Introduction to Linear Programming

1 Preliminaries

1.1 Different forms of linear programs

There are a variety of ways to write linear programs, and a variety of names to refer to them. We shall stick to two forms: the standard and the canonical forms. Different authors have different opinions on what standard is and what canonical is. Each form has two versions: the maximization and the minimization versions. Fortunately, all versions and forms are equivalent.

The min version of the standard form generally reads

$$\begin{array}{rccccccccc}
\min & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{subject to} & a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\
& a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\
& \vdots & & \vdots & & \dots & & \vdots & = & \vdots \\
& a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & = & b_m \\
& & & & & & & \multicolumn{3}{r}{x_i \geq 0, \forall i = 1, \dots, n,}
\end{array}$$

where the a_{ij}, c_j, and b_i are given real constants, and the x_j are the variables. The linear function $c_1x_1 + c_2x_2 + \cdots + c_nx_n$ is called the *objective function*. To solve a linear program is to find some combination of x_j satisfying the constraint set, at the same time minimize the objective function. The constraints $x_i \geq 0$ are also referred to as the *non-negativity constraints*. If the objective is to maximize instead of minimize, we have the max version of the standard form.

In canonical form, the min version reads

$$\begin{array}{rccccccccc}
\min & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{subject to} & a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \geq & b_1 \\
& a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \geq & b_2 \\
& \vdots & & \vdots & & \dots & & \vdots & \geq & \vdots \\
& a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \geq & b_m \\
& & & & & & & \multicolumn{3}{r}{x_i \geq 0, \forall i = 1, \dots, n \ ,}
\end{array}$$

and the max version is nothing but

$$\begin{array}{rccccccccc}
\max & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{subject to} & a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \leq & b_1 \\
& a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \leq & b_2 \\
& \vdots & & \vdots & & \dots & & \vdots & \leq & \vdots \\
& a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \leq & b_m \\
& & & & & & & \multicolumn{3}{r}{x_i \geq 0, \forall i = 1, \dots, n \ .}
\end{array}$$

One of the reasons we change $\geq$ to $\leq$ when moving from the min to the max version is that it might be intuitively easier to remember: if we are trying to minimize some function of x, there should be some "lower bound" on how small x can get, and vice versa. Obviously, exchanging the terms on both sides and the inequalities are reversed. Another reason for changing $\geq$ to $\leq$ has to do with the notion of duality, as we will see later.

Henceforth, when we say "vector" we mean column vector, unless explicitly specify otherwise. To this end, define the following vectors and a matrix

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \dots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

(We shall use bold-face letters to denote vectors and matrices.) Then, we can write the min and the max versions of the standard form as

$$\min \left\{ \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \right\}, \text{ and } \max \left\{ \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \right\}.$$

You get the idea? The versions for the canonical form are

$$\min \left\{ \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0} \right\}, \text{ and } \max \left\{ \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0} \right\}.$$

A vector $\mathbf{x}$ satisfying the constraints is called a *feasible solution*. Feasible solutions are not necessarily optimal. An *optimal solution* is a feasible vector $\mathbf{x}$ which, at the same time, also minimizes (or maximizes) the objective function. A linear program (LP) is *feasible* if it has a feasible solution. Later, we shall develop conditions for an LP to be feasible.

1.2 Converting general LPs to standard and canonical forms

In general, a linear program could be of any form and shape. There may be a few equalities, inequalities; there may not be enough non-negativity constraints, there may also be non-positivity constraints; the objective might be to maximize instead of minimize; etc.

We resort to the following rules to convert one LP to another.

- $\max \mathbf{c}^T\mathbf{x} = \min(-\mathbf{c})^T\mathbf{x}$
- $\sum_j a_{ij}x_j = b_i$ is equivalent to $\sum_j a_{ij}x_j \leq b_i$ and $\sum_j a_{ij}x_j \geq b_i$.
- $\sum_j a_{ij}x_j \leq b_i$ is equivalent to $-\sum_j a_{ij}x_j \geq -b_i$
- $\sum_j a_{ij}x_j \leq b_i$ is equivalent to $\sum_j a_{ij}x_j + s_i = b_i, s_i \geq 0$. The variable s_i is called a *slack variable*.
- When $x_j \leq 0$, replace all occurrences of x_j by $-x'_j$, and replace $x_j \leq 0$ by $x'_j \geq 0$.
- When x_j is not restricted in sign, replace it by $(u_j - v_j)$, and $u_j, v_j \geq 0$.

Exercise 1. Write

$$\begin{array}{lrrrrcr} \min & x_1 & - \; x_2 & + \; 4x_3 & & & \\ \text{subject to} & 3x_1 & - \; x_2 & & & = & 3 \\ & & - \; x_2 & & + \; 2x_4 & \geq & 4 \\ & x_1 & & + \; x_3 & & \leq & -3 \\ & & & & x_1, x_2 & \geq & 0 \end{array}$$

in all four forms.

Exercise 2. Write

$$\max \left\{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} \le \mathbf{b} \right\}$$

in all four forms.

Exercise 3. Write

$$\min \left\{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} \ge \mathbf{b} \right\}$$

in all four forms.

Exercise 4. Write

$$\max \left\{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b} \right\}$$

in all four forms.

Exercise 5. Convert each form to each of the other three forms.

Exercise 6. Consider the following linear program

$$\begin{array}{ll} \max & \mathbf{a^T x} + \mathbf{b^T y} + \mathbf{c^T z} \\ \text{subject to} & \begin{array}{ccccccc} \mathbf{A}_{11}\mathbf{x} & + & \mathbf{A}_{12}\mathbf{y} & + & \mathbf{A}_{13}\mathbf{z} & = & \mathbf{d} \\ \mathbf{A}_{21}\mathbf{x} & + & \mathbf{A}_{22}\mathbf{y} & + & \mathbf{A}_{23}\mathbf{z} & \le & \mathbf{e} \\ \mathbf{A}_{31}\mathbf{x} & + & \mathbf{A}_{32}\mathbf{y} & + & \mathbf{A}_{33}\mathbf{z} & \ge & \mathbf{f} \end{array} \\ & \mathbf{x} \ge \mathbf{0}, \mathbf{y} \le \mathbf{0}. \end{array}$$

Note that $\mathbf{A_{ij}}$ are matrices and $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{x}, \mathbf{y}, \mathbf{z}$ are vectors. Rewrite the linear program in standard form (max version) and in canonical form (max version).

Because the forms are all equivalent, without loss of generality we can work with the min version of the standard form. The reason for choosing this form is technical, as shall be seen in later sections.

2 A geometric view of linear programming

2.1 Polyhedra

Consider an LP in canonical form with two variables, it is easy to see that the feasible points lie in a certain region defined by the inequalities. The objective function defines a *direction of optimization*. Consequently, if there is an optimal solution, there is a vertex on the feasible region which is optimal. We shall develop this intuition into more rigorous analysis in this section.

Definition 2.1. A *polyhedron* is the set of points satisfying $\mathbf{A}\mathbf{x} \le \mathbf{b}$ (or equivalently $\mathbf{A}'\mathbf{x} \ge \mathbf{b}'$) for some $m \times n$ matrix $\mathbf{A}$, and $\mathbf{b} \in \mathbb{R}^m$. In other words, A *polyhedron* in $\mathbb{R}^n$ is the intersection of a finite set of half spaces of $\mathbb{R}^n$.

Consider the standard form of an LP:

$$\min \left\{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \ge \mathbf{0} \right\}.$$

Let $P := \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$, i.e. P consists of all feasible solutions to the linear program; then, P is a polyhedron in $\mathbb{R}^n$. For, we can rewrite P as

$$P = \left\{ \mathbf{x} : \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \\ -\mathbf{I} \end{bmatrix} \mathbf{x} \le \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\}.$$

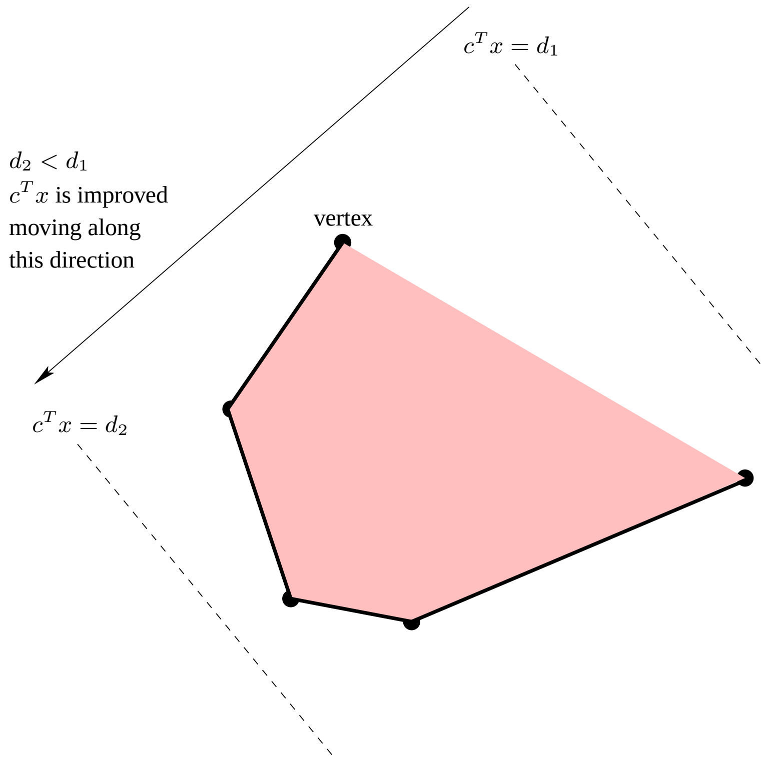

Figure 1: Polyhedron, vertices, and direction of optimization

(Actually, this polyhedron lies in an $(n-1)$-dimensional space since each equality in $Ax = b$ reduces the dimension by one.)

Refer to Figure 1 following the discussion below. Geometrically, each equation in the system $\mathbf{Ax} = \mathbf{b}$ defines a hyperplane of dimension $n-1$. In general, a vector $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}$ lies in the intersection of all m hyperplanes defined by $\mathbf{Ax} = \mathbf{b}$. The intersection of two $(n-1)$-dimensional hyperplanes is generally a space of dimension $n-2$. On the same line of reasoning, the solution space to $\mathbf{Ax} = \mathbf{b}$ is generally an $(n-m)$-dimensional space. The non-negativity condition $\mathbf{x} \geq \mathbf{0}$ restricts our region to the non-negative orthant of the original n-dimensional space. The part of the $(n-m)$-dimensional space which lies in the non-negative orthant is a polyhedral-shaped region, which we call a polyhedron. For example, when $n = 3$ and $m = 1$, we look at the part of a plane defined by $\mathbf{Ax} = \mathbf{b}$ which lies in the non-negative orthant of the usual three dimensional space. This part is a triangle if the (only) equality in $\mathbf{Ax} = \mathbf{b}$ is, say, $x_1 + x_2 + x_3 = d > 0$.

It is sometime easier to look at the LP in its canonical form $\min\left\{\mathbf{c}^T\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\right\}$. Each inequality in $\mathbf{Ax} \geq \mathbf{b}$ defines a half space. (See Figure 2.) Each inequality in $\mathbf{x} \geq \mathbf{0}$ also defines a half space. Hence, the feasible region is the intersection of $m + n$ half spaces.

Now, let us take into account the objective function $\mathbf{c}^T\mathbf{x}$. For each real constant d, $\mathbf{c}^T\mathbf{x} = d$ defines a plane. As d goes from $-\infty$ to ∞, $\mathbf{c}^T\mathbf{x} = d$ defines a set of parallel planes. The first plane which hits the feasible region defines the optimal solution(s). Think of sweeping a line from left to right until it touches a polygon on a plane. Generally, the point of touching is a vertex of the polygon. In some cases, we might touch an edge of the polygon first, in which case we have infinitely many optimal solutions. In the case the polygon degenerates into an infinite band, we might not have any optimal solution at all.

Definition 2.2. A *vertex* of a polyhedron P is a point $\mathbf{x} \in P$ such that there is no non-zero vector $\mathbf{y}$ for which $\mathbf{x} + \mathbf{y}$ and $\mathbf{x} - \mathbf{y}$ are both in P. A polyhedron which has a vertex is called a *pointed polyhedron*.

Exercise 7. We can define a point $\mathbf{v}$ in a polyhedron P to be a vertex in another way: $\mathbf{v} \in P$ is a vertex

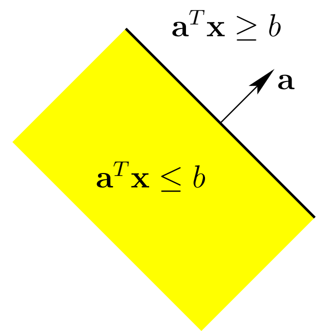

Figure 2: Each inequality $\mathbf{c}^T\mathbf{x} \leq d$ defines a halfspace.

if and only if there are no distinct points $\mathbf{u}, \mathbf{w} \in P$ such that $\mathbf{v} = (\mathbf{u} + \mathbf{w})/\mathbf{2}$. Show that this definition is equivalent to the definition given in Definition 2.2.

The following exercise confirms a different intuition about vertices: a vertex is at the intersection of n linearly independent hyperplanes of the polyhedron $\mathbf{Ax} \leq \mathbf{b}$. Henceforth, for any positive integer m we use $[m]$ to denote the set $\{1, \ldots, m\}$.

Exercise 8. Let $P = \{\mathbf{x} \mid \mathbf{Ax} \leq \mathbf{b}\}$, where $\mathbf{A}$ is an $m \times n$ matrix. For each $i \in [m]$, let $\mathbf{a}^{(i)}$ denote the ith row vector of $\mathbf{A}$. Show that $\mathbf{v} \in P$ is a vertex iff rank $\left\{\mathbf{a}^{(i)} \mid \mathbf{a}^{(i)}\mathbf{v} = b_i\right\} = n$.

We now can convert our observation about an optimal solution at a vertex into rigorous analysis. We would like to know a few things:

1. When is an LP feasible? Or, equivalently, when is a polyhedron not empty?
2. When is a polyhedron pointed?
3. When is a point in a polyhedron a vertex? Characterize vertices.
4. If a polyhedron is pointed, and if it is bounded at the direction of optimization, is it true that there is an optimal vertex?
5. If there is an optimal vertex, how do we find one?

We shall put off the first and the fifth questions for later. Let us attempt to answer the middle three questions.

Theorem 2.3. *A non-empty polyhedron is pointed if and only if it does not contain a line.*

Proof. We give a slightly intuitive proof. The proof can be turned completely rigorous easily.

Consider a non-empty polyhedron $P = \{\mathbf{x} \mid \mathbf{Ax} \leq \mathbf{b}\}$ which does not contain any line. Let S be the set of m hyperplanes defined by $\mathbf{Ax} = \mathbf{b}$. Consider a particular point $\bar{\mathbf{x}} \in P$. Note that $\bar{\mathbf{x}}$ must lie on or strictly on one side of each of the hyperplanes in S. Suppose $\bar{\mathbf{x}}$ lies on precisely k ($0 \leq k \leq m$) of the hyperplanes in S. Call this set of hyperplanes S'. If $\bar{\mathbf{x}}$ is not a vertex, then there is some $\mathbf{y} \neq \mathbf{0}$ such that both $(\bar{\mathbf{x}} - \mathbf{y})$ and $(\bar{\mathbf{x}} + \mathbf{y})$ are in P. It follows that the line $\bar{\mathbf{x}} + \alpha y$, $\alpha \in \mathbb{R}$, must be entirely on all hyperplanes of S'. Since P does not contain the line $\bar{\mathbf{x}} + \alpha\mathbf{y}$, this line must cut a plane in $S - S'$ at a point $\mathbf{x}'$. (Note, this argument also shows $S - S' \neq \emptyset$.) Now, replace $\bar{\mathbf{x}}$ by $\mathbf{x}'$, then the set S' for x' is increased by at least 1. Keep doing this at most m times and we get to a vertex.

(To be "rigorous", we must carefully pick a value of α so that there is at least one more equality in the system $\mathbf{A}(\bar{\mathbf{x}} + \alpha\mathbf{y}) \leq \mathbf{b}$ than in the system $\mathbf{A}\bar{\mathbf{x}} \leq \mathbf{b}$.)

Conversely, suppose P has a vertex $\mathbf{v}$ and also contains some line $\mathbf{x}+\alpha\mathbf{y}$, $\mathbf{y} \neq \mathbf{0}$, which means $\mathbf{A}(\mathbf{x}+\alpha\mathbf{y}) \leq \mathbf{b}, \forall\alpha$. This can only happen when $\mathbf{A}\mathbf{y} = \mathbf{0}$ (why?). But then $\mathbf{A}(\mathbf{v}+\mathbf{y}) = \mathbf{A}(\mathbf{v}-\mathbf{y}) = \mathbf{A}\mathbf{v} \leq \mathbf{b}$, contradicting the fact that $\mathbf{v}$ is a vertex. In fact, if $\mathbf{x}+\alpha\mathbf{y}$ is a line contained in P, then for any point $\mathbf{z} \in P$, the line $\mathbf{z}+\alpha\mathbf{y}$ (parallel with the other line) has to also be entirely in P. $\square$

Corollary 2.4. *A non-empty polyhedron $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ is pointed if and only if* $\operatorname{rank}(\mathbf{A}) = n$.

Proof. We only need to show that $\operatorname{rank}(\mathbf{A}) = n$ if and only if P contains no line.

Firstly, assume $\operatorname{rank}(\mathbf{A}) = n$. If P has a line $\mathbf{x}+\alpha\mathbf{y}$, for $\mathbf{y} \neq \mathbf{0}$, then it is necessary that $\mathbf{A}\mathbf{y} = \mathbf{0}$, which means $\operatorname{rank}(\mathbf{A}) < n$ (why?), which is a contradiction.

Conversely, if $\operatorname{rank}(\mathbf{A}) < n$, then the columns of $\mathbf{A}$ are linearly dependent, i.e. there is a non-zero vector $\mathbf{y}$ such that $\mathbf{A}\mathbf{y} = \mathbf{0}$. If $\mathbf{x}$ is any point in P, then $\mathbf{A}(\mathbf{x}+\alpha\mathbf{y}) = \mathbf{A}\mathbf{x} \leq \mathbf{b}, \forall\alpha \in \mathbb{R}$, implying P contains the line $\mathbf{x}+\alpha\mathbf{y}$. $\square$

Exercise 9. Prove Corollary 2.4 directly using the vertex definition in Exercise 8.

Corollary 2.5. *A non-empty polyhedron $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is always pointed.*

Proof. Rewrite P as

$$P = \left\{ \mathbf{x} : \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \\ -\mathbf{I} \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\}.$$

Then, P as a vertex by the previous corollary since

$$\operatorname{rank}\left(\begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \\ -\mathbf{I} \end{bmatrix} \right) = n.$$

$\square$

Exercise 10. Show that a non-empty polyhedron $P = \{\mathbf{x} \mid \mathbf{A}_1\mathbf{x} = \mathbf{b}_1, \mathbf{A}_2\mathbf{x} \leq \mathbf{b}_2, \mathbf{x} \geq \mathbf{0}\}$ is pointed. Moreover, suppose k is the total number of rows of $\mathbf{A}_1$ and $\mathbf{A}_2$. Show that a vertex $\mathbf{x}^*$ of P has at most m positive components.

The following theorem characterizes the set of vertices of the polyhedron $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$.

Theorem 2.6. *Let $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. Then $\mathbf{v} \in P$ is a vertex if and only if the column vectors of A corresponding to non-zero coordinates of $\mathbf{v}$ are linearly independent.*

Proof. Let J be the index set of non-zero coordinates of $\mathbf{v}$. Let $\mathbf{a_j}$ be the jth column vector of $\mathbf{A}$.

Suppose $\mathbf{v}$ is a vertex. We want to show that $\{\mathbf{a_j} \mid j \in J\}$ is a set of independent vectors. This is equivalent to saying that the system $\sum_{j \in J} \mathbf{a_j} x_j = \mathbf{b}$ has a unique solution. If $\mathbf{y}$ is another solution (other than v restricted to J) of this system, then adding more 0-coordinates to $\mathbf{y}$ corresponding to the indices not in J, we get an n-dimensional vector $\mathbf{z}$ with $\mathbf{A}\mathbf{z} = \mathbf{b}$ and $\mathbf{z} \neq \mathbf{v}$. With sufficiently small α, both $\mathbf{v}+\alpha(\mathbf{v}-\mathbf{z})$ and $\mathbf{v}-\alpha(\mathbf{v}-\mathbf{z})$ are feasible (why?), contradicting the fact that $\mathbf{v}$ is a vertex.

Conversely, suppose $\sum_{j \in J} \mathbf{a_j} x_j = \mathbf{b}$ has a unique solution. If there is a $\mathbf{y} \neq \mathbf{0}$ such that $\mathbf{v}+\mathbf{y}$ and $\mathbf{v}-\mathbf{y}$ are both in P, then $y_j = 0$ whenever $j \notin J$ (why?). Hence, $\mathbf{b} = \mathbf{A}(\mathbf{v}+\mathbf{y}) = \sum_{j \in J} \mathbf{a_j}(v_j + y_j)$, contradicting the uniqueness of the solution to $\sum_{j \in J} \mathbf{a_j} x_j = \mathbf{b}$. $\square$

Exercise 11. Prove Corollary 2.4 directly using Theorem 2.6.

Lemma 2.7. *Let $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. If $\min\left\{\mathbf{c}^{\mathbf{T}}\mathbf{x} \mid \mathbf{x} \in P\right\}$ is bounded (i.e. it has an optimal solution), then for all $\mathbf{x} \in P$, there is a vertex $\mathbf{v} \in P$ such that $\mathbf{c}^T\mathbf{v} \leq \mathbf{c}^T\mathbf{x}$.*

Proof. We proceed in much that same way as in the proof of Theorem 2.3, where we start from a point $\mathbf{x}$ inside P, and find a vertex by keep considering lines going through $\mathbf{x}$.

A slight difference is that here we already have m hyperplanes $\mathbf{Ax} = \mathbf{b}$. These planes play the role of S' in Theorem 2.3's proof. The n half spaces $\mathbf{x} \geq \mathbf{0}$ play the role of $S - S'$. Another difference is that, starting from a point $\mathbf{x}$ in P, we now have to find a vertex with better cost. Hence, we have to be more careful in picking the direction to go.

What do I mean by "direction to go"? Suppose $\mathbf{x} \in P$ is not a vertex. We know there is $\mathbf{y} \neq \mathbf{0}$ such that $\mathbf{x} + \mathbf{y}, \mathbf{x} - \mathbf{y} \in P$. From $\mathbf{x}$, we could either go along the $+\mathbf{y}$ direction or the $-\mathbf{y}$ direction, hoping to improve the cost function, while wanting to meet another plane defined by $\mathbf{x} \geq \mathbf{0}$. The $+\mathbf{y}$ direction is better iff $\mathbf{c^T}(\mathbf{x} + \mathbf{y}) \leq \mathbf{c^T x}$, or $\mathbf{c^T y} \leq \mathbf{0}$. The $-\mathbf{y}$ direction is better iff $\mathbf{c^T}(-\mathbf{y}) \leq \mathbf{0}$. Let $\mathbf{z} \in \{\mathbf{y}, -\mathbf{y}\}$ be the better direction, i.e. $\mathbf{c^T z} \leq \mathbf{0}$.

Note that $\mathbf{A}(\mathbf{x} + \mathbf{y}) = \mathbf{A}(\mathbf{x} - \mathbf{y}) = \mathbf{b}$ implies $\mathbf{Az} = \mathbf{0}$.

We shall go along the ray $\mathbf{x} + \alpha\mathbf{z}$, $\alpha > 0$. We knew going along this ray would improve the objective function. The problem is that we might not meet any bounding face of P. When would this happen? Firstly, note that $\mathbf{A}(\mathbf{x} + \alpha\mathbf{z}) = \mathbf{Ax} = \mathbf{b}$, implying that the ray $(\mathbf{x} + \alpha\mathbf{z})$ is entirely on each of the m planes defined by $\mathbf{Ax} = \mathbf{b}$. Now, let's look at the hyperplanes $x_1 = 0$, $x_2 = 0$, ..., $x_n = 0$. Suppose $\mathbf{x}$ is already on k of them, where $0 \leq k \leq n$. Without loss of generality, assume $x_1 = \cdots = x_k = 0$, and the rest of the coordinates are positive. Since $\mathbf{x} + \mathbf{y}, \mathbf{x} - \mathbf{y} \in P$, we know $x_j + y_j \geq 0$ and $x_j - y_j \geq 0$, $\forall j = 1, \dots, k$. Thus, $x_j + \alpha z_j = 0, \forall j = 1, \dots, k, \alpha > 0$. The line $\mathbf{x} + \alpha\mathbf{z}$ is also on all of those k planes.

How about the indices $i = k + 1, \dots, n$?

If $z_j \geq 0$ for all $j = k + 1, \dots, n$, then $x_j + \alpha z_j \geq 0$ for all $i = k + 1, \dots, n$, also. This means $(\mathbf{x} + \alpha\mathbf{z}) \in P$ for all $\alpha > 0$. This is the case where we do not meet any boundary face. If $\mathbf{c^T z} < \mathbf{0}$, then $\mathbf{c}^T(\mathbf{x} + \alpha\mathbf{z})$ goes to $-\infty$: the LP is not bounded. If $\mathbf{c^T z} = \mathbf{0}$, then replace $\mathbf{z}$ by $-\mathbf{z}$ to avoid $\mathbf{z}$ having all non-negative coordinates. (Note that $\mathbf{y} \neq \mathbf{0}$ implies $\mathbf{y}$ or $-\mathbf{y}$ has negative coordinates.) What's happening here is that, when $\mathbf{c^T z} = \mathbf{0}$, going to the $\mathbf{z}$ direction is perpendicular to the direction of optimization, meaning we don't get any improvement on the objective function. However, we must still meet one of the bounding faces if we go the right way. And, the right way is to the $\mathbf{z}$ with some negative coordinates.

If $z_j < 0$ for some $j = k + 1, \dots, n$, then $x_j + \alpha z_j$ cannot stay strictly positive forever. Thus, we will meet one (or a few) more of the planes $\mathbf{x} = \mathbf{0}$ when α is sufficiently large. Let $\mathbf{x}'$ be the first point we meet, and replace $\mathbf{x}$ by $\mathbf{x}'$. (You should try to define $\mathbf{x}'$ precisely.) The new point $\mathbf{x}$ has more 0-coordinates. The process cannot go on forever, since the number of 0-coordinates is at most n. Thus, eventually we shall meet a vertex. □

Exercise 12. Let $P = \{\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}\}$ be a pointed polyhedron. Suppose the LP $\min\{\mathbf{c^T x} \mid \mathbf{x} \in P\}$ has an optimal solution. Show that the LP has an optimal solution at a vertex. Note that this exercise is a slight generalization of Lemma 2.7.

Theorem 2.8. *The linear program* $\min\{\mathbf{c^T x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ *either*

1. *is infeasible,*
2. *is unbounded, or*
3. *has an optimal solution at a vertex.*

Proof. If the LP is feasible, i.e. $P = \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is not empty, then its objective function is either bounded or unbounded. If the objective function is bounded and P is not empty, starting from a point $\mathbf{x} \in P$, we can find a vertex with better cost. Exercise 19 shows that there can only be a finite number of vertices, hence a vertex with the best cost would be optimal. □

Exercise 13. A set S of points in $\mathbb{R}^n$ is said to be *convex* if for any two points $\mathbf{x}, \mathbf{y} \in S$, all points on the segment from $\mathbf{x}$ to $\mathbf{y}$, i.e. points of the form $\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x}), 0 \leq \alpha \leq 1$, are also in S.

Show that each of the following polyhedra are convex:

1. $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$
2. $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}\}$
3. $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$
4. $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$

Thus, in fact the feasible set of solutions of any LP is convex.

Exercise 14 (Convex Hull). Let S be a (finite or infinite) set of points (or vectors) in $\mathbb{R}^n$. Let H denote the set of all points $\mathbf{h} \in \mathbb{R}^n$ such that, for each $\mathbf{h} \in H$, there is some positive integer k, some points $\mathbf{v_1}, \dots, \mathbf{v_k} \in S$, and some *positive* numbers $\alpha_1, \dots, \alpha_k$ such that

$$\mathbf{h} = \sum_{i=1}^{k} \alpha_i \mathbf{v_i} \quad \text{and} \quad \sum_{i=1}^{k} \alpha_i = 1.$$

(The vector $\mathbf{h}$ is expressed as a *convex combination* of the vectors $\mathbf{v_1}, \dots, \mathbf{v_k}$.) Show that

(i) $S \subseteq H$.

(ii) H is convex.

(iii) Every convex set containing S also contains H.

The set H is unique for each S, and H is called the *convex hull* of S.

Exercise 15 (Carathéodory, 1907). Prove that, if $S \subseteq \mathbb{R}^n$ then a point $\mathbf{v}$ belongs to the convex hull of S if and only if $\mathbf{v}$ is a convex combinations of at most $n+1$ points in S.

Exercise 16. Let S be any subset of $\mathbb{R}^n$. Prove that the convex hull of S is the set of all convex combinations of affinely independent vectors from S. Use this result to prove Carathéodory's theorem.

Exercise 17. Show that, if a system $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ on n variables has no solution, then there is a subsystem $\mathbf{A}'\mathbf{x} \leq \mathbf{b}'$ of at most $n+1$ inequalities having no solution.

Exercise 18. In $\mathbb{R}^2$, the polyhedron

$$P = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} : 0 \leq x_1 \leq 1 \right\}$$

has no vertex. (Why?)

Consider a linear program $\min\{x_1 \mid \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \in P\}$.

1. Rewrite the LP in standard form: $\min\{\bar{\mathbf{c}}^{\mathbf{T}}\mathbf{z} \mid \mathbf{z} \in P'\}$ for $P' = \{\mathbf{z} \mid \mathbf{A}\mathbf{z} = \mathbf{b}, \mathbf{z} \geq \mathbf{0}\}$. (You are to determine what $\bar{\mathbf{c}}$, $\mathbf{A}$ and $\mathbf{b}$ are.)
2. Does P' has a vertex? If it does, specify one and show that it is indeed a vertex of P'.

Exercise 19. Consider the polyhedron $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. Suppose the dimension of $\mathbf{A}$ is $m \times n$. We assume that rank$(\mathbf{A}) = m \leq n$. (Otherwise some equations are redundant.) Show that

1. If $\mathbf{v}$ is a vertex, then $\mathbf{v}$ has at least $n - m$ zero coordinates.
2. Show that P has at most $\binom{n}{n-m}$ vertices.

Exercise 20. Show that every vertex of a pointed polyhedron is the unique optimal solution over P of some linear cost function.

3 The Simplex Method

3.1 A high level description

Let us consider the LP $\min\{\mathbf{c^T x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. We shall answer the feasibility question later. Let us assume for now that the convex polyhedron $P = \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is not empty. From previous sections, we know that P is pointed. Moreover, if $\min\{\mathbf{c^T x} \mid \mathbf{x} \in P\}$ is bounded, i.e. the LP has an optimal solution, then there is an optimal solution at a vertex.

We shall not discuss the simplex method in all its rigor. The main ideas are needed to gain a solid understanding of the linear algebra of convex polyhedra, which is essential to apply linear programming methods to design approximation algorithms.

The idea of the simplex method is quite simple. We start off from a vertex, which is also called a *basic feasible solution*, then we attempt to move along an edge of P to another vertex toward the direction of optimization. We shall make sure that each move does not increase the objective function.

(Terminologically, an $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{b}$ is a *solution*. If $\mathbf{x} \geq \mathbf{0}$ also holds, then the solution is *feasible* for the LP. A feasible solution is *basic* iff the columns of A corresponding to non-zero components of x are linearly independent.)

In general, a vertex is the intersection of exactly n different (affine) hyperplanes. (In the so-called *degenerate cases*, a vertex might be at the intersection of more than n hyperplanes.) An edge is the intersection of $n-1$ hyperplanes. Removing one hyperplane from the n planes which defines a vertex $\mathbf{v}$, and we have an edge at which $\mathbf{v}$ is on. Thus, in most cases $\mathbf{v}$ is incident to n edges. We need to pick an edge to move along from $\mathbf{v}$ until we meet another hyperplane, which would be another vertex $\mathbf{v'}$. The main idea is to find $\mathbf{v'}$ such that $\mathbf{c^T v'} \leq \mathbf{c^T v}$. The algorithm terminates when no move would improve the objective function.

3.2 An example

Example 3.1. To put the idea of the simplex method into place, let us consider an example.

$$\begin{array}{lccccccc}
\max & 3x_1 & + & 2x_2 & + & 4x_3 & & \\
\text{subject to} & x_1 & + & x_2 & + & 2x_3 & \leq & 4 \\
 & 2x_1 & & & + & 3x_3 & \leq & 5 \\
 & 4x_1 & + & x_2 & + & 3x_3 & \leq & 7 \\
 & & & & & x_1, x_2, x_3 & \geq & 0.
\end{array}$$

We first convert it to standard form, by adding a few slack variables.

$$\begin{array}{lcccccccc}
\max & 3x_1 & +2x_2 & +4x_3 & & & & & \\
\text{subject to} & x_1 & +x_2 & +2x_3 & +x_4 & & & = & 4 \\
 & 2x_1 & & +3x_3 & & +x_5 & & = & 5 \\
 & 4x_1 & +x_2 & +3x_3 & & & +x_6 & = & 7 \\
 & & & & & & x_1, x_2, x_3, x_4, x_5, x_6 & \geq & 0.
\end{array} \tag{1}$$

The first question is, *how do we find a vertex?* We will give a complete answer to this later. Let us attempt an ad hoc method to find a vertex for this problem.

Recall that, for a polyhedron $P = \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, a point $\mathbf{v} \in P$ is a vertex iff the columns of $\mathbf{A}$ corresponding to the non-zero components of $\mathbf{x}$ are linearly independent. If $\mathbf{A}$ is an $m \times n$ matrix, we assume rank$(\mathbf{A}) = m$ (and thus $m \leq n$), otherwise some equation(s) in $\mathbf{Ax} = \mathbf{b}$ is redundant or inconsistent with the rest. If it is inconsistent then $\mathbf{P}$ is empty. To check rank$(\mathbf{A}) = m$, Gaussian elimination can be employed.

Assume the index set for non-zero components of $\mathbf{v}$ is B, and $N = [n] - B$. The columns of $\mathbf{A}$ corresponding to B are independent, hence $|B| \leq m$. If $|B| < m$, we can certainly move a few members of N into B such that $|B| = m$ and the columns of $\mathbf{A}$ corresponding to B are still independent (extending the set of independent vectors into a basis). Conversely, if we can find m independent columns of $\mathbf{A}$ whose index set is B, then, setting all x's coordinates not in B to be 0 and solve for $\mathbf{A}_B\mathbf{x}_B = b$, we would get a vertex if $\mathbf{x}_B \geq 0$.

Let us now come back to the sample problem. The last 3 columns of A are independent. In fact, they form an identity matrix. So, if we set $B = \{4, 5, 6\}$, $N = \{1, 2, 3\}$, $x_1 = x_2 = x_3 = 0$, and $x_4 = 4$, $x_5 = 5$, $x_6 = 7$, then we have a vertex! The variables x_i, $i \in N$ are called *free variables*. The x_i with $i \in B$ are *basic variables*.

(Note that, if an LP is given in canonical form, such as $\max\{\mathbf{x} \mid \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, then after adding m slack variables we automatically obtain m independent columns of A, which would be a good place to start looking for a vertex. When an LP is given in standard form, we have to work slightly harder. One way to know if the columns are independent is to apply Gaussian elimination on the system $\mathbf{Ax} = \mathbf{b}$. The columns with non-zero pivots are independent.)

To this end, we have to find a way to realize our intuition of moving along an edge of the polyhedron to get to a vertex with better cost. The current vertex has cost $3x_1 + 2x_2 + 4x_3 = 0$. This can only be increased if we increase one or more of the free variables x_1, x_2, x_3. (Now you know why they are called free variables).

The objective function is increased with highest rate if we increase x_3, whose coefficient 4 is positive and largest among the free variables. The thing is, the three equations in $\mathbf{Ax} = \mathbf{b}$ have to be satisfied, and we also have to maintain the non-negativity of vector $\mathbf{x}$. For example, when $x_3 = \delta > 0$, the variable x_4 has to be changed to $x_4 = 4 - 2\delta$. If we want $x_4 \geq 0$, then we must have $\delta \leq 2$. Thus, with respect to the first equation, x_3 cannot be increased to more than 2. Similarly, the second and third equations restrict $\delta \leq 5/3$ and $\delta \leq 7/3$. In summary, x_3 can only be at most $5/3$, which forces

$$
\begin{aligned}
x_4 &= 4 - 2\frac{5}{3} = \frac{2}{3} \\
x_5 &= 0 \\
x_6 &= 7 - 3\frac{5}{3} = 2
\end{aligned}
$$

We then get to a new point $\mathbf{x} \in P$, where

$$\mathbf{x}^T = \begin{bmatrix} 0 & 0 & 5/3 & 2/3 & 0 & 2 \end{bmatrix}.$$

The new objective value is $4\frac{5}{3} = \frac{20}{3}$. Is this point $\mathbf{x}$ a new vertex? Indeed, the vectors

$$a_3 = \begin{bmatrix} 2 \\ 3 \\ 3 \end{bmatrix}, a_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, a_6 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

are linearly independent. The second component of a_3 is not zero, while the the other two vectors are unit vectors corresponding to the first and the third coordinates. You can see why it is very easy to check for independence when the column vectors corresponding to the basic variables are unit vectors.

To this end, we are looking at $B = \{3, 4, 6\}$, $N = \{1, 2, 5\}$. The basic variables have been changed, and the free variables are changed also. The free variable x_3 is said to *enter the basis*, and the basic variable x_5 is *leaving the basis*.

Note also that the reasoning is fairly straightforward, as we have just done, when the objective function depends only on the free variables, and the column vectors corresponding to the basic variables are unit vectors. Now, we want to turn (1) into an equivalent system in which a_3, a_4, a_6 are unit vectors. In fact, we only need to turn a_3 into $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$. This is simple: divide the second equation by 3, then subtract 2 times the second from the first, and 3 times the second from the third, we obtain

$$\begin{array}{lrrrrrrcl}
\max & 3x_1 & +2x_2 & +4x_3 & & & & & \\
\text{subject to} & -\frac{1}{3}x_1 & +x_2 & & +\mathbf{x_4} & & & = & \frac{2}{3} \\
& \frac{2}{3}x_1 & & +\mathbf{x_3} & & +\frac{1}{3}x_5 & & = & \frac{5}{3} \\
& 2x_1 & +x_2 & & & & +\mathbf{x_6} & = & 2 \\
& & & & & & x_1, x_2, x_3, x_4, x_5, x_6 & \geq & 0.
\end{array}$$

Since we want the objective function to contain only free variables, we do not want x_3 in the objective function. Replace

$$x_3 = \frac{5}{3} - \frac{2}{3}x_1 - \frac{1}{3}x_5$$

in the objective function, we get

$$\begin{aligned}
3x_1 + 2x_2 + 4x_3 &= 3x_1 + 2x_2 + 4\left(\frac{5}{3} - \frac{2}{3}x_1 - \frac{1}{3}x_5\right) \\
&= \frac{1}{3}x_1 + 2x_2 - \frac{4}{3}x_5 + \frac{\mathbf{20}}{\mathbf{3}}
\end{aligned}$$

Note that the value $20/3$ is precisely the cost of the new vertex. You can also see that the replacement of x_3 was so convenient after we turn the vector a_3 into a unit vector. Our new system becomes

$$\begin{array}{lrrrrrrcl}
\max & \frac{1}{3}x_1 & +2x_2 & & & -\frac{4}{3}x_5 & & + & \frac{\mathbf{20}}{\mathbf{3}} \\
\text{subject to} & -\frac{1}{3}x_1 & +x_2 & & +\mathbf{x_4} & & & = & \frac{2}{3} \\
& \frac{2}{3}x_1 & & +\mathbf{x_3} & & +\frac{1}{3}x_5 & & = & \frac{5}{3} \\
& 2x_1 & +x_2 & & & & +\mathbf{x_6} & = & 2 \\
& & & & & & x_1, x_2, x_3, x_4, x_5, x_6 & \geq & 0.
\end{array}$$

Now, to further improve our solution, x_2 should be increased as its coefficient in the objective function is the largest among positive ones. The most it can be increased up to is $2/3$, in which case x_2 enters the basis and x_4 leaves the basis. The new system is

$$\begin{array}{lrrrrrrcl}
\max & x_1 & & & -2x_4 & & & + & \mathbf{8} \\
\text{subject to} & -\frac{1}{3}x_1 & +\mathbf{x_2} & & +x_4 & & & = & \frac{2}{3} \\
& \frac{2}{3}x_1 & & +\mathbf{x_3} & & +\frac{1}{3}x_5 & & = & \frac{5}{3} \\
& \frac{7}{3}x_1 & & & -x_4 & -\frac{1}{3}x_5 & +\mathbf{x_6} & = & \frac{4}{3} \\
& & & & & & x_1, x_2, x_3, x_4, x_5, x_6 & \geq & 0.
\end{array}$$

Now, we want to increase x_1. In the first equation, increasing x_1 does not affect the non-negativity of x_2 at all. In fact, if we have only equations in which the coefficients of x_1 are negative (or there's no x_1), then the LP is certainly *unbounded*.

In this case, however, we can only increase x_1 to $4/7$, due to the restriction of the first and the third equation. Now x_6 leaves the basis, and x_1 enters. The new system is

$$\begin{array}{lrrrrrrcl}
\max & & & & -\frac{11}{7}x_4 & +\frac{1}{7}x_5 & -\frac{3}{7}x_6 & + & \frac{\mathbf{60}}{\mathbf{7}} \\
\text{subject to} & & +\mathbf{x_2} & & +\frac{6}{7}x_4 & -\frac{5}{7}x_5 & +\frac{1}{7}x_6 & = & \frac{6}{7} \\
& & & +\mathbf{x_3} & +\frac{2}{7}x_4 & +\frac{3}{7}x_5 & -\frac{2}{7}x_6 & = & \frac{9}{7} \\
& \mathbf{x_1} & & & -\frac{3}{7}x_4 & -\frac{1}{7}x_5 & +\frac{3}{7}x_6 & = & \frac{4}{7} \\
& & & & & & x_1, x_2, x_3, x_4, x_5, x_6 & \geq & 0.
\end{array}$$

To this end, x_5 reenters the basis and x_3 leaves:

$$\begin{array}{lrrrrrrcr}
\max & & & -\frac{1}{3}x_3 & -\frac{34}{21}x_4 & & -\frac{1}{3}x_6 & + & \mathbf{9} \\
\text{subject to} & & +\mathbf{x_2} & +\frac{49}{15}x_3 & +\frac{188}{105}x_4 & & -\frac{1}{3}x_6 & = & 3 \\
& & & +\frac{7}{3}x_3 & +\frac{2}{3}x_4 & +\mathbf{x_5} & -\frac{2}{3}x_6 & = & 3 \\
& \mathbf{x_1} & & +\frac{1}{3}x_3 & -\frac{1}{3}x_4 & & +\frac{1}{3}x_6 & = & 1 \\
& & & & & & x_1,x_2,x_3,x_4,x_5,x_6 & \geq & 0.
\end{array}$$

Clearly no more improvement is possible. The optimal value is **9**, at the vertex

$$\mathbf{v} = \begin{bmatrix}1 & 3 & 0 & 0 & 3 & 0\end{bmatrix}^T.$$

3.3 Rigorous description of a simplex iteration

Consider $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, and the linear program

$$\min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in P\}.$$

Let's assume we have a vertex $\mathbf{v} \in P$. As we have discussed earlier, we can partition $[n] = B \cup N$ such that $|B| = m$ and the columns of A corresponding to B are independent, while $v_i = 0, \forall i \in N$. Conversely, any $\mathbf{v} \in P$ satisfying this condition is a vertex.

Let $\mathbf{A}_B, \mathbf{A}_N$ be the submatrices of $\mathbf{A}$ obtained by taking the columns corresponding to B and N, respectively. Similarly, up to rearranging the variables we can write every vector $\mathbf{x} \in \mathbb{R}^n$ as $\mathbf{x} = \begin{bmatrix}\mathbf{x}_B & \mathbf{x}_N\end{bmatrix}^T$, and $\mathbf{c}^T = \begin{bmatrix}\mathbf{c}_B & \mathbf{c}_N\end{bmatrix}$. The LP is equivalent to

$$\begin{array}{lrcl}
\min & \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N & & \\
\text{subject to} & \mathbf{A}_B\mathbf{x}_B + \mathbf{A}_N\mathbf{x}_N & = & \mathbf{b} \\
& \mathbf{x} & \geq & 0.
\end{array}$$

How do we turn the columns of $\mathbf{A}_B$ into unit vectors? Easy, just multiply both sides of $\mathbf{A}x = b$ by $\mathbf{A}_B^{-1}$, which exists since the columns of $\mathbf{A}_B$ are independent. We have

$$\begin{array}{lrcl}
\min & \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N & & \\
\text{subject to} & \mathbf{x}_B + \mathbf{A}_B^{-1}\mathbf{A}_N x_N & = & \mathbf{A}_B^{-1}\mathbf{b} \\
& \mathbf{x} & \geq & 0.
\end{array}$$

We also want the objective function to depend only on free variables. Thus, we should replace $\mathbf{x}_B$ by $\mathbf{A}_B^{-1}\mathbf{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N$ in the objective function:

$$\begin{array}{rcl}
\mathbf{c}^T\mathbf{x} & = & \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N \\
& = & \mathbf{c}_B^T\left(\mathbf{A}_B^{-1}\mathbf{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N\right) + \mathbf{c}_N^T\mathbf{x}_N \\
& = & \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{A}_N\right)\mathbf{x}_N.
\end{array}$$

Let $\mathbf{y}_B^T = c_B^T A_B^{-1}$, the LP can be written as

$$\begin{array}{lrcccl}
\min & & & \left(\mathbf{c}_N^T - \mathbf{y}_B^T\mathbf{A}_N\right)\mathbf{x}_N & + & \mathbf{y}_B^T\mathbf{b} \\
\text{subject to} & \mathbf{x}_B & + & \mathbf{A}_B^{-1}\mathbf{A}_N x_N & = & \mathbf{A}_B^{-1}\mathbf{b} \\
& & & \mathbf{x} & \geq & 0.
\end{array}$$

The constant $\mathbf{y}_B^T\mathbf{b}$ is the cost of vertex $\mathbf{v}$. (In the first step of the example in the previous section, $\mathbf{y}_B^T\mathbf{b} = 20/3$.) In the objective function the coefficient of x_j is $\left(c_j - \mathbf{y}_B^T\mathbf{a}_j\right)$, for $j \in N$. For $j \in B$ we have $c_j - \mathbf{y}_B^T\mathbf{a}_j = c_j - \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{a}_j = 0$, which is the coefficient of x_j also.

Case 1. If $\left(c_j - \mathbf{y}_B^T\mathbf{a}_j\right) \geq 0$ for all $j \in N$, then we cannot further reduce the objective value, because $\mathbf{x}_N \geq 0$. The optimal value is thus $\mathbf{y}_B^T\mathbf{b}$, which is attained by vertex $\mathbf{v}$.

Case 2. If for some $j \in N$, $\left(c_j - \mathbf{y}_B^T\mathbf{a}_j\right) < 0$, then we want to increase v_j to get a better objective value. When having a few choices, which j should be picked? There are several strategies that work. For reasons that will become clear later, we use the so-called *Bland's pivoting rule* and pick the least candidate j.

Having chosen j, the next step is to decide how much we can increase v_j to. (Think of the variable x_3 at the beginning of Example 3.1.) We have to know the coefficient of x_j in each of the equations of the system $\mathbf{x}_B + \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N = \mathbf{A}_B^{-1}\mathbf{b}$. The system has m equations, each of which corresponds to a basic variable x_i, $i \in B$. For each $i \in B$, the corresponding equation is

$$x_i + \sum_{j \in N} \left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i x_j = (\mathbf{A}_B^{-1}\mathbf{b})_i.$$

Consequently, when $\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i \leq 0$, increasing v_j does not affect the non-negativity of v_i. On the other hand, if $\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i > 0$, then v_j can only be increased to as much as $\frac{(\mathbf{A}_B^{-1}\mathbf{b})_i}{\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i}$.

Case 2a If $\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i \leq 0$ for all $i \in B$, then the LP is unbounded, because we can increase v_j to be as large as we want, while keeping $\mathbf{v}$ feasible. If this is the case, the simplex algorithm stops and reports **unbounded**.

Case 2b If there is some $i \in B$ such that $\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i > 0$, then the new value of v_j can only be as large as

$$v_j = \min\left\{ \frac{(\mathbf{A}_B^{-1}\mathbf{b})_i}{\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i} : \left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i > 0 \right\} = \frac{(\mathbf{A}_B^{-1}\mathbf{b})_k}{\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_k}.$$

Here, again using Bland's rule, we choose k to be the least index which minimizes the fraction.

Knowing such a k, x_k now leaves the basis and x_j enters the basis: $B = B \cup \{j\} - \{k\}$, $N = N \cup \{k\} - \{j\}$. We have a new vertex and can go to the next iteration.

3.4 Termination and running time

You may be having a few doubts:

1. How do we know that the algorithm terminates? (Either indicating unboundedness or stop with an optimal vertex.) Can it loop forever?

2. If the algorithm terminates, how long does it take?

It turns out that without a specific rule of picking the entering and leaving variables, the algorithm might loop forever. Since we are moving from vertex to vertex of P, and there are only finitely many vertices ($\leq \binom{n}{m}$), if the algorithm does not terminate than it must cycle back to a vertex we have visited before. See [4, 11] for examples of LPs where the method cycles. There are quite a few methods to prevent cycling: the *perturbation method* [10], *lexicographic rule* [13], and *smallest subscript rule* or *Bland's pivoting rule* [5], etc.

The *smallest subscript rule*, or *Bland's pivoting rule*, simply says that we should pick the smallest candidate j to leave the basis, and then smallest candidate i to enter the basis. That was the rule we chose to present the simplex iteration in the previous section.

If each iteration increases the objective function positively, then there cannot be cycling. Thus, we can only cycle around a set of vertices with the same cost. This only happens when v_j cannot be increased at all, which means that the leaving candidates i all satisfy the conditions that $\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i > 0$ and $\mathbf{A}_B^{-1}\mathbf{b})_i = 0$. This is the case when the basic variable v_i is also 0: we have what called a *degenerate* case. What happens is that the current vertex is at the intersection of more than n hyperplanes.

Theorem 3.2. *Under the Bland's pivoting rule, cycling does not happen.*

Proof. Note that, for any basis B during the execution of the simplex algorithm, we have

$$c_B - \mathbf{y}_B^T\mathbf{A}_B = c_B - \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{A}_B = 0.$$

We thus have our first observation:

(i) For any $j \in B$, $c_j - \mathbf{y}_B^T\mathbf{a}_j = 0$, where B is any basis.

Suppose cycling happens. During cycling, an index j is "fickle" if $\mathbf{a}_j$ enters some basis at some point, and thus leaves some other basis at some other point.

Let p be the largest fickle index, where $\mathbf{a}_p$ leaves some basis B and enters another basis B' during cycling. Suppose $\mathbf{a}_q$ enters B in place of $\mathbf{a}_p$. Thus, $q < p$ because q is also fickle. We make three basic observations:

(ii) Because q enters B, q is the least index among $1, \dots, n$ for which $c_q - \mathbf{y}_B^T\mathbf{a}_q < 0$.

(iii) Because p leaves B, p is the least index in B satisfying $\left(\mathbf{A}_B^{-1}\mathbf{a}_q\right)_p > 0$ and $\left(\mathbf{A}_B^{-1}\mathbf{b}\right)_p = 0$.

(iv) Because p enters B', p is the least index among $1, \dots, n$ satisfying $c_p - \mathbf{y}_{B'}^T\mathbf{a}_p < 0$.

(v) Since $q < p$, we have $c_q - \mathbf{y}_{B'}^T\mathbf{a}_q \geq 0$.

From (ii) and (v) we get

$$\begin{array}{rcl} 0 & < & (c_q - \mathbf{y}_{B'}^T\mathbf{a}_q) - (c_q - \mathbf{y}_B^T\mathbf{a}_q) \\ & = & \mathbf{y}_B^T\mathbf{a}_q - \mathbf{y}_{B'}^T\mathbf{a}_q \\ & = & \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{a}_q - \mathbf{y}_{B'}^T\mathbf{A}_B\mathbf{A}_B^{-1}\mathbf{a}_q \\ & = & \left(\mathbf{c}_B^T - \mathbf{y}_{B'}^T\mathbf{A}_B\right)\left(\mathbf{A}_B^{-1}\mathbf{a}_q\right) \\ & = & \displaystyle\sum_{r\in B}\left(\mathbf{c}_r^T - \mathbf{y}_{B'}^T\mathbf{a}_r\right)\left(\mathbf{A}_B^{-1}\mathbf{a}_q\right)_r \end{array}$$

Thus, there is some index $r \in B$ where

$$\left(\mathbf{c}_r^T - \mathbf{y}_{B'}^T\mathbf{a}_r\right)\left(\mathbf{A}_B^{-1}\mathbf{a}_q\right)_r > 0. \tag{2}$$

Consider three cases, all of which leads to contradiction.

- If $r > p$, then r is not fickle, and thus $r \in B'$ also. This implies $c_r - \mathbf{y}_{B'}^T\mathbf{a}_r = 0$ due to (i).
- If $r = p$, then $c_r - \mathbf{y}_{B'}^T\mathbf{a}_r < 0$ because of (iv) and $\left(\mathbf{A}_B^{-1}\mathbf{a}_q\right)_r > 0$ because of (iii).

- If $r < p$, then $c_r - \mathbf{y}_{B'}^T \mathbf{a}_r \geq 0$ because of (iv), and thus $c_r - \mathbf{y}_{B'}^T \mathbf{a}_r > 0$ due to inequality (2). Hence, $r \notin B'$ because of (i). This means r is also fickle. Thus, $\left(\mathbf{A}_B^{-1}\mathbf{b}\right)_r = 0$ because $\left(\mathbf{A}_B^{-1}\mathbf{b}\right)_r$ is exactly the value of the coordinate v_r of a vertex during cycling, which does not change its value. But then, this means that $\left(\mathbf{A}_B^{-1}\mathbf{a}_q\right)_r \leq 0$ because of (iii).

□

It was an important longstanding open problem concerning the running time of the simplex method. In 1972, Klee and Minty [21] constructed an example in which the simplex method goes through all vertices of a polyhedron, showing that it is an exponential algorithm, under the assumption that we use the largest coefficient rule.

Exercise 21 (Klee-Minty). Consider the following linear program.

$$
\begin{array}{lrcl}
\min & \sum_{j=1}^m -10^{m-j} x_j & & \\
\text{subject to} & \left(2\sum_{j=1}^{i-1} 10^{i-j} x_j\right) + x_i + z_i & = & 100^{i-1},\ i = 1, \dots, m, \\
& \mathbf{x} \geq \mathbf{0},\ \mathbf{z} \geq \mathbf{0} & &
\end{array}
$$

Show that, the simplex method using the largest coefficient rule performs $(2^m - 1)$ iterations before terminating.

We can also pick the x_j which increases the objective function the most, i.e. applying the *largest increase* rule. The largest increase rule does not fare any better as Jeroslow (1973, [16]) found a similar exponential example. Since the largest coefficient rule takes less work, it is often preferred.

In practice, however, the simplex method works rather well for many practical problems. To explain this phenomenon, researchers have tried to show that, under some certain probabilistic distributions of linear programs, the simplex method takes a polynomial number of iterations on average. See, for example, Borgwardt [6–9], Smale [27, 28], Spielman and Teng [29–33].

3.5 The revised simplex method

The simplex method with a certain computation optimization is called the *revised simplex method,* as briefly described below.

In a typical iteration of the method described in the previous section, we have to compute the following vectors:

- $\mathbf{d}_N = \mathbf{c}_N - \mathbf{c}_B^T \mathbf{A}_B^{-1} \mathbf{A}_N$: this is the coefficient vector of $\mathbf{x}_N$
- $\mathbf{f} = \mathbf{A}_B^{-1} \mathbf{a}_j$ (after j is chosen): this is the coefficient (column) vector of x_j in the system
- $\mathbf{g} = \mathbf{A}_B^{-1} \mathbf{b}$: this is the vector on the right hand side.

If we know $\mathbf{A}_B^{-1}$, we can actually get away with re-computing the inverse $\mathbf{A}_B^{-1}$ and the product $\mathbf{A}_B^{-1}\mathbf{A}_N$ at each step by noticing that the difference between the old A_B and the new A_B is only a replacement of one column ($\mathbf{a}_k$) by another ($\mathbf{a}_j$).

Let $B' = B \cup \{j\} - \{k\}$ be the new index set of the basis. Without loss of generality, assume the leaving vector $\mathbf{a}_k$ is the last column in A_B. Noting that $\mathbf{A}_B \mathbf{f} = \mathbf{a}_j$, it is not difficult to see that

$$
A_{B'}^{-1} = \begin{bmatrix} 1 & 0 & \dots & f_1 \\ 0 & 1 & \dots & f_2 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & f_m \end{bmatrix}^{-1} A_B^{-1} = F^{-1} A_B^{-1}.
$$

It is computationally very easy to compute F^{-1}. In practical implementation, we do not have to even compute $\mathbf{A}_B^{-1}$ (which is very much subject to numerical errors). We can write A_B as an LU factorization, then the desired vectors such as $\mathbf{f}, \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{A}_N$, and $\mathbf{g}$ can be computed mostly by "backward substitution." For instance, we can solve the system $\mathbf{A}_B\mathbf{f} = \mathbf{a}_j$ for $\mathbf{f}$, solve $\mathbf{A}_B\mathbf{g} = \mathbf{b}$ for $\mathbf{g}$, and so on.

We will not delve deeper into this. The key idea is that, by storing the old A_B^{-1}, it is easy (and quick) to compute the new A_B^{-1}.

3.6 Summary of the simplex method

In the following summary, we use Bland's pivoting rule.

1. Start from a vertex $\mathbf{v}$ of P.
2. Determine the basic index set B and free index set N. Let $\mathbf{y}_B^T = \mathbf{c}_B^T\mathbf{A}_B^{-1}$.
3. If $\left(c_N^T - \mathbf{y}_B^T\mathbf{a}_j\right) \geq 0$, then the optimal value is $\mathbf{y}_B^T\mathbf{b}$. We have found an optimal vertex $\mathbf{v}$. STOP!
4. Else, let
$$j = \min\left\{j' \in N : \left(c_{j'} - \mathbf{y}_B^T\mathbf{a}_{j'}\right) < 0\right\}.$$
5. If $\mathbf{A}_B^{-1}\mathbf{a}_j \leq 0$, then report UNBOUNDED LP and STOP!
6. Otherwise, pick smallest $k \in B$ such that $\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_k > 0$ and that
$$\frac{(\mathbf{A}_B^{-1}\mathbf{b})_k}{\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_k} = \min\left\{\frac{(\mathbf{A}_B^{-1}\mathbf{b})_i}{\left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i} \;:\; i \in B,\;\; \left(\mathbf{A}_B^{-1}\mathbf{a}_j\right)_i > 0\right\}.$$
7. x_k now leaves the basis and x_j enters the basis: $B = B \cup \{j\} - \{k\}$, $N = N \cup \{k\} - \{j\}$.
   GO BACK to step 3.

We thus have the following fundamental theorem of the simplex method.

Theorem 3.3. *Given a linear program under standard form and a basic feasible solution, the simplex method reports "unbounded" if the LP has no optimal solution. Otherwise, the method returns an optimal solution at a vertex.*

Exercise 22. We discussed the simplex method for the min version of the standard form. Write down the simplex method for the max version, but do not use the fact that $\max \mathbf{c}^T\mathbf{x} = \min(-\mathbf{c})^T\mathbf{x}$. Basically, I want you to reverse some of the $\min$ and $\max$, and inequalities in Section 3.6.

3.7 The two-phase simplex method

So far, we have assumed that we can somehow get a hold of a vertex of the polyhedron. What if the polyhedron is empty? Even when it is not, how do we find a vertex to start the simplex loop? This section answer those questions.

Let $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. By multiplying some equation(s) by -1, we can assume that $\mathbf{b} \geq \mathbf{0}$. As usual, $\mathbf{A} = (a_{ij})$ is an $m \times n$ matrix. Let $\mathbf{A}' = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix}$, then $\mathbf{A}'$ is an $m \times (n+m)$ matrix. Let $P' = \{\mathbf{z} \mid \mathbf{A}'\mathbf{z} = \mathbf{b}, \mathbf{z} \geq \mathbf{0}\}$. (Note that the vectors in P' lie in $\mathbb{R}^{n+m}$.) It is straightforward to see that the linear program $\min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in P\}$ is feasible if and only if the linear program $\min\left\{\sum_{i=1}^m z_{n+i} \mid \mathbf{z} \in P'\right\}$ is feasible with optimal value 0. Moreover, let $\mathbf{z}$ be any vertex of P', and let $\mathbf{x} \in \mathbb{R}^n$ be formed by the first n coordinates of $\mathbf{z}$, then $\mathbf{x}$ is a vertex of P.

It is easy to see that $\mathbf{z} = [0, \ldots, 0, b_1, \ldots, b_m]$ is a vertex of P'. We can start the simplex algorithm from this vertex and find an optimal vertex $\mathbf{z}^*$ of the second linear program, which induces an optimal vertex of the first linear program.

Exercise 23. Solve the following linear program using the Simplex method.

$$\begin{array}{rrcrcrcrcr} \max & 3x_1 & + & x_2 & + & 5x_3 & + & 4x_4 & & \\ \text{subject to} & 3x_1 & - & 3x_2 & + & 2x_3 & + & 8x_4 & \leq & 50 \\ & 4x_1 & + & 6x_2 & - & 4x_3 & - & 4x_4 & \leq & 40 \\ & 4x_1 & - & 2x_2 & + & x_3 & + & 3x_4 & \leq & 20 \\ & & & & & & & x_1, x_2, x_3, x_4 & \geq & 0. \end{array}$$

Exercise 24. Solve the following linear program using the simplex method:

$$\begin{array}{rrcrcrcrcr} \max & 3x_1 & + & 6x_2 & + & 9x_3 & + & 8x_4 & & \\ \text{subject to} & x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \leq & 5 \\ & x_1 & + & x_2 & + & 2x_3 & + & 3x_4 & \leq & 3 \\ & & & & & & & x_1, x_2, x_3, x_4 & \geq & 0. \end{array}$$

Exercise 25. Show that the following linear program is infeasible

$$\begin{array}{rrcrcrcr} \max & x_1 & - & 3x_2 & + & 2x_3 & & \\ \text{subject to} & x_1 & + & 2x_2 & + & 3x_3 & \leq & 5 \\ & 2x_1 & + & 3x_2 & + & 2x_3 & \leq & 4 \\ & \multicolumn{7}{r}{2 \leq x_1 \leq 4, x_2 \leq -1, 3 \leq x_3 \leq 8} \end{array}$$

Exercise 26. Show that the following linear program is feasible but unbounded

$$\begin{array}{rrcrcrcr} \min & x_1 & - & 3x_2 & + & 2x_3 & & \\ \text{subject to} & x_1 & + & 2x_2 & + & x_3 & \leq & 2 \\ & 2x_1 & + & x_2 & + & 4x_3 & \leq & 4 \\ & \multicolumn{7}{r}{0 \leq x_1 \leq 2, x_2 \leq 0, -2 \leq x_3 \leq 2} \end{array}$$

Exercise 27. In this exercise, we devise a way to solve the linear program $\max\{\mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ "directly," i.e. without first converting it to standard form. Recall that $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ is pointed iff $\text{rank}(\mathbf{A}) = n$. More specifically, from Exercise 8, $\mathbf{v} \in P$ is a vertex iff $\text{rank}\left\{\mathbf{a}^{(i)} \mid \mathbf{a}^{(i)}\mathbf{v} = b_i\right\} = n$. Basically, there must be a subsystem $\mathbf{A}_B\mathbf{x} \leq \mathbf{b}$ with n inequalities for which $\mathbf{A}_B$ has full rank and $\mathbf{A}_B\mathbf{v} = \mathbf{b}_B$.

1. Write $\mathbf{A} = \begin{bmatrix} \mathbf{A}_B \\ \mathbf{A}_N \end{bmatrix}$, then our linear program is equivalent to $\max\{\mathbf{c}^T\mathbf{x} \mid \mathbf{A}_B\mathbf{x} \leq \mathbf{b}_B,\ \mathbf{A}_N\mathbf{x} \leq \mathbf{b}_N\}$. Intuitively, if $\mathbf{c}$ is in the cone generated by the row vectors of $\mathbf{A}_B$, then v is optimal. (Going along $\mathbf{c}$ will take us outside of the polyhedron.) Formally, let $\mathbf{u}_B$ be the vector such that $\mathbf{A}_B^T\mathbf{u}_B = \mathbf{c}$. **Prove that,** if $\mathbf{u}_B \geq \mathbf{0}$, then $\mathbf{v}$ is optimal.

2. Next, if $\mathbf{v}$ is not optimal, we try to find a ray $\mathbf{v} + \alpha\mathbf{z}$ ($\alpha \geq 0$) to move along so as to improve the objective value. The ray should be on an edge of the polyhedron. If the ray is entirely in P, then the program is unbounded. Otherwise, we will meet a better vertex and thus can go to the next iteration.

   An edge incident to $\mathbf{v}$ is on $n-1$ of the n hyperplanes $\mathbf{A}_B\mathbf{x} = \mathbf{b}_B$. Hence, $\mathbf{z}$ is the vector perpendicular to $n-1$ of the row vectors of $\mathbf{A}_B$. The vector $\mathbf{a}^{(i)}$ that $\mathbf{z}$ is not perpendicular to should be such that $u_i < 0$. Moreover, $\mathbf{z}$ should point away from $\mathbf{a}^{(i)}$.

   Formally, using Bland's pivoting rule, let i^* be the least index so that $u_{i^*} < 0$. Let $\mathbf{z}$ be the vector such that $\mathbf{a}^{(i)}\mathbf{z} = 0$ for all $i \in B - \{i^*\}$, and that $\mathbf{a}^{(i^*)}\mathbf{z} = -1$. Then, $\mathbf{v} + \alpha\mathbf{z}$ ($\alpha \geq 0$) traverses an edge of P. **Show that** there is uniquely one such vector $\mathbf{z}$.

3. Suppose $\mathbf{a}^{(i)}\mathbf{z} \leq 0, \forall i \in [m]$. **Show that** the linear program is unbounded.

4. Otherwise, let $\overline{\alpha}$ be the largest α such that $\mathbf{v} + \alpha\mathbf{z}$ is still in P, namely

$$\overline{\alpha} = \min_{i \in [n]} \left\{ \frac{b_i - \mathbf{a}^{(i)}\mathbf{v}}{\mathbf{a}^{(i)}\mathbf{z}} \mid \mathbf{z}^T\mathbf{a}^{(i)} > 0 \right\}.$$

   Let k^* be the least index attaining this minimum.

   Replace $\mathbf{v}$ by $\mathbf{v} + \overline{\alpha}\mathbf{z}$. **Show that** the new $\mathbf{v}$ is still a vertex of P.

   Replace B by $B \cup \{k^*\} - \{i^*\}$. Go back to step 1.

Finally, **show that** the above algorithm terminates. (*Hint*: suppose the algorithm does not terminate. During cycling, suppose h is the highest index for which h as been removed from some basis B, and thus it is added during cycling to some basis B^*. Show that $\mathbf{u}_B\mathbf{A}_B\mathbf{z}_{B^*} > 0$, which implies that there is some $i \in B$ for which $(u_B)_i(\mathbf{a}^{(i)}\mathbf{z}_{B^*}) > 0$. Derive a contradiction.)

Jumping ahead a little bit, we have the following exercises.

Exercise 28. State and prove a strong duality theorem from the above algorithm where $\max\{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ is the primal program.

Exercise 29. Prove a variance of Farkas' lemma from the above algorithm.

Exercise 30. Describe and prove necessary results for a 2-phase simplex method based on the above algorithm.

4 Feasibility and the fundamental theorem of linear inequalities

Definition 4.1 (Cones). A set C of points in a Euclidean space is called a (convex) *cone* if it is closed under non-negative linear combinations, namely $\alpha\mathbf{x} + \beta\mathbf{y} \in C$ whenever $\mathbf{x}, \mathbf{y} \in C$, and $\alpha, \beta \geq 0$.

Definition 4.2 (Finitely generated cones). Given vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$ in some Euclidean space, the set

$$\text{cone}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\} := \{\alpha_1\mathbf{a}_1 + \cdots + \alpha_n\mathbf{a}_n \mid \alpha_j \geq 0, \forall j \in [n]\}$$

is obviously a cone, and is called the cone generated by the vectors $\mathbf{a}_j$. A cone generated this way is said to be *finitely generated*.

We give two proofs of the following "separation theorem."

Theorem 4.3 (Fundamental theorem of linear inequalities). *Let $\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}, \mathbf{b}$ be vectors in $\mathbb{R}^m$. Then, exactly one of the following statements holds:*

(1) $\mathbf{b}$ *is in the cone generated by some linearly independent vectors from* $\mathbf{a_1}, \ldots, \mathbf{a_n}$.

(2) *there exists a hyperplane* $\{\mathbf{x} \mid \mathbf{c^T x = 0}\}$ *containing* $r - 1$ *independent vectors from* $\mathbf{a_1}, \ldots, \mathbf{a_n}$, *such that* $\mathbf{c^T b < 0}$, *and* $\mathbf{c^T a_j \geq 0}, \forall j \in [n]$, *where* $r = \text{rank}\{\mathbf{a_1}, \ldots, \mathbf{a_n}, \mathbf{b}\}$.

Direct proof. We first show that the two statements are mutually exclusive. Suppose $b = \sum \alpha_j\mathbf{a_j}$, with $\alpha_j \geq 0$, then $\mathbf{c^T b} = \sum \alpha_j \mathbf{c^T a_j} \geq 0$ whenever $\mathbf{c^T a_j} \geq \mathbf{0}, \forall j$. Thus (1) and (2) are mutually exclusive.

To show that one of them must hold, we shall describe a procedure which will either produce a non-negative combination as in (1), or a vector $\mathbf{c}$ satisfying (2).

Note that if $\mathbf{b}$ is not in the span of the $\mathbf{a_j}$, then there is a hyperplane $\{\mathbf{x} \mid \mathbf{c^T x = 0}\}$ which contains all the $\mathbf{a_j}$ but does not contain $\mathbf{b}$. That plane serves our purpose. (Such vector $\mathbf{c}$ lies in the null space of span$\{\{\mathbf{a_1}, \ldots, \mathbf{a_n}\}\}$ but not in the null space of span$\{\{\mathbf{a_1}, \ldots, \mathbf{a_n}, \mathbf{b}\}\}$.) Hence, we can assume that $r = \text{rank}\{\mathbf{a_1}, \ldots, \mathbf{a_n}\}$. In fact, we can also assume $r = m$, because if $r < m$, then we can add into $\{\mathbf{a_1}, \ldots, \mathbf{a_n}\}$ a few vectors to make the rank equal m.

Now, consider the following procedure:

(0) Choose m linearly independent vectors $B = \{\mathbf{a_{j_1}}, \ldots, \mathbf{a_{j_m}}\}$

1. Write $\mathbf{b} = \sum_i \alpha_{j_i} \mathbf{a_{j_i}}$. If $\alpha_{j_i} \geq 0, \forall i \in [m]$, then (1) holds. STOP.
2. Otherwise, chose the smallest $p \in \{j_1, \ldots, j_m\}$ so that $\alpha_p < 0$. Let $\{\mathbf{x} \mid \mathbf{c^T x = 0}\}$ be the hyperplane spanned by $m-1$ vectors $B \setminus \{\mathbf{a_p}\}$, where we normalize $\mathbf{c}$ such that $\mathbf{c^T a_p = 1}$. It is easy to see that such a $\mathbf{c}$ uniquely exists and that $\mathbf{c}^T \mathbf{b} < 0$.
3. If $\mathbf{c^T a_1}, \ldots, \mathbf{c^T a_n} \geq \mathbf{0}$, then (2) holds. STOP.
4. Otherwise, choose the smallest q such that $\mathbf{c^T a_q} < \mathbf{0}$. Replace B by $B \cup \{\mathbf{a_q}\} - \{\mathbf{a_p}\}$, and go back to step 1.

We shall show that the procedure must stop. Note that $\mathbf{a}_q$ is independent of the vectors $B - \{\mathbf{a}_p\}$, since otherwise $\mathbf{c}^T\mathbf{a}_q = 0$, a contradiction. Thus, when replacing B by $B \cup \{\mathbf{a}_q\} - \{\mathbf{a}_p\}$ and go back to step 1 we still have a set of independent vectors.

To this end, let B_0 denote the original B, and B_i the set B after the ith iteration. Consider any B_k. If the procedure does not terminate, then there must be a smallest $l > k$ such that $B_l = B_k$, because the number of different B's is at most $\binom{n}{m}$.

Consider the highest index h such that $\mathbf{a}_h$ has been removed from B at the end of one of the iterations $k, k+1, \ldots, l-1$. Whether or not $\mathbf{a}_h$ was in B_k, there must be some iterations s and t, $k \leq s, t < l$, in which $\mathbf{a}_h$ was removed from B_s and $\mathbf{a}_h$ was added into B_t. It is easy to see that

$$B_s \cap \{\mathbf{a}_j \mid j > h\} = B_t \cap \{\mathbf{a}_j \mid j > h\} = B_k \cap \{\mathbf{a}_j \mid j > h\}.$$

Without loss of generality, assume $B_s = \{\mathbf{a}_{j_1}, \ldots, a_{j_m}\}$. Write $\mathbf{b} = \sum_{i=1}^m \alpha_{j_i} \mathbf{a}_{j_i}$. Let $\bar{\mathbf{c}}$ be the vector $\mathbf{c}$ at iteration t. Then,

$$\bar{\mathbf{c}}^T \mathbf{b} < 0,$$

as we have shown. However,

$$\bar{\mathbf{c}}^T \mathbf{b} = \sum_{i=1}^m \alpha_{j_i} \bar{\mathbf{c}}^T \mathbf{a}_{j_i} > 0,$$

because

- when $j_i < h$, we have $\alpha_{j_i} \geq 0$ because h was the least index for which $\alpha_h < 0$ so that $\mathbf{a}_h$ is to be removed from B_s, and $\bar{\mathbf{c}}^T \mathbf{a}_{j_i} \geq \mathbf{0}$ because $\bar{\mathbf{c}}$ is the vector $\mathbf{c}$ at the point we added $\mathbf{a}_h$ into B, and at that point h was the least index such that $\bar{\mathbf{c}}^T \mathbf{a}_{j_i} < \mathbf{0}$.
- when $j_i = h$, $\alpha_{j_i} < 0$ and $\bar{\mathbf{c}}^T \mathbf{a}_{j_i} < 0$.
- when $j_i > h$, $\bar{\mathbf{c}}^T \mathbf{a}_{j_i} = \mathbf{0}$ because of step 2.

We got a contradiction! $\square$

The fundamental theorem basically says that either $\mathbf{b}$ is in the cone generated by the $\mathbf{a}_j$, or it can be *separated* from the $\mathbf{a}_j$ by a hyperplane containing $r-1$ independent $\mathbf{a}_j$. The following result states the same fact but it is less specific.

Lemma 4.4 (Farkas' lemma). *The system* $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ *is feasible iff the system* $\mathbf{A^T y} \geq \mathbf{0}, \mathbf{b^T y} < \mathbf{0}$ *is infeasible.*

Constructive proof from the simplex algorithm. If $\mathbf{A^T y} \geq \mathbf{0}, \mathbf{b^T y} < \mathbf{0}$ is feasible, it is easy to see that $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ is infeasible. We will show the converse: assuming $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ is infeasible, we want to find a vector $\mathbf{y}$ such that $\mathbf{A^T y} \geq \mathbf{0}, \mathbf{b^T y} < \mathbf{0}$.

Let $\mathbf{A}' = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix}$, then $\mathbf{A}'$ is an $m \times (n+m)$ matrix. Let $P' = \{\mathbf{z} \mid \mathbf{A}'\mathbf{z} = \mathbf{b}, \mathbf{z} \geq \mathbf{0}\}$. Recall the two-phase simplex method, where we noted that $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ is infeasible if and only if the linear program

$$\min \left\{ \mathbf{d}^T\mathbf{z} = \sum_{i=1}^{m} z_{n+i} \mid \mathbf{z} \in P' \right\}$$

is feasible with optimal value > 0. Let $\mathbf{z}^*$ be an optimal vertex of P' returned by the simplex method. Let $\mathbf{A}'_B$ be the corresponding basis, which consists of some columns from $\mathbf{A}$ and some columns from $\mathbf{I}$. When the simplex method returns $\mathbf{z}^*$, two conditions hold

$$\begin{aligned} \mathbf{d}^T\mathbf{z}^* = \mathbf{y}_B^T\mathbf{b} &> 0 \\ \mathbf{d}_N - \mathbf{y}_B^T\mathbf{A}'_N &\geq \mathbf{0}, \end{aligned}$$

where $\mathbf{y}_B^T = \mathbf{d}_B^T\mathbf{A}'^{-1}_B$. It is easy to see that the vector $-\mathbf{y}_B^T$ serves our purpose. $\square$

Proof from the fundamental theorem of linear inequalities. Geometrically, this is saying that if $\mathbf{b}$ is in the cone generated by the column vectors of $\mathbf{A}$ iff there is no hyperplane separating $\mathbf{b}$ from the column vectors of $\mathbf{A}$. It should be no surprise that we can derive Farkas' lemma and its variations from the fundamental theorem. Below is a sample proof.

Necessity is obvious. For sufficiency, assume the first system is infeasible, i.e. $\mathbf{b}$ is not in the cone generated by the column vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$ of $\mathbf{A}$. By the fundamental theorem, there is a vector $\mathbf{c}$ such that $\mathbf{c}^T\mathbf{a}_j \geq 0, \forall j$, and $\mathbf{c}^T\mathbf{b} < 0$. Obviously, $\mathbf{y} = \mathbf{c}$ is a solution to the second system. $\square$

Exercise 31 (Farkas' lemma (variation)). The system $\mathbf{Ax} \leq \mathbf{b}$ is infeasible iff the system

$$\mathbf{A^T y} = \mathbf{0}, \mathbf{b^T y} < \mathbf{0}, \mathbf{y} \geq \mathbf{0}$$

is feasible.

Exercise 32 (Gordan, 1873). Show that the system $\mathbf{Ax} < \mathbf{0}$ is unsolvable iff the system

$$\mathbf{A^T y} = \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \mathbf{y} \neq \mathbf{0}$$

is solvable.

Exercise 33 (Stiemke, 1915). Show that the system $\mathbf{Ax} = \mathbf{0}, \mathbf{x} > \mathbf{0}$ is unsolvable iff the system

$$\mathbf{A^T y} \geq \mathbf{0}, \mathbf{A^T y} \neq \mathbf{0}$$

is solvable.

Exercise 34 (Ville, 1938). Show that the system $\mathbf{Ax} < \mathbf{0}, \mathbf{x} \geq \mathbf{0}$ is unsolvable iff the system

$$\mathbf{A^T y} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \mathbf{y} \neq \mathbf{0}$$

is solvable.

Farkas' lemma deals with non-strict inequalities. There is a even more general result dealing with non-strict and strict inequalities, due to Fourier (1826, [15]), Kuhn (1956, [23]), and Motzkin (1936, [25]).

Theorem 4.5 (Motzkin's transposition theorem). *The system*

$$\mathbf{Ax} < \mathbf{b}, \ \ \mathbf{Bx} \le \mathbf{c}$$

is feasible if and only if

$$\mathbf{y}, \mathbf{z} \ge \mathbf{0}, \ \ \mathbf{A^T y} + \mathbf{B^T z} = \mathbf{0}, \ \ \Rightarrow \ \ \mathbf{b^T y} + \mathbf{c^T z} \ge \mathbf{0}, \tag{3}$$

and

$$\mathbf{y}, \mathbf{z} \ge \mathbf{0}, \ \ \mathbf{A^T y} + \mathbf{B^T z} = \mathbf{0}, \ \ \mathbf{y} \ne \mathbf{0}, \ \ \Rightarrow \ \ \mathbf{b^T y} + \mathbf{c^T z} > \mathbf{0}. \tag{4}$$

Proof. Note that (3) is equivalent to the fact that

$$\mathbf{y}, \mathbf{z} \ge \mathbf{0}, \ \ \begin{bmatrix} \mathbf{A^T} & \mathbf{B^T} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \mathbf{0}, \ \ \begin{bmatrix} \mathbf{b^T} & \mathbf{c^T} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} < \mathbf{0}$$

is infeasible, and (4) is equivalent to the fact that

$$\mathbf{y}, \mathbf{z} \ge \mathbf{0}, \ \ \begin{bmatrix} \mathbf{A^T} & \mathbf{B^T} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \mathbf{0}, \ \ \mathbf{y} \ne \mathbf{0}, \ \ \begin{bmatrix} \mathbf{b^T} & \mathbf{c^T} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \le \mathbf{0} \tag{5}$$

is infeasible.

For necessity, suppose there is some $\mathbf{x}$ such that $\mathbf{Ax} < \mathbf{b}$, and $\mathbf{Bx} \le \mathbf{c}$. When $\mathbf{A^T y} + \mathbf{B^T z} = \mathbf{0}$, we have $\mathbf{0} = \mathbf{x^T A^T y} + \mathbf{x^T B^T z} \le \mathbf{b^T y} + \mathbf{c^T z}$, (3) is proved. When $\mathbf{y} \ne \mathbf{0}$, we have strict inequality and (4) is shown.

For sufficiency, (3) and Exercise 31 imply that there is an $\mathbf{x}$ with $\mathbf{Ax} \le \mathbf{b}$ and $\mathbf{Bx} \le \mathbf{c}$. Let $\mathbf{a_1}, \dots, \mathbf{a_m}$ be the row vectors of $\mathbf{A}$. Condition (5) implies that, for each $i \in [m]$, the system

$$\mathbf{y}, \mathbf{z} \ge \mathbf{0}, \ \ \begin{bmatrix} \mathbf{A^T} & \mathbf{B^T} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = -\mathbf{a_i^T}, \ \ \begin{bmatrix} \mathbf{b^T} & \mathbf{c^T} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \le -\mathbf{b_i} \tag{6}$$

is infeasible. Or, the system

$$\mathbf{y}, \mathbf{z}, \bar{\mathbf{w}} \ge \mathbf{0}, \ \ \begin{bmatrix} \mathbf{A^T} & \mathbf{B^T} & \mathbf{0} \\ \mathbf{b^T} & \mathbf{c^T} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \\ \bar{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} -\mathbf{a_i^T} \\ -\mathbf{b_i} \end{bmatrix} \tag{7}$$

is infeasible. By Farkas' lemma, this is equivalent to the fact that the system

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{B} & \mathbf{c} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix} \le \mathbf{0}, \ \ \begin{bmatrix} -\mathbf{a_i} & -\mathbf{b_i} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix} > \mathbf{0} \tag{8}$$

is feasible. Since $\mathbf{Ax} \le \mathbf{b}, \mathbf{Bx} \le \mathbf{c}, \mathbf{a_i x} \le \mathbf{b_i}$, we have

$$\begin{aligned} \mathbf{A}(\mathbf{x} + \mathbf{v}) &\le (-\bar{\mathbf{v}} + \mathbf{1})\mathbf{b} \\ \mathbf{B}(\mathbf{x} + \mathbf{v}) &\le (-\bar{\mathbf{v}} + \mathbf{1})\mathbf{c} \\ \mathbf{a_i}(\mathbf{x} + \mathbf{v}) &< (-\bar{\mathbf{v}} + \mathbf{1})\mathbf{b_i} \\ -\bar{\mathbf{v}} + \mathbf{1} &\ge 1. \end{aligned}$$

Let $\mathbf{x^{(i)}} = (\mathbf{x} + \mathbf{v})/(\mathbf{1} - \bar{\mathbf{v}})$, then we have $\mathbf{Ax^{(i)}} \le \mathbf{b}, \mathbf{Bx^{(i)}} \le \mathbf{c}, \mathbf{a_i x^{(i)}} < \mathbf{b_i}$. The barycenter of the $\mathbf{x^{(i)}}$ is an $\mathbf{x}$ we are looking for. $\square$

Corollary 4.6 (Gordan, 1873). $\mathbf{Ax} < \mathbf{0}$ *is infeasible iff* $\mathbf{A^T y} = \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \mathbf{y} \neq \mathbf{0}$ *is feasible.*

Corollary 4.7 (Stiemke, 1915). $\mathbf{Ax} = \mathbf{0}, \mathbf{x} > \mathbf{0}$ *is infeasible iff* $\mathbf{A^T y} \geq \mathbf{0}, \mathbf{A^T y} \neq \mathbf{0}$ *is feasible.*

Corollary 4.8 (Ville, 1938). $\mathbf{Ax} < \mathbf{0}, \mathbf{x} \geq \mathbf{0}$ *is infeasible iff* $\mathbf{A^T y} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \mathbf{y} \neq \mathbf{0}$ *is feasible.*

Corollary 4.9 (Carver, 1921). $\mathbf{Ax} < \mathbf{b}$ *is feasible iff* $\mathbf{y} \neq \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \mathbf{A^T y} = \mathbf{0}, \mathbf{b^T y} \leq$ *is infeasible.*

Exercise 35. In this exercise, we devise a method to either find a solution to the system $\mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$ ($\mathbf{A}$ is an $m \times n$ matrix of rank m), or gives proof that the system in infeasible. The method consists of the following steps:

1. Start with any set of m linearly independent columns $\mathbf{A}_B$ of $\mathbf{A}$. Rewrite the system as
$$\mathbf{x}_B + \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N = \mathbf{A}_B^{-1}\mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}.$$
2. If $\mathbf{A}_B^{-1}\mathbf{b} \geq 0$, then the system is feasible with $\mathbf{x}_B = \mathbf{A}_B^{-1}\mathbf{b}$, and $\mathbf{x}_N = 0$.
Report FEASIBLE and STOP.
3. Else, let $p = \min\{i \mid i \in B, (\mathbf{A}_B^{-1}\mathbf{b})_i < 0\}$.
For each $i \in B$, let $\mathbf{r}^{(i)}$ be the ith row vector of the $m \times (n-m)$ matrix $\mathbf{A}_B^{-1}\mathbf{A}_N$.
Consider the equation corresponding to $\mathbf{x}_p$:
$$x_p + \mathbf{r}^{(p)}\mathbf{x}_N = (\mathbf{A}_B^{-1}\mathbf{b})_p.$$
4. If $\mathbf{r}^{(p)} \geq 0$, then the system is infeasible. Report INFEASIBLE and STOP.
5. Else, let $q = \min\{j \mid j \in N,\ r_j^{(p)} < 0\}$, let $B = B \cup \{q\} - \{p\}$, and go back to step 1.

Questions:

(a) Show that the procedure terminates after a finite number of steps.

(b) Show that the procedure reports feasible/infeasible iff the system is feasible/infeasible

(c) Prove Farkas' lemma from this procedure. Specifically, show that the system $\mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$ is feasible iff the system $\mathbf{A}^T\mathbf{y} \geq 0, \mathbf{b}^T\mathbf{y} < 0$ is infeasible.

Exercise 36. Consider the system $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$, where $\mathbf{A}$ is an $m \times n$ matrix, and $\text{rank}(\mathbf{A}) = m$. We shall try devise a procedure to test if the system is feasible, slightly different than what we have seen so far. For any $j = 1, \ldots, n$, let $\mathbf{a}_j$ denote the jth column vector of $\mathbf{A}$.

(0) $B = \{j_1, \ldots, j_m\}$ such that $\{\mathbf{a}_j \mid j \in B\}$ form a basis for $\mathbb{R}^m$.

1. Write $\mathbf{b} = \sum_{j \in B} \alpha_j \mathbf{a}_j$. This is unique.
2. If $\alpha_j \geq 0, \forall j \in B$, then STOP. We have found a solution: $\mathbf{x}_j = \alpha_j, \forall j \in B$, $\mathbf{x}_j = 0, \forall j \notin B$.
3. Otherwise, pick the smallest $p \in B$ such that $\alpha_p < 0$. We want to find a $q \in [n] - B$ such that after replacing $\mathbf{a}_p$ by $\mathbf{a}_q$, we get $\alpha_q \geq 0$. (The new B has to also form a basis.) Consider any $h \in [n] - B$. What is the coefficient of $\mathbf{a}_h$ when expressing $\mathbf{b}$ as a linear combination of vectors in $\mathbf{A}_B \cup \{\mathbf{a}_h\} - \{\mathbf{a}_p\}$? How do we know if this is even a basis? Express $\mathbf{a}_h = \sum_{j \in B} \beta_j \mathbf{a}_j$, then $\mathbf{A}_B \cup \{\mathbf{a}_h\} - \{\mathbf{a}_p\}$ is a basis iff $\beta_p \neq 0$. Moreover,
$$\mathbf{a}_p = \sum_{j \in B, j \neq p} (-\beta_j/\beta_p)\mathbf{a}_j + (1/\beta_p)\mathbf{a}_h.$$

Thus, the coefficient of $\mathbf{a}_h$ when expressing $\mathbf{b}$ as a linear combination of $\mathbf{A}_B \cup \{\mathbf{a}_h\} - \{\mathbf{a}_p\}$ is α_p/β_p. We want this to be positive. If there are many such h, we pick the smallest indexed one. If there are none, we should have a certificate for the system being infeasible. The infeasibility is quite easy to see, since if $\mathbf{b} = \sum_{j\in[n]} x_j \mathbf{a}_j$, $x_j \geq 0, \forall j$, and all the $\beta_p^{(h)}$ are none negative, then $\alpha_p \geq 0$.

In conclusion, if there is no such h, then the system is infeasible.

4. Otherwise, pick a smallest q for which $\beta_q < 0$ and exchange p and q. Then, go back to step one.

Questions:

(i) Prove that this procedure will terminate.

(ii) If the system terminates in step 3, find a vector y such that $\mathbf{A^T y} \leq \mathbf{0}, \mathbf{b^T y} > \mathbf{0}$ (Farkas' lemma).

5 Duality

5.1 The basics

Let us consider the following LP:

$$\begin{array}{rl} \min & x_1 - 2x_2 + 4x_3 \\ \text{subject to} & \begin{array}{ccccccc} x_1 & - & 3x_2 & & & = & 3 \\ -2x_1 & + & x_2 & + & 2x_3 & = & 4 \\ x_1 & & & + & x_3 & = & -3 \end{array} \\ & \qquad\qquad x_1, x_2, x_3 \geq 0 \ . \end{array}$$

Adding the first two equalities and two times the third we get

$$(x_1 - 3x_2) + (-2x_1 + x_2 + 2x_3) + 2(x_1 + x_3) = 3 + 4 - 2 \cdot 3,$$

or

$$x_1 - 2x_2 + 4x_3 = 1.$$

This is exactly the objective function. Hence, any feasible solution would also be an optimal solution, and the optimal objective value is 1.

Although in general we will not be that lucky, we could and should try to find a lower bound for the objective function. Basically, when trying to minimize something, we would like to know how much we could minimize it to. If no lower bound exists for a minimization problem, then the LP is *infeasible*. Consider the following LP:

$$\begin{array}{rl} \min & 3x_1 - 2x_2 + 4x_3 + x_4 \\ \text{subject to} & \begin{array}{ccccccccc} x_1 & - & 3x_2 & & & + & 2x_4 & = & 3 \\ -2x_1 & + & x_2 & + & 2x_3 & & & = & 4 \\ -2x_1 & + & x_2 & + & 2x_3 & - & x_4 & = & -2 \end{array} \\ & \qquad\qquad x_1, x_2, x_3, x_4 \geq 0. \end{array} \tag{9}$$

Suppose we multiply the ith equality by a number y_i, then add them all up we get

$$y_1(x_1 - 3x_2 + 2x_4) + y_2(-2x_1 + x_2 + 2x_3) + y_3(-2x_1 + x_2 + 2x_3 - x_4) = 3y_1 + 4y_2 - 2y_3.$$

Equivalently,

$$(y_1 - 2y_2 - 2y_3)\mathbf{x_1} + (-3y_1 + y_2 + y_3)\mathbf{x_2} + (2y_2 + 2y_3)\mathbf{x_3} + (2y_1 - y_3)\mathbf{x_4} = 3y_1 + 4y_2 - 2y_3.$$

Maximization problem	**Minimization problem**
Constraints	Variables
ith constraint $\leq$	ith variable ≥ 0
ith constraint $\geq$	ith variable ≤ 0
ith constraint $=$	ith variable unrestricted
Variables	Constraints
jth variable ≥ 0	jth constraint $\geq$
jth variable ≤ 0	jth constraint $\leq$
jth variable unrestricted	jth constraint $=$

Table 1: Rules for converting between primals and duals.

So, if

$$
\begin{aligned}
y_1 - 2y_2 - 2y_3 &\leq 3 \\
-3y_1 + y_2 + y_3 &\leq -2 \\
2y_2 + 2y_3 &\leq 4 \\
2y_1 - y_3 &\leq 1,
\end{aligned} \tag{10}
$$

then

$$
\begin{aligned}
& 3\mathbf{x_1} - 2\mathbf{x_2} + 4\mathbf{x_3} + \mathbf{x_4} \\
\geq\ & (y_1 - 2y_2 - 2y_3)\mathbf{x_1} + (-3y_1 + y_2 + y_3)\mathbf{x_2} + (2y_2 + 2y_3)\mathbf{x_3} + (2y_1 - y_3)\mathbf{x_4} \\
=\ & 3y_1 + 4y_2 - 2y_3.
\end{aligned}
$$

Consequently, for every triple (y_1, y_2, y_3) satisfying (10), we have a lower bound $3y_1 + 4y_2 - 2y_3$ for the objective function. Since we would like the lower bound to be as large as possible, finding a good triple is equivalent to solving the following LP:

$$
\begin{array}{rrcrcrcr}
\max & 3y_1 + 4y_2 - 2y_3 & & & & & & \\
\text{subject to} & y_1 & - & 2y_2 & - & 2y_3 & \leq & 3 \\
 & -3y_1 & + & y_2 & + & y_3 & \leq & -2 \\
 & & & 2y_2 & + & 2y_3 & \leq & 4 \\
 & 2y_1 & & & - & y_3 & \leq & 1.
\end{array} \tag{11}
$$

The LP (9) is called the *primal LP*, while the LP (11) is the *dual LP* of (9).

Applying the principle just described, every LP has a *dual*. We list here several primal-dual forms. The basic rules are given in table 1.

In **standard form**, the primal and dual LPs are

$$
\begin{array}{rcl}
\min & \mathbf{c}^T\mathbf{x} & \text{(primal program)} \\
\text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} & \\
 & \mathbf{x} \geq \mathbf{0} &
\end{array}
$$

$$
\begin{array}{rcl}
\max & \mathbf{b}^T\mathbf{y} & \text{(dual program)} \\
\text{subject to} & \mathbf{A}^T\mathbf{y} \leq \mathbf{c} & \text{no non-negativity restriction!.}
\end{array}
$$

In **canonical form**, the primal and dual LPs are

$$\begin{array}{rll} \min & \mathbf{c}^T\mathbf{x} & \text{(primal program)} \\ \text{subject to} & \mathbf{A}\mathbf{x} \geq \mathbf{b} & \\ & \mathbf{x} \geq \mathbf{0} & \end{array}$$

$$\begin{array}{rll} \max & \mathbf{b}^T\mathbf{y} & \text{(dual program)} \\ \text{subject to} & \mathbf{A}^T\mathbf{y} \leq \mathbf{c} & \\ & \mathbf{y} \geq \mathbf{0}. & \end{array}$$

Exercise 37. Show that the standard and canonical primal-dual forms above are equivalent.

Exercise 38. Why in canonical form the dual program has the non-negativity constraints?

Exercise 39. Write the dual program of an LP in the max version of the standard form.

Exercise 40. Write the dual program of an LP in the max version of the canonical form.

Exercise 41. Show that the dual program of the dual program is the primal program.

Exercise 42. Write the dual program of the following linear programs:

$$\max\{\mathbf{c^T x} \mid \mathbf{Ax} = \mathbf{b}\}$$

$$\min\{\mathbf{c^T x} \mid \mathbf{Ax} \leq \mathbf{b}\}$$

$$\min\{\mathbf{c^T x} \mid \mathbf{Ax} \geq \mathbf{b}\}$$

$$\min\{\mathbf{c^T x} \mid \mathbf{A_1 x} = \mathbf{b_1}, \mathbf{A_2 x} \leq \mathbf{b_2}, \mathbf{A_3 x} \geq \mathbf{b_3}\}$$

Exercise 43. Write the dual program of the following linear program:

$$\begin{array}{rlllllll} \max & \mathbf{a^T x} + \mathbf{b^T y} + \mathbf{c^T z} & & & & & & \\ \text{subject to} & \mathbf{A}_{11}\mathbf{x} & + & \mathbf{A}_{12}\mathbf{y} & + & \mathbf{A}_{13}\mathbf{z} & = & \mathbf{d} \\ & \mathbf{A}_{21}\mathbf{x} & + & \mathbf{A}_{22}\mathbf{y} & + & \mathbf{A}_{23}\mathbf{z} & \leq & \mathbf{e} \\ & \mathbf{A}_{31}\mathbf{x} & + & \mathbf{A}_{32}\mathbf{y} & + & \mathbf{A}_{33}\mathbf{z} & \geq & \mathbf{f} \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{y} \leq \mathbf{0}. & & & & & & \end{array}$$

5.2 Primal dual relationship

Consider the standard form of the primal and dual programs:

$$\begin{array}{rl} \text{Primal LP:} & \min\{\mathbf{c^T x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \\ \text{Dual LP:} & \max\{\mathbf{b^T y} \mid \mathbf{A^T y} \leq \mathbf{c}\}. \end{array}$$

We have seen, as an example in the previous section, how $\mathbf{b}^T\mathbf{y}$ is the lower bound for the optimal objective value of the primal LP. Let us formalize this observation:

Theorem 5.1 (Weak Duality). *Suppose $\mathbf{x}$ is primal feasible, and $\mathbf{y}$ is dual feasible for the LPs defined above, then*

$$\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}.$$

In particular, if $\mathbf{x}^*$ is an optimal solution to the primal LP, and $\mathbf{y}^*$ is an optimal solution to the dual LP as defined above, then

$$\mathbf{c}^T\mathbf{x}^* \geq \mathbf{b}^T\mathbf{y}^*.$$

Proof. Noticing that $\mathbf{x} \geq 0$, we have

$$\mathbf{c}^T\mathbf{x} \geq \left(\mathbf{A}^T\mathbf{y}\right)^T \mathbf{x} = (\mathbf{y}^T\mathbf{A})\mathbf{x} = \mathbf{y}^T(\mathbf{A}\mathbf{x}) = \mathbf{y}^T\mathbf{b}.$$

□

Exercise 44. State and prove the weak duality property for the primal and dual programs written in canonical form:

$$\begin{aligned} \text{Primal LP:} \quad & \min\{\mathbf{c^T x} \mid \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \\ \text{Dual LP:} \quad & \max\{\mathbf{b^T y} \mid \mathbf{A^T y} \leq \mathbf{c}, \mathbf{y} \geq \mathbf{0}\}. \end{aligned}$$

Would your proof still work if one or both of the non-negativity constraints for $\mathbf{x}$ and $\mathbf{y}$ were removed?

The following result is almost immediate from the previous proof and Theorem 5.5, yet it is extremely important:

Corollary 5.2 (Complementary Slackness - standard form). *Let $\mathbf{x}^*$ and $\mathbf{y}^*$ be feasible for the primal and the dual programs (written in standard form as above), respectively. Then, $\mathbf{x}^*$ and $\mathbf{y}^*$ are optimal for their respective LPs if and only if*

$$\left(\mathbf{c} - \mathbf{A^T y}^*\right)^{\mathbf{T}} \mathbf{x}^* = \mathbf{0}. \tag{12}$$

Equation (12) can be written explicitly as follows:

$$\left(c_j - \sum_{i=1}^{m} y_i^* a_{ij}\right) x_j^* = 0, \;\; \forall j = 1, \ldots, n.$$

Also, since $\left(\mathbf{c} - \mathbf{A^T y}^*\right)^T \geq 0$ and $\mathbf{x}^* \geq 0$, we can write the condition as

$$\text{for all } j = 1, \ldots, n, \text{ if } \; c_j - \sum_{i=1}^{m} y_i^* a_{ij} > 0 \; \text{ then } \; x_j = 0, \;\; \text{and vice versa.}$$

After doing Exercise 44, we get the following easily:

Corollary 5.3 (Complementary Slackness - canonical form). *Given the following programs*

$$\begin{aligned} \textit{Primal LP:} \quad & \min\{\mathbf{c^T x} \mid \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \\ \textit{Dual LP:} \quad & \max\{\mathbf{b^T y} \mid \mathbf{A^T y} \leq \mathbf{c}, \mathbf{y} \geq \mathbf{0}\}. \end{aligned}$$

Let $\mathbf{x}^*$ and $\mathbf{y}^*$ be feasible for the primal and the dual programs, respectively. Then, $\mathbf{x}^*$ and $\mathbf{y}^*$ are optimal for their respective LPs if and only if

$$\left(\mathbf{c} - \mathbf{A^T y}^*\right)^{\mathbf{T}} \mathbf{x}^* = \mathbf{0}, \;\; \textit{and} \;\; (\mathbf{b} - \mathbf{Ax})^{\mathbf{T}} \mathbf{y}^* = \mathbf{0}. \tag{13}$$

Again, condition (13) can be written explicitly as

$$\left(c_j - \sum_{i=1}^{m} y_i^* a_{ij}\right) x_j^* = 0, \;\; \forall j = 1, \ldots, n,$$

and

$$\left(b_i - \sum_{j=1}^{n} x_i^* a_{ij}\right) y_i^* = 0, \;\; \forall i = 1, \ldots, m.$$

Exercise 45. Derive the complementary slackness condition for each of the following LPs and their corresponding duals.

(i) $\min\{\mathbf{c^T x} \mid \mathbf{Ax} = \mathbf{b}\}$.

(ii) $\max\{\mathbf{c^T x} \mid \mathbf{Ax} \leq \mathbf{b}\}$.

The weak duality property already tells us some thing about the unboundedness of the LPs involved:

Corollary 5.4. *If the primal and the dual are both feasible, then they are both bounded, and thus both have optimal solutions.*

In fact, we can say much more than that. The relationship between the primal and the dual is best illustrated by the following table:

			Dual		
			Feasible		Infeasible
			Optimal	Unbounded	
Primal	Feasible	Optimal	X	O	O
		Unbounded	O	O	X
	Infeasible		O	X	X

The X's are possible, the O's are impossible to happen. The previous corollary already proved four entries in the table, namely if both the dual and the primal are feasible, then they both have optimal solutions. We shall show the rest of the O entries by a stronger assertion that if either the dual or the primal has an optimal solution, then the other has an optimal solution with the same objective value. (Notice that the dual of the dual is the primal.)

Theorem 5.5 (Strong Duality). *If the primal LP has an optimal solution $\mathbf{x}^*$, then the dual LP has an optimal solution $\mathbf{y}^*$ such that*

$$\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*.$$

Proof. By weak duality, we only need to find a feasible $\mathbf{y}^*$ such that $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$. Without loss of generality, assume $\mathbf{x}^*$ is a vertex of the polyhedron $P = \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ returned by the simplex algorithm, where $\mathbf{A}$ has dimension $m \times n$, with $m \leq n$, and $\text{rank}(\mathbf{A}) = m$. Let $\mathbf{A}_B, \mathbf{A}_N$ denote the parts of $\mathbf{A}$ corresponding to the basis and non-basis columns, i.e. $\mathbf{A}_B$ is an $m \times m$ invertible matrix and $x_j = 0, \forall j \in N$. When the simplex algorithm stop, the cost of $\mathbf{x}^*$ is

$$\mathbf{c}^T\mathbf{x}^* = \mathbf{y}_B^T b,$$

where $\mathbf{y}_B^T = \mathbf{c}_B^T\mathbf{A}_B^{-1}$. It seems that $\mathbf{y}_B$ is a good candidate for $\mathbf{y}^*$. We only need to verify its feasibility:

$$\mathbf{A}^T\mathbf{y}_B = \begin{bmatrix}\mathbf{A}_B^T \\ \mathbf{A}_N^T\end{bmatrix}\mathbf{y}_B = \begin{bmatrix}\mathbf{c}_B \\ \mathbf{A}_N^T\mathbf{y}_B\end{bmatrix} \leq \begin{bmatrix}\mathbf{c}_B \\ \mathbf{c}_N\end{bmatrix}.$$

The last inequality holds because, when the simplex method outputs the optimum vertex, we have $\mathbf{c}_N^T - \mathbf{y}_B^T\mathbf{A}_N \geq 0$. □

Exercise 46. Consider the linear program $\min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}, \mathbf{I}'\mathbf{x} \geq \mathbf{0}\}$, where $\begin{bmatrix}\mathbf{A} \\ \mathbf{I}'\end{bmatrix}$ is a square matrix, and $\mathbf{I}'$ is a subset of rows of an identity matrix. Suppose $\mathbf{x}^*$ is the unique optimal solution to this linear program that satisfies all constraints with equality. Construct a dual solution $\mathbf{y}*$ that certifies the optimality of $\mathbf{x}^*$.

Exercise 47. Prove that the system $\mathbf{Ax} \le \mathbf{b}$ can be partitioned into two subsystems $\mathbf{A}_1\mathbf{x} \le \mathbf{b}_1$ and $\mathbf{A}_2\mathbf{x} \le \mathbf{b}_2$ such that

$$\max\{\mathbf{c}^T\mathbf{x} \mid \mathbf{A}_1\mathbf{x} < \mathbf{b}_1,\ \mathbf{A}_2\mathbf{x} = \mathbf{b}_2\} = \min\{\mathbf{y}_2^T\mathbf{b}_2 \mid \mathbf{y}_2 > \mathbf{0},\ \mathbf{A}_2^T\mathbf{y}_2 = \mathbf{c}\}.$$

Use this result to prove the Fourier-Motzkin transposition theorem (Theorem 4.5).

Exercise 48. Given a system $\mathbf{Ax} \le \mathbf{b}$ of linear inequalities, describe a linear program whose optimal solution immediately tells us which inequalities among $\mathbf{Ax} \le \mathbf{b}$ are always satisfied with equality.

Exercise 49. Prove the strong duality theorem using Farkas' lemma instead of using the simplex algorithm as we have shown.

5.3 Intepreting the notion of dualily

There are many ways to intepret the meaning of primal-dual programs. In economics, for instance, dual variables correspond to *shadow prices*. In optimization, they correspond to *Lagrange multipliers*. We briefly give a geometric intepretation here.

Consider our favorite primal program $\min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$, and its dual $\max\{\mathbf{b}^T\mathbf{y} \mid \mathbf{A}^T\mathbf{y} \le \mathbf{c}\}$. A feasible solution $\mathbf{x}$ to the primal program simply indicates that $\mathbf{b}$ is in the cone generated by the column vectors $\mathbf{a}_j$ of $\mathbf{A}$. At an optimal vertex $\mathbf{x}^*$, there are m linearly independent columns $\mathbf{a}_{j_1}, \dots, \mathbf{a}_{j_m}$ of $\mathbf{A}$ such that $\mathbf{b} = \sum_{i=1}^m x^*_{j_i}\mathbf{a}_{j_i}$. Let $d = \mathbf{c}^T\mathbf{x} = \sum_{i=1}^m x^*_{j_i} c_{j_i}$.

TBD:

6 More on polyhedral combinatorics (very much incomplete)

6.1 Decomposing a polyhedron

Definition 6.1 (Polyhedral cones). A cone C is *polyhedral* if $C = \{\mathbf{x} \mid \mathbf{Ax} \le \mathbf{0}\}$ for some real matrix $\mathbf{A}$, i.e. C is the intersection of finitely many linear half spaces.

Theorem 6.2 (Farkas-Minkowski-Weyl). *A convex cone is polyhedral if and only if it is finitely generated.*

Proof. Let $C = \text{cone}\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$, i.e. C is finitely generated. We shall show that C is polyhedral. Suppose $\mathbf{a}_j \in \mathbb{R}^m, \forall j$. Without loss of generality, assume that the $\mathbf{a}_j$ span $\mathbb{R}^m$. (If not, we can always extend a half-space in the span of the $\mathbf{a}_j$ to a half-space of $\mathbb{R}^m$.) If $C = \mathbb{R}^m$, then there is nothing to show. Otherwise, let b be a vector not in C, then by the fundamental theorem there is a hyperplane $\{\mathbf{x} \mid \mathbf{c}^T\mathbf{x} = 0\}$ containing $m-1$ independent vectors from $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ such that $\mathbf{c}^T\mathbf{a}_j \ge 0$ for all j. In other words, the $\mathbf{a}_j$ belongs to a half space defined by $\mathbf{c}$. The number of such half-spaces is at most $\binom{n}{m-1}$. It is easy to see that C is the intersection of all such half-spaces.

Conversely, consider a polyhedral cone $C = \{\mathbf{x} \mid \mathbf{Ax} \le \mathbf{0}\}$. Let $\mathbf{a}_1, \dots, \mathbf{a}_m$ denote the row vectors of $\mathbf{A}$, then C is the intersection of the half-spaces $\{\mathbf{x} \mid \mathbf{a}_i^T\mathbf{x} \le \mathbf{0}\}$. As we have just shown above, there is a matrix $\mathbf{B}$ with row vectors $\mathbf{b}_1, \dots, \mathbf{b}_k$ such that

$$\text{cone}(\mathbf{a}_1, \dots, \mathbf{a}_m) = \{\mathbf{y} \mid \mathbf{By} \le \mathbf{0}\}.$$

In particular, $\mathbf{b}_j^T\mathbf{a}_i \le 0, \forall i, j$, since $\mathbf{a}_i \in \text{cone}(\mathbf{a}_1, \dots, \mathbf{a}_m)$. Thus, $\mathbf{Ab}_j \le 0, \forall j$.

We shall show that

$$\text{cone}(\mathbf{b}_1, \dots, \mathbf{b}_k) = \{\mathbf{x} \mid \mathbf{Ax} \le \mathbf{0}\}.$$

Consider $\mathbf{x} = \sum_j \alpha_j\mathbf{b}_j$, where $\alpha_j \ge 0, \forall j$. Then, $\mathbf{Ax} = \sum_j \alpha_j\mathbf{Ab}_j \le 0$. Conversely, consider a vector $\mathbf{x}$ such that $\mathbf{Ax} \le \mathbf{0}$. Assume $\mathbf{x} \notin \text{cone}(\mathbf{b}_1, \dots, \mathbf{b}_k)$, then the fundamental theorem implies that there is a vector $\mathbf{c}$ such that $\mathbf{c}^T\mathbf{x} > 0$ and $\mathbf{Bc} \le \mathbf{0}$. Thus $\mathbf{c} \in \text{cone}(\mathbf{a}_1, \dots, \mathbf{a}_m)$, implying that $\mathbf{c}$ can be written as a non-negative combination $\mathbf{c} = \sum_i \beta_i\mathbf{a}_i$. But then $\mathbf{c}^T\mathbf{x} = \sum_i \beta_i\mathbf{a}_i\mathbf{x} \ge 0$, a contradiction. $\square$

Exercise 50 (Finite basis theorem for polytopes). Show that a set of points is a polytope if and only if it is the convex hull of finitely many vectors.

Exercise 51 (Decomposition theorem for polyhedra). Show that, a set P of vectors in a Euclidean space is a polyhedron if and only if $P = Q + C$ for some polytope Q and some polyhedral cone C.

6.2 Faces and facets

Let $P = \{\mathbf{x} \mid \mathbf{Ax} \leq \mathbf{b}\}$. Let $\mathbf{c}$ be a non-zero vector, and $d = \max\{\mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in P\}$. Then the hyperplane $\mathbf{c}^T\mathbf{x} = d$ is called a *supporting hyperplane* of P. Let H be a supporting hyperplane of P, then $H \cap P$ is called a *face* of P. For convenience, P is also called a face of itself. Basically, a face can be thought of as the set of optimal solution to some linear program on P. (P is the set of solutions when $\mathbf{c} = \mathbf{0}$.)

Exercise 52. Show that F is a face of P if and only if $F \neq \emptyset$ and $F = \{\mathbf{x} \mid \mathbf{x} \in P,\ \mathbf{A}'\mathbf{x} = \mathbf{b}'\}$ for some subsystem $\mathbf{A}'\mathbf{x} \leq \mathbf{b}'$ of $\mathbf{Ax} \leq \mathbf{b}$.

Exercise 53. Show that

(i) P has finitely many faces

(ii) Each face is a non-empty polyhedron

(iii) If F is a face of P, then $F' \subseteq F$ is a face of F iff F' is a face of P.

Exercise 54. A *facet* is a maximal face other than P. Show that the dimension of every facet is one less than the dimension of P.

TBD:

7 The Ellipsoid Algorithm

We briefly sketch the idea of the Ellipsoid algorithm by Khachian in this section. What we will need in designing a variety of approximation algorithms is a way to find optimal solution to linear programs with an exponential number of constraints. The notion of a *separation oracle* will sometimes help us accomplish this task.

Given a positive definite matrix $\mathbf{D}$ of order n and a point $\mathbf{z} \in \mathbb{R}^n$, the set

$$E(\mathbf{z}, \mathbf{D}) = \{\mathbf{x} \mid (\mathbf{x} - \mathbf{z})^T\mathbf{D}^{-1}(\mathbf{x} - \mathbf{z}) \leq 1\}$$

is called an *ellipsoid* with center $\mathbf{z}$.

Exercise 55. Show that $E(\mathbf{z}, \mathbf{D}) = \mathbf{D}^{1/2}E(\mathbf{0}, \mathbf{I}) + \mathbf{z}$. In other words, every ellipsoid is an affine transformation of the unit sphere $E(\mathbf{0}, \mathbf{I})$.

The basic ellipsoid algorithm finds a point $\mathbf{z}$ in the polyhedron $P = \{\mathbf{Ax} \leq \mathbf{b}\}$, or reports that P is empty. The algorithm runs in polynomial time. To use the ellipsoid algorithm to solve linear programs, we can add appropriate upper and lower bounds on the objective function as constraints, then do a binary search. (More details on this later.)

In the following algorithm, we assume that the polyhedron is full-dimensional and bounded, and that computation with inifite precisions can be carried out. Let ν be the maximum number of bits required to describe a vertex of $\mathbf{Ax} \leq \mathbf{b}$. (We can set $\nu = n^2\phi$, where ϕ is the maximum number of bits required to describe a constraint in the system $\mathbf{Ax} \leq \mathbf{b}$.) Set $r = 2^\nu$. ELLIPSOID ALGORITHM($\mathbf{A}$, $\mathbf{b}$)

Start with the ellipsoid $E_0 = E(\mathbf{0}, r^2\mathbf{I})$ which contains P.
$k = 0$
while The center $\mathbf{z}_k$ of E_k is not in P **do**
  if Vol$(E_k) \leq$ Vol(P) **then**
    (since $2^{-2n\nu} \leq$ Vol(P), we can check if Vol$(E_k) \leq 2^{-2n\nu}$ instead)
    Return INFEASIBLE
  end if
  Find a constraint $\mathbf{a}^{(i)}\mathbf{x} \leq b_i$ violated by z_k
  (Note that the plain $\mathbf{a}^{(i)}\mathbf{x} \leq b_i$ is a hyperplane separating z_k from the polyhedron:

$$\mathbf{a}^{(i)}\mathbf{x} \leq b_i < \mathbf{a}^{(i)}\mathbf{z}_k$$

  for all $\mathbf{x} \in P$.)
  Let E_{k+1} be the minimum ellipsoid containing the half $E_k \cap \{\mathbf{x} \mid \mathbf{a}^{(i)}\mathbf{x} \leq \mathbf{a}^{(i)}\mathbf{z}_k\}$.
  (Note that $P \subseteq E_{k+1}$, still.)
  $k \leftarrow k + 1$
end while
Return $\mathbf{z}_k$

The analysis of the ellipsoid algorithm is based on the following theorem, whose proof can be found in [26].

Theorem 7.1. *Let $E = E(\mathbf{z}, \mathbf{D})$ be an ellipsoid in $\mathbb{R}^n$, and let $\mathbf{a}$ be a vector in $\mathbb{R}^n$. Let E' be an ellipsoid with minimum volume containing $E \cap \{\mathbf{x} \mid \mathbf{a}^T\mathbf{x} \leq \mathbf{a}^T\mathbf{z}\}$. Then, $E' = E(\mathbf{z}', \mathbf{D}')$, where*

$$\mathbf{z}' = \mathbf{z} - \frac{1}{n+1}\frac{\mathbf{D}\mathbf{a}}{\sqrt{\mathbf{a}^T\mathbf{D}\mathbf{a}}} \tag{14}$$

$$\mathbf{D}' = \frac{n^2}{n^2-1}\left(\mathbf{D} - \frac{2}{n+1}\frac{\mathbf{D}\mathbf{a}\mathbf{a}^T\mathbf{D}}{\mathbf{a}^T\mathbf{D}\mathbf{a}}\right). \tag{15}$$

In particular, E' is unique. Furthermore,

$$\text{Vol}(E') < e^{-\frac{1}{2n+2}}\,\text{Vol}(E).$$

From the theorem, it can be shown that the number of iterations of the ellipsoid algorithm is at most $N = 16n^2\nu$. Note that N does not depend on the number of constraints of the system $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. Consequently, the running time of the algorithm would still be polynomial if we have a polynomial time procedure to confirm if $\mathbf{z}_k \in P$, and otherwise return a separation hyperplane separating P from $\mathbf{z}_k$. Such a procedure is called a *separation oracle*.

Exercise 56. Suppose we use the ellipsoid method to solve a linear program whose corresponding polyhedron is $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$, which is pointed. The optimal solution $\mathbf{x}^*$ returned by the ellipsoid algorithm may not be a vertex of P. How do we find an optimal vertex of P efficiently from $\mathbf{x}^*$?

Example 7.2. The MINIMUM-COST ARBORESCENCE PROBLEM, also called the MINIMUM-COST BRANCHING PROBLEM is defined as follows. Given a directed, edge-weighted graph $G = (V, E)$ with a special vertex r called the root. Find a minimum-cost spanning tree where edges are directed away from r. Basically, every cut $[S, \bar{S}]$ with $r \in S$ must contain at least one edge of the tree. Thus, an equivalent integer linear program is

$$\begin{aligned} \min \quad & \sum_{e \in E} w_e x_e \\ \text{subject to} \quad & \sum_{e \in [S,\bar{S}]} x_e \geq 1, \quad \forall S \subseteq V, r \in S \\ & x_e \in \{0,1\} \qquad \forall e \in E. \end{aligned} \tag{16}$$

Relaxing the integral constraints to $0 \leq x_e \leq 1$, and we have a linear program. Edmonds showed that the set of feasible solutions to the LP is exactly the convex hull of characteristic vectors of arborescences of G. Thus, if we can find an optimal vertex of the corresponding polyhedron in polynomial time, then this problem can be solved in polynomial time.

Since the number of constraints of the LP is exponential, we devise an efficient separation oracle and apply the ellipsoid algorithm. Given any vector $\mathbf{z} \in \mathbb{R}^{|E|}$, checking $0 \leq z_e \leq 1$ is trivial. Now, think of x_e as the capacity of edge e. Checking $\displaystyle\sum_{e \in [S, \bar{S}]} x_e \geq 1$ is the same as checking if the minimum cut from r to any vertex u of G is of capacity at least 1. This can certainly be done by invoking the max-flow min-cut algorithm $n-1$ times. In case the min-cut capacity is less than 1, the max-flow min-cut algorithm also returns such a cut $[S, \bar{S}]$ which gives us the separation hyperplane!

Example 7.3. In the MULTIWAY NODE CUT problem, we are given a vertex-weight graph $G = (V, E)$ with weight function $w : V \to \mathbb{Z}^+$, and an independent set of *terminals* $T \subset G$. The objective is to find a subset of $V - T$ whose removal disconnect the terminals from each other. Let $\mathcal{P}$ be the set of all paths connecting the terminals, then an equivalent integer linear program is

$$\begin{aligned}
\min \quad & \sum_{v \in V - T} w_v x_v \\
\text{subject to} \quad & \sum_{v \in P \setminus T} x_v \geq 1, \quad \forall P \in \mathcal{P} \\
& x_v \in \{0, 1\} \quad \forall v \in V.
\end{aligned} \tag{17}$$

In the corresponding linear program, we turn the integral constraints into $0 \leq x_v \leq 1$. To check if $\mathbf{x}$ is feasible, construct a directed graph D by turning each edge uv of G into two edges (u, v) and (v, u) of D. Assign a weight of x_v to edge (u, v) and a weight of x_v to edge (v, u). (For convenience, we set $x_v = 0$ if $v \in T$.) Then, find all shortest paths among all pairs of terminals in D. If one such shortest path has length < 1, then we have found a separating hyperplane. Otherwise the solution is feasible.

Exercise 57. The MINIMUM MULTICUT problem can be defined as follows. We are given a graph $G = (V, E)$ where each edge e has a non-negative integral capacity c_e. We are also given k pairs of vertices $(s_1, t_1), \ldots, (s_k, t_k)$, where each pair consists of two different vertices, but the vertices from different pairs are not necessarily different. The problem is to find a minimum capacity subset of edges whose removal separates each of the given pairs. Let $\mathcal{P}_i$ be the set of all paths connecting s_i to t_i, and $\mathcal{P}$ be the union of all $\mathcal{P}_i$. The problem is equivalent to the following ILP:

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{subject to} \quad & \sum_{e \in P} x_e \geq 1, \quad \forall P \in \mathcal{P} \\
& x_e \in \{0, 1\} \quad \forall e \in E.
\end{aligned} \tag{18}$$

Show that the relaxed LP of this ILP can be solved efficiently using the ellipsoid method.

Exercise 58. The GROUP STEINER TREE problem can be defined as follows. We are given a graph $G = (V, E)$ and non-negative integral cost c_e for each edge e. There are k disjoint groups of vertices $X_1, \ldots, X_k$. The objective is to find a minimum-cost subgraph T of G which contains at least one vertex from each group. Clearly, T only needs to be a tree, which is called a *Steiner tree*. (In the *Steiner Tree* problem, each group contains one vertex.)

Note 1: although it does not concern us in this problem, it is worth mentioning that we can assume that the cost function c satisfies the triangle inequality. For if an edge $e = (u, v)$ in the optimal solution has greater cost than some path between u and v, we can replace e by this path.

Note 2: the assumption that the groups are disjoint can also be relaxed. If there was a vertex v contained in m groups, we can add m new vertices connected to v with new edge costs equal to zero. Then, add each new vertex to a distinct group in that set of m groups, and remove v from all the groups.

We will consider a version of this problem where the Steiner tree has to contain a given "root" $r \in V$. If we can solve this version of the problem, the original version can be solved by running the algorithm for the rooted version over all $r \in X_1$, then take the best resulting Steiner tree.

The rooted version can be formulated as follows.

$$\begin{array}{lll} \min & \sum_{e \in E} c_e x_e & \\ \text{subject to} & \sum_{e \in [S, \bar{S}]} x_e \geq 1, & \forall S \subseteq V, \text{ such that } r \in S, \text{ and } \bar{S} \cap (\bigcup_{i=1}^{k} X_i) \neq \emptyset \\ & x_e \in \{0, 1\} & \forall e \in E. \end{array} \tag{19}$$

Show that the relaxed LP of this ILP can be solved efficiently using the ellipsoid method.

Lastly, we describe two (of several) ways to solve a linear program using the ellipsoid algorithm:

- Consider the primal dual pair of linear programs:

  $$\min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

  and

  $$\max\{\mathbf{b}^T\mathbf{y} \mid \mathbf{A}^T\mathbf{y} \leq \mathbf{c}\}.$$

  To solve both programs at the same time, simply find a feasible solution to the polyhedron

  $$P = \left\{ \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \mid \mathbf{b}^T\mathbf{y} = \mathbf{c}^T\mathbf{x}, \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{A}^T\mathbf{y} \leq \mathbf{c} \right\}.$$

  For numerical accuracy consideration, some perturbation might need to be done.

- We could also apply the so-called *sliding objective* method as follows. Suppose we try to minimize a linear objective $\mathbf{c}^T\mathbf{x}$ over a polyhedron P. Find $\mathbf{x}_0 \in P$. At iteration k, apply the ellipsoid algorithm to $P \cap \{\mathbf{x} \mid \mathbf{c}^T\mathbf{x} < \mathbf{c}^T\mathbf{x}_k\}$.

Historical Notes (Very much incomplete)

Standard texts on linear algebra and algebra are [2] and [34]. Texts on linear programming are numerous, of which I recommend [11] and [26].

The idea (moving along edges of the feasible polyhedron from vertex to vertex) for the *simplex method* dated back to Fourier (1826), and mechanized algebraically by George Dantzig in 1947 (published in 1951 [12]), who also acknowledged fruitful conversation with von Neumann. This worst-case exponential algorithm has proved to work very well for most practical problems. Even now, when we know of many other polynomial time algorithms [18,19,36] to solve linear programs, the simplex method is still among the best when it comes to practice. The worst-case complexity of the simplex method was determined to be exponential when Klee and Minty [21] found an example where the method actually visits all vertices of the feasible polyhedron.

The quest for a provably good algorithm continues, until Khachian [19] devised the *ellipsoid method* in 1979. The method performs poorly in practice, however. A breakthrough was made by Karmarkar in 1984 [18], when he found a method which works in provably polynomial time, and also 50 times faster than the simplex method in his experiments. Karmarkar's method was of the *interior point* type of method, where one keeps moving a point strictly inside the feasible region toward an optimal vertex. This method applies to non-linear programming as well. For a recent discussion on interior methods, see [14]. In fact, the simplex method is still the most popular method to be applied in practice. Somehow it runs in polynomial time on most inputs. To explain this phenomenon, researchers have tried to show that, under some certain probabilistic distributions of linear programs, the simplex method takes a polynomial number of iterations on average. See, for example, Borgwardt [6–9], Smale [27, 28], Spielman and Teng [29–33]. Recently, Kelner and Spielman gave the first polynomial time randomized simplex algorithm to solve linear programs [].

In 1957, Warren Hirsch conjectured that the diameter of an n-dimensional polytope with m facets is at most $m - n$. The conjecture does not hold for unbounded polyhedra (Klee and Walkup []). Kalai and Kleitman [17] proved a quasi-polynomial upper bound on the shortest path between any pair of vertices: $m^{\log_2 n+2}$. Larman [24] showed the upper bound $2^{n-2}m$. See [1, 20, 22] for related results.

The concise surveys [3, 35] on linear programming contain many good references and interesting discussions.

References

[1] N. AMENTA AND G. M. ZIEGLER, *Deformed products and maximal shadows of polytopes*, in Advances in discrete and computational geometry (South Hadley, MA, 1996), vol. 223 of Contemp. Math., Amer. Math. Soc., Providence, RI, 1999, pp. 57–90.

[2] M. ARTIN, *Algebra*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1991.

[3] M. L. BALINSKI, *Mathematical programming: journal, society, recollections*, in History of Mathematical Programming, Elsevier Science Publisher, Amsterdam, 1991, pp. 5–18.

[4] E. M. L. BEALE, *Cycling in the dual simplex algorithm*, Naval Res. Logist. Quart., 2 (1955), pp. 269–275 (1956).

[5] R. G. BLAND, *New finite pivoting rules for the simplex method*, Math. Oper. Res., 2 (1977), pp. 103–107.

[6] K.-H. BORGWARDT, *The average number of pivot steps required by the simplex-method is polynomial*, Z. Oper. Res. Ser. A-B, 26 (1982), pp. A157–A177.

[7] K.-H. BORGWARDT, *Some distribution-independent results about the asymptotic order of the average number of pivot steps of the simplex method*, Math. Oper. Res., 7 (1982), pp. 441–462.

[8] K.-H. BORGWARDT, *Probabilistic analysis of optimization algorithms—some aspects from a practical point of view*, Acta Appl. Math., 10 (1987), pp. 171–210.

[9] K.-H. BORGWARDT, *The simplex method*, vol. 1 of Algorithms and Combinatorics: Study and Research Texts, Springer-Verlag, Berlin, 1987. A probabilistic analysis.

[10] A. CHARNES, *Optimality and degeneracy in linear programming*, Econometrica, 20 (1952), pp. 160–170.

[11] V. CHVÁTAL, *Linear programming*, A Series of Books in the Mathematical Sciences, W. H. Freeman and Company, New York, 1983.

[12] G. B. DANTZIG, *Maximization of a linear function of variables subject to linear inequalities*, in Activity Analysis of Production and Allocation, Cowles Commission Monograph No. 13, John Wiley & Sons Inc., New York, N. Y., 1951, pp. 339–347.

[13] G. B. DANTZIG, A. ORDEN, AND P. WOLFE, *The generalized simplex method for minimizing a linear form under linear inequality restraints*, Pacific J. Math., 5 (1955), pp. 183–195.

[14] A. FORSGREN, P. E. GILL, AND M. H. WRIGHT, *Interior methods for nonlinear optimization*, SIAM Review, 44 (2002), pp. 525–598.

[15] J. B. J. FOURIER, *Solution d'une question particulière du calcul des inégalités*, Nouveau Bulletin des Sciences par la Société philomathique de Paris, (1826), pp. 99–100.

[16] R. G. JEROSLOW, *The simplex algorithm with the pivot rule of maximizing criterion improvement*, Discrete Math., 4 (1973), pp. 367–377.

[17] G. KALAI AND D. J. KLEITMAN, *A quasi-polynomial bound for the diameter of graphs of polyhedra*, Bull. Amer. Math. Soc. (N.S.), 26 (1992), pp. 315–316.

[18] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[19] L. G. KHACHIAN, *A polynomial algorithm for linear programming*, Dokl. Akad. Nauk SSSR, 244 (1979), pp. 1093–1096. English translation in Soviet Math. Dokl. 20, 191-194, 1979.

[20] V. KLEE AND P. KLEINSCHMIDT, *The d-step conjecture and its relatives*, Math. Oper. Res., 12 (1987), pp. 718–755.

[21] V. KLEE AND G. J. MINTY, *How good is the simplex algorithm?*, in Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin), Academic Press, New York, 1972, pp. 159–175.

[22] P. KLEINSCHMIDT, *The diameter of polytopes and related applications*, in Polytopes: abstract, convex and computational (Scarborough, ON, 1993), vol. 440 of NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., Kluwer Acad. Publ., Dordrecht, 1994, pp. 467–492.

[23] H. W. KUHN, *Solvability and consistency for linear equations and inequalities*, Amer. Math. Monthly, 63 (1956), pp. 217–232.

[24] D. G. LARMAN, *Paths of polytopes*, Proc. London Math. Soc. (3), 20 (1970), pp. 161–178.

[25] T. S. MOTZKIN, *Beiträge zur theorie der linearen ungleichungen*, 1936. (Inaugural Dissertation Basel) Azriel, Jerusalem.

[26] A. SCHRIJVER, *Theory of linear and integer programming*, Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons Ltd., Chichester, 1986. A Wiley-Interscience Publication.

[27] S. SMALE, *On the average number of steps of the simplex method of linear programming*, Math. Programming, 27 (1983), pp. 241–262.

[28] S. SMALE, *The problem of the average speed of the simplex method*, in Mathematical programming: the state of the art (Bonn, 1982), Springer, Berlin, 1983, pp. 530–539.

[29] D. SPIELMAN AND S.-H. TENG, *Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time*, in Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, New York, 2001, ACM, pp. 296–305 (electronic).

[30] D. A. SPIELMAN AND S.-H. TENG, *Smoothed analysis of algorithms*, in Proceedings of the International Congress of Mathematicians, Vol. I (Beijing, 2002), Beijing, 2002, Higher Ed. Press, pp. 597–606.

[31] ———, *Smoothed analysis: motivation and discrete models*, in Algorithms and data structures, vol. 2748 of Lecture Notes in Comput. Sci., Springer, Berlin, 2003, pp. 256–270.

[32] ———, *Smoothed analysis of termination of linear programming algorithms*, Math. Program., 97 (2003), pp. 375–404. ISMP, 2003 (Copenhagen).

[33] ———, *Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time*, J. ACM, 51 (2004), pp. 385–463 (electronic).

[34] G. STRANG, *Linear algebra and its applications*, Academic Press [Harcourt Brace Jovanovich Publishers], New York, second ed., 1980.

[35] M. J. TODD, *The many facets of linear programming*, Math. Program., 91 (2002), pp. 417–436. ISMP 2000, Part 1 (Atlanta, GA).

[36] Y. Y. YE, *Extensions of the potential reduction algorithm for linear programming*, J. Optim. Theory Appl., 72 (1992), pp. 487–498.