CLASSICAL FORMS OF LINEAR PROGRAMS, CONVERSION TECHNIQUES, AND SOME NOTATION

Linear programming is the minimization (maximization) of a linear objective, say \( c_1x_1 + c_2x_2 + \ldots + c_nx_n \), subject to linear constraints. The constraints can be of two types, equations or inequalities, \( a_1x_1 + a_2x_2 + \ldots + a_nx_n \leq b \), \( \geq b \), or equality, \( a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \). Although the non-negativity constraint, \( x_j \geq 0 \), is a special case of the inequality constraint, it is usually treated separately.

It would seem that there is a combinatorially large number of different types of linear programs depending on whether the objective is to minimize or to maximize the objective, whether the constraints are individually equality or inequality, or if the variables are sign restricted or unrestricted in sign. Fortunately, there are simple transformations which can convert the general case into simple classical forms.

**Transformation 1:** Maximizing \( f(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n \) is equivalent to minimizing \( g(x) = -c_1x_1 - c_2x_2 - \ldots - c_nx_n \); i.e., \(-g(x) = f(x)\). If \( f(x^1) > f(x^2) \) then \( g(x^1) = -f(x^1) < -f(x^2) = g(x^2) \). Thus if \( x^* \) maximizes \( f(x) \), then it minimizes \( g(x) \) and conversely.

**Transformation 2:** An inequality in one sense, say \( a_1x_1 + a_2x_2 + \ldots + a_nx_n \leq b \), can be replaced by an inequality, \(-a_1x_1 - a_2x_2 - \ldots - a_nx_n \geq -b\), in the other sense.

**Transformation 3:** An equality constraint, \( a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \), can be replaced by two inequality constraints, \( a_1x_1 + a_2x_2 + \ldots + a_nx_n \geq b \) and \(-a_1x_1 - a_2x_2 - \ldots - a_nx_n \geq -b\).

**Transformation 4:** An inequality constraint, \( a_1x_1 + a_2x_2 + \ldots + a_nx_n \leq b \) can be replaced by an equality constraint, \( a_1x_1 + a_2x_2 + \ldots + a_nx_n + s = b \) by adding a non-negative variable \( s (s \geq 0) \) called the slack variable.

**Transformation 5:** A variable \( x \), unrestricted in sign, can be replaced using two non-negative variables, \( x^+ \) and \( x^- \), by \( x = x^+ - x^- \).

Any place that \( x \) appears in either the objective or in a constraint, say as \( bx \), we have \( bx = b(x^+ - x^-) = bx^+ - bx^- \). Of course, many values of \( x^+ \) and \( x^- \) can give rise to the same value of \( x \) as long as their difference is the same. But as long as they appear together as a difference in linear expressions their contributions is the same.

Using these devices any linear program can be converted into any one of several classical formats.
The Standard Form:

The standard form of a linear program is to express it as the minimum of a linear form, subject to 
m equality constraints in n non-negative decision variables. We can also assume, without loss of 
generality, that the coefficients in the right hand side of the equality constraints are non-negative 
(b_i ≥ 0, i = 1,...,m). That is, we have:

minimize \[ c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \]  
in \( x_1, x_2, \ldots, x_n \)
subject to \[ a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n = b_1 \]  
\[ a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n = b_2 \]  
\[ \ldots \]  
\[ a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n = b_m \]  
and \( x_1 ≥ 0, x_2 ≥ 0, \ldots, x_n ≥ 0 \).

Note that the a_i's, c_i's, and the b_i's are given fixed constants, and the x_i's are variables whose 
values are to be determined in the optimizations process.

The standard form is what we will be most concerned with for the moment. Later we will 
consider another classical form of linear programs that we will find useful in our discussion of 
duality is called the symmetric form.

\[ z = \text{minimum} \ c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \]  
in \( x_1, x_2, \ldots, x_n \)
subject to \[ a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n ≥ b_1 \]  
\[ a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n ≥ b_2 \]  
\[ \ldots \]  
\[ a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n ≥ b_m \]  
and \( x_1 ≥ 0, x_2 ≥ 0, \ldots, x_n ≥ 0 \).

We also will deal with a canonical form when we start discussing solution methods for linear 
programs.

Note these transformations do not suffice to convert strict inequalities into standard form, or 
variables constrained to be strictly positive.
We now consider an example illustrating the conversion of a general linear program into one in standard form.

\[
\begin{align*}
\text{z} &= \text{Maximize} \quad 3x_1 - 2x_2 + x_4 \\
\text{Subject to} \quad 6x_1 + x_2 + x_3 + x_4 & \geq 2 \\
&\quad x_2 + x_4 \leq 4 \\
&\quad x_1 + x_2 + x_3 + x_4 = 4 \\
\end{align*}
\]

and \( x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \)

Applying Transformation 1 to the objective, and Transformations 2 and 4 for first and second constraints, and, finally, Transformation 5 to convert \( x_4 \) into the difference of two sign restricted variables we get:

\[
\begin{align*}
\text{z} &= \text{minimize} \quad -3x_1 + 2x_2 - x_4^+ + x_4^- \\
\text{subject to} \quad 6x_1 + x_2 + x_3 - s_1 &= 2 \\
&\quad x_2 + x_4^+ - x_4^- + s_2 = 4 \\
&\quad x_1 + x_2 + x_3 + x_4^+ - x_4^- = 4 \\
\end{align*}
\]

and \( x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4^+ \geq 0, x_4^- \geq 0, s_1 \geq 0, s_2 \geq 0. \)

Note that instead of using Transformation 5 on \( x_4 \), one could solve the third constraint for \( x_4 \) and thereby eliminate it from the problem. This gives a simpler standard form.

**SOME NOTATION:**

The explicit representation of linear programs used above can be rather clumsy. It can also obscure important features. For these reasons a number of representations in terms of vectors, matrices, tableaus, and block diagrams have been proposed to represent linear programs. In fact, entire computer modeling systems and languages are available just for converting from formulations which are convenient for applications to standard linear programming input. Unfortunately, no single representation is best for all purposes. In fact, for many purposes the explicit representation is the most useful. So for a solid understanding of mathematical programming it is necessary to be fluent in a number of representations. We introduce a few here which we will find immediately useful.

The explicit representation can be compacted by using summation notation. In the case of the standard form we get:
Most commonly we use a matrix notation. We represent the coefficients of the constraint as a matrix $A$ with $m$ rows and $n$ columns. This implies that the vector of decision variables, usually denoted $x$ is a column vector with $n$ components. Similarly we represent the coefficients of the cost function which we are trying to minimize as an $n$ dimensional row vector. Finally, the constants in the constraint relations are generally represented as a $m$ dimensional column vector, $b$. We will usually not distinguish in our notation between row and column vectors; the context usually makes this clear.

Thus this notation, the standard form can be written:

\[
\begin{align*}
\text{minimize} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j = b_i \quad i=1, \ldots, m \\
& \quad x_j \geq 0 \quad j=1, \ldots, n
\end{align*}
\]

Similarly, the symmetric form is given in our notation as:

\[
\begin{align*}
z = \text{minimum} \quad & cx \\
\text{subject to} & \quad Ax \geq b \\
& \quad x \geq 0
\end{align*}
\]

On occasion, it is useful to consider the matrix $A$ as a collection of columns or rows. We denote by $A_i$ the $i$th row of $A$, and by $A^j$ the $j$th column of $A$. Using the column vector notation the standard form becomes:

\[
\begin{align*}
\text{minimize} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} A^j x_j = b \\
& \quad x_j \geq 0
\end{align*}
\]

Sometimes we include the coefficients of the objective form in the column vectors yielding:
where $\hat{A}^i$ extends $A^i$ by adding a zeroth component of $c_i$. Similarly $b$ is extended by adding a 0 as the zeroth component. \( U_0 \) is a unit vector except with one 0 in the 0th component.

The symmetric form using the row vector notation becomes:

$$
\begin{align*}
\text{minimize} & \quad x_0 \\
\text{subject to} & \quad -U_0 x + \sum_{j=1}^{n} \hat{A}^j x_j = \hat{b} \\
& \quad \text{and} \quad x_j \geq 0 \quad j = 1, \ldots, n
\end{align*}
$$

These show that linear programs can be considered as finding the cheapest combination of column vectors, or as finding a vector which satisfies a number of linear relations with the cheapest cost. Shortly we will see geometric representations of both these interpretations.
Exercises:

1. Convert the investment problem of the Aunt's money into standard form.

2. Convert the following linear program to standard form.

\[
\begin{align*}
\text{maximize} & \quad -x_1 + x_2 + x_3 + x_4 \\
\text{subject to} & \quad x_1 + 2x_3 &= -1 \\
& \quad x_2 + x_3 + x_4 \leq 4 \\
& \quad x_1 + x_2 + x_3 + x_4 \geq 0 \\
\text{and} & \quad x_2 \geq 0, \quad x_4 \geq 0.
\end{align*}
\]