Chapter 4

# Linear Programming

We assume that vectors are defined as column vectors. Also vector inequalities are understood componentwise, that is, $x \geq 0$ for $x \in \mathbb{R}^n$ means $x_i \geq 0$ for $i = 1, \ldots, n$.

**Definition 4.1.** A *linear program* is an optimization problem of the form

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

where $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $A$ is an $m \times n$ matrix.

We call $c^T x$ the *objective function*, and $Ax \leq b$ are constraints. We refer to this linear program as (LP).

The constraint $x \geq 0$ can also be included in constraints of the form $A'x \leq b'$ by setting $A' = \begin{bmatrix} A \\ -I \end{bmatrix}$ where $I_n$ is the identity matrix and $b' = \begin{bmatrix} b \\ 0_n \end{bmatrix}$.

Note that the above form describes a linear objective function in $x$ with linear constraints in $x$ without loss of generality: if some $x_j \leq 0$, then set a new variable $x'_j = -x_j \geq 0$, if some $x_j \geq d$, for some constant $d$, then set a new variable $x'_j = x_j - d \geq 0$, if some $x_j \leq d$, for some constant $d$, then set a new variable $x'_j = d - x_j \geq 0$, and if a constraint is of the form $a_j x \geq b_j$, then write instead $-a_j x \leq -b_j$, and to minimize $c^T x$, maximize $-c^T x$. Finally, a variable $x_j$ may appear with no constraint on being positive or negative, we call such a variable a *free (or unrestricted) variable.* A free variable can be replaced by $x_j = u_j - v_j$, $u_j, v_j \geq 0$.

A linear program can also be stated as

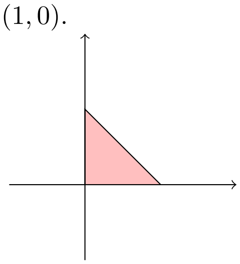
$$\begin{aligned} \min \quad & y^T b \\ s.t. \quad & y^T A \geq c^T \\ & y \geq 0 \end{aligned}$$

where $c \in \mathbb{R}^n$, $b, y \in \mathbb{R}^m$ and $A$ is an $m \times n$ matrix. We will discuss more this form in Section 4.3.

**Example 4.1.** Consider the linear program:

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ s.t. \quad & x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

where $c^T = [2, 1]$ and $A = [1, 1]$. The constraints delimit a portion of $\mathbb{R}^2$, namely a triangle whose vertices are (0,0), (0,1) and (1,0). To maximize the objective function, we notice that since $x_1$, $x_2$ should be as large as possible, we are looking at points on the line $x_1 + x_2 = 1$, and the maximum is reached at $(1, 0)$.



## 4.1 Feasible and Basic Solutions

**Definition 4.2.** The set

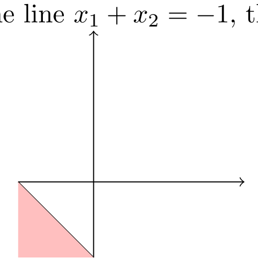
$$\{x \in \mathbb{R}^n,\ Ax \leq b, x \geq 0\}$$

is called the *feasible region* of (LP). A point $x$ in the feasible region is called a feasible solution. An LP is said to be feasible if the feasible region is not empty, and infeasible otherwise.

**Example 4.2.** In the above example, the feasible region is the triangle whose vertices are (0,0), (0,1) and (1,0). Consider the linear program:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 + x_2 \leq -1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

The constraints are requesting that $x_1$, $x_2$ are non-negative, but also below the line $x_1 + x_2 = -1$, this LP is thus infeasible.



**Definition 4.3.** A feasible maximum (respectively minimum) LP is said to be *unbounded* if the objective function can assume arbitrarily large positive (respectively negative) values at feasible points. Otherwise it is said to be *bounded*.

**Example 4.3.** The linear program

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1, x_2 \geq 0 \end{aligned}$$

is unbounded, no optimal solution exists.

We thus have 3 possibilities for a LP:

- It is feasible and bounded, see Example 4.1.
- It is feasible and unbounded, see Example 4.3.
- It is infeasible, see Example 4.2.

**Definition 4.4.** A *slack variable* is a variable that is added to an inequality constraint in $Ax \leq b$ to transform it into an equality.

**Example 4.4.** Consider the linear program:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 + 3x_2 \leq 9 \\ & 2x_1 + x_2 \geq 8 \\ & x_1, x_2 \geq 0. \end{aligned}$$

We can rewrite it as

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 + 3x_2 + s_1 = 9 \\ & 2x_1 + x_2 - s_2 = 8 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{aligned}$$

and $s_1, s_2$ are slack variables. Indeed, $2x_1+x_2-s_2 = 8 \iff 2x_1+x_2 = 8+s_2 \geq 8$ for $s_2 \geq 0$, and similarly $x_1 + 3x_2 + s_1 = 9 \iff x_1 + 3x_2 = 9 - s_1 \leq 9$ for $s_1 \geq 0$.

With slack variables the problem

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

has the *standard form*

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

and in fact, as discussed above, any LP can be converted to this standard form. The converse is true. Suppose we have an LP of the form

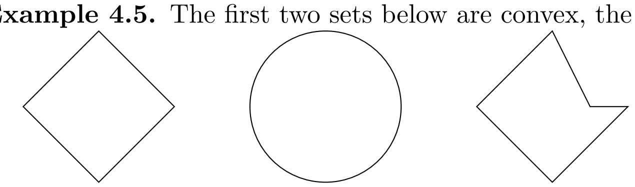
$$\begin{aligned} \max \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

then we can replace $Ax = b$ by $Ax \leq b$ and $-Ax \leq -b$.

In a linear program in standard form, we assume that the $m \times n$ matrix $A$ has rank $m$, with $m \leq n$. The case $m > n$ corresponds to having an over determined system, where the number of constraints $m$ is more than the number of unknowns $n$. In this case, the system typically does not have a solution, and this requires a different approach, which we will not consider. We will see in the next chapter that sometimes having an $m \times n$ matrix $A$ of rank less than $m$ can be exploited to introduce degrees in freedom in how to solve the corresponding LP, but for this chapter, if we have a redundant equality, we will just assume that we remove it.

**Definition 4.5.** A set $S \subset \mathbb{R}^n$ is called *convex* if for all $u, v \in S$, $\lambda u + (1-\lambda)v \in S$ for all $\lambda \in ]0,1[$ (the notation $(0,1)$ is also often found instead of $]0,1[$).

**Example 4.5.** The first two sets below are convex, the 3rd one is not.



**Proposition 4.1.** *The feasible region $S = \{x \in \mathbb{R}^n,\ Ax = b,\ x \geq 0\}$ of (LP) is convex.*

*Proof.* Suppose $u, v \in S$, $\lambda \in ]0,1[$. Set $w = \lambda u + (1-\lambda)v$. Then

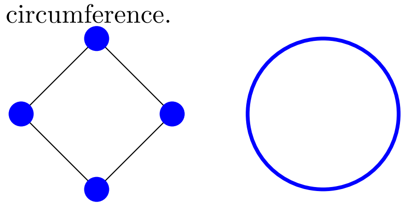
$$Aw = \lambda Au + (1-\lambda)Av = \lambda b + (1-\lambda)b = b$$

so $w$ satisfies $Aw = b$. Then $w = \lambda u + (1-\lambda)v$ where $\lambda$ lives in $]0,1[$, and $u, v \geq 0$ so $w \geq 0$. $\square$

**Definition 4.6.** A point $x$ in a convex set $S$ is called an *extreme point* of $S$ is there are no distinct points $u, v \in S$ and $\lambda \in ]0,1[$ such that $x = \lambda u + (1-\lambda)v$.

In words, this is saying that an extreme point is not in the interior of any line segment in $S$.

**Example 4.6.** In the example above, for the circle, the extreme points are on its circumference.



Extreme points of convex sets are particularly important in the context of linear programming because of the following theorem, which tells us that optimal solutions of (LP) are found among extreme points.

**Theorem 4.2.** *If an LP has an optimal solution (an optimal solution is a feasible solution that optimizes the objective function), then it has an optimal solution at an extreme point of the feasible set* $S = \{x \in \mathbb{R}^n,\ Ax = b, x \geq 0\}$.

*Proof.* Since there exists an optimal solution, there exists an optimal solution $x$ with a minimal number of nonzero components.

Suppose $x$ is not extreme, then by definition there exist $u, v \in S,\ u \neq v$ and $\lambda \in ]0,1[$ such that

$$x = \lambda u + (1-\lambda)v \in S.$$

Since $x$ is optimal and we want to maximize the objective function, then

$$c^T u \leq c^T x,\ c^T v \leq c^T x.$$

But also

$$c^T x = \lambda c^T u + (1-\lambda)c^T v \leq \lambda c^T x + (1-\lambda)c^T x = c^T x$$

which forces the inequality to be an equality and since $\lambda \in ]0,1[$, $\lambda(c^T x - c^T u) + (1-\lambda)(c^T x - c^T v) = 0$ means that $c^T x = c^T u = c^T v$.

Now consider the line

$$x(\epsilon) = x + \epsilon(u - v),\ \epsilon \in \mathbb{R}.$$

We start by showing in (a) that the vector $x(\epsilon)$ satisfies the constraints defined by $A$ for all $\epsilon$, in (b) that it has the same objective function as $x$ for all $\epsilon$, and in (c),(d) that its coefficients are non-negative for values of $\epsilon$ around 0:

(a) $Ax = Au = Av = b$ since $x, u, v$ all are in the feasible region, thus $Ax(\epsilon) = Ax + \epsilon(Au - Av) = b$ for all $\epsilon$.

(b) $c^T x(\epsilon) = c^T x + \epsilon(c^T u - c^T v) = c^T x$ for all $\epsilon$, since we showed above that $c^T x = c^T u = c^T v$.

(c) If $x_i = 0$, since $x_i = \lambda u_i + (1-\lambda)v_i$ with $u_i, v_i \geq 0$, we must have $0 = \lambda u_i + (1-\lambda)v_i$ and thus $u_i = v_i = 0$. So $x(\epsilon)_i = x_i + \epsilon(u_i - v_i) = x_i = 0$.

(d) If $x_i > 0$, then $x(0) = x$ and $x(0)_i = x_i > 0$. Also, by continuity of $x(\epsilon)_i$ in $\epsilon$, $x(\epsilon)_i$ will remain positive for a suitable range of values of $\epsilon$ around 0.

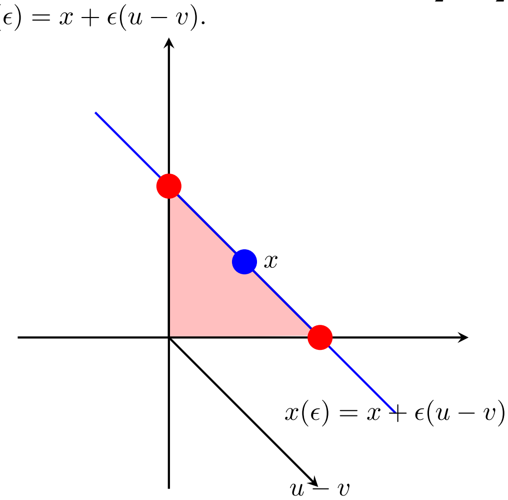
So we just showed that if $x$ is not extreme, then it is on the line $x(\epsilon)$, and every point on this line satisfies the constraints defined by $A$, and has the same optimal objective function $c^T x$, furthermore, $x(\epsilon)_i > 0$ for values of $\epsilon$ around 0. Now invoking again the continuity of $x(\epsilon)_i$ in $\epsilon$, we can increase $\epsilon$ from 0, in a positive or negative direction as appropriate (depending on the sign of $u - v$), until at least one extra component of $x(\epsilon)$ becomes 0 (since we start at $x_i > 0$ along a line, we are in the feasible region, and then need to go through 0 before getting something negative). This gives an optimal solution with fewer nonzero components than $x$, a contradiction, so $x$ must be extreme.

□

**Example 4.7.** Consider the linear program:

$$\begin{array}{rl} \max & x_1 + x_2 \\ s.t. & x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{array}$$

where $c^T = (2, 1)$ and $A = (1, 1)$. The feasible region $S$ is shown below. The point $x = (0.5, 0.5)$ is optimal, but not extreme. We can find two vectors $u = (1, 0), v = (0, 1) \in S$ such that $x = \frac{1}{2}u + \frac{1}{2}v$ ($\lambda = \frac{1}{2}$). Consider then the line $x(\epsilon) = x + \epsilon(u - v)$.




**Definition 4.7.** Given the $m \times n$ matrix $A$ which we assumed is of rank $m$, select $m$ linearly independent columns, whose indices are put in a set $B$. Then we can solve

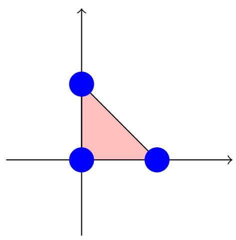
$$A_B x_B = b$$

by inverting the matrix $A_B$ created by selecting the columns of $A$ in $B$, to find an $m$-dimensional vector $x_B$, which contains the coefficients of $x$ whose indices are in $B$. Then set the coefficients of $x$ whose indices are not in $B$ to be zero. Then $x$ is called a *basic solution.* A basic solution satisfying $x \geq 0$ is called a *basic feasible solution (BFS).*

**Example 4.8.** In Example 4.1, we considered the linear program:

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ s.t. \quad & x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

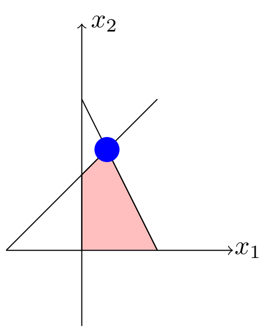
whose maximum is reached at $(1, 0)$.



Using a slack variable $s_1$, we can write the constraint $x_1 + x_2 \leq 1$ as $x_1 + x_2 + s_1 = 1$ for $s_1 \geq 0$ so $A = [1, 1, 1]$ and $b = 1$. We can have a single linearly independent column, so if we choose $B = \{1\}$, we get $A_B x_B = x_B = 1$ and the basic solution $[1, 0, 0]$, if we choose the second column, $B = \{2\}$, $A_B x_B = x_B = 1$ and we get the basic solution $[0, 1, 0]$ and if we choose the third column, $A_B x_B = x_B = 1$, we obtain the basic solution $[0, 0, 1]$. All the basic solutions are feasible.

**Example 4.9.** Consider the linear program:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & -x_1 + x_2 \leq 1 \\ & 2x_1 + x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$




Using slack variables $s_1, s_2$, we can write the constraints so

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

If we choose $B = \{1, 2\}$, we get

$$A_B x_B = \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Then

$$A_B^{-1} = \frac{1}{-3} \begin{bmatrix} 1 & -1 \\ -2 & -1 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{-3} \begin{bmatrix} 1 & -1 \\ -2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 4/3 \end{bmatrix}$$

and $[1/3, 4/3, 0, 0]$ is a basic feasible solution.

**Theorem 4.3.** *We have that $x$ is an extreme point of $S = \{x,\ Ax = b,\ x \geq 0\}$ if and only if $x$ is a basic feasible solution (BFS).*

*Proof.* Let $I = \{i,\ x_i > 0\}$.

($\Leftarrow$) Suppose that $x$ is a basic feasible solution, and write $x = \lambda u + (1-\lambda)v$ for $u, v \in S$, $\lambda \in ]0, 1[$. To show that $x$ is extreme, we need to show that $u = v$.

(a) If $i \notin I$, then $x_i = 0$ (since $x$ is a BFS, we cannot have $x_i < 0$), which implies $x_i = \lambda u_i + (1-\lambda)v_i = 0$ and since $u_i, v_i \geq 0$ (recall that $u, v \in S$), it must be that $u_i = v_i = 0$, so we know that $u$ and $v$ coincide on indices not in $I$.

(b) Since $Au = Av = b$, we have $A(u - v) = 0$. For $A_i$ the $i$th column of $A$, this is equivalent to say that

$$\sum_{i=1}^{n} (u_i - v_i) A_i = 0 \Rightarrow \sum_{i \in I} (u_i - v_i) A_i = 0$$

since by (a), $u_i - v_i = 0$ for all $i \notin I$. But $x$ is a BFS, this means that the $x_i$ which are zero may be coming either from solving the system $A_B x_B = b$ for some choice $B$ of indices, or by being coefficients whose index is not in $B$, but the $x_i$ which are not zero are necessarily coming from solving the system $A_B x_B = b$. This implies $u_i - v_i = 0$ for all $i \in I$ since the columns $A_i$ of $A_B$ are linearly independent.

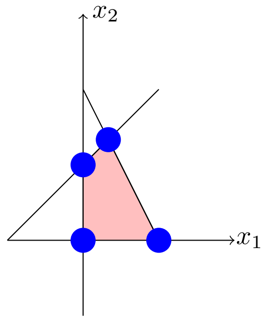
Hence $u = v$ and $x$ is an extreme point.

($\Rightarrow$) Suppose that $x$ is not a BFS, that is, $\{A_i,\ i \in I\}$ are linearly dependent. Then there exists $u \neq 0$ with $u_i = 0$ for $i \notin I$ such that $Au = 0$. For small enough $\epsilon$, $x \pm \epsilon u$ are feasible since

- $A(x \pm \epsilon u) = Ax \pm \epsilon Au = Ax = b$ using that $Au = 0$ and $x \in S$,
- $x \pm \epsilon u \geq 0$ since $x \geq 0$, and when $x_i = 0$, then $i \notin I$ and $u_i = 0$ for $i \notin I$, while when $x_i > 0$, we take $\epsilon$ small enough,

and $x = \frac{1}{2}(x + \epsilon u) + \frac{1}{2}(x - \epsilon u)$, so $x$ is not extreme. $\square$

**Example 4.10.** In Example 4.9, we already saw that $B = \{1, 2\}$ gives the point $[1/3, 4/3, 0, 0]$. By choosing $B = \{1, 3\}$, we get $[1, 0, 2, 0]$, by choosing $B = \{2, 4\}$, we get $[0, 1, 0, 1]$, and for $B = \{3, 4\}$, we get $[0, 0, 1, 2]$. The other choices of $B$ are not feasible. We thus get points $(x_1, x_2) \in \{(1/3, 4/3), (1, 0), (0, 1), (0, 0)\}$.




**Corollary 4.4.** *If there is an optimal solution, then there is an optimal BFS.*

*Proof.* By Theorem 4.2, we know that if an LP has an optimal solution, then it has an optimal solution at an extreme point of the feasible set. Then by Theorem 4.3, we have that if $x$ is an extreme point of the feasible set $S = \{x,\ Ax = b,\ x \geq 0\}$, then $x$ is a basic feasible solution (BFS). $\square$

In words, we have reduced our search space considerably: we started by trying to find an optimal solution for our optimization problem by looking at all points of our feasible region, while it turns out that it is enough to look at basic feasible solutions (which are extreme points of the feasible region). We also have an algorithm to do so, based on the definition of BFS. Sometimes the set $B$ is called *a basis*.

1. Choose $n - m$ of the variables to be 0 ($x_i = 0$ for $i \notin B$). They are called the *non-basic variables*. We may group them into the vector $x_N$ where $N = \{1, \dots, n\} \backslash B$.

2. Look at the remaining $m$ columns $\{A_i,\ i \in B\}$. Are they linearly independent? if so, we have an invertible $m \times m$ matrix $A_B$ and we can solve to find $x_B$ and thus $x$. We call these $x_i$ (those in $x_B$), the *basic variables*.

If we try all possible choices of $n - m$ variables, we get at most $\binom{n}{m}$ of them. This is actually a bad algorithm... First of all, $\binom{n}{m}$ grows quickly when $n$ and $m$ grow, e.g. $\binom{20}{11} = 167960$ and solving the $m \times m$ system of equations to find $x_m$ also costs a Gaussian elimination.

A classical method that provides a better alternative to the above algorithm is the so-called *Simplex Algorithm*.

## 4.2 The Simplex Algorithm

We argued above that trying all the possible basic feasible solutions is too expensive. The idea of the Simplex Algorithm is to start from one BFS (an extreme

Figure 4.1: G.B. Dantzig (1914-2005) proposed the Simplex Algorithm in 1947, on the above picture, he is awarded the National Medal of Science.

point of the feasible region), and then to try out an *adjacent* (or *neighbouring*) BFS in such a way that we improve the value of the objective function. By adjacent, we mean that the two BFS differ by exactly one basic (or non-basic) variable. This would lead to an algorithm of the following form:

**Algorithm 10** General Simplex Algorithm

**Input:** an LP in standard form $\max c^T x$, such that $Ax = b,\ x \geq 0$.
**Output:** a vector $x^*$ that maximizes the objective function $c^T x$.

```
Start with an initial BFS.
while (the current BFS is not optimal) do
    Move to an improved adjacent BFS.
return x* =BFS;
```

This needs a lot of clarifications.

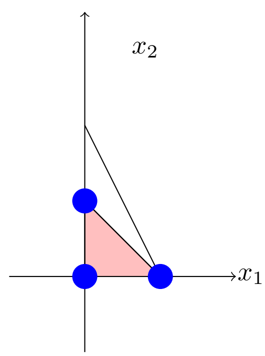
**Non-Degeneracy.** To start with, we are relying on the fact that a BFS corresponds to extreme points of the feasible region. We need to make sure that when we are moving from one BFS to another BFS, we actually go from one extreme point to another extreme point, the risk being that several BFS correspond to the same extreme point, and we could get stuck trying to improve a BFS which actually remains the same extreme point: this results in *stalling* if we eventually move to another solution, or worse, *cycling*, if we return to a tried degenerate BFS.

**Example 4.11.** Consider the linear program:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 + x_2 \leq 1 \\ & 2x_1 + x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

thus, with slack variables $s_1, s_2$, we get

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$




We thus have the following basic feasible solutions.

$$\begin{array}{ll} B = \{1,2\} & [1,0,0,0] \\ B = \{1,3\} & [1,0,0,0] \\ B = \{1,4\} & [1,0,0,0] \\ B = \{2,4\} & [0,1,0,1] \\ B = \{3,4\} & [0,0,1,2] \end{array}$$

**Definition 4.8.** We say that a basic feasible solution is *degenerate* if there exists at least one basic variable which is 0. It is called non-degenerate otherwise. An LP is non-degenerate if every basic feasible solution is non-degenerate.

Indeed, in order to compute $x$, we are given a choice $B$ of columns of $A$ which are linearly independent, and

$$A_B x_B = b.$$

Suppose that we solve for $x_B$, and find that the $i$th coefficient of $x_B$ is zero. Then this means that the $i$th column of $A_B$ contributes 0 to obtain $b$, therefore it can be replaced by a column of $A_N$ such that the resulting matrix $\tilde{A}_B$ remains full rank. The $i$th column of $A_B$ then gets moved to form a new matrix $\tilde{A}_N$, which does not interfere in the computation of the extreme point.

This is exactly what happened in our previous example. Once the choice of $B = \{1,2\}$ gives a basic feasible solution where the basic variable $x_2 = 0$, then the second column of $A_B$ where column 2 of $A$ is could be replaced by the column 3 and 4 of $A$ to create the same extreme point:

$$\begin{array}{ll} B=\{1,2\} & [1,0,0,0] \\ B=\{1,3\} & [1,0,0,0] \\ B=\{1,4\} & [1,0,0,0] \end{array}$$

We thus need non-degenerate BFS to look for improved BFS in our algorithm.

**Pivoting.** Once we have a non-degenerate BFS, we need to compute an adjacent BFS. We recall that two BFS are adjacent if they have $m-1$ basic variables in common (or equivalently, they differ by exactly one basic variable). Given a BFS, an adjacent BFS can be reached by increasing one non-basic variable from zero to positive, and decreasing one basic variable from positive to zero. This process is called *pivoting*. As a result, one non-basic variable "enters" $B$ (that is, its index is put in $B$), while one basic variable "leaves" $B$.

Formally, suppose that we have a BFS $x$, and let $x_q$ denote the non-basic variable of $x$ which we would like to increase by a coefficient of $\lambda \geq 0$ (the other non-basic variables are kept to 0). Then $x_N$ will be updated to

$$x_N + \lambda \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

where $\lambda$ multiplies a vector containing only zeroes, but for a 1 in the position of $x_q$. To simplify the notation, let us suppose that $x_q$ is the $q$th non-basic variable, and write $e_q$ for the vector with only zero coefficients but a 1 in the $q$th component (we could alternatively index $x_q$ with respect to its position in $A$ instead of $A_N$).

We need to decrease correspondingly a basic variable. Since $A_B x_B + A_N x_N = b$, we have that

$$x_B = A_B^{-1}(b - A_N x_N).$$

When $x_q$ increases by $\lambda$, we just saw above that $x_N$ gets updated to $x_N + \lambda e_q$, and by similarly updating $x_N$ in $A_N x_N$, we get

$$A_N(x_N + \lambda e_q)$$

and

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} A_B^{-1}(b - A_N x_N) \\ x_N \end{bmatrix}$$

gets updated to

$$\tilde{x} = \begin{bmatrix} A_B^{-1}b - A_B^{-1}A_N\,(x_N + \lambda e_q) \\ x_N + \lambda e_q \end{bmatrix}$$

and in conclusion

$$\tilde{x} = x + \lambda \begin{bmatrix} -A_B^{-1} A_N e_q \\ e_q \end{bmatrix}.$$

Write $(A_N)_q$ for the $q$th column of $A_N$. Then

$$\tilde{x} = x + \lambda \begin{bmatrix} -A_B^{-1} (A_N)_q \\ e_q \end{bmatrix}.$$

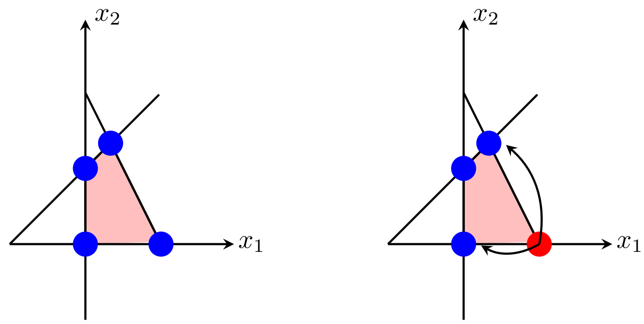
It is clear that multiplying $\tilde{x}$ by $A = [A_B, A_N]$ gives $A\tilde{x} = Ax = b$. For the non-degenerate case, $x_B > 0$, thus for $\lambda \geq 0$ small enough, $\tilde{x} \geq 0$. Note that this is not true for the degenerate case, if some $x_i = 0$, no matter how small $\lambda$ is, the negative sign in $-A_B^{-1} A_N$ could create a negative coefficient. This confirms that there exist choices of $\lambda \geq 0$ such that the pivoting operation sent one BFS to a feasible solution, however do note that we still have to discuss the actual choice of $\lambda$.

**Example 4.12.** Let us continue Example 4.9, for which

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

and basic feasible solutions are:

$$\begin{array}{ll} B = \{1, 2\} & [1/3, 4/3, 0, 0] \\ B = \{1, 3\} & [1, 0, 2, 0] \\ B = \{2, 4\} & [0, 1, 0, 1] \\ B = \{3, 4\} & [0, 0, 1, 2] \end{array}$$




Suppose we start with the BFS $[1, 0, 2, 0]$. Then $x_B, x_N$ satisfy

$$x_B = A_B^{-1}(b - A_N x_N) = \begin{bmatrix} 0 & 1/2 \\ 1 & 1/2 \end{bmatrix} \left( \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ s_2 \end{bmatrix} \right).$$

We want to update $x_N$, so we can update $x_2$ or $s_2$.

- If we update $s_2$ while keeping $x_2 = 0$, the constraint $2x_1 + x_2 + s_2 = 2$ tells us that $s_2$ can be increased to $s_2 = 2$, which in turn sends $x_1$ to 0. The constraint $-x_1 + x_2 + s_1 = 1$ with $x_1 = x_2 = 0$ means that $s_1$ is updated to 1. Replace $s_2 = 2$ and $x_2 = 0$ in the above equation gives $x_1 = 0$ and $s_1 = 1$ as desired. Thus this instance of pivoting sends the BFS $[1, 0, 2, 0]$ to $[0, 0, 1, 2]$.

- If we update $x_2$ while keeping $s_2 = 0$, the constraint $2x_1 + x_2 + s_2 = 2$ tells us that $2x_1 + x_2 = 2$, but while we could increase $x_2$ to 2 by sending $x_1$ to 0, this would violate the constraint $-x_1 + x_2 + s_1 = 1$. Replacing $x_2 = 2 - 2x_1$ in this latter constraint gives $-3x_1 + s_1 = -1$ and we can send $s_1$ to 0, $x_1$ to $1/3$, and $x_2$ to $4/3$. Replace $x_2 = 4/3$ and $s_2 = 0$ in the above question gives $x_1 = 1/3$ and $s_1 = 0$ as desired. Thus this instance of pivoting sends the BFS $[1, 0, 2, 0]$ to $[1/3, 4/3, 0, 0]$.

Alternatively, for the BFS $x = [1, 0, 2, 0]$, use the formula

$$\tilde{x} = \begin{bmatrix} x_B \\ x_N \end{bmatrix} + \lambda \begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix}$$

for $q = 1$ ($x_2$) and $\lambda = 4/3$ to get

$$\tilde{x} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} + \frac{4}{3} \begin{bmatrix} -\begin{bmatrix} 0 & 1/2 \\ 1 & 1/2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 0 \\ 4/3 \\ 0 \end{bmatrix}$$

and for $q = 2$ ($s_2$) and $\lambda = 2$:

$$\tilde{x} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} -\begin{bmatrix} 0 & 1/2 \\ 1 & 1/2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \end{bmatrix}$$

which gives the desired result.

We may want to look at the geometrical interpretation of the above formula: $x$ is an extreme point, we add the vector $\begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix}$ to it, so this vector gives the direction in which we move away from $x$, and $\lambda$ tells us how far we move away from $x$ along the given direction. We notice that we move while remaining in the feasible set.

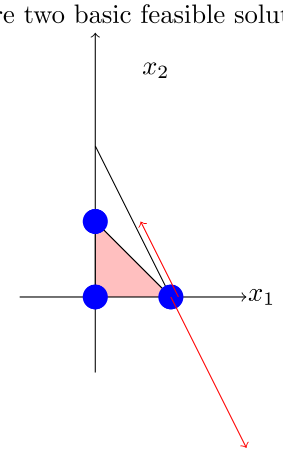
**Example 4.13.** Consider the linear program

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 + x_2 \leq 1 \\ & 2x_1 + x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

of Example 4.11, for which we saw that

$$\begin{array}{ll} B = \{1, 2\} & [1, 0, 0, 0] \\ B = \{1, 3\} & [1, 0, 0, 0] \end{array}$$

are two basic feasible solutions corresponding to the extreme point $[1, 0, 0, 0]$.




Let us compute the vector $\begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix}$ that gives the direction in the pivoting process, for $B = \{1, 2\}, q = 1$:

$$\begin{bmatrix} -A_B^{-1}(A_N)_1 \\ e_1 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 1 \\ 0 \end{bmatrix}$$

and $B = \{1, 3\}, q = 1$:

$$\begin{bmatrix} -A_B^{-1}(A_N)_1 \\ e_1 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -1/2 \\ 1 \\ 0 \end{bmatrix}$$

which since $B = \{1, 3\}$, corresponds to $[-1/2, 1, -1/2, 0]$.

This illustrates the problem that with degenerate BFS, we have no guarantee to remain within the feasible region.

**Reduced Cost.** We now know (or rather, almost know, we are still left to figure out how to compute $\lambda$) how to move from one BFS (which uniquely determines an extreme point) to an adjacent BFS. We next need to discuss how to look for an adjacent BFS which improves the objective function. For that, we observe how it changes from $x$ to $\tilde{x}$:

$$\begin{aligned} c^T\tilde{x} &= c^T\left(x + \lambda \begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix}\right) \\ &= c^T x + \lambda[c_B^T, c_N^T]\begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix} \\ &= c^T x + \lambda((c_N^T)_q - c_B^T A_B^{-1}(A_N)_q). \end{aligned}$$

**Definition 4.9.** The quantity

$$r_q = (c_N^T)_q - c_B^T A_B^{-1}(A_N)_q$$

is called a *reduced cost* with respect to the non-basic variable $x_q$.

If $r_q < 0$, then $c^T\tilde{x} = c^T x + \lambda r_q \leq c^T x$, which improves the cost function if it is a minimization, or in our case, where we chose a maximization of the cost function, we want $r_q > 0$.

**Algorithm Termination.** Since the reduced cost characterizes the improvement in the cost function by moving from one BFS to an adjacent one, we expect that once the reduced cost cannot improve anymore, the algorithm terminates, and we found an optimal solution. This is under the assumption that BFS are not degenerate.

**Theorem 4.5.** *Given a basic feasible solution $x^*$ with respect to a given $B$, that is $x^* = \begin{bmatrix} A_B^{-1}b \\ 0_{n-m} \end{bmatrix}$, if $r_q \leq 0$ for all non-basic variables $x_q^*$ (we assume the objective function is a maximization), then $x^*$ is optimal.*

*Proof.* Consider an arbitrary other feasible solution $x$, that is $x$ is such that $Ax = b$ and $x \geq 0$. Write $x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$. We have

$$\begin{aligned} \begin{bmatrix} A_B & A_N \\ 0_{(n-m)\times m} & I_{n-m} \end{bmatrix} (x - x^*) &= \begin{bmatrix} A_B & A_N \\ 0_{(n-m)\times m} & I_{n-m} \end{bmatrix} \begin{bmatrix} x_B - A_B^{-1}b \\ x_N \end{bmatrix} \\ &= \begin{bmatrix} A_B x_B - b + A_N x_N \\ x_N \end{bmatrix} = \begin{bmatrix} 0_m \\ x_N \end{bmatrix} \end{aligned}$$

since $Ax = [A_B, A_N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b$. Thus

$$x - x^* = \begin{bmatrix} A_B & A_N \\ 0_{(n-m)\times m} & I_{n-m} \end{bmatrix}^{-1} \begin{bmatrix} 0_m \\ x_N \end{bmatrix} = \begin{bmatrix} A_B^{-1} & -A_B^{-1}A_N \\ 0_{(n-m)\times m} & I_{n-m} \end{bmatrix} \begin{bmatrix} 0_m \\ x_N \end{bmatrix} = \begin{bmatrix} -A_B^{-1}A_N \\ I_{n-m} \end{bmatrix} x_N$$

that is

$$x = x^* + \sum_{q\in N} x_q \begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix}.$$

We can now compare the objective function of both $x$ and $x^*$:

$$\begin{aligned} c^T x &= c^T x^* + \sum_{q\in N} x_q [c_B^T, c_N^T] \begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix} \\ &= c^T x^* + \sum_{q\in N} x_q(-c_B^T A_B^{-1}(A_N)_q + c_N^T e_q) \\ &= c^T x^* + \sum_{q\in N} x_q((c_N^T)_q - c_B^T A_B^{-1}(A_N)_q) \\ &= c^T x^* + \sum_{q\in N} x_q r_q \end{aligned}$$

where $r_q \leq 0$ for all $q \in N$, showing as desired that $c^T x \leq c^T x^*$. $\square$

**Example 4.14.** In Example 4.12, where we want to maximize $x_1 + x_2$, we showed that we can start from the BFS $x = [1, 0, 2, 0]$, which for $B = \{1, 3\}$, is rewritten as $\begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix}$, and reach two new extreme points $\tilde{x}$ using

$$\tilde{x} = \begin{bmatrix} x_B \\ x_N \end{bmatrix} + \lambda \begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix}.$$

The corresponding objective function to maximize is thus

$$x_1 + x_2 = \underbrace{[1, 0, 1, 0]}_{c^T} \begin{bmatrix} x_1 \\ s_1 \\ x_2 \\ s_2 \end{bmatrix}.$$

We can compute the corresponding reduced cost $r_q = (c_N^T)_q - c_B^T A_B^{-1}(A_N)_q$. For $q = 1$ ($x_2$) and $\lambda = 4/3$:

$$\tilde{x} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} + \frac{4}{3} \begin{bmatrix} -\begin{bmatrix} 0 & 1/2 \\ 1 & 1/2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 0 \\ 4/3 \\ 0 \end{bmatrix}$$

thus

$$r_1 = (c_N^T)_1 - c_B^T A_B^{-1}(A_N)_1 = 1 - 1/2 = 1/2 > 0.$$

There is thus an improvement in the cost function by moving into this direction, given by $\lambda r_q$ and

$$c^T \tilde{x} = c^T x + \lambda r_q = 1 + \frac{4}{3}\frac{1}{2} = 5/3.$$

For $q = 2$ ($s_2$) and $\lambda = 2$:

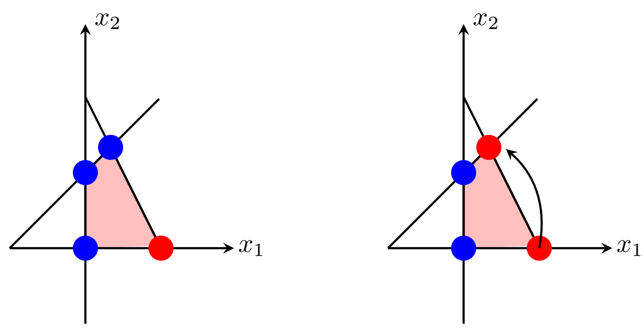
$$\tilde{x} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} -\begin{bmatrix} 0 & 1/2 \\ 1 & 1/2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \end{bmatrix}$$

thus

$$r_2 = (c_N^T)_2 - c_B^T A_B^{-1}(A_N)_2 = 0 - 1/2 = -1/2 \leq 0,$$

and there is thus no improvement in the cost function by moving into this direction. In fact, the objective function decreases:

$$c^T \tilde{x} = c^T x + \lambda r_2 = 1 + 2\frac{-1}{2} = 0.$$




From the BFS $[1, 0, 2, 0]$, we thus go the BFS $[1/4, 4/3, 0, 0]$. We can check by computing the reduced costs (see Exercise 36) that this solution is optimal

**Choice of $\lambda$.** We saw that given a BFS $x$, we can compute a new feasible solution $\tilde{x}$ given by

$$\tilde{x} = \begin{bmatrix} x_B \\ x_N \end{bmatrix} + \lambda \begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix},$$

where we will denote by $d_q$ the "direction" vector $\begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix}$ and we next determine $\lambda > 0$ (if the BFS is not degenerate which we assume, $\lambda > 0$ always exists). We choose $q$ (among the possible non-basic variables) such that the reduced cost $r_q > 0$ to guarantee that the cost function increases.

The choice of $\lambda$ tells us "how far" we will go away from the BFS $x$ in the direction given by $d_q$.

If $d_q \geq 0$, then $\tilde{x} = x + \lambda d_q \geq 0$ for any choice of $\lambda$, so $\lambda$ can be chosen arbitrarily big. Then

$$c^T \tilde{x} = c^T x + \lambda c^T d_q = c^T x + \lambda r_q$$

becomes arbitrarily big and the LP is unbounded.

Thus we may assume that $d_q$ has at least one negative component $(d_q)_i$ (and $i$ must be in $B$ since indices in $N$ correspond to $e_q$). For the $i$th component $\tilde{x}_i$ of $\tilde{x}$ which should be non-negative so $\tilde{x}$ remains feasible, we have

$$\tilde{x}_i = x_i + \lambda (d_q)_i \geq 0 \Rightarrow \lambda \leq \frac{x_i}{-(d_q)_i}.$$

and since there may be several negative components in $d_q$, we choose

$$\lambda = \min_{i \in B} \left\{ \frac{x_i}{-(d_q)_i},\ (d_q)_i < 0 \right\}.$$

Then for the index $i$ that achieves the minimum, we get

$$\tilde{x}_i = x_i - \frac{x_i}{(d_q)_i}(d_q)_i$$

and the corresponding basic variable $x_i$ is set to 0. Thus this choice of $\lambda$ takes us from one BFS to another BFS. We emphasize again that if the BFS were degenerate, we could have $x_i = 0$, this would achieve the above minimum, $\lambda$ would be set to zero, and we would remain at the same extreme point.

We can now revisit Algorithm 10.

---

**Algorithm 11** Simplex Algorithm

**Input:** an LP in standard form $\max c^T x$, such that $Ax = b,\ x \geq 0$.

**Output:** a vector $x^*$ that maximizes the objective function $c^T x$ (or that the LP is unbounded).

```
Start with an initial BFS x with basis B and N = {1...n}\B;
For q ∈ N, compute r_q = c^T d_q = c_q − c_B^T A_B^{-1} (A_N)_q.
while (there is a q such that r_q > 0) do
    if (d_q ≥ 0) then
        the LP is unbounded, stop.
    else
        Compute λ = min_{i∈B} { x_i / −(d_q)_i ,  (d_q)_i < 0 }.
        x ← x + λ d_q.
        Update B and N.
```

---

**An Initial Basic Feasible Solution.** The first step of the algorithm consists of finding one BFS. Remark that if the matrix $A$ contains $I_m$ as an $m \times m$ submatrix, and $b \geq 0$, then there is an obvious BFS $x'$: choose $A_B = I_m$, then $x_B = A_B^{-1} b = b$ and

$$x' = \begin{bmatrix} 0_{n-m} \\ b \end{bmatrix} \Rightarrow Ax' = [A_N,\ I_m] \begin{bmatrix} 0_{n-m} \\ b \end{bmatrix} = b.$$

Now an $m \times m$ submatrix which is the identity will be present if the LP is such that we need $m$ slack variables to transform the $m$ inequalities (of the form $\leq$) defining the constraints into $m$ equalities.

**Tableau.** The Simplex Algorithm can be computed by writing a linear program in the form of a *tableau*. Suppose we have an initial BFS given by setting the slack variables to be $s = x_B = b$, and $x = x_N = 0$, and create an array of the form:

$$\begin{array}{cc|c} A_N & I_m & b \\ \hline c^T & 0_m & -0 \end{array}$$

representing the linear system of equations

$$\begin{bmatrix} 0 & A_N & I_m \\ -1 & c^T & 0_m \end{bmatrix} \begin{bmatrix} f \\ x \\ s \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \iff s = b - A_N x, \ \ -f + c^T x = 0$$

so the first $n$ columns of the tableau correspond to $x_1, \ldots, x_n$, while the next $n + m$ columns correspond to $s_1, \ldots, s_m$. When $x = 0$ and $s = b$, $c^T x = 0$, and $f = -0$, this is the value of the objective function in the BFS $x = 0, s = b$, shown in the right bottom of the tableau. The BFS can be read from the tableau, since the variables corresponding to the columns of $A_N$ are 0, and we read $x_B = s = b$, $x_N = 0$.

The next step of the algorithm is to compute the reduced costs $r_q = c_q - c_B^T A_B^{-1}(A_N)_q$, where $A_B = I_m$, but since $B$ contains the indices of the slack variables, $c_B = 0_m$, and $r_q = c_q$. The condition "there is a $q$ such that $r_q > 0$" then simplifies to check whether we have $c_q > 0$ ($c_i = 0$ for all $i \in B$), so we pick a column $q$ of the tableau (called *pivot column*) corresponding to a coefficient $c_q > 0$. We then compute

$$\begin{aligned} d_q &= \begin{bmatrix} -A_B^{-1}(A_N)_q \\ e_q \end{bmatrix} = \begin{bmatrix} -(A_N)_q \\ e_q \end{bmatrix}, \\ \lambda &= \min_{i \in B} \left\{ \frac{x_i}{-(d_q)_i},\ (d_q)_i < 0 \right\} = \min_{i \in B} \left\{ \frac{b_i}{a_i},\ a_i \in (A_N)_q \right\} \end{aligned}$$

since $A_B = I_m$, $x_B = b$. Note that if $d_q \geq 0$, the LP is unbounded. So there should be coefficients in $-(A_N)_q$ which are negative, that is, coefficients in $(A_N)_q$ which are positive. This means that given the choice of the column $q$, we look at the coefficients $a_{iq} > 0$, and choose $i$ that minimizes $b_i/a_{iq}$, this gives the limit on how much we can increase $x_q$, and $i$ is the *pivot row*. The last step is the update of $x, B, N$, and $x$ is updated to $\tilde{x}$:

$$\tilde{x} = \begin{bmatrix} b \\ 0_{n-m} \end{bmatrix} + \frac{b_i}{a_{iq}} \begin{bmatrix} -(A_N)_q \\ e_q \end{bmatrix}$$

so the $i$th row of $\tilde{x}$ becomes 0 as it should be, the $q$th non-basic variable is increased by $b_i/a_i$, thus $i$ goes form $B$ to $N$, while $q$ goes from $N$ to $B$. We denote by $\tilde{B}$ the new basis, and by $\tilde{N}$ the other indices.

Next we wish to read the new basic variables $x_{\tilde{B}}$ from the tableau, and how the objective function changes as a function of $\tilde{B}$. The current tableau allows us to read $[A_N, I_m] \begin{bmatrix} x \\ s \end{bmatrix} = b$, so given the new $\tilde{B}$, we get a corresponding matrix $A_{\tilde{B}}$, we multiply the above equation by $A_{\tilde{B}}^{-1}$ to get

$$A_{\tilde{B}}^{-1}[A_N, I_m] \begin{bmatrix} x \\ s \end{bmatrix} = A_{\tilde{B}}^{-1} b.$$

Now some columns of $[A_N, I_m]$ correspond to the matrix $A_{\tilde{B}}$ and to the vector $x_{\tilde{B}}$. From them, we get $x_{\tilde{B}}$. The other columns of $[A_N, I_m]$ correspond to $A_{\tilde{N}}$ and to the vector $x_{\tilde{N}}$, from them, we get $A_{\tilde{B}}^{-1} A_{\tilde{N}} x_{\tilde{N}}$, that is we can rewrite the above equation as

$$x_{\tilde{B}} + A_{\tilde{B}}^{-1} A_{\tilde{N}} x_{\tilde{N}} = A_{\tilde{B}}^{-1} b$$

where setting $x_{\tilde{N}} = 0$ gives $x_{\tilde{B}} = A_{\tilde{B}}^{-1} b$ as the current BFS. The cost $c^T x$ for the current BFS is given by $c_{\tilde{B}}^T A_{\tilde{B}}^{-1} b$. More generally, we evaluate $c^T x$ in $[x_{\tilde{B}}, x_{\tilde{N}}]^T$ to get

$$c^T x = c_{\tilde{B}}^T x_{\tilde{B}} + c_{\tilde{N}}^T x_{\tilde{N}} = c_{\tilde{B}}^T A_{\tilde{B}}^{-1} b + (c_{\tilde{N}}^T - c_{\tilde{B}}^T A_{\tilde{B}}^{-1} A_{\tilde{N}}) x_{\tilde{N}}.$$

The linear system of equations is updated by:

$$\begin{bmatrix} A_{\tilde{B}}^{-1} & 0 \\ -c_{\tilde{B}}^T A_{\tilde{B}}^{-1} & 1 \end{bmatrix} \begin{bmatrix} 0 & A_N & I_m \\ -1 & c^T & 0_m \end{bmatrix} \begin{bmatrix} f \\ x \\ s \end{bmatrix} = \begin{bmatrix} A_{\tilde{B}}^{-1} & 0 \\ -c_{\tilde{B}}^T A_{\tilde{B}}^{-1} & 1 \end{bmatrix} \begin{bmatrix} b \\ 0 \end{bmatrix} = \begin{bmatrix} A_{\tilde{B}}^{-1} b \\ -c_{\tilde{B}}^T A_{\tilde{B}}^{-1} b \end{bmatrix}.$$

We just computed the upper part, for the lower part, the same principle applies: $c_{\tilde{B}}^T A_{\tilde{B}}^{-1}$ multiplies $[A_N, I_m]$ so the columns corresponding to $\tilde{B}$ and the corresponding coefficients $x_{\tilde{B}}$ of $[x,s]^T$ will give $c_{\tilde{B}}^T x_{\tilde{B}}$, while the other columns will give $c_{\tilde{B}}^T A_{\tilde{B}}^{-1} A_{\tilde{B}} x_{\tilde{B}}$, and $-f - c_{\tilde{B}}^T x_{\tilde{B}} - c_{\tilde{B}}^T A_{\tilde{B}}^{-1} A_{\tilde{N}} x_{\tilde{N}} + c_{\tilde{B}}^T x_{\tilde{B}} + c_{\tilde{N}}^T x_{\tilde{N}} = -c_{\tilde{B}}^T A_{\tilde{B}}^{-1} b$. So

$$s = b - A_N x, \;\; f = c^T x$$

got updated to

$$x_{\tilde{B}} = A_{\tilde{B}}^{-1} b - A_{\tilde{B}}^{-1} A_{\tilde{N}} x_{\tilde{N}}, \;\; f = c^T x = c_{\tilde{B}}^T A_{\tilde{B}}^{-1} b + (c_{\tilde{N}}^T - c_{\tilde{B}}^T A_{\tilde{B}}^{-1} A_{\tilde{N}}) x_{\tilde{N}}. \quad (4.1)$$

The multiplication by $A_{\tilde{B}}^{-1}$ is done by a Gaussian elimination, in such a way that the column corresponding to the new basic variable gets a 1 in row $i$ and 0 elsewhere, elsewhere includes the objective function row:

- Multiply row $i$ of $A_N$ by $1/a_{iq}$, where $a_{iq}$ is the coefficient in the $i$th row and $q$th column of $A_N$.
- For row $i' \neq i$ of $A$, add $-a_{i'q}/a_{iq}$ times row $i$ to row $i'$, where $a_{i'q}$ is the coefficient in the $i'$th row and $q$th column of $A_N$.
- Add $-c_q/a_{iq}$ times row $i$ to the last (objective function) row.

For an explicit computation, the tableau from the initial stage gets updated to:

| $x_1$ column | | $x_q$ column | $s_i$ column | | |
|---|---|---|---|---|---|
| $\frac{a_{i1}}{a_{iq}}$ | $\ldots$ | $\frac{a_{iq}}{a_{iq}} = 1$ | $\frac{1}{a_{iq}}$ | $\frac{b_i}{a_{iq}}$ | row $i$ |
| $a_{i'1} - \frac{a_{i'q}}{a_{iq}} a_{i1}$ | $\ldots$ | $a_{i'q} - \frac{a_{i'q}}{a_{iq}} a_{iq} = 0$ | $0$ | $b_{i'} - \frac{a_{i'q}}{a_{iq}} b_i$ | row $i'$ |
| $c_1 - a_{i1} \frac{c_q}{a_{iq}}$ | $\ldots$ | $c_q - a_{iq} \frac{c_q}{a_{iq}} = 0$ | $-\frac{c_q}{a_{iq}}$ | $-0 - \frac{c_q}{a_{iq}} b_i$ | |

and the matrix $A_{\tilde{B}}$ at this stage is an identity matrix where the $i$th column got replaced by the column $[a_{1q}, \ldots, a_{iq}, \ldots, a_{mq}]^T$, so its inverse is also an identity matrix, where the $i$th column got replaced this time by the column $[-\frac{a_{1q}}{a_{iq}}, \ldots, \frac{1}{a_{iq}}, \ldots, -\frac{a_{mq}}{a_{iq}}]^T$.

The coefficients $c_{\tilde{N}}^T - c_{\tilde{B}}^T A_{\tilde{B}}^{-1} A_{\tilde{N}}$ appear in the last row of the tableau for the non-basic variables (we have zeroes for basic variables), $-c_{\tilde{B}}^T A_{\tilde{B}}^{-1} b$ in the last column of the last row, as a result of the multiplication $[-c_{\tilde{B}}^T A_{\tilde{B}}^{-1}, 1] \begin{bmatrix} A_N & I_m \\ c^T & 0_m \end{bmatrix} x = -c_{\tilde{B}}^T A_{\tilde{B}}^{-1} b$.

At this step of the tableau, the last column containing $-0$ gets updated to $-0 - \frac{b_i}{a_{iq}} c_q = -0 - \lambda r_q$, so we can read the updated value of the objective function by negating the coefficient contained in the last row last column. Also, $c_q$ becomes 0 for the new basic variable $x_q$, and 0 becomes $-\frac{c_q}{a_{iq}}$ for the new non-basic variable $s_i$. The last row of the tableau thus contains $c_j - a_{ij}\frac{c_q}{a_{iq}}$ for $j = 1, \ldots, n$, and 0 after that but for the term that corresponds to $s_i$, for which it is $-\frac{c_q}{a_{iq}}$ and

$$\sum_j \left(c_j - a_{ij}\frac{c_q}{a_{iq}}\right) x_j - s_i \frac{c_q}{a_{iq}} = \sum_j c_j x_j - \frac{c_q}{a_{iq}}\left(\sum_j a_{ij} x_j + s_i\right) = c^T x - \frac{c_q}{a_{iq}} b_i$$

by looking at the $i$th row of $[A_N, I_m]\begin{bmatrix} x \\ s \end{bmatrix}$.

We can now iterate the process (call $A_N, b, c$ the newly obtained matrix and vectors), which gives the simplex algorithm in tableau form.

---

**Algorithm 12** Simplex Algorithm

**Input:** an LP in standard form $\max c^T x$, such that $Ax = b,\ x \geq 0$.

**Output:** a vector $x^*$ that maximizes the objective function $c^T x$ (or that the LP is unbounded).

Start with an initial BFS $x$ with basis $B$ and $N = \{1 \ldots n\} \backslash B$;
Create the corresponding tableau.
**while** (there is a $q$ such that $c_q > 0$) **do**
  Choose the pivot column $q$.
  **if** ($a_{iq} \leq 0$ for all $i$) **then**
    the LP is unbounded, stop.
  Choose a pivot row $i$, that is among $a_{iq} > 0$, choose $i$ such that $b_i/a_{iq}$ is minimized.
  Multiply row $i$ by $1/a_{iq}$.
  For $i' \neq i$ add $-a_{i'q}/a_{iq}$ times row $i$ to row $i'$.
  Add $-c_q/a_{iq}$ times row $i$ to the objective function row.

---

**Example 4.15.** Consider the following LP:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 + 3x_2 \leq 9 \\ & 2x_1 + x_2 \leq 8 \\ & x_1, x_2 \geq 0 \end{aligned}$$

and introduce the slack variables $s_1, s_2$. A BFS is given by $[0, 0, 9, 8]$, and for this BFS, the objective function is taking the value 0. Using slack variables, we write the initial tableau

| $x_1$ | $x_2$ | $s_1$ | $s_2$ | $b$ |
|---|---|---|---|---|
| 1 | 3 | $\boxed{1}$ | $\boxed{0}$ | 9 |
| 2 | 1 | $\boxed{0}$ | $\boxed{1}$ | 8 |
| 1 | 1 | 0 | 0 | 0 |

We are looking for a column $q$ for which $c_q > 0$, pick $q = 1$. This corresponds to the non-basic variable $x_1$ which we want to increase. For choosing the row, compute $b_1/a_{11} = 9$ and $b_2/a_{21} = 8/2 = 4$, so the minimum is given by choosing the row $i = 2$.

You may want to keep in mind how this step of the algorithm relates to the constraints of the LP: since $x_1$ and $x_2$ are constrained by $x_1 + 3x_2 \leq 9$, $2x_1 + x_2 \leq 8$, the first equation says that $x_1$ could be at most 9 if $x_2 = 0$, the second equation says that $2x_1$ could be at most 8, that is $x_1$ could be at most $8/2 = 4$ if $x_2 = 0$. In order not to violate any constraint, we choose the smallest increment for $x_1$, which is $x_1 = 4$.

If the rows are called $\rho_1, \rho_2, \rho_3$, $\rho_2' = \rho_2/2$, $\rho_1' = \rho_1 - \rho_2'$, $\rho_3' = \rho_3 - \rho_2'$:

$$\begin{array}{cccc|c} x_1 & x_2 & s_1 & s_2 & b \\ \boxed{0} & 5/2 & \boxed{1} & -1/2 & 5 \\ \boxed{1} & 1/2 & \boxed{0} & 1/2 & 4 \\ \hline 0 & 1/2 & 0 & -1/2 & -4 \end{array}$$

The objective function increased from 0 to 4. The new BFS is $[4, 0, 5, 0]$. Let us also illustrate (4.1) on this example. The last row of the tableau is

$$\begin{aligned} 0 \cdot x_1 + \tfrac{1}{2}x_2 - \tfrac{1}{2}s_2 &= (1-1)x_1 + (1-\tfrac{1}{2})x_2 - \tfrac{1}{2}s_2 \\ &= x_1 + x_2 - \tfrac{1}{2}(2x_1 + x_2 + s_2) \\ &= x_1 + x_2 - 4 \end{aligned}$$

since $2x_1 + x_2 + s_2 = 8$. Thus we have that the objective function $c^T x = x_1 + x_2$ can be read from the last row of the tableau:

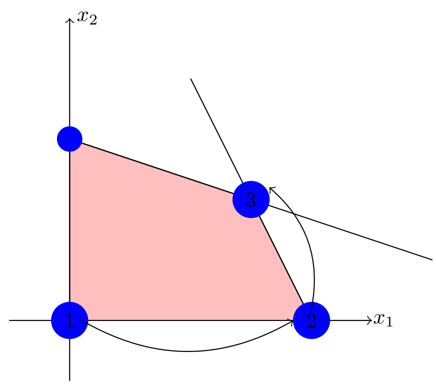
$$c^T x = \underbrace{(0 \cdot x_1 + \tfrac{1}{2}x_2 - \tfrac{1}{2}s_2)}_{\text{last row}} + \underbrace{4}_{\text{–last row, last column}} .$$

Next for the column pivot 2, we choose the pivot row to be 1, since $5/(5/2) = 2$ while $4/(1/2) = 8$. Then $\rho_1'' = (2/5)\rho_1'$, $\rho_2'' = \rho_2' - \rho_1''/2$, $\rho_3'' = \rho_3' - \rho_1'/2$:

$$\begin{array}{cccc|c} x_1 & x_2 & s_1 & s_2 & b \\ \boxed{0} & \boxed{1} & 2/5 & -1/5 & 2 \\ \boxed{1} & \boxed{0} & -1/5 & 3/5 & 3 \\ \hline 0 & 0 & -1/5 & -2/5 & -5 \end{array}$$

We see that $c_q \leq 0$ for all $q$, thus the algorithm stops. The objective function has value 5. The optimal solution is given by $x_1 = 3$, $x_2 = 2$.

Geometrically, the algorithm goes from one BFS to another as shown below: it starts at $(0, 0)$, then goes to $(4, 0)$, and then to $(3, 2)$.




**Artificial Variables.** We argued above that if the matrix $A$ contains $I_m$ as an $m \times m$ submatrix, and $b \geq 0$, then we have an immediate BFS. That $b \geq$ is not a restriction, one can always multiply the corresponding equality in the LP standard form so that $-b_i \leq 0$ becomes $b_i \geq 0$. Finding an $m \times m$ submatrix inside $A$ may not be easy, if we can cannot just take the columns corresponding to the slack variables. Of course, we can try some exhaustive search, pick $m$ columns of $A$ until we get a choice which is linearly independent, and then we inverse this $m \times m$ submatrix. Alternatively, we can introduce *artificial variables* $w_1, \ldots, w_m$ and solve the LP:

$$\begin{aligned} \min_{x,w} \quad & \textstyle\sum_{i=1}^m w_i \\ s.t. \quad & Ax + w = b \\ & x, w \geq 0. \end{aligned}$$

We can solve this LP using the Simplex Algorithm (and note the presence of the identity matrix in front of $w$).

- If the original problem $Ax = b$, $x \geq 0$, has a feasible solution, then the above LP has for optimal value 0 and $w = 0$. This optimal solution with $w = 0$ gives the desired BFS for $Ax = b$, $x \geq 0$.
- If the above LP has an optimal value which is strictly positive, then the original LP is not feasible.

This is called the *Two Phase Simplex Algorithm*. See Exercise 38 for an Example.

## 4.3 Duality

**Definition 4.10.** Given a *primal program*

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

where $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $A$ is an $m \times n$ matrix of rank $m$, $m \leq n$, the *dual* of $P$ is defined by

$$\begin{array}{rl} D : \min & y^T b \\ s.t. & y^T A \geq c^T \\ & y \geq 0 \end{array}$$

where $c \in \mathbb{R}^n$, $b, y \in \mathbb{R}^m$ and $A$ is an $m \times n$ matrix of rank $m$, $m \leq n$.

It is not hard to see that the dual of the dual is the primal (see Exercise 39).

**Theorem 4.6. (Weak Duality Theorem)** *If $x$ is feasible for $P$ and $y$ is feasible for $D$, then $c^T x \leq b^T y$.*

*Proof.* Since $y$ is feasible for $D$, $y^T A \geq c^T$, and since $x$ is feasible for $P$, $x \geq 0$ so

$$c^T x \leq (y^T A)x.$$

We flip the same argument. Since $x$ is feasible for $P$, $Ax \leq b$, and since $y$ is feasible for $D$, $y \geq 0$ so

$$y^T Ax \leq y^T b.$$

Hence

$$c^T x \leq (y^T A)x \leq b^T y.$$

□

**Corollary 4.7.** *If $y$ is a feasible solution for $D$, then $P$ is bounded. Similarly, if $x$ is a feasible solution for $P$, then $D$ is bounded.*

*Proof.* If $y$ is a feasible solution for $D$, then any feasible solution $x$ for $P$ satisfies $c^T x \leq b^T y$ so $b^T y$ is an upper bound for any $x$ feasible.

If $x$ is a feasible solution for $P$, then any feasible solution $y$ for $d$ satisfies $b^T y \geq c^T x$, so $c^T x$ is a lower bound for any $y$ feasible. □

**Corollary 4.8.** *If $x^*$ is feasible for $P$, and $y^*$ is feasible for $D$ and $c^T x^* = b^T y^*$, then $x^*$ is optimal for $P$ and $y^*$ is optimal for $D$.*

*Proof.* For all $x$ feasible for $P$, the Weak Duality theorem tells us that

$$c^T x \leq b^T y^* = c^T x^*.$$

This shows that $x^*$ is optimal (maximal) for $P$.

Similarly, for all $y$ feasible for $D$, the Weak Duality theorem tells us that

$$b^T y \geq c^T x^* = b^T y^*.$$

This shows that $y^*$ is optimal (minimal) for $D$. □

**Example 4.16.** Consider the primal program

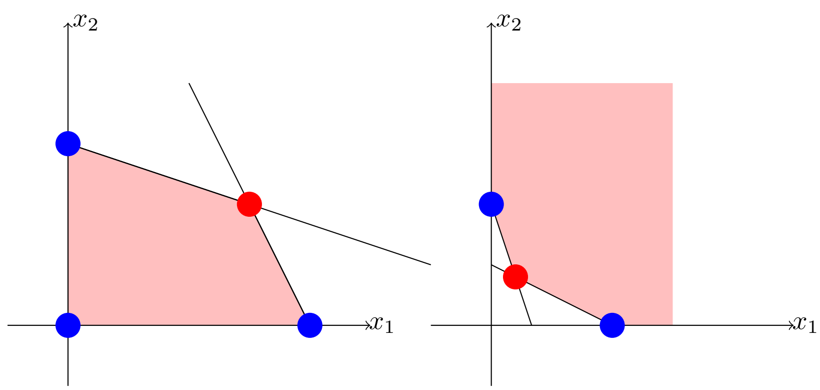
$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 + 3x_2 \leq 9 \\ & 2x_1 + x_2 \leq 8 \\ & x_1, x_2 \geq 0 \end{aligned}$$

with $b^T = [9, 8]$ and $c^T = [1, 1]$.

We compute its dual:

$$\begin{aligned} \max \quad & 9y_1 + 8y_2 = b^T y \\ s.t. \quad & y_1 + 2y_2 \geq 1 \\ & 3y_1 + y_2 \geq 1 \\ & y_1, y_2 \geq 0. \end{aligned}$$

The points $x^* = (3, 2)$ and $y^* = (1/5, 2/5)$ are both feasible, $c^T x^* = 5 = b^T y$ so both are optimal.




Weak duality states than any feasible solution of the dual gives an upper bound on any solution of the primal (and vice-versa). There is a stronger version stated next, that says that values of the optimal solutions for the primal and dual match.

**Theorem 4.9. (Strong Duality Theorem)** *If $P$ has an optimal solution $x^*$, then $D$ has an optimal solution $y^*$ such that $c^T x^* = b^T y^*$.*

*Proof.* Write the constraints of $P$ as

$$Ax + s = b, \; x, s \geq 0,$$

where $s$ is the vector of slack variables (an LP already in the form $Ax = b$ can be written with inequalities by noting that $Ax = b \iff Ax \leq b, \; -Ax \leq -b$).

Consider the bottom row in the final tableau of the Simplex Algorithm applied to $P$:

| $x$ columns | $s$ columns | $b$ column |
|---|---|---|
| $c_1^* \dots c_n^*$ | $-y_1^* \dots - y_m^*$ | $-f^*$ |

where for now $c_1^*, \dots, c_n^*, -y_1^* \dots - y_m^*$ are just some notations for the reduced costs that appear at this stage of the tableau computations. What we know is that $f^*$ is by construction of the tableau the optimal value of the objective function $c^T x$, and that all the reduced costs are negative or zero, since we assumed that this is the final tableau, thus:

$$c_j^* \leq 0 \text{ for all } j, \quad -y_i^* \leq 0 \text{ for all } i.$$

Recall (see (4.1)) that the last row of the tableau can be read as $c^T x = (c^*)^T x - (y^*)^T s + f^*$ where $s = b - Ax$, thus

$$c^T x = f^* - b^T y^* + ((c^*)^T + (y^*)^T A)x$$

and using $x = 0$ in this expression gives $f^* = b^T y^*$. But if this is the case, then

$$c^T x = ((c^*)^T + (y^*)^T A)x \Rightarrow c = c^* + A^T y^*. \tag{4.2}$$

Since $c_j^* \leq 0$ for all $j$, we get

$$A^T y^* \geq c.$$

This combined with $y_j^* \geq 0$ shows that $y^*$ is feasible for $D$, and $f^* = b^T y^*$ gives the objective function of $D$ at $y^*$, namely it is the optimal value of $P$. Using the Weak Duality Theorem, $y^*$ is optimal for $D$. □

**Example 4.17.** Consider the primal problem

$$\begin{array}{rc} \max & x_1 + x_2 \\ s.t & x_1 + 3x_2 + s_1 = 9 \\ & 2x_1 + x_2 + s_2 = 8 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{array}$$

whose dual problem is

$$\begin{array}{rc} \min & 9y_1 + 8y_2 \\ s.t & y_1 + 2y_2 \geq 1 \\ & 3y_1 + y_2 \geq 1 \\ & y_1, y_2 \geq 0 \end{array}$$

The initial tableau for the primal is:

| $x_1$ | $x_2$ | $s_1$ | $s_2$ | $b$ |
|---|---|---|---|---|
| 1 | 3 | $\boxed{1}$ | $\boxed{0}$ | 9 |
| 2 | 1 | $\boxed{0}$ | $\boxed{1}$ | 8 |
| 1 | 1 | 0 | 0 | 0 |

corresponding to the initial BFS $x = [0, 0, 9, 8]$.

We already computed in Example 4.15 that the final tableau is

$$\begin{array}{cccc|c} x_1 & x_2 & s_1 & s_2 & b \\ \boxed{0} & \boxed{1} & 2/5 & -1/5 & 2 \\ \boxed{1} & \boxed{0} & -1/5 & 3/5 & 3 \\ \hline 0 & 0 & \underbrace{-1/5}_{-y_1^*} & \underbrace{-2/5}_{-y_2^*} & -5 \end{array}$$

corresponding to the BFS $x^* = [3, 2, 0, 0]$, with value 5 for the objective function. According to the proof of the Strong Duality Theorem, the coefficients $y_1^*, y_2^*$ read in the last row of the tableau form an optimal solution for the dual. We can check that $9\frac{1}{5} + 8\frac{2}{5} = 5$, thus the primal has an optimal solution $x^* = [3, 2, 0, 0]$, and the dual has an optimal solution $y^* = [1/5, 2/5]$ such that both their objective functions take value 5 at their respective optimal solutions. This confirms that $y^* = [1/5, 2/5]$ is indeed an optimal solution for the dual.

We saw an LP may be feasible and bounded, feasible and unbounded, and infeasible. Thus for a primal and its dual, there are a priori 9 possibilities:

| | $P$ feasible bounded | $P$ feasible unbounded | $P$ infeasible |
|---|---|---|---|
| $D$ feasible bounded | ✓ | × | × |
| $D$ feasible unbounded | × | × | ✓ |
| $D$ infeasible | × | ✓ | ✓ |

- The Weak Duality Theorem says that if a primal and its dual are both feasible, then both are bounded feasible.
- The Strong Duality Theorem says if an LP has an optimal solution (thus feasible and bounded), then its dual cannot be infeasible.

The following theorem can be proven as a corollary of the Strong Duality Theorem.

**Theorem 4.10** (The Equilibrium Theorem)**.** *Let $x^*$ and $y^*$ be feasible solutions for a primal and its dual respectively. Then $x^*$ and $y^*$ are optimal if and only if*

*(1) $\sum_{j=1}^{n} a_{ij}x_j^* < b_i$ (or $(Ax^*)_i < b_i$) $\Rightarrow y_i^* = 0$ for all $i$,*

*(2) $\sum_{i=1}^{m} a_{ij}y_i^* > c_j$ (or $(y^*A)_j > c_j$) $\Rightarrow x_j^* = 0$ for all $j$.*

*Proof.* ($\Leftarrow$) (1) The sum $\sum_{i=1}^{m} y_i^* b_i$ contains terms for which $\sum_{j=1}^{n} a_{ij}x_j^* = b_i$, and terms for which $\sum_{j=1}^{n} a_{ij}x_j^* < b_i$ but in this case $y_i^* = 0$:

$$\sum_{i=1}^{m} y_i^* b_i = \sum_{i=1}^{m} y_i^* (\sum_{j=1}^{n} a_{ij}x_j^*) = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_j^* y_i^*.$$

(2) Repeating the above argument, namely that when $\sum_{i=1}^{m} a_{ij} y_i^* > c_j$, $x_j^* = 0$:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j^* y_i^* = \sum_{j=1}^{n} c_j x_j^*.$$

Putting both equalities together gives

$$\sum_{i=1}^{m} y_i^* b_i = \sum_{j=1}^{n} c_j x_j^* \iff b^T y^* = c^T x^*$$

and by Corollary 4.8, $x^*$ and $y^*$ are optimal.

($\Rightarrow$) We repeat the proof of the Weak Duality Theorem. Since $y^*$ is feasible, $(y^*)^T A \geq c^T$, and since $x^*$ is feasible, $x^* \geq 0$, thus

$$c^T x^* \leq (y^*)^T A x^*.$$

Since $x^*$ is feasible, $Ax \leq b$, and since $y^*$ is feasible, $y \geq 0$, so

$$(y^*)^T A x^* \leq (y^*)^T b \Rightarrow c^T x^* \leq (y^*)^T A x^* \leq b^T y^*,$$

or equivalently

$$\sum_{j=1}^{n} c_j x_j^* \leq \sum_{i=1}^{m} \sum_{j=1}^{n} y_i^* a_{ij} x_j^* \leq \sum_{i=1}^{m} y_i^* b_i.$$

But we know more, we know that $x^*$ and $y^*$ optimal, so we may invoke the Strong Duality Theorem to tell us that $c^T x^* = b^T y^*$ and in fact all the above inequalities and equalities.

The first inequality thus becomes

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

Since $x^*$ is feasible, each $x_j^*$ is nonnegative, and the constraints $y^T A \geq c^T$ on the dual forces $c_j - \sum_{i=1}^{m} \sum_{j=1}^{n} y_i^* a_{ij} \leq 0$, thus for this sum (composed of only non-positive terms) to be zero, for each $j$, either one of the terms needs to be zero. Hence if $\sum_{i=1}^{m} \sum_{j=1}^{n} y_i^* a_{ij} > c_j$, $x_j^* = 0$.

We repeat the same argument. The second inequality gives the equality

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

Since $y^*$ is feasible, each $y_i^*$ is nonnegative, and the constraints $Ax \leq b$ on the primal forces $\sum_{j=1}^{n} a_{ij} x_j^* - b_i \leq 0$, thus for the sum to be zero, for each $i$, either one of the terms needs to be zero. Hence, if $\sum_{j=1}^{n} a_{ij} x_j^* < b_i$, then $y_i^* = 0$. $\square$

(1) and (2) are sometimes called the *complementary slackness conditions*. They require that a strict inequality (slackness) in a variable in a standard problem implies that the complementary constraint in the dual be satisfied with equality.

**Example 4.18.** Consider the primal problem

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t \quad & x_1 + 2x_2 \le 4 \\ & 4x_1 + 2x_2 \le 12 \\ & -x_1 + x_2 \le 1 \\ & x_1, x_2 \ge 0 \end{aligned}$$

whose dual problem is

$$\begin{aligned} \min \quad & 4y_1 + 12y_2 + y_3 \\ s.t \quad & y_1 + 4y_2 - y_3 \ge 1 \\ & 2y_1 + 2y_2 + y_3 \ge 1 \\ & y_1, y_2, y_3 \ge 0 \end{aligned}$$

We are given that $(x_1^*, x_2^*) = (8/3, 2/3)$, and $x_1^* + x_2^* = 10/3$. Since $x_1^* > 0$, $x_2^* > 0$, this means that constraints on $y^*$ must be met with equality, namely

$$y_1^* + 4y_2^* - y_3^* = 1,\ 2y_1^* + 2y_2^* + y_3^* = 1.$$

Now using $(x_1^*, x_2^*) = (8/3, 2/3)$, we have

$$\begin{aligned} x_1^* + 2x_2^* &= 4 \le 4 \\ 4x_1^* + 2x_2^* &= 12 \le 12 \\ -x_1^* + x_2^* &= -2 < 1. \end{aligned}$$

Since we have two constraints with equality, and one with strict inequality, the one with strict inequality means $y_3^* = 0$. Thus the two equalities in $y_1^*, y_2^*$ become

$$y_1^* + 4y_2^* = 1,\ 2y_1^* + 2y_2^* = 1.$$

We can solve for $y_1^*, y_2^*$: $(y_1^*, y_2^*) = (1/3, 1/6)$. Since this vector is feasible, the if part of the Equilibrium Theorem implies it is optimal. Furthermore $4(1/3) + 12(1/6) = 10/3$, which is the same as the optimal value for the primal, another check for optimality!

## 4.4 Two Person Zero Sum Games

We give an application of linear programming to game theory, for zero sum games with two players.

A two person zero sum game is a game with two players, where one player wins and the other player loses.

**Example 4.19.** Consider the game of "Odds and Evens" (also known as matching pennies). Suppose that Player 1 takes evens, and Player 2 takes odds. Each player simultaneously shows either one finger or two fingers, if the number of fingers matches, the result is even, Player 1 wins (say 2 dollars) otherwise it is odd, Player 2 wins (say 2 dollars). Each player has 2 strategies, show one finger or two fingers.

| | Player 2, 1 finger | Player 2, 2 fingers |
|---|---|---|
| Player 1, 1 finger | 2 | -2 |
| Player 1, 2 fingers | -2 | 2 |

This table is read from the view point of Player 1, that is a 2 is a gain of 2 for Player 1, and a -2 is a loss of 2 for Player 1. It is read the other way round for Player 2, a 2 is bad for Player 2, this is what he has to pay to Player 1, while a -2 is good, it means Player 1 owes him 2.

We get a *pay off matrix*

$$A = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

We consider games where players move simultaneously. For two players, we have a general *pay off matrix* $A = (a_{ij})$ which summarizes the gains/losses of both players. Suppose Player 1 indexes the rows of $A$, and Player 2 its columns, which means that there is one row of $A$ for every move of Player 1, and one column of $A$ for every move of Player 2. We say that Player 1 wins $a_{ij} > 0$ from Player 2, so $a_{ij} > 0$ is good for Player 1, and bad for Player 2.

Suppose we are given the pay off matrix

$$\begin{bmatrix} -5 & 3 & 1 & 20 \\ 5 & 5 & 4 & 6 \\ -4 & 6 & 0 & -5 \end{bmatrix}$$

For Player 1, each row consists of a move, so for the first move, the worst is -5, for the second move, the worst is 4, for the last move, the worst is -5. For Player 1, it makes sense to play move 2, because this moves ensures a win.

For Player 2, each column consists of a move, and the worst for him is the largest gain for Player 1, so for the first move, the worst is 5, for the second move, the worst is 6, for the 3rd move, the worst is 4, and for the last move, the worst is 20. So it makes sense for Player 2 to use move 3, this ensures the least loss.

It turns out that for this example $a_{23} = 4$ is both the smallest entry in row 2, and the largest entry in column 3. The game is thus considered "solved", or we say that the game has an *equilibrium* which has value 4 (4 is sometimes called a saddle point), because there is exists a strategy which is best for both players: at the same time, it maximizes Player 1's win, and minimizes Player 2's loss.

**Example 4.20.** The game "Scissors-Paper-Stone" has no saddle point. In this game, each player shows either scissors, paper or stone with a hand: scissors cut paper, paper covers stone, stone breaks scissors,

| | Scissors | Paper | Stone |
|---|---|---|---|
| Scissors | 0 | 1 | -1 |
| Paper | -1 | 0 | 1 |
| Stone | 1 | -1 | 0 |

The worst for Player 1 is -1 on each row. The worst for Player 2 is 1 on each column, so there is no saddle point.

The game "Odds and Evens" has no saddle point either.

We speak of "mixed" strategy when each move of a player is probabilistic. In contrast, a "pure" strategy is when each move is deterministic. In the above solved game, there was a pure strategy leading to an equilibrium: Player 1 chooses move 2, and Player 2 chooses move 3.

Consider a "mixed" strategy. Player 1 has a set of moves, move $i$, $i = 1, \ldots, m$, whose gains/losses are specified by the coefficients $a_{ij}$ of the pay off matrix $A$. Each move is done with probability $p_i$. If Player 2 plays $j$, Player 1's expected pay off is

$$\sum_{i=1}^{m} a_{ij} p_i.$$

Player 1 wishes to maximize (over $p = (p_1, \ldots, p_m)$) his minimal expected pay off:

$$\min_j \sum_{i=1}^{m} a_{ij} p_i.$$

Similarly, Player 2 has a set of moves, and move $j$, $j = 1, \ldots, n$ is attached a probability $q_j$. Player 2 wishes to minimize over $q = (q_1, \ldots, q_n)$ his maximal expected loss:

$$\max_i \sum_{j=1}^{n} a_{ij} q_j.$$

There may be other ways to define what both players are interested in doing, but we will see that this formulation is meaningful in that it will lead to a saddle point.

Let us write the Player's optimization problem as a linear program. Player 2 wants to minimize its maximal expected loss, that is

$$\min_q \left( \max_i \sum_{j=1}^{n} a_{ij} q_j \right)$$

such that

$$\sum_{j=1}^{n} q_j = 1, \ q \geq 0.$$

An equivalent formulation is

$$\begin{array}{ll} \min\limits_{q,v} & v \\ s.t. & \sum_{j=1}^m a_{ij}q_j \leq v, \ i = 1, \ldots, m \\ & \sum_{j=1}^n q_j = 1 \\ & q \geq 0. \end{array}$$

This is an equivalent formulation because the first constraint $\sum_{j=1}^m a_{ij}q_j \leq v, \ i = 1, \ldots, m$ ensures that $\sum_{j=1}^m a_{ij}q_j$ will take the highest possible value $v$, while $v$ itself will be minimized. The second and third constraints tell us that $q$ defines a probability distribution. Similarly, for Player 1, we have

$$\begin{array}{ll} \max\limits_{p,v} & v \\ s.t. & \sum_{i=1}^m a_{ij}p_i \geq v, \ j = 1, \ldots, n \\ & \sum_{i=1}^m p_i = 1 \\ & p \geq 0. \end{array}$$

Now we add a constant $k$ to each $a_{ij}$ so that $a_{ij} > 0$ for all $i, j$, this does not change the nature of the game (if the constraints $\sum_{j=1}^m a_{ij}q_j \leq v$ is replaced by the constraints $\sum_{j=1}^m (a_{ij} + k)q_j = \sum_{j=1}^m a_{ij}q_j + k \leq v \Rightarrow \sum_{j=1}^m a_{ij}q_j \leq v - k$, then just set $v' = v - k$ and the objection function becomes to minimize $v' + k$, this thus does not change the optimal solution, though the value of the objective function is of course shifted by $k$), but it guarantees $v > 0$. For example, if all $a_{ij} < 0$, $v = 0$ could be a candidate for an upper bound. So without loss of generality, assume $a_{ij} > 0$ for all $i, j$. Then do the following change of variables $x_j = q_j/v$ (now $v$ cannot be zero, and $x_j$ is not infinite) inside Player 2's LP:

$$\begin{array}{ll} \min\limits_{x,v} & v \\ s.t. & \sum_{j=1}^m a_{ij}x_j \leq 1, \ i = 1, \ldots, m \\ & \sum_{j=1}^n x_j = \frac{1}{v} \\ & x \geq 0 \end{array}$$

and since $v = 1/\sum_{j=1}^n x_j$ from the second constraint, we finally get

$$\begin{array}{rl} (P) : \max & \sum_{j=1}^n x_j \\ s.t. & Ax \leq 1 \\ & x \geq 0 \end{array}$$

which is a primal LP in the form we are familiar with. We do the same transformation for Player 1. Assume $a_{ij} > 0$ and set $y_i = p_i/v$:

$$\begin{array}{ll} \max\limits_{p,v} & v \\ s.t. & \sum_{i=1}^m a_{ij}y_i \geq v, \ j = 1, \ldots, n \\ & \sum_{i=1}^m y_i = \frac{1}{v} \\ & y \geq 0, \end{array}$$

which becomes

$$\begin{aligned}(D):\min_{y} \quad & \textstyle\sum_{i=1}^{m} y_i\\ s.t. \quad & A^T y \geq 1\\ & y \geq 0,\end{aligned}$$

and we see that $P$ and $D$ are dual, hence they have the same optimal value (that is, assuming they are feasible and bounded), reached respectively for $x^*$ and $y^*$.

Thus Player 1 can guarantee an expected gain of a least

$$v = 1/\sum_{i=1}^{m} y_i^*$$

using the strategy $p = vy^*$.

Player 2 can guarantee an expected loss of a most

$$v = 1/\sum_{j=1}^{n} x_j^*$$

using the strategy $q = vx^*$.

The game is thus solved, and has value $v$.

**Example 4.21.** Consider the game of "Odds and Evens", given by the pay off matrix:

| | Player 2, 1 finger, $q_1$ | Player 2, 2 fingers, $q_2$ |
|---|---|---|
| Player 1, 1 finger, $p_1$ | 2 | -2 |
| Player 1, 2 fingers, $p_2$ | -2 | 2 |

We saw there is no saddle point for a pure strategy, so we consider a mixed strategy where each move is assigned a probability.

If we look at Player 2, if Player 1 chooses move 1, his expected loss is $2q_1 - 2q_2$, if Player 1 chooses move 2, his expected loss is $-2q_1 + 2q_2$. So Player 2 will try to minimize the maximum loss between $2q_1 - 2q_2$ and $-2q_1 + 2q_2$. The corresponding LP is

$$\begin{aligned}\min_{q,v} \quad & v\\ s.t. \quad & 2q_1 - 2q_2 \leq v\\ & -2q_1 + 2q_2 \leq v\\ & q_1 + q_2 = 1\\ & v, q \geq 0,\end{aligned}$$

and in order to have only pay off coefficients that are positive, we add a constant $k = 3$ to every coefficient, to get

$$\begin{aligned} \min_{q,v} \quad & v \\ s.t. \quad & 5q_1 + q_2 \leq v \\ & q_1 + 5q_2 \leq v \\ & q_1 + q_2 = 1 \\ & v, q \geq 0. \end{aligned}$$

We do the change of variables $x_j = q_j/v$ to get

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & 5x_1 + x_2 \leq 1 \\ & x_1 + 5x_2 \leq 1 \\ & x \geq 0. \end{aligned}$$

We then solve this LP. The initial tableau is:

$$\begin{array}{cccc|c} 5 & 1 & 1 & 0 & 1 \\ 1 & 5 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 \end{array}$$

for the BFS $[0, 0, 1, 1]$ whose objective function takes value 0. Pick $q = 1$ as column pivot, then the row pivot is $i = 1$:

$$\begin{array}{cccc|c} 1 & 1/5 & 1/5 & 0 & 1/5 \\ 0 & 24/5 & -1/5 & 1 & 4/5 \\ \hline 0 & 4/5 & -1/5 & 0 & -1/5 \end{array}$$

This is the BFS $[1/5, 0, 0, 4/5]$ whose objective function takes value $1/5$. Pick $q = 2$ as column pivot, then pick the row pivot $i = 2$:

$$\begin{array}{cccc|c} 1 & 0 & 1/5 + 1/120 & -1/24 & 1/5 - 1/30 \\ 0 & 1 & -1/24 & 5/24 & 1/6 \\ \hline 0 & 0 & -1/5 + 1/30 & -1/6 & -1/5 - 4/30 \end{array}$$

This is the BFS $[1/6, 1/6, 0, 0]$ whose objective function takes value $1/3 = 1/v$. To $x_1 = 1/6 = q_1/v$ corresponds the probability $q_1 = vx_1 = 3/6 = 1/2$. Thus Player 2's optimal strategy is to choose move 1 with probability $q_1 = 1/2$, and move 2 with probability $q_2 = 1/2$. A posteriori, this is the strategy that makes most sense, due to the symmetry of the pay off matrix.

## 4.5 Exercises

**Exercise 35.** Compute all the basic feasible solutions of the following LP:

$$\begin{aligned} \min \quad & x_1 + x_2 \\ s.t. \quad & x_1 + 2x_2 \geq 3 \\ & 2x_1 + x_2 \geq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

**Exercise 36.** Consider the linear program:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & -x_1 + x_2 \leq 1 \\ & 2x_1 + x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Prove by computing reduced costs that $[1/3, 4/3, 0, 0]$ is an optimal solution for this LP.

**Exercise 37.** Solve the following LP using the simplex algorithm:

$$\begin{aligned} \max \quad & 6x_1 + x_2 + x_3 \\ s.t. \quad & 9x_1 + x_2 + x_3 \leq 18 \\ & 24x_1 + x_2 + 4x_3 \leq 42 \\ & 12x_1 + 3x_2 + 4x_3 \leq 96 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

**Exercise 38.** Solve the following LP using the simplex algorithm:

$$\begin{aligned} \max \quad & 3x_1 + x_3 \\ s.t. \quad & x_1 + 2x_2 + x_3 = 30 \\ & x_1 - 2x_2 + 2x_3 = 18 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

**Exercise 39.** Show that the dual of the dual is the primal.

**Exercise 40.** Consider the following LP:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ s.t. \quad & x_1 + 2x_2 \leq 4 \\ & 4x_1 + 2x_2 \leq 12 \\ & -x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Compute its dual.

**Exercise 41.** Consider the following LP:

$$\begin{array}{rc} \max & 2x_1 + 4x_2 + x_3 + x_4 \\ s.t. & x_1 + 3x_2 + x_4 \le 4 \\ & 2x_1 + x_2 \le 3 \\ & x_2 + 4x_3 + x_4 \le 3 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{array}$$

Compute its dual. Show that $x = [1, 1, 1/2, 0]$ and $y = [11/10, 9/20, 1/4]$ are optimal solutions for respectively this LP and its dual.

**Exercise 42.** In Exercise 40, we computed the dual of the LP:

$$\begin{array}{rc} \max & x_1 + x_2 \\ s.t. & x_1 + 2x_2 \le 4 \\ & 4x_1 + 2x_2 \le 12 \\ & -x_1 + x_2 \le 1 \\ & x_1, x_2 \ge 0. \end{array}$$

Solve both this LP and its dual.

**Exercise 43.** Solve the “Scissors-Paper-Stone” whose pay off matrix is

$$\begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}.$$

**Exercise 44.** Solve the game whose pay off matrix is

$$\begin{bmatrix} 2 & -1 & 6 \\ 0 & 1 & -1 \\ -2 & 2 & 1 \end{bmatrix}.$$