10-725: Convex Optimization **Fall 2013**

Lecture 2: August 29

Lecturer: Barnabás Póczos *Scribes: Samrachana Adhikari, Mattia Ciollaro, Fabrizio Lecci*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Linear Programming (part I)

2.1 Goals of Lectures on Linear Programming

The goals of this and the following lecture on Linear Programming are:

1. to define what a Linear Program is
2. to solve linear programs using the Simplex Algorithm
3. to understand why linear programs are useful by means of motivating examples and applications to Machine Learning
4. to understand the difficulties associated to linear programs (e.g. convergence of algorithms, solvability in polynomial vs. exponential many operations, approximate vs. exact solutions, ...).

2.2 Simplest Optimization Problems

The simplest optimization problems concern the optimization (i.e. maximization or minimization) of a given 1-dimensional constant or linear function f when no constraints or only bound constraints are placed on its argument. The optimization of f is trivial in these cases. Examples are:

- $\min_x f(x) = c\,; \quad c \in \mathbb{R}$ (f constant: the optimal value c is achieved at any x in the domain of f)
- $\min_x f(x) = ax + b\,; \quad a, b \in \mathbb{R}$ (f linear: the optimal values are $+\infty$ if $a > 0$ or $-\infty$ if $a < 0$)
- $\max_x f(x) = ax + b\,; \quad c \leq x \leq d \quad a > 0 \quad b, c, d \in \mathbb{R}$ (f linear with bound constraint: the optimal value $ad + b$ is achieved at $x = d$). See figure 2.1.

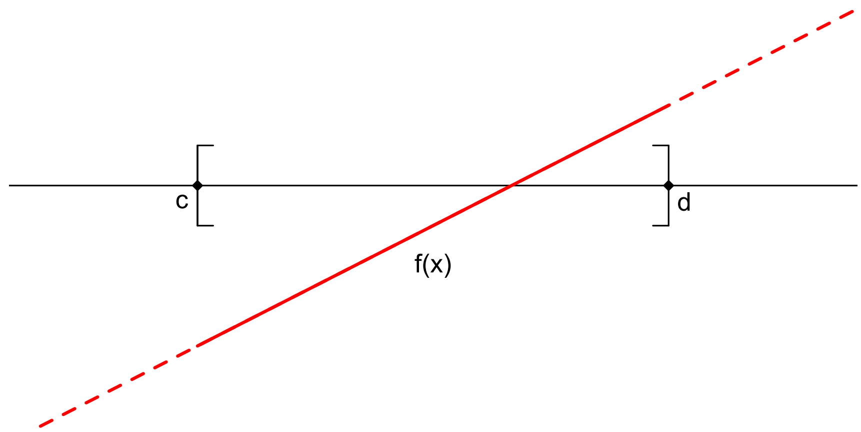

Figure 2.1: A linear objective function with bound constraints. The function f is maximized at $x = d$.

2.3 Linear Programs

A more interesting class of optimization problems involves the optimization of a n-dimensional linear function when m linear constraints are placed on its arguments and the arguments are subject to bound constraints. For instance, the following problem corresponds to the minimization of a linear cost function f subject to m linear constraints and n bound constraints:

$$\min_{x_1,\ldots,x_n} f(x_1,\ldots,x_n) = c_1x_1 + \cdots + c_nx_n$$

$$\text{s.t.}$$

$$\left.\begin{array}{l} a_{11}x_1 + \cdots + a_{1n}x_n \le b_1 \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \le b_m \end{array}\right. \qquad \text{(linear constraints)}$$

$$\left.\begin{array}{l} l_1 \le x_1 \le u_1 \\ \vdots \\ l_n \le x_n \le u_n\,. \end{array}\right. \qquad \text{(bound constraints)}$$

An optimization problem such as the one above is termed *Linear Program*. In principle, one can express a linear program in different ways. In this case, the linear program above is expressed in *inequality form*. More conveniently, the same problem can be rewritten in matrix notation as

$$\min_{x_1,\ldots,x_n} f(x_1,\ldots,x_n) = c^T x$$

$$\text{s.t.}$$

$$A^T x \le b$$
$$l \le x \le u$$

where

$$\begin{aligned} c &= (c_1, \ldots, c_n)^T \in \mathbb{R}^n \\ x &= (x_1, \ldots, x_n)^T \in \mathbb{R}^n \\ b &= (b_1, \ldots, b_m)^T \in \mathbb{R}^m \\ l &= (l_1, \ldots, l_n)^T \in \mathbb{R}^n \\ u &= (u_1, \ldots, u_n)^T \in \mathbb{R}^n \end{aligned}$$

and $A^T \in \mathbb{R}^{m \times n}$ is the $m \times n$ matrix of coefficients appearing in the linear constraints.

Example 2.1. The following linear program is expressed in inequality form:

$$\min_{x_1, x_2} \; - 2x_1 - x_2$$

$$\text{s.t.}$$

$$\begin{aligned} x_1 + x_2 &\leq 5 \\ 2x_1 + 3x_2 &\leq 4 \end{aligned}$$

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0\,. \end{aligned}$$

In this case,

$$\begin{aligned} c &= [-2, -1]^T \\ b &= [5, 4]^T \\ A^T &= \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix}\,. \end{aligned}$$

Example 2.2. Let's consider the following linear program:

$$\min_{x_1, x_2} z = -2x_1 - x_2$$

$$\text{s.t.}$$

$$\begin{aligned} x_1 + x_2 &\leq 5 \\ 2x_1 + 3x_2 &\leq 12 \end{aligned}$$

$$\begin{aligned} x_1 &\leq 4 \\ x_1 &\geq 0 \\ x_2 &\geq 0. \end{aligned}$$

The set of points (x_1, x_2) that simultaneously satisfy all the linear and bound constraint is called the set of *feasible points* or *feasible set*. The set of points lying at the boundary of the set of feasible points corresponding to intersections of linear and bound constraints is called the set of *corner points* or the set of *corners*.

The set of feasible points which are not solutions of the linear program is called the set of *suboptimal points*.

Figure 2.2 gives a graphical representation of the linear program. The set of feasible points is marked in pink. It is easy to guess that the the objective function $z = z(x_1, x_2)$ is minimized at the point C. Notice that at the point C the constraints $x_1 + x_2 \leq 5$ and $x_1 \leq 4$ are *active* (i.e. they hold with exact equality) whereas the remaining constraints are *inactive* (i.e. they hold as strict inequalities).

Roughly speaking, the Simplex Algorithm 'jumps' on the corners of the linear program and evaluates the objective function $z = z(x_1, x_2)$ until the optimum is found.

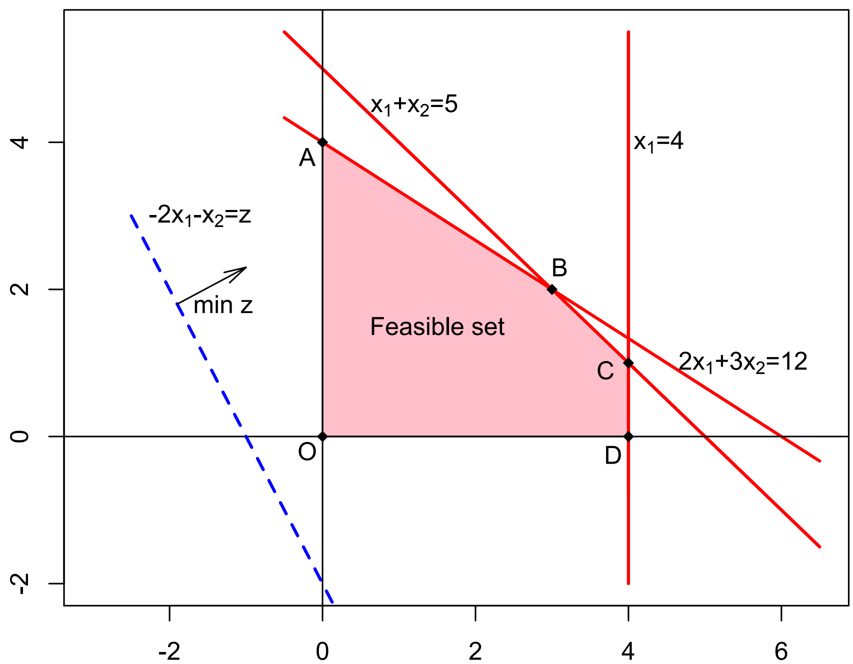

Figure 2.2: Graphical representation of the linear program of Example 2.2.

From Figure 2.2 one can see that some difficulties can arise in a Linear Program. In fact,

- the feasible set might be empty
- there might be infinitely global optima if the optimum lies on an edge of the feasible set
- the optimum could be $\pm\infty$.

Finally, notice that in higher dimensional linear programs (such as the general one described at the beginning of this Section), the feasible set is a polytope and the objective function is an hyperplane.

2.4 History and Motivation

The interest in *Linear Programming* began during World War II in order to deal with problems of transportation, scheduling, and allocation of resources subject to certain restrictions such as costs and availability. Examples of these problems are:

- how to efficiently allocate 70 men to 70 jobs (job scheduling)
- how to produce a particular blend (e.g. 30% Lead, 30% Zinc, 40% Tin) out of 9 different alloys that have different mixtures and different costs in such a way that the total cost is minimize
- how to optimize a flow network (the *max flow min cut* problem).

Why *Linear Programming*? Because

- the objective functions to be optimized are linear and so are the constraints on their arguments
- programming = scheduling, and efficient scheduling is one of the focuses of this kind of optimization problems. Also, at the time of the development of the first computers, *programming* was a fancy word that attracted funds and grants from both military and non-military sources.

George Bernard Dantzig (1914-2005) is considered to be the father of Linear Programming. He was the inventor of the Simplex Algorithm.

There are many reasons which make Linear Programming an interesting topic. First of all, linear programs represent the simplest (yet nontrivial) optimization problems. Also, many complex systems can be well approximated by linear equations and therefore Linear Programming is a key tool for a number of important applications. Last but not least, linear programs are very appealing from a computational perspective, especially because nowadays there exists many toolboxes that can solve them efficiently.

Example 2.3 (The product mix problem)**.** Here is an example of a real life problem that can be framed as a Linear Program. Suppose that a certain company manufactures four models of desks. In order to produce each desk, a certain amount of man hours are needed in the carpentry shop and in the finishing shop. For instance, desks of type 1 need 4 man hours of manufacturing in the carpentry shop and 1 man hour in the finishing shop before their production is complete. Each sold desk guarantees a certain profit to the company depending on the model. For instance, the company makes a 4$ profit each time a desk of type 1 is sold. However, the number of available man hours in the carpentry shop and in the finishing shop are limited to 6000 and 4000 respectively. What is the optimal number of each type of desk that the company should produce?

	Desk 1	Desk 2	Desk 3	Desk 4	Available hrs
Carpentry shop hrs	4	9	7	10	6000
Finishing shop hrs	1	1	3	40	4000
Profit	$12	$20	$18	$40	

Figure 2.3: The number of man hours and the profits for the product mix problem.

Based on Figure 2.3 which displays the parameters of this problem, we can state the question of interest in terms of a Linear Program. Let $x_1, \ldots, x_4$ indicate the number of produced units of each of the four models of desks and let $f(x_1, x_2, x_3, x_4) = 12x_1 + 20x_2 + 18x_3 + 40x_4$ be the company's profit function. The company needs to solve

$$\max_{x_1,x_2,x_3,x_4} f(x_1,x_2,x_3,x_4) = 12x_1 + 20x_2 + 18x_3 + 40x_4$$

s.t.

$$4x_1 + 9x_2 + 7x_3 + 10x_4 \leq 6000$$
$$x_1 + x_2 + 3x_3 + 40x_4 \leq 4000$$

$$x_1 \geq 0$$
$$x_2 \geq 0$$
$$x_3 \geq 0$$
$$x_4 \geq 0.$$

2.5 Application: Pattern Classification via Linear Programming

This section presents an application of Linear Programming to Pattern Classification[1]. The intent is to demonstrate how Linear Programming can be used to perform linear classification (for instance as an alternative to Linear Discriminant Analysis).

Here are our goals. Given two sets $H = \{H^1, \ldots, H^h\} \subseteq \mathbb{R}^n$ and $M = \{M^1, \ldots, M^m\} \subseteq \mathbb{R}^n$, we want to

Problem 1) check if H and M are linearly separable

Problem 2) if H and M are linearly separable, find a separating hyperplane.

Before proceeding, it is worth clarifying what it means for two sets to be linearly separable.

Definition 2.4 (linear separability)**.** *Two sets $H \subseteq \mathbb{R}^n$ and $M \subseteq \mathbb{R}^n$ are said to be (strictly) linearly separable if $\exists a \in \mathbb{R}^n, b \in \mathbb{R} : H \subseteq \{x \in \mathbb{R}^n : a^T x > b\}$ and $M \subseteq \{x \in \mathbb{R}^n : a^T x \leq b\}$.*

In words, H and M are linearly separable if there exists an hyperplane that leaves all the elements of H on one side and all the elements on M on the other side. Figure 2.4 illustrates linear separability.

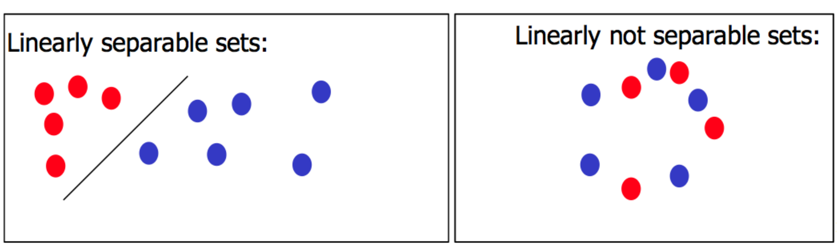

Figure 2.4: An example of linearly separable sets (left) and not linearly separable sets (right).

The following Lemma gives an alternative characterization of linear separability.

[1] See also `http://cgm.cs.mcgill.ca/~beezer/cs644/main.html` for more information

Lemma 2.5. *Two sets $H = \{H^1, \ldots, H^h\} \subseteq \mathbb{R}^n$ and $M = \{M^1, \ldots, M^m\} \subseteq \mathbb{R}^n$ are linearly separable if and only if there exist $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that $a^T H^i - b \geq 1$ and $a^T M_j - b \leq -1$ for all $i \in \{1, \ldots, h\}$ and for all $j \in \{1, \ldots, m\}$.*

Proof.
($\Longleftarrow$) Trivial.

($\Longrightarrow$) By Definition 2.4 there exist $c \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that $c^T x > b$ for $x \in H$ and $c^T x \leq b$ for $x \in M$. Define $p = \min_{x \in H} c^T x - \max_{x \in M} c^T x$. It is clear from above that such p is strictly greater than 0. Let

$$a = \frac{2}{p} c \qquad \text{and} \qquad b = \frac{1}{p}\left(\min_{x\in H} c^T x + \max_{x \in M} c^T x\right).$$

Now,

$$\begin{aligned}\min_{x\in H} a^T x = \min_{x \in H} \frac{2}{p} c^T x = \min_{x\in H} \frac{1}{p}\left(c^T x + c^T x\right) &\overset{(*)}{=} \min_{x \in H} \frac{1}{p}\left(c^T x + \max_{x\in M} c^T x + p\right) = \\ = \frac{1}{p}\left(\min_{x \in H} c^T x + \max_{x\in M} c^T x + p\right) &\overset{(**)}{=} \frac{1}{p}(pb + p) = b + 1\,,\end{aligned}$$

which equivalently reads

$$a^T x - b \geq 1 \quad \forall x \in H\,.$$

Similarly,

$$\begin{aligned}\max_{x\in M} a^T x = \max_{x \in M} \frac{2}{p} c^T x = \max_{x\in M} \frac{1}{p}\left(c^T x + c^T x\right) &\overset{(*)}{=} \max_{x \in M} \frac{1}{p}\left(c^T x + \min_{x\in H} c^T x - p\right) = \\ = \frac{1}{p}\left(\max_{x \in M} c^T x + \min_{x\in H} c^T x - p\right) &\overset{(**)}{=} \frac{1}{p}(pb - p) = b - 1\end{aligned}$$

which equivalently reads

$$a^T x - b \leq -1 \quad \forall x \in M\,.$$

Notice that the $(*)$ follow from the definition of p and the $(**)$ follow from the definition of b. ☐

Figure 2.5 depicts the geometry of Lemma 2.5.

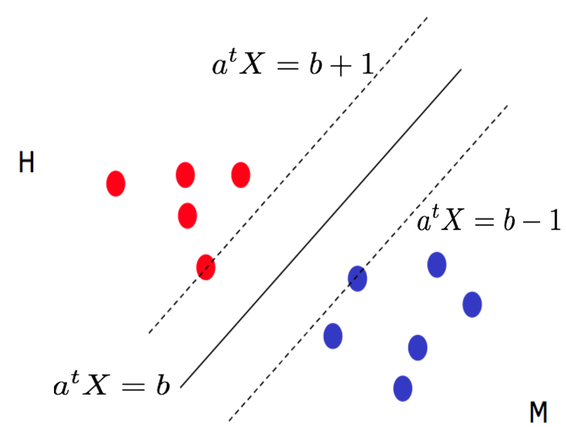

Figure 2.5: Geometrical representation of Lemma 2.5.

Now we show how Linear Programming can be used to solve **Problem 1** and **Problem 2** above for two given sets $H = \{H^1, \dots, H^h\} \subseteq \mathbb{R}^n$ and $M = \{M^1, \dots, M^m\} \subseteq \mathbb{R}^n$. Consider the following linear program:

$$\min_{y,z,a,b} \ \frac{1}{h}[y_1 + y_2 + \cdots + y_h] + \frac{1}{m}[z_1 + z_2 + \cdots + z_m]$$

s.t.

$$\begin{aligned} y_i &\geq -a^T H^i + b + 1 && \text{for } i = 1, \dots, h \\ z_i &\geq \ \ a^T M^i - b + 1 && \text{for } j = 1, \dots, m \end{aligned}$$

$$\begin{aligned} y_i &\geq 0 && \text{for } i = 1, \dots, h \\ z_i &\geq 0 && \text{for } j = 1, \dots, m \end{aligned}$$

where $y \in \mathbb{R}^h$, $z \in \mathbb{R}^m$, $a \in \mathbb{R}^n$, $b \in \mathbb{R}$.

In the following we refer to the above program as **LP**.

Theorem 2.6. *H and M are linearly separable if and only if the optimal value of **LP** is 0.*

Theorem 2.7. *If H and M are linearly separable and y^*, z^*, a^*, b^* is an optimal solution of **LP**, then $f(x) = a^{*T}x + b^*$ is a separating hyperplane.*

Proof of Theorems 2.6 and 2.7.

$$\text{The optimal value of } \mathbf{LP} \text{ is } 0 \iff \begin{aligned} & y^* = 0 \\ & z^* = 0 \\ & a^{*T}H^i \geq b^* + 1 && \text{for } i = 1, \dots, h \\ & a^{*T}M^i \leq b^* - 1 && \text{for } j = 1, \dots, m \end{aligned}$$

$$\iff \begin{aligned} & H \text{ and } M \text{ are linearly separable and} \\ & a^{*T}x - b^* = 0 \text{ is a separating hyperplane.} \end{aligned}$$

The second implication follows from Lemma 2.5. □

The authors of [1] performed linear classification via Linear Programming for breast cancer diagnosis. They have been able to classify benign lumps and malignant lumps with 97.5% accuracy.

Example 2.8 (Linearly Separable Case)**.** Let $H = \{(0,0), (1,0)\} \subset \mathbb{R}^2$ and $M = \{(0,2), (1,2)\} \subset \mathbb{R}^2$. See

Figure 2.6 (left). The associated linear program is

$$\min_{y,z,a,b} \frac{1}{2}[y_1 + y_2] + \frac{1}{2}[z_1 + z_2]$$

s.t.

$$\begin{aligned} y_1 &\geq -[a_1, a_2]\begin{bmatrix}0\\0\end{bmatrix} + b + 1 = b + 1 \\ y_2 &\geq -[a_1, a_2]\begin{bmatrix}1\\0\end{bmatrix} + b + 1 = -a_1 + b + 1 \\ z_1 &\geq \quad [a_1, a_2]\begin{bmatrix}0\\2\end{bmatrix} - b + 1 = 2a_2 - b + 1 \\ z_1 &\geq \quad [a_1, a_2]\begin{bmatrix}1\\2\end{bmatrix} - b + 1 = a_1 + 2a_2 - b + 1 \end{aligned}$$

$$\begin{aligned} y_i &\geq 0 \qquad \text{for } i = 1, 2 \\ z_i &\geq 0 \qquad \text{for } j = 1, 2. \end{aligned}$$

An optimal solution for this program is

$$\begin{aligned} &y_1 = y_2 = z_1 = z_2 = 0 \\ &a^T = [1, -2],\ b = -1. \end{aligned}$$

By Theorem 2.7, $x_1 - 2x_2 + 1 = 0$ is a separating hyperplane.

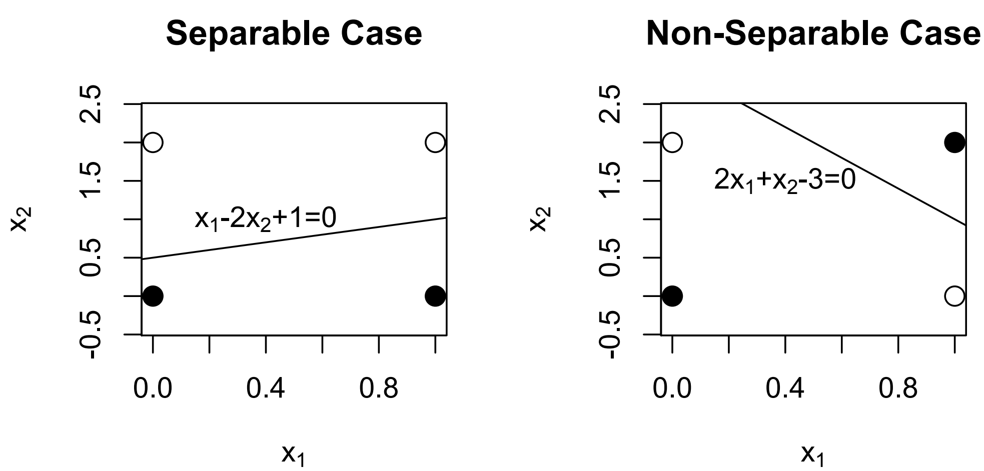

Figure 2.6: Left: the separable case of Example 2.8. Right: the non-separable case of Example 2.9

Example 2.9 (Linearly Nonseparable Case)**.** Let $H = \{(0,0), (1,2)\} \subset \mathbb{R}^2$ and $M = \{(0,2), (1,0)\} \subset \mathbb{R}^2$.

See Figure 2.6 (right). The associated linear program is

$$\min_{y,z,a,b} \ \frac{1}{2}[y_1 + y_2] + \frac{1}{2}[z_1 + z_2]$$

s.t.

$$\begin{aligned}
y_1 &\geq -[a_1, a_2]\begin{bmatrix}0\\0\end{bmatrix} + b + 1 = b + 1\\
y_2 &\geq -[a_1, a_2]\begin{bmatrix}1\\2\end{bmatrix} + b + 1 = -a_1 - 2a_2 + b + 1\\
z_1 &\geq \quad [a_1, a_2]\begin{bmatrix}0\\2\end{bmatrix} - b + 1 = 2a_2 - b + 1\\
z_1 &\geq \quad [a_1, a_2]\begin{bmatrix}1\\0\end{bmatrix} - b + 1 = a_1 - b + 1
\end{aligned}$$

$$\begin{aligned}
y_i &\geq 0 \qquad \text{for } i = 1, 2\\
z_i &\geq 0 \qquad \text{for } j = 1, 2.
\end{aligned}$$

An optimal solution for this program is

$$\begin{aligned}
&y_1 = 4, y_2 = z_1 = z_2 = 0\\
&a^T = [2, 1], b = 3.
\end{aligned}$$

$2x_1 + x_2 - 3 = 0$ is the proposed non-separating hyperplane.

2.6 Inequality Form vs. Standard Form

We have already seen how a linear program is written in the inequality form. In matrix notation:

$$\min_{x_1,\ldots,x_n} c^T x$$

s.t.

$$\begin{aligned}
A^T x &\leq b\\
l \leq x &\leq u
\end{aligned}$$

where

$$\begin{aligned}
c &= (c_1, \ldots, c_n)^T \in \mathbb{R}^n\\
x &= (x_1, \ldots, x_n)^T \in \mathbb{R}^n\\
b &= (b_1, \ldots, b_m)^T \in \mathbb{R}^m\\
l &= (l_1, \ldots, l_n)^T \in \mathbb{R}^n\\
u &= (u_1, \ldots, u_n)^T \in \mathbb{R}^n
\end{aligned}$$

and $A^T \in \mathbb{R}^{m\times n}$ is the $m \times n$ matrix of coefficients appearing in the linear constraints.

We say that a linear program is written in he *standard form* if the inequalities in the linear constraints are replaced by equalities and the variables are constrained to be non-negative. In matrix notation:

$$\min_{x_1,\dots,x_n} c^T x$$

$$\text{s.t.}$$

$$\begin{aligned} &A^T x = b \\ &x \geq 0 \end{aligned}$$

where

$$\begin{aligned} c &= (c_1,\dots,c_n)^T \in \mathbb{R}^n \\ x &= (x_1,\dots,x_n)^T \in \mathbb{R}^n_+ \\ b &= (b_1,\dots,b_m)^T \in \mathbb{R}^m_+ \quad \text{(sometimes } b \geq 0 \text{ is not required)} \end{aligned}$$

Theorem 2.10. *Any linear program can be rewritten to an equivalent standard linear program.*

The proof of the previous theorem consists in a series of rules to convert constraints from one form to the other one:

- **Getting rid of inequalities (except variable bounds)**. An inequality like

  $$x_1 + x_2 \leq 4$$

  can be converted into an equality by introducing a non-negative *slack* variable x_3, i.e.

  $$\begin{aligned} x_1 + x_2 + x_3 &= 4 \\ x_3 &\geq 0. \end{aligned}$$

- **Getting rid of equalities**. An equality can be converted into two inequalities. For example

  $$\begin{aligned} x_1 + 2x_2 = 4 \iff &x_1 + 2x_2 \leq 4 \\ &x_1 + 2x_2 \geq 4. \end{aligned}$$

- **Getting rid of negative variables**. A negative variable can be converted into the difference of two non-negative variables. For example

  $$x \in \mathbb{R} \implies x = u - v, \quad \text{for some } u \geq 0, v \geq 0.$$

- **Getting rid of bounded variables**.

  $$\begin{aligned} l \leq x \leq u \iff &x \geq l \\ &x \leq u. \end{aligned}$$

- **Max to Min**. A max problem can be easily converted into a min problem by noting that

  $$\max c^T x = -\min(-c)^T x$$

- **Getting rid of negative b**. If b is negative in the constraint

$$ax = b,$$

we can consider the equivalent constraint

$$-ax = -b.$$

Remark. If the standard form has n variables and m equalities, then the inequality form has $n-m$ variables and $m + (n - m) = n$ inequalities.

Example 2.11. Consider the following linear program, expressed in inequality form:

$$\max_{x_1,x_2} 2x_1 + 3x_2$$

$$\text{s.t.}$$

$$\begin{aligned} &x_1 + x_2 \leq 4 \\ &2x_1 + 5x_2 \leq 12 \\ &x_1 + 2x_2 \leq 5 \end{aligned}$$

$$\begin{aligned} &x_1 \geq 0 \\ &x_2 \geq 0\,. \end{aligned}$$

The equivalent standard form is

$$\min_{x_1,x_2} - 2x_1 - 3x_2$$

$$\text{s.t.}$$

$$\begin{aligned} &x_1 + x_2 + u = 4 \\ &2x_1 + 5x_2 + v = 12 \\ &x_1 + 2x_2 + w = 5 \end{aligned}$$

$$x_1, x_2, u, v, w \geq 0.$$

References

[1] Mangasarian, Olvi L., W. Nick Street, and William H. Wolberg. *Breast cancer diagnosis and prognosis via linear programming.* Operations Research 43.4 (1995): 570-577.