finally, there are algorithms designed to solve specific LPs. In fact you already have seen such algorithms: the maximum flow problem can be written as an LP, and the different variants of the Ford-Fulkerson algorithm solve this LP. Soon we will see other examples, when we talk about primal-dual algorithms. While these algorithms do not work for general LPs, they are usually simple and efficient.

## 4 Duality

Consider the LP in (1)–(5). If you wanted to convince someone that the optimal value of this LP is *at most* $3/2$, then all you need to do is to present them with a feasible solution $x$ which achieves this value. For example, you can take $x_1 = x_2 = x_3 = 1/2$. However, how would you convince someone that this is in fact an optimal solution, i.e. that the optimal value of the LP is also *at least* 3/2? No single feasible solution proves that, because the optimal value of the LP is the minimum over *all* feasible solutions. A powerful idea is to show that the inequality $x_1 + x_2 + x_3 \geq 3/2$ is *implied* by the constraints of the LP. In this case this is actually quite easy. You can add up the three constraints (2)–(4) and you get the new (implied) constraint $2x_1 + 2x_2 + 2x_3 \geq 3$. Now divide both sides of this inequality by 2 and you are done.

For another example, let us take the LP which maximizes $x_3$ subject to the constraints (6)–(7). As we mentioned before, geometrically, it seems clear that the optimal value is 1. To show more formally that the value is at least 1, we just need to exhibit a feasible solution: $x_1 = x_2 = 0$ and $x_3 = 1$. To show that $x_3 \leq 1$, observe that (6) and the non-negativity constraints (7) imply that $x_3 \leq x_1 + x_2 + x_3 \leq 1$.

Let us try to formalize this technique, and put it in as general terms as possible. Recall that we write a generic maximization LP as:

$$\max c^\intercal x \tag{8}$$
$$\text{s.t.}$$
$$Ax \leq b \tag{9}$$
$$x \geq 0 \tag{10}$$

Let us call this the *primal* LP. To it corresponds a *dual* LP:

$$\min b^\intercal y \tag{11}$$
$$\text{s.t.}$$
$$A^\intercal y \geq c \tag{12}$$
$$y \geq 0 \tag{13}$$

Notice that in this program the variables are $y \in \mathbb{R}^m$, where $m$ is the number of rows of the matrix $A$. To see how we this LP corresponds to what we did above, observe that $A^\intercal y$, for any $y \geq 0$ is a non-negative combination of the left hand sides of the constraints $Ax \leq b$, and $b^\intercal y$ is the corresponding non-negative combination of the right hand sides.

**Exercise 5.** *Write the dual linear program to the following linear program:*

$$\begin{aligned} & \max c^\intercal x \\ s.t.\quad & Ax = b \\ & x \geq 0 \end{aligned}$$

The following theorem, known as the *weak duality theorem*, proves that the dual LP indeed gives upper bounds on the optimal value of the primal LP.

**Theorem 3** (Weak Duality)**.** *Let $x$ satisfy the primal constraints* (9)–(10)*, and let $y$ satisfy the dual constraints* (12)–(13)*. Then $c^\intercal x \leq b^\intercal y$.*

*Proof.* The main observation we use is that if $u, v, w \in \mathbb{R}^n$, and $u \geq v$, $w \geq 0$, then $u^\intercal w \geq v^\intercal w$. (**Make sure you understand this.**)

Using $c \leq A^\intercal y$ and $x \geq 0$, we have $c^\intercal x \leq y^\intercal Ax$. In turn, using $Ax \leq b$ and $y \geq 0$, we have $y^\intercal Ax \leq y^\intercal b$. Combining the two inequalities proves the theorem. □

A surprising, important, and very powerful fact is that this simple way to bound the optimal value of an LP gives a tight bound for *every* LP:

**Theorem 4** (Strong Duality)**.** *Suppose that the primal program* (8)–(10) *and the dual program* (11)–(13) *are both feasible. Then their optimal values are equal.*

The heart of the proof of this theorem is a useful lemma, known as Farkas's lemma. There are many different version of it. We will state one which is amenable to a geometric proof.

**Lemma 5** (Farkas's Lemma)**.** *For any $m \times n$ matrix $A$ and any $m \times 1$ vector $b$, exactly one of the following two statements is true:*

1. *There exists a $x \in \mathbb{R}^n$, $x \geq 0$, such that $Ax = b$.*
2. *There exists a $y \in \mathbb{R}^m$ such that $A^\intercal y \leq 0$ and $b^\intercal y > 0$.*

You can view Farkas's lemma as characterizing when the constraints $Ax = b, x \geq 0$ are (in)feasible: there either exists a simple "obstacle" to feasibility, i.e. a vector $y$ such that $A^\intercal y \leq 0$ and $b^\intercal y > 0$, or the constraints are feasible. It should be clear that the existence of such a $y$ contradicts feasibility of $Ax = b, x \geq 0$. The lemma shows that this is the only possible reason the constraints are not feasible.

**Exercise 6.** *Prove that if there exists a vector $y \in \mathbb{R}^m$ such that $A^\intercal y \leq 0$ and $b^\intercal y > 0$, then there exists no $x \geq 0$ such that $Ax = b$.*

It may be helpful to see an analogue to Farkas's lemma for linear equalities.

**Exercise 7.** *Using what you know from linear algebra, prove that for any $m \times n$ matrix $A$ and any $m \times 1$ vector $b$, exactly one of the following two statements is true:*

1. *There exists a $x \in \mathbb{R}^n$ such that $Ax = b$.*
2. *There exists a $y \in \mathbb{R}^m$ such that $A^\intercal y = 0$ and $y^\intercal b \neq 0$.*

*Proof of Lemma 5.* From Exercise 6, only one of the two statements can be true. It only remains to show that if the first statement is not true, the second must be. I.e. we need to show that the only possible reason that $Ax = b$ is not feasible is that there exists a $y$ as in the second statement.

In the proof, we will adopt a geometric viewpoint. Let $C = \{Ax : x \geq 0\}$. Such a set is called a *cone*. If the first statement of the Lemma does not hold, then we have that $b \notin C$. Observe that if $A^\intercal y \leq 0$, then every $z \in C$ satisfies $y^\intercal z = y^\intercal Ax \leq 0$. Then, geometrically what we are trying to show is that if $b \notin C$, then there exists a hyperplane $H = \{z : y^\intercal z = 0\}$ through the origin, such that all of $C$ is on one side of $H$, and $b$ is on the other. This is a simple geometric certificate that $b \notin C$, and a special case of the *hyperplane separation theorem.* See Figure 9 for an illustration.

In the proof we will work with the (Euclidean) norm $\|\cdot\|$ defined for a vector $z \in \mathbb{R}^m$ by $\|z\| = \sqrt{z^\intercal z} = \sqrt{z_1^2 + \ldots + z_m^2}$. Notice that in two or three dimensions this gives the usual distance from the origin 0 to $z$. It is clear from the definition that $\|z\| > 0$ for every nonzero $z$, and $\|0\| = 0$.

Let now $z$ be the closest point in $C$ to $b$: $z = \arg\min\{\|b - z'\|^2 : z' \in C\}$. We will need the following important claim.

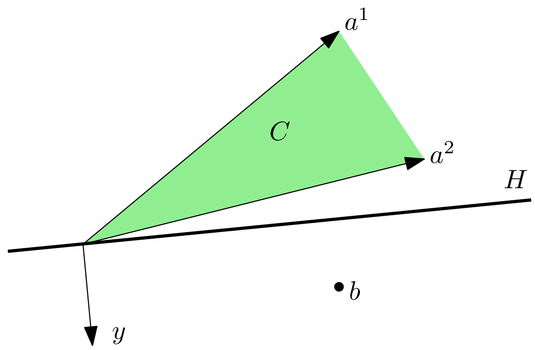


Figure 9: An illustration of the Farkas lemma for a 2 by 2 matrix $A = (a^1, a^2)$.

**Claim 6.** *We have* $z^\intercal(b - z) = 0$.

*Proof.* If $z = 0$ the claim is trivially true, so let us assume $z \neq 0$. Let us write the function $f(t) = \|b - tz\|^2$. From the definition of $z$, we have that $f(1)$ is a minimum of this function over $t \geq 0$, and, therefore $f'(1) = 0$ by elementary calculus. Expanding $f(t)$, we have

$$f(t) = (b - tz)^\intercal(b - tz) = \|b\|^2 + t^2\|z\|^2 - 2tz^\intercal b.$$

Therefore, $f'(t) = 2t\|z\|^2 - 2z^\intercal b$, and $f'(1) = 0$ implies $2z^\intercal(z - b) = 2\|z\|^2 - 2z^\intercal b = 0$. Multiplying both sides by $-1/2$ gives $z^\intercal(b - z) = 0$. $\square$

Geometrically, the claim means that $z$ and $b - z$ must form a right angle.

Because $b \notin C$, we must have $\|b - z\|^2 > 0$: otherwise $b - z = 0$, so $b = z \in C$, a contradiction. Let us define $y = b - z$. Then $\|b - z\|^2 > 0$ implies

$$0 < (b - z)^\intercal(b - z) = b^\intercal(b - z) - z^\intercal(b - z) = b^\intercal y.$$

Here in the final equality we used the definition of $y$ and Claim 6. We have then found a $y$ such that $b^\intercal y > 0$. It remains to show that that $A^\intercal y \leq 0$. We shall prove this by contradiction. Assume that there exists some column $a^i$ of $A$ such that $(a^i)^\intercal y > 0$. We will show that this means that there exists a $z' \in C$ for which $\|b - z'\|^2 < \|b - z\|$, contradicting the choice of $z$.

Let us take $z' = (1 - \alpha)z + \alpha a^i$, for a real number $\alpha \in [0, 1]$ to be chosen later. Then, using $z' = z + \alpha(a^i - z)$, we have

$$\begin{aligned}
\|b - z'\|^2 &= ((b - z) - \alpha(a^i - z))^\intercal((b - z) - \alpha(a^i - z)) \\
&= (y - \alpha(a^i - z))^\intercal(y - \alpha(a^i - z)) \\
&= \|y\|^2 + \alpha^2\|a^i - z\|^2 - 2\alpha y^\intercal(a^i - z).
\end{aligned}$$

By Claim 6, we have $y^\intercal z = z^\intercal y = z^\intercal(b - z) = 0$. So, the equation above simplifies to

$$\|b - z'\|^2 = \|y\|^2 + \alpha^2\|a^i - z\|^2 - 2\alpha y^\intercal a^i.$$

If $\alpha < \frac{2y^\intercal a^i}{\|a^i - z\|^2}$, the equation above gives $\|b - z'\|^2 < \|y\|^2 = \|b - z\|^2$. Moreover, because $\frac{2y^\intercal a^i}{\|a^i - z\|^2} = \frac{2(a^i)^\intercal y}{\|a^i - z\|^2} > 0$, we can choose such an $\alpha$ in $[0, 1]$. But, if we choose some $x \geq 0$ such that $z = Ax$

(which exists because $z \in C$), then we have $z' = Ax'$ for $x'$ defined by $x'_j = (1-\alpha)x_j$ for $j \neq i$, and $x'_i = (1-\alpha)x_i + \alpha$. It is obvious that $x' \geq 0$, which implies that $z' = Ax' \in C$. Then, we have found a $z'$ for which $\|b - z'\|^2 < \|b - z\|^2$, contradicting the choice of $z$. This completes the proof of the lemma. □

Farkas's lemma is also known as a *theorem of the alternative*, because it gives two statements exactly one of which is true. Many other proofs are known: for example, one based on a variant of Gaussian elimination known as Fourier-Motzkin elimination, and another based on the simplex method.

While Lemma 5 was convenient for our proof method, the following version is more useful in proving Theorem 4:

**Lemma 7** (Farkas's Lemma, variant)**.** *For any $m \times n$ matrix $A$ and any $m \times 1$ vector $b$, exactly one of the following two statements is true:*

1. *There exists a $x \in \mathbb{R}^n$ such that $x \geq 0$ and $Ax \leq b$.*
2. *There exists a $y \in \mathbb{R}^m$ such that $y \geq 0$, $A^\intercal y \geq 0$ and $b^\intercal y < 0$.*

**Exercise 8.** *Derive Lemma 7 from Lemma 5.*

*Proof of Theorem 4.* The theorem has a very short proof once we have established Farkas's lemma. We will opt for a slightly longer proof, in the hope that it provides some geometric intuition.

For a given $t \in \mathbb{R}$, define the halfspace $H_t^+ = \{x \in \mathbb{R}^n : c^\intercal x \geq t\}$, and let $P = \{x \in \mathbb{R}^n : Ax \geq b, x \geq 0\}$ be the feasible region of the primal LP (8)–(10). Recall that we assumed that $P \neq \emptyset$. Geometrically, the optimal value of this LP is the largest value $t$ such that $H_t^+ \cap P \neq \emptyset$ (see Figure 10). Farkas's lemma gives us a necessary and sufficient condition for $H_t^+ \cap P = \emptyset$. To see this, define:

$$\tilde{A} = \begin{pmatrix} A \\ -c^\intercal \end{pmatrix}; \quad \tilde{b} = \begin{pmatrix} b \\ -t \end{pmatrix}.$$

Notice that the system of inequalities $\tilde{A}x \leq \tilde{b}, x \geq 0$ is feasible if and only if $H_t^+ \cap P \neq \emptyset$. Then, by Lemma 7, $H_t^+ \cap P = \emptyset$ if and only if there exists a $y \in \mathbb{R}^m$, $y \geq 0$, such that $A^\intercal y \geq c$ and $y^\intercal b < t$. (**Make sure you understand how this follows from the lemma: we have skipped one or two steps here.**) In other words, $H_t^+ \cap P = \emptyset$ if and only if the optimal value of the dual LP is *strictly* less than $t$.

We are now ready to finish the proof. Let $v_P$ be the value of the primal LP, and $v_D$ the value of the dual LP. Because the optimal value of the dual LP is $v_D$, there does not exist any $y \in \mathbb{R}^m$ such that $y \geq 0$, $A^\intercal y \geq c$, and $y^\intercal b < v_D$. This implies, by our observations above, that $H_{v_D}^+ \cap P \neq \emptyset$, and, therefore, $v_P \geq v_D$. By Theorem 3, $v_P \leq v_D$, and, therefore, $v_P = v_D$. □

**Exercise 9.** *Using Lemma 7, show that, assuming $\{x : Ax \leq b, x \geq 0\} \neq \emptyset$, the set $\{x : c^\intercal x \geq t, Ax \leq b, x \geq 0\}$ is empty if and only if there exists a $y \in \mathbb{R}^m$, $y \geq 0$, such that $A^\intercal y \geq c$ and $y^\intercal b < t$.*

In some cases we treat the minimization LP (11)–(13) as the primal program. In these cases, we say that the maximization LP (8)–(10) is the dual of the minimization LP (11)–(13). Theorems 3,

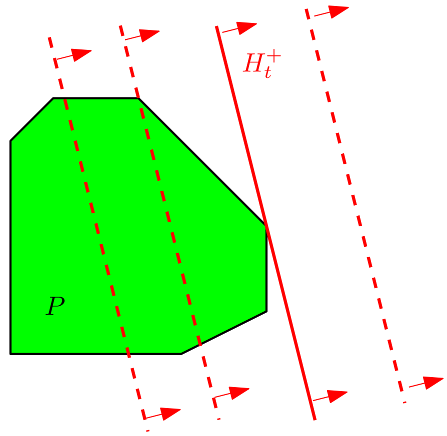


Figure 10: Geometric proof of strong duality: the feasible region $P$ and the halfspace $H_t^+$.

and 4 hold with the obvious modifications. Moreover, notice that if we take the dual of an LP twice, we get back the original LP. In other words, it's not really important which LP we treat as the primal and which we treat as the dual: if they are both feasible, then their optimal values are equal.

In passing, we mention that the concept of dual pairs is pervasive in modern mathematics. Some examples of duality in mathematics are: dual vector spaces, point-line duality in projective geometry, duality of planar graphs, duality of lattices, the duality between a function and its Fourier transform, etc.

# 5 Complementary Slackness

Complementary slackness is an easy but very useful consequence of LP duality. It gives a combinatorial condition for the optimality of an LP solution. We will make extensive use of complementary slackness when we discuss primal-dual algorithms.

**Theorem 8.** *Let $x$ be a feasible solution to the primal LP* (8)–(10)*, and let $y$ be a feasible solution to the dual LP* (11)–(13)*. Then $x$ and $y$ are both optimal if and only if the following conditions are satisfied:*

$$\forall i \in \{1, \dots, m\} : (b_i - (Ax)_i)y_i = 0$$
$$\forall i \in \{1, \dots, n\} : ((A^\intercal y)_j - c_j)x_j = 0$$

*Proof.* By Theorem 4, $x$ and $y$ are optimal if and only if $c^\intercal x = b^\intercal y = v$. By the feasibility of $x$ and $y$, we have

$$y^\intercal Ax \leq y^\intercal b = v$$
$$y^\intercal Ax \geq c^\intercal x = v$$

Therefore,

$$y^\intercal Ax = c^\intercal x = y^\intercal b \tag{14}$$

(14) implies that $y^\intercal(b - Ax) = 0$. Because, by the feasibility of $y$ and $x$, each coordinate of $y$ and each coordinate of $b - Ax$ is non-negative, their inner product $y^\intercal(b - Ax)$ can be 0 if and only if $(b_i - (Ax)_i)y_i = 0$ for every $i \in \{1, \dots, m\}$. This is the first condition of the theorem. Similarly, (14) implies $(y^\intercal A - c^\intercal)x = 0$, which gives the second condition of the theorem. □

Let us spell out what Theorem 8 says in words: a primal feasible solution $x$ and a dual feasible solution $y$ are optimal if and only if:

- whenever a dual variable $y_i$ is positive, the corresponding primal constraint is tight, i.e. $(Ax)_i = b_i$; conversely, if the primal constraint is "slack", i.e. $(Ax)_i < b_i$, then the corresponding dual variable is 0.
- whenever a primal variable $x_j$ is positive, the corresponding dual constraint is tight, i.e. $(A^\intercal y)_j = c_j$; conversely, if the dual constraint is slack, i.e. $(A^\intercal y)_j > c_j$, then the corresponding dual variable is 0.

# 6 Max Flow - Min Cut via LP

We will prove the Max Flow - Min Cut theorem via LP duality and complementary slackness. Let us recall some notation first. We consider a *directed* graph $G = (V, E)$, in which each edge $e \in E$ is given a capacity $c_e$. We are also given two vertices $s, t \in V$. In the maximum flow problem, our goal is compute a flow from $s$ to $t$, i.e. a flow vector $f \in \mathbb{R}^E$, such that:

- the flow leaving $s$, $\sum_{(s,v)\in E} f_{sv} - \sum_{(v,s)\in E} f_{vs}$ is maximized;
- the flow satisfies the non-negativity and capacity constraints: $0 \le f_e \le c_e$ for every $e \in E$;
- the flow satisfies the flow conservation constraints: for every $u \in V \setminus \{s, t\}$, $\sum_{(v,u)\in E} f_{vu} = \sum_{(u,v)\in E} f_{uv}$.

An $s$-$t$ cut in the graph is a partition of $V$ into two sets $S, \bar{S}$, such that $s \in S$ and $t \in \bar{S}$. The value of the cut is $c(S, \bar{S}) = \sum_{(u,v)\in E: u\in S, v\in \bar{S}} c_{uv}$.

**Theorem 9** (Max Flow - Min Cut)**.** *In any directed graph $G = (V, E)$, for any $s, t \in V$, the maximum flow from $s$ to $t$ equals the minimum value of an $s$-$t$ cut.*

To prove this theorem, let us write the maximum flow problem as a linear program:

$$\max \sum_{(s,v)\in E} f_{sv} - \sum_{(u,s)\in E} f_{us} \tag{15}$$

s.t.

$$\forall u \in V \setminus \{s, t\} : \sum_{(v,u)\in E} f_{vu} - \sum_{(u,v)\in E} f_{uv} = 0 \tag{16}$$

$$\forall e \in E : 0 \le f_e \le c_e. \tag{17}$$

**Verify that this linear program captures the maximum flow problem.** The dual of this linear program is:

$$
\begin{aligned}
&\min \sum_{e \in E} c_e y_e && (18)\\
&\text{s.t.}\\
&\forall (u,v) \in E : x_v - x_u + y_{uv} \geq 0 && (19)\\
&x_s = 1; x_t = 0 && (20)\\
&\forall e \in E : y_e \geq 0 && (21)
\end{aligned}
$$

The dual variables $y$ correspond to the primal constraints $f_e \leq c_e$. It may be helpful to observe that, in an optimal solution $x, y$, we will always have $y_{uv} = \max\{x_u - x_v, 0\}$, so, in fact, the dual LP is equivalent to minimizing $\sum_{(u,v)\in E} c_e \max\{x_u - x_v, 0\}$ subject to $x_s = 1$ and $x_t = 0$.

**Exercise 10.** *Show that the linear programs*

$$
\begin{aligned}
&\max b^{\mathsf{T}} f\\
&s.t.\\
&\quad Af = 0\\
&\quad 0 \leq f \leq c
\end{aligned}
$$

*and*

$$
\begin{aligned}
&\min c^{\mathsf{T}} y\\
&s.t.\\
&\quad A^{\mathsf{T}} x + y \geq b\\
&\quad y \geq 0
\end{aligned}
$$

*are dual to each other. Here the first program has variables $f$, and the second program has variables $x$ and $y$. To verify the duality, write the first program in standard form, dualize it, and simplify.*

**Exercise 11.** *Use the previous exercise to show that the linear program* (15)–(17) *is dual to the program*

$$
\begin{aligned}
&\min \sum_{e \in E} c_e y_e\\
&s.t.\\
&\forall (u,v) \in E, u, v \notin \{s,t\} : x_v - x_u + y_{uv} \geq 0\\
&\forall (u,s) \in E : \ -x_u + y_{us} \geq -1\\
&\forall (s,v) \in E : x_v + y_{sv} \geq 1\\
&\forall (u,t) \in E : \ -x_u + y_{ut} \geq 0\\
&\forall (t,v) \in E : x_v + y_{tv} \geq 0\\
&\forall e \in E : y_e \geq 0
\end{aligned}
$$

*which has a variable $x_u$ for every vertex of $G$* except *$s$ and $t$, and a variable $y_e$ for every edge of $G$. Then show that this program is equivalent to* (18)–(21).

*Proof of Theorem 9.* Let $F$ be the value of the maximum flow, which is also equal to the optimal value of (15)–(17), and let $C$ be the minimum value of an $s$-$t$ cut.

The inequality $F \le C$ is easy to prove directly, but let us verify that it also follows from weak duality. Indeed let $(S, \bar{S})$ be a minimum $s$-$t$ cut, i.e. an $s$-$t$ cut s.t. $C = c(S, \bar{S})$. Define a feasible solution $x', y'$ of (18)–(21) by setting $x'_u = 1$ for all $u \in S$, and $x'_u = 0$ for all $u \in \bar{S}$, and setting $y_{uv} = 1$ if and only if $(u, v) \in E$, $u \in S$, $v \in \bar{S}$. **Verify that $x'$ and $y'$ are feasible.** This solution has value $c(S, \bar{S}) = C$, and, therefore, the optimal solution of (18)–(21) has value at most $C$, and, by Theorem 3, $F$, the value of (15)–(17), is at most $C$ as well.

So far we have seen that the "easy" part of the Max Flow - Min Cut theorem, the inequality $F \le C$, follows from the "easy" part of LP duality, weak duality. Next, we will see that the "hard" part of the theorem follows from strong duality, in the form of complementary slackness. Let $f, x, y$ be the optimal solutions of (15)–(17) and (18)–(21). Define an $s$-$t$ cut $(S, \bar{S})$ by $S = \{u \in V : x_u \ge 1\}$, $\bar{S} = \{u \in V : x_u < 1\}$. We make the following observations:

- $s \in S$;
- If $(u, v) \in E$, $u \in S$, $v \in \bar{S}$, then $y_{uv} \ge x_u - x_v > 0$, so, by Theorem 8, the corresponding primal constraint is tight, i.e. $f_e = c_e$;
- If $(u, v) \in E$, $u \in \bar{S}$, $v \in S$, then $x_v - x_u + y_{uv} \ge x_v - x_u > 0$. Therefore, by Theorem 8, the corresponding primal variable is 0, i.e. $f_{uv} = 0$.

By (16) (i.e. flow conservation), $\sum_{(u,v)\in E} f_{uv} - \sum_{(v,u)\in E} f_{vu} = 0$ for all $u \in S \setminus \{s\}$. We can then write

$$F = \sum_{(s,v)\in E} f_{sv} - \sum_{(v,s)\in E} f_{vs} = \sum_{u\in S} \sum_{(u,v)\in E} f_{uv} - \sum_{u\in S} \sum_{(v,u)\in E} f_{vu}.$$

Observe now that, on the right hand side, for any edge $(u, v)$ such that $u$ and $v$ are both in $S$, $f_{uv}$ appears twice – once with a positive sign, and once with a negative – and therefore, cancels. On the other hand, for any edge $(u, v)$ such that $u$ and $v$ are both in $\bar{S}$, $f_{uv}$ does not appear at all. So, the only terms that remain are $f_{uv}$, where $(u, v) \in E$, $u \in S$, $v \in \bar{S}$, and $-f_{uv}$, where $(u, v) \in E$, $u \in \bar{S}$, $v \in S$. I.e.

$$F = \sum_{(u,v)\in E: u\in S, v\in \bar{S}} f_{uv} - \sum_{(u,v)\in E: u\in \bar{S}, v\in S} f_{uv} = \sum_{(u,v)\in E: u\in S, v\in \bar{S}} c_e = c(S, \bar{S}).$$

Here in the second equation we used the observations based on complementary slackness we made in the previous paragraph. Since $c(S, \bar{S}) \ge C$, we have shown that $F \ge C$. Combining with the easy inequality $F \le C$ that we already proved, we have $F = C$. $\square$

Notice that in the proof of Theorem 9 we have in fact shown that the the optimal value of (18)–(21) equals the minimum value of an $s$-$t$-cut, and is achieved by an *integral solution*, i.e. one in which all coordinates of $x$ and $y$ are integers, and, in fact are either 0 or 1.