

Notes on Linear Programming
CE 377K
Stephen D. Boyles
Spring 2015

1 Introduction

Linear programs are a special class of optimization problem, with restrictions that both the objective function and constraints are linear (technically, affine¹) functions of the vector of decision variables \mathbf{x} . The special structure of linear programs allows us to use specialized methods which are fast, and proven to find the exact optimal solution in a finite number of iterations.

The quintessential algorithm for solving linear programs is the simplex method, which has an intuitive geometric interpretation and is amenable to efficient computer implementation. A compact representation of this algorithm demands a new standard form for linear programs, described in the following section. The simplex method works by considering extreme points of the feasible region (a well-defined notion if all of the constraints are linear). This is not restrictive: an examination of the geometry of linear programs shows that if a finite optimal solution exists, one of the extreme points must be optimal. The simplex tableau is a compact and efficient form for performing the simplex algorithm's manipulations.

One other advantage of linear programs is the ease of performing sensitivity analysis: that is, how would the objective function and optimal decision change if some of the input problem data were to change; this is covered at the end of these notes.

Certain examples in these notes were adapted from *Introduction to Linear Optimization*, by Bertsimas and Tsitsiklis.

¹The difference is that a linear function cannot include a constant term, while an affine function can. For instance, $f(x) = 3x + 2$ is affine but not linear, while $f(x) = 3x$ is both affine and linear. However the word “linear” is often used for both cases.

2 What is a linear program?

The standard form of a linear program is

$$\begin{aligned} \min_{x_1, \dots, x_n} \quad & \sum_{i=1}^n c_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^n a_{i1} x_i = b_1 \\ & \vdots \\ & \sum_{i=1}^n a_{im} x_i = b_m \\ & x_1, \dots, x_n \geq 0 \end{aligned}$$

Notice that all of the constraints are equalities (rather than inequalities), and that each decision variable is required to be nonnegative. Depending on the problem, it may take some effort to transform a given linear problem to standard form. The following are common “tricks” for doing so:

Inequality constraints: A constraint of the form $\sum_{i=1}^n a_{ij} x_i \leq b_j$ can be converted to standard form by adding a new slack variable x_{n+1} , rewriting the constraint as $\sum_{i=1}^n a_{ij} x_i + x_{n+1} = b_j$ and requiring $x_{n+1} \geq 0$. Similarly, if $\sum_{i=1}^n a_{ij} x_i \geq b_j$, then the constraint can be rewritten as $\sum_{i=1}^n a_{ij} x_i - x_{n+1} = b_j$ with $x_{n+1} \geq 0$. The slack variable can be interpreted as the “distance” between the actual value of the left hand side, and its upper or lower bound. We are justified in making this a decision variable: when we decide one, we decide the other, so the original problem is essentially unchanged.

Free variables: If a variable x_i is not constrained to be nonnegative in the original problem, we can introduce a new pair of nonnegative decision variables x_i^+ and x_i^- , and substitute $x_i = x_i^+ - x_i^-$ throughout the problem. One can interpret x_i^+ as the “positive” component of x_i and x_i^- the “negative component.” If we want x_i to be positive, we could set $x_i^+ = x_i$ and $x_i^- = 0$. If x_i is negative, we could set $x_i^+ = 0$ and $x_i^- = |x_i|$. Either way, these two nonnegative variables can represent the original free variable.

The standard form is important because of the simplex algorithm; so unless we’re using this algorithm, there is no need to write a linear program in standard form.

A more compact way to write a standard-form linear program is to use vector/matrix notation:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c} \cdot \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

where $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$, \mathbf{A} is a $m \times n$ matrix, and $\mathbf{b} \in \mathbb{R}^m$ with m and n denoting the number of constraints and decision variables, respectively.

3 Linear Programming Geometry

Because all of the constraints are linear functions, the feasible region will have well-defined corner points defined by the intersection points of the constraint equations. (Figure 1). The corner points are usually called *extreme points*, which are defined as points not belonging to the interior of any line segment in the feasible set. Figure 1 shows some examples of these: points A and B are not extreme points, since they lie on the interior of feasible line segments, while points C and D are extreme points since they must be an endpoint of any feasible line segment containing them. Clearly, if the problem is feasible, the number of extreme points is finite. Extreme points are important because of the following result:

Theorem 1. *If a linear program has a global optimal solution and the feasible region has an extreme point, at least one of the extreme points is globally optimal.*

Here is some quick (but imprecise) justification for the theorem: let \mathbf{x}^* represent a globally optimal solution to a linear programming problem, and let $\bar{\mathbf{x}}$ represent any extreme point. We assume \mathbf{x}^* is not an extreme point, or else the theorem is trivial. Construct the function $f(\lambda) = \mathbf{c} \cdot ((1-\lambda)\mathbf{x}^* + \lambda\bar{\mathbf{x}})$. When $\lambda \in [0, 1]$, $(1-\lambda)\mathbf{x}^* + \lambda\bar{\mathbf{x}}$ is feasible. Further, f is differentiable. There are three possibilities: if $f'(0) < 0$, \mathbf{x}^* is not optimal, because $f(\delta) < f(0)$ for some small positive δ . Likewise, if $f'(0) > 0$, $f(-\delta) < f(0)$ for some small δ and again \mathbf{x}^* is suboptimal. Thus $f'(0) = 0$. By linearity of the objective function, f' is uniformly equal to zero on $[0, 1]$, and thus $f(0) = f(1) \iff \mathbf{c} \cdot \mathbf{x}^* = \mathbf{c} \cdot \bar{\mathbf{x}}$ and $\bar{\mathbf{x}}$ is optimal as well.

This argument is not entirely rigorous (why?), but illustrates the general idea. Curious readers are referred to Bertsimas & Tsitsiklis' book for a formal proof.

Because of this theorem, this means that we only need to look at extreme points of the feasible region, and it is precisely this property which lets us solve linear programs exactly. Even though there may be infinitely many feasible solutions, there are only finitely many extreme points, which we can check one at a time. The simplex method provides a more efficient way of doing this, but in principle you can solve any linear program by performing the following two steps:

1. Identify all of the extreme points.
2. Calculate the value of the objective function at each extreme point.

This will allow us to find the optimum solution. So how do we find the extreme points?

Recall that extreme points must (1) be feasible and (2) lie at the intersection of multiple constraints. How many constraints? In two dimensions, the constraints are lines, and two lines are needed to identify an intersection point. In three dimensions, linear constraints are planes, and three planes are needed to identify an intersection point (the intersection of two planes is a line, not a point; in general, the intersection of four or more planes is empty). In general, if there are n decision variables, we need n active constraints to identify an intersection point.

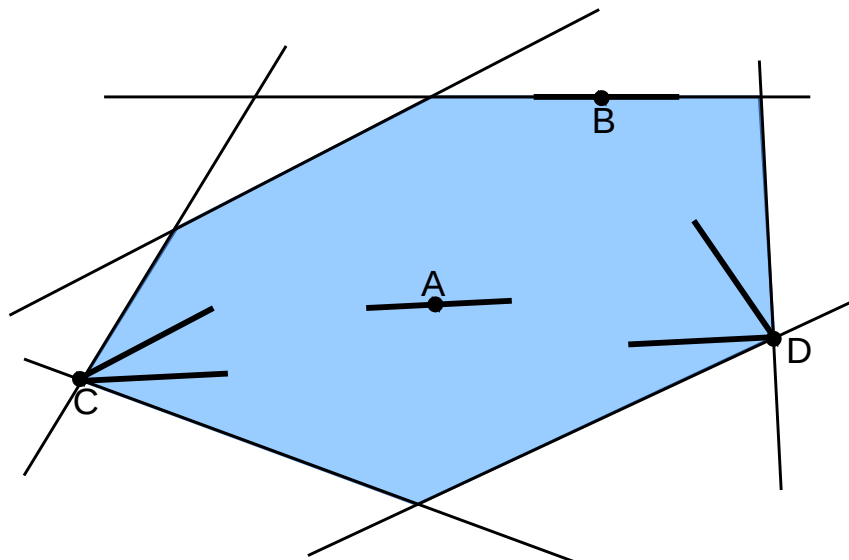


Figure 1: Extreme points of a linear program.

Any equality constraints ('=') must be satisfied no matter what; let's say there are m of these. The remaining $n - m$ constraints can be chosen from the inequality constraints. Replacing the inequality with an equality, we can solve for the point x_1, \dots, x_n which solves these n equations. If this point satisfies all of the remaining constraints, it is feasible, and thus an extreme point.

Example 1. Identify all of the extreme points of the feasible region in \mathbb{R}^2 defined by the constraints

$$x_1 + x_2 \leq 9 \tag{1}$$

$$x_1 \leq 6 \tag{2}$$

$$x_2 \leq 6 \tag{3}$$

$$x_1 \geq 0 \tag{4}$$

$$x_2 \geq 0 \tag{5}$$

In this example, extreme points correspond to the intersection of two lines, because there are two decision variables. Furthermore, there are no equality constraints, so both of these lines are inequality constraints which we convert to equalities. There are 10 possible pairs:

- (1) and (2). Solving $x_1 + x_2 = 9$ and $x_1 = 6$ simultaneously gives $(x_1, x_2) = (6, 3)$. This satisfies the other constraints, so this is an extreme point.
- (1) and (3). Solving $x_1 + x_2 = 9$ and $x_2 = 6$ simultaneously gives $(x_1, x_2) = (3, 6)$. This satisfies the other constraints, so this is an extreme point.
- (1) and (4). Solving $x_1 + x_2 = 9$ and $x_1 = 0$ simultaneously gives $(x_1, x_2) = (0, 9)$. This violates constraint (3), so this is **not** an extreme point.

- (1) and (5). Solving $x_1 + x_2 = 9$ and $x_2 = 0$ simultaneously gives $(x_1, x_2) = (9, 0)$. This violates constraint (2), so this is **not** an extreme point.
- (2) and (3). Solving $x_1 = 6$ and $x_2 = 6$ simultaneously gives $(x_1, x_2) = (6, 6)$. This violates constraint (1), so this is **not** an extreme point.
- (2) and (4). $x_1 = 6$ and $x_1 = 0$ cannot be solved simultaneously, so this is **not** an extreme point.
- (2) and (5). $x_1 = 6$ and $x_2 = 0$ satisfies all of the constraints, so this is an extreme point.
- (3) and (4). $x_2 = 6$ and $x_1 = 0$ satisfies all of the constraints, so this is an extreme point.
- (3) and (5). $x_2 = 6$ and $x_2 = 0$ cannot be solved simultaneously, so this is **not** an extreme point.
- (4) and (5). $x_1 = 0$ and $x_2 = 0$ satisfies all of the constraints, so this is an extreme point.

Thus, the five extreme points are $(0, 0)$, $(0, 6)$, $(6, 0)$, $(6, 3)$ and $(3, 6)$.

As we can see, extreme points can be identified by the active constraints at that point: for instance, $(0, 0)$ is defined by the intersection of (4) and (5). This leads to the concept of *adjacent* extreme points. Two extreme points are called adjacent if all but one of the active constraints at these points are the same. For instance, $(0, 0)$ is adjacent to $(0, 6)$, because the sets of active constraints are (4) and (5) in the first case, and (4) and (3) in the second. On the other hand, $(6, 0)$ is *not* adjacent to $(0, 6)$, because the set of active constraints for the former is (2) and (5), neither of which is active at $(0, 6)$.

4 Simplex Method: Algebra

The simplex method is one of the most common algorithms for solving linear programs. Carefully implemented, it is guaranteed to terminate with an optimal solution after a finite number of iterations, and performs very well in practice. Theorem 1 is the key to the simplex method: since there must be an optimal extreme point² we can confine our attention to extreme points. We did this in the previous section, but it is tedious to solve multiple systems of equations, especially as the dimension of the problem grows large.

The simplex method moves from one extreme point to another, adjacent extreme point where the objective function is either the same, or better. The simplex method is highly based in matrix algebra, and solutions of simultaneous equations. In fact, this turns out to be the motivating factor behind writing the constraints as $\mathbf{Ax} = \mathbf{b}$ in standard form. Recall that \mathbf{A} is an $m \times n$ matrix, representing the m constraints and n variables. We can safely assume that $m \leq n$: if $m > n$, some

²Assuming that the problem is feasible and that a global optimum exists, of course. We take these assumptions throughout these notes unless otherwise stated.

of the constraints are linearly dependent, meaning the problem is either infeasible, or that one or more constraints is redundant and can be eliminated.

Now, recall that (a) an extreme point consists of the solution to n simultaneous equations; (b) $n \geq m$; and (c) all m equality constraints must be satisfied for feasibility. This means we must pick $n - m$ of the nonnegativity constraints to be binding. Equivalently, we force $n - m$ decision variables to equal zero, and allow the remaining m decision variables to take nonzero values. These m variables are called a *basis*, and for a standard-form linear program, these characterize the set of extreme points.

Let \mathbf{x}^B and \mathbf{x}^{NB} represent the vectors of basic and nonbasic decision variables. By definition, $\mathbf{x}^{NB} \equiv \mathbf{0}$. We can also separate the columns of \mathbf{A} according to whether the corresponding decision variable is basic or nonbasic, producing matrices \mathbf{B} and \mathbf{N} . Because we are free to rearrange the columns of \mathbf{A} in any order we choose (so long as we rearrange \mathbf{x} and \mathbf{c} in the same way), without loss of generality we can assume that $\mathbf{A} = [\mathbf{B}|\mathbf{N}]$, $\mathbf{x} = [\mathbf{x}^B|\mathbf{x}^{NB}]$, and $\mathbf{c} = [\mathbf{c}^B|\mathbf{c}^{NB}]$ (that is, that we list the basic variables first).

With this convention, the constraint equation $\mathbf{Ax} = \mathbf{b}$ can be written $\mathbf{Bx}^B + \mathbf{Nx}^{NB} = \mathbf{b}$. Because $\mathbf{x}^{NB} = \mathbf{0}$, we simply have $\mathbf{Bx}^B = \mathbf{b}$ or $\mathbf{x}^B = \mathbf{B}^{-1}\mathbf{b}$. This represents exactly the same procedure applied in the previous section; we are solving a system of m equations in m unknowns, where the coefficients are given by the basic columns B .

Assume we have a feasible basis (that is, a basis matrix B such that $\mathbf{x}^B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$). This represents an extreme point; we want to know if we can improve the objective by moving to an adjacent extreme point. Recall that all but one of the binding constraints at an adjacent extreme point is the same. From the standpoint of bases, this means all but one of the basic variables is the same; that is, a nonbasic decision variable must enter the basis (become nonzero), and a basic decision variable must leave the basis (become zero).

Let's say that we pick some nonbasic decision variable x^k to enter the basis. We need to see whether this will improve the objective function. It's not obvious how to do this, because we can't just change x^k and leave the other variables at their old values. Instead, to maintain the constraints $\mathbf{Ax} = \mathbf{b}$, we will have to change the basic variables x^B simultaneously, that is, to $\mathbf{x}^B + \mathbf{d}^B$. If $x^k = \lambda$, we have to satisfy $\mathbf{B}(\mathbf{x}^B + \mathbf{d}^B) + \mathbf{A}^k\lambda = \mathbf{b}$, where \mathbf{A}^k is the k -th column of the constraint matrix. Simplifying, we obtain

$$\begin{aligned}\mathbf{B}(\mathbf{x}^B + \mathbf{d}^B) + \mathbf{A}^k\lambda &= \mathbf{b} \\ \iff \mathbf{Bd}^B + \lambda\mathbf{A}^k &= \mathbf{0} \\ \iff \mathbf{d}^B &= -\lambda\mathbf{B}^{-1}\mathbf{A}^k\end{aligned}$$

providing a formula showing how the basic variables must change if we set $x^k = \lambda$. For shorthand, we write $\Delta\mathbf{x}^B = \mathbf{B}^{-1}\mathbf{A}^k$.

This allows us see how the objective function will change. If x^k increases from 0 to λ , certainly the objective function will increase by $c^k\lambda$. At the same time, the change in the basic variables will decrease the objective function by $\lambda\mathbf{c}^B\mathbf{B}^{-1}\mathbf{A}^k$. If the net effect is negative, the objective function

improves by adding x^k to the basis. This can be written as $\lambda \bar{c}^k$ where $\bar{c}^k = c^k - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{A}^k$ is called the *reduced cost* of the k -th decision variable.

Of course, if x^k enters the basis, some other decision variable should leave it. Because the problem is linear, if it is beneficial to increase x^k from zero to some positive number, we want to make x^k as large as possible (unlike nonlinear programs, there is no worry about taking too far a step). Taking into account the change in the basic variables, we can continue to increase x^k as long as each element of \mathbf{x}^B is strictly positive. Once any of the basic variables equals zero, we can't move any further. That variable then leaves the basis, and we finish moving to the new extreme point.

The first basic variable to equal zero can be found by examining the elements of \mathbf{x}^B and $\Delta \mathbf{x}^B$. The value of the i -th basic variable is x_i ; if $\Delta x_i < 0$ (that is, x_i decreases when x^k is added to the basis), we need $x_i + \lambda \Delta x_i \geq 0 \iff \lambda \leq -x_i / \Delta x_i$. If $\Delta x_i \geq 0$, x_i will never run afoul of its nonnegativity constraint, and plays no role in bounding λ . Because these restrictions must hold for all i , we have $x^k = \lambda = \min_{i: \Delta x_i < 0} x_i / |\Delta x_i|$, and the decision variable leaving the basis is one where this minimum value is obtained.

Thus, the basic simplex method can be presented as follows:

1. Identify an initial feasible basis $\{x_1^B, \dots, x_m^B\}$.
2. Calculate the reduced cost $\bar{c}^k = c^k - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{A}^k$ for each nonbasic decision variable.
3. If all of the reduced costs are nonnegative, the current basis is optimal. **STOP**.
4. Otherwise, choose some nonbasic decision variable x^k with a negative reduced cost to enter the basis.
5. Set $x^k = \min_{i: \Delta x_i < 0} x_i / |\Delta x_i|$; the basic variable leaving the basis is one where this minimum is obtained.
6. Return to step 2.

Example 2. *Solve the linear program*

$$\begin{array}{ll}
 \min_{x_1, x_2, x_3} & -10x_1 - 12x_2 - 12x_3 \\
 \text{s.t.} & x_1 + 2x_2 + 2x_3 \leq 20 \\
 & 2x_1 + x_2 + 2x_3 \leq 20 \\
 & 2x_1 + 2x_2 + x_3 \leq 20 \\
 & x_1, x_2, x_3 \geq 0
 \end{array}$$

using the simplex method.

Solution. We first have to write the problem in standard form:

$$\begin{array}{ll}
\min_{x_1, \dots, x_6} & -10x_1 - 12x_2 - 12x_3 \\
\text{s.t.} & x_1 + 2x_2 + 2x_3 + x_4 = 20 \\
& 2x_1 + x_2 + 2x_3 + x_5 = 20 \\
& 2x_1 + 2x_2 + x_3 + x_6 = 20 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{array}$$

By inspection, a feasible basis is $\{x_4, x_5, x_6\}$, which corresponds to the the extreme point $(0, 0, 0, 20, 20, 20)$. We have

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

so $\mathbf{B}^{-1} = \mathbf{B}$ and

$$\begin{aligned}
\bar{\mathbf{c}} = \mathbf{c} - \mathbf{c}^{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A} &= [-10 \quad -12 \quad -12 \quad 0 \quad 0 \quad 0] - [0 \quad 0 \quad 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 2 & 0 & 1 & 0 \\ 2 & 2 & 1 & 0 & 0 & 1 \end{bmatrix} \\
&= [-10 \quad -10 \quad -12 \quad 0 \quad 0 \quad 0]
\end{aligned}$$

The reduced cost of x_1 is negative, so we choose it to enter the basis. We calculate

$$\Delta \mathbf{x}^{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{A}^1 = \mathbf{A}^1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

Checking the ratio test, we see that $x_4/\Delta x_4 = 20$, $x_5/\Delta x_5 = 10$, and $x_6/\Delta x_6 = 10$, so one of x_5 and x_6 must leave the basis; let's assume that x_5 leaves, so the new basis is $\{x_4, x_1, x_6\}$. Now

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \end{bmatrix} \quad \mathbf{B}^{-1} = \begin{bmatrix} 1 & -0.5 & 0 \\ 0 & 0.5 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

so the current solution is $\mathbf{B}^{-1}\mathbf{b} = [10 \quad 0 \quad 0 \quad 10 \quad 0 \quad 0]$ and

$$\begin{aligned}
\bar{\mathbf{c}} &= [-10 \quad -12 \quad -12 \quad 0 \quad 0 \quad 0] - [0 \quad -10 \quad 0] \begin{bmatrix} 1 & -0.5 & 0 \\ 0 & 0.5 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 2 & 0 & 1 & 0 \\ 2 & 2 & 1 & 0 & 0 & 1 \end{bmatrix} \\
&= [0 \quad -7 \quad -2 \quad 0 \quad 5 \quad 0]
\end{aligned}$$

The reduced cost of x_3 is negative, so we choose it to enter the basis. We calculate

$$\Delta \mathbf{x}^{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{A}^3 = \begin{bmatrix} 1 & -0.5 & 0 \\ 0 & 0.5 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

Checking the ratio test, we see that $x_4/\Delta x_4 = 10$ and $x_1/\Delta x_1 = 10$, so one of x_4 and x_1 must leave the basis; let's assume that x_4 leaves, so the new basis is $\{x_3, x_1, x_6\}$. Now

$$\mathbf{B} = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 2 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \mathbf{B}^{-1} = \begin{bmatrix} 1 & -0.5 & 0 \\ -1 & 1 & 0 \\ 1 & -1.5 & 1 \end{bmatrix}$$

so the current solution is $\mathbf{B}^{-1}\mathbf{b} = [0 \ 0 \ 10 \ 0 \ 0 \ 10]$ and

$$\begin{aligned} \bar{\mathbf{c}} &= [-10 \ -12 \ -12 \ 0 \ 0 \ 0] - [-12 \ -10 \ 0] \begin{bmatrix} 1 & -0.5 & 0 \\ -1 & 1 & 0 \\ 1 & -1.5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 2 & 0 & 1 & 0 \\ 2 & 2 & 1 & 0 & 0 & 1 \end{bmatrix} \\ &= [0 \ -4 \ 0 \ 2 \ 4 \ 0] \end{aligned}$$

The reduced cost of x_2 is negative, so we choose it to enter the basis. We calculate

$$\Delta \mathbf{x}^{\mathbf{B}} = \mathbf{B}^{-1}A^2 = \begin{bmatrix} 1 & -0.5 & 0 \\ -1 & 1 & 0 \\ 1 & -1.5 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1.5 \\ -1 \\ 2.5 \end{bmatrix}$$

Checking the ratio test, we see that $x_3/\Delta x_3 = 6.67$ and $x_6/\Delta x_6 = 4$, so x_6 must leave the basis, and the new basis is $\{x_3, x_1, x_2\}$. Now

$$\mathbf{B} = \begin{bmatrix} 2 & 1 & 2 \\ 2 & 2 & 1 \\ 1 & 2 & 2 \end{bmatrix} \quad \mathbf{B}^{-1} = \begin{bmatrix} 0.4 & 0.4 & -0.6 \\ -0.6 & 0.4 & 0.4 \\ 0.4 & -0.6 & 0.4 \end{bmatrix}$$

so the current solution is $\mathbf{B}^{-1}\mathbf{b} = [4 \ 4 \ 4 \ 0 \ 0 \ 0]$ and

$$\begin{aligned} \bar{\mathbf{c}} &= [-10 \ -12 \ -12 \ 0 \ 0 \ 0] - [-12 \ -10 \ -12] \begin{bmatrix} 0.4 & 0.4 & -0.6 \\ -0.6 & 0.4 & 0.4 \\ 0.4 & -0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 2 & 0 & 1 & 0 \\ 2 & 2 & 1 & 0 & 0 & 1 \end{bmatrix} \\ &= [0 \ 0 \ 0 \ 3.6 \ 1.6 \ 1.6] \end{aligned}$$

so all reduced costs are nonnegative, and the current basis is optimal. Thus the optimal solution is $\mathbf{x}^* = [4 \ 4 \ 4 \ 0 \ 0 \ 0]$, and the optimal objective function value is $\mathbf{c} \cdot \mathbf{x}^* = -136$. ■

5 Simplex Method: Tableau

As presented in the previous section, the simplex method still involves a lot of computational overhead, namely, inverting the basis matrix repeatedly. Much of this work is redundant, because $m - 1$ columns of the basis matrix are identical to the previous iteration. The tableau method provides a compact and efficient way to calculate these without duplicating effort. The tableau method maintains a table containing the following elements:

$-Z$	\dots	$c^k - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{A}^k$	\dots
\vdots			
x_i^B	\dots	$\mathbf{B}^{-1} \mathbf{A}^k$	\dots
\vdots			

Key features of the simplex tableau are

- The tableau contains $n + 1$ columns and $m + 1$ rows, where n is the number of decision variables and m the number of constraints.
- We will start counting rows and columns from zero, so the top row and left column are the zero-th. This way, the n -th column corresponds to x_n , and the m -th row corresponds to the m -th basic variable. We describe the entry in the i -th row and j -th column by \mathcal{T}_{ij} .
- The upper left cell contains the *negative* of the objective function value.
- The remainder of the top row contains the reduced cost of each decision variable.
- After the top row, each row corresponds to one of the basic variables. The leftmost column contains the value of that basic variable, and the other columns contain the appropriate element of $\mathbf{B}^{-1} \mathbf{A}^k$ (the amount by which that basic variable *decreases* if x^k were to enter the basis).

To see how to use the simplex tableau, recall the step-by-step algebraic version of the simplex method given in the previous section. After initialization, each of these steps can be performed using the tableau.

1. An initial feasible basis must still be found; the big- M method in the next section shows how to provide this.
2. The reduced costs for all variables are present in the first row of the tableau.
3. We can visually see whether all the reduced costs are positive, in which case we can stop.
4. Otherwise, we can visually pick one of the x_k with a negative reduced cost to enter the basis.
5. To find out which variable leaves, we can calculate the ratio between the current value of each basic variable (in the leftmost column) and the amount by which that variable decreases when x_k enters (in the column corresponding to x_k).
6. Update the reduced costs and the $\mathbf{B}^{-1} \mathbf{A}^k$, and return to step 2.

All of these steps should be clear, except for the last. Let's assume that x_k is entering the basis, and x_ℓ is leaving. Because x_k is taking the place of x_ℓ in the basis, we want to make the k -th column look like the ℓ -th column currently does. We do this through the use of matrix row manipulations:

1. Divide every element in the ℓ -th row by $\mathcal{T}_{\ell k}$. This sets $\mathcal{T}_{\ell k} = 1$
2. For each of the other rows i , multiply the ℓ -th row by \mathcal{T}_{ik} and subtract it from the i -th row. This sets $\mathcal{T}_{ik} = 0$ for all $i \neq \ell$.

Caution! Although this resembles the row reductions used to solve a system of equations, it is not quite the same procedure, and these matrix operations must be performed in this order. A different sequence of row reductions will result in a different tableau, even if the k -th column looks correct.

Example 3. Resolve Example 2 using a tableau.

Solution. Starting as before, we can write the initial tableau

0	-10	-12	-12	0	0	0
20	1	2	2	1	0	0
20	2	1	2	0	1	0
20	2	2	1	0	0	0

Examining the reduced costs in the first row, we see that the reduced cost of x_1 is negative, so it enters the basis. Checking the ratio test involves calculating $\mathcal{T}_{10}/\mathcal{T}_{11}$, $\mathcal{T}_{20}/\mathcal{T}_{21}$, and $\mathcal{T}_{30}/\mathcal{T}_{31}$, or 20, 10, and 10 respectively. We thus choose x_5 to leave the basis. The first row operation thus requires dividing the second row by 2:

0	-10	-12	-12	0	0	0
20	1	2	2	1	0	0
10	1	0.5	1	0	0.5	0
20	2	2	1	0	0	0

We then add ten times the second row to the zero-th row; subtract the second row from the first row; and subtract twice the second row from the third row:

100	0	-7	-2	0	5	0
10	0	1.5	1	1	-0.5	0
10	1	0.5	1	0	0.5	0
0	0	1	-1	0	-1	1

This completes the first iteration. Notice that we can tell which variables are basic by seeing which columns form the identity matrix in the lower-right section of the tableau. Taking the fourth, first, and sixth columns (in that order) forms the identity matrix, so the basis is $\{x_4, x_1, x_6\}$ and their values are given by the zero-th column (in that order). Returning to the second step, we see that the reduced cost of x_3 is negative, so it enters the basis. Checking the ratio test, we calculate $10/1 = 10$ for x_4 and $10/1 = 10$ for x_1 , ignoring x_6 because $\mathcal{T}_{33} < 0$. We force x_1 to leave the basis. Thus, we divide the first row by one (no change), add twice the first row to the zero-th; subtract the first row from the second; and add the first row to the third:

120	0	-4	0	2	4	0
10	0	1.5	1	1	-0.5	0
0	1	-1	0	-1	1	0
10	0	2.5	0	1	-1.5	1

This completes the second iteration. The reduced cost of x_2 is negative, so it enters the basis. Checking the ratio test, we calculate $10/1.5 = 6.67$ for x_3 and $10/2.5 = 4$ for x_6 . Thus x_6 leaves the basis; and we divide the third row by 2.5, then add four times the third row to the zero-th, subtract 1.5 times the third row from the first, and add the third row to the second:

136	0	0	0	3.6	1.6	1.6
4	0	0	1	0.4	0.4	-0.6
4	1	0	0	-0.6	0.4	0.4
4	0	1	0	0.4	-0.6	0.4

All reduced costs are nonnegative, so we have found the optimal solution. The identity matrix can be found in the third, first, and second columns, so the basis is $\{x_3, x_1, x_2\}$, the final optimal solution is $\mathbf{x} = [4 \ 4 \ 4 \ 0 \ 0 \ 0]$, and the optimal objective function value is -136 . ■

6 The Big- M Method

Whether we needed a tableau or not, we needed a feasible basic solution. For large problems, this may not be easy or trivial; and if possible, we'd like to never have to worry about analytically inverting a matrix at all. The big- M method gets around this problem. Consider the optimization problem

$$\begin{array}{llll}
\min_{x_1, \dots, x_4} & x_1 + x_2 + x_3 & & \\
\text{s.t.} & x_1 + 2x_2 + 3x_3 & = & 3 \\
& -x_1 + 2x_2 + 6x_3 & = & 2 \\
& 4x_2 + 9x_3 & = & 5 \\
& 3x_3 + x_4 & = & 1 \\
& x_1, x_2, x_3, x_4 & \geq & 0
\end{array}$$

The constraint matrix for this problem is

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 0 \\ -1 & 2 & 6 & 0 \\ 0 & 4 & 9 & 0 \\ 0 & 0 & 3 & 1 \end{bmatrix}$$

and to find an initial basis, we want a 4×4 identity matrix somewhere in \mathbf{A} . The last column of \mathbf{A} can serve as the fourth column of the identity matrix, but we need the other three columns.

The easiest way to do this is to add three new decision variables to the problem, with the right constraints to form an identity matrix:

$$\begin{aligned}
\text{s.t. } x_1 + 2x_2 + 3x_3 + x_5 &= 3 \\
&- x_1 + 2x_2 + 6x_3 + x_6 &= 2 \\
4x_2 + 9x_3 + x_7 &= 5 \\
3x_3 + x_4 &= 1 \\
x_1, x_2, x_3, x_4 &\geq 0
\end{aligned}$$

so the new constraint matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 0 & 1 & 0 & 0 \\ -1 & 2 & 6 & 0 & 0 & 1 & 0 \\ 0 & 4 & 9 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 1 & 0 & 0 & 0 \end{bmatrix}$$

However, we don't want these new variables to appear in the solution, since they aren't part of the original problem. Since it's a minimization problem, we multiply these variables by a very large, positive constant M in the objective function:

$$\min_{x_1, \dots, x_7} x_1 + x_2 + x_3 + Mx_5 + Mx_6 + Mx_7$$

The exact value of M isn't important so long as it's much larger than all of the problem data. Alternately, you don't need to fix a value for M , as long as you treat M as larger than any other number whenever performing any comparisons.

Returning to this problem, we can now see that $\{x_5, x_6, x_7, x_4\}$ forms a basis, corresponding to the extreme point $(0, 0, 0, 1, 3, 2, 5)$. The objective function value corresponding to this solution is $10M$, which is extremely large. However, the first few iterations of the simplex method will reduce this dramatically. We still have to calculate the initial reduced costs for starting the simplex method:

$$\begin{aligned}
\bar{\mathbf{c}} &= \mathbf{c} - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{A} = \\
&[1 \quad 1 \quad 1 \quad 0 \quad M \quad M \quad M] - [M \quad M \quad M \quad 0] \mathbf{I} \begin{bmatrix} 1 & 2 & 3 & 0 & 1 & 0 & 0 \\ -1 & 2 & 6 & 0 & 0 & 1 & 0 \\ 0 & 4 & 9 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 1 & 0 & 0 & 0 \end{bmatrix} \\
&= [-1 \quad -8M + 1 \quad -18M + 1 \quad 0 \quad 0 \quad 0 \quad 0]
\end{aligned}$$

so the initial tableau is

$-10M$	1	$-8M + 1$	$-18M + 1$	0	0	0	0
3	1	2	3	0	1	0	0
2	-1	2	6	0	0	1	0
5	0	4	9	0	0	0	1
1	0	0	3	1	0	0	0

The reduced cost of x_3 is negative (because M is very large), so x_3 enters the basis. Using the ratio test, we see that x_4 (the last row) must exit. Thus, we divide by fourth row by three; then add $(18M - 1)$ times the fourth row to the zero-th, subtract thrice the fourth row from the first, subtract six times the fourth row from the second, and nine times the fourth row from the third:

$-4M - 1/3$	1	$-8M + 1$	0	$6M - 1/3$	0	0	0
2	1	2	0	-1	1	0	0
0	-1	2	0	-2	0	1	0
2	0	4	0	-3	0	0	1
1/3	0	0	1	1/3	0	0	0

The reduced cost of x_2 is negative, so it enters the basis and x_6 exits from the ratio test (the second row). So, we divide the second row by two; add $(8M - 1)$ times the second row from the zero-th; subtract twice the second row from the first; and subtract four times the second row from the third:

$-4M - 1/3$	$-4M + 3/2$	0	0	$-2M + 2/3$	0	$4M - 1/2$	0
2	2	0	0	-1	1	-1	0
0	-1/2	1	0	-1	0	1/2	0
2	2	0	0	1	0	-2	1
1/3	0	0	1	1/3	0	0	0

Now, x_1 enters and x_5 leaves:

$-11/6$	0	0	0	$-1/12$	$2M - 3/4$	$2M + 1/4$	0
1	1	0	0	1/2	1/2	-1/2	0
1/2	0	1	0	-3/4	1/4	1/4	0
0	0	0	0	0	-1	-1	1
1/3	0	0	1	1/3	0	0	0

Then x_4 enters and x_3 leaves:

$-7/4$	0	0	1/4	0	$2M - 3/4$	$2M + 1/4$	0
1/2	1	0	-3/2	0	1/2	-1/2	0
5/4	0	1	9/4	0	1/4	1/4	0
0	0	0	0	0	-1	-1	1
1	0	0	3	1	0	0	0

All of the reduced costs are now nonnegative (since $2M \gg 3/4$), so this is an optimal basis. The identity matrix can now be found in the first, second, seventh, and fourth columns, so the optimal basis is $\{x_1, x_2, x_7, x_4\}$ and the optimal solution is $\{1/2, 5/4, 0, 1, 0, 0, 0\}$. Note that M drove all of the “artificial” variables out of the optimal solution. (Although x_7 is in the optimal basis, its value is zero.) If any of the artificial variables were *positive* in an optimal solution, the original problem was infeasible, because M is so large that literally any feasible solution to the original problem is better than any solution involving a strictly positive artificial variable.

7 Sensitivity Analysis

One of the main advantages linear programs have over nonlinear programs is the ease of performing sensitivity analysis, that is, studying how optimal solutions change if the problem input data (such as the objective function, constraint matrix, or requirement vector \mathbf{b}) change. With linear programs, analytical formulas are readily available, and insightful theoretical results can be obtained. While this may be true for certain nonlinear programs, oftentimes there is nothing better than simply re-solving the problem from scratch.

This section discusses the two most common types of sensitivity analysis needed for linear programs: changes in the cost vector \mathbf{c} , and changes in the requirement vector \mathbf{b} . Changes to the constraint matrix \mathbf{A} are less common in practice (\mathbf{A} usually represents factors more under your control, and more predictable, while \mathbf{c} and \mathbf{b} depend more on external factors which are less well known). Further, studying the impact of these changes is more complicated mathematically, and not treated in these notes.

The basic question can be formulated like this: we have an optimal basis B and an optimal solution \mathbf{x}^* to the standard-form linear program $\min_{\mathbf{x}} \mathbf{c} \cdot \mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$. If the objective function changes slightly (say, to $\mathbf{c} + \delta \mathbf{e}_i$, where \mathbf{e}_i represents the i -th unit vector $[e_1 = 0 \ \dots \ e_{i-1} = 0 \ e_i = 1 \ e_{i+1} = 0 \ \dots]$), is B still an optimal basis, or do we need to redo the simplex method? Likewise, if the requirement vector changes to $\mathbf{b} + \delta \mathbf{e}_j$, do we need to change the basis? In the latter case, the actual optimal solution $\mathbf{x}^* = \mathbf{B}^{-1}\mathbf{b}$ will almost certainly change, but the same basis may still be optimal.

In short, there are two things which can cause the previous optimal basis to lose their optimality.

1. **The original basis is no longer feasible.** If $\mathbf{x}^B = \mathbf{B}^{-1}\mathbf{b} \not\geq \mathbf{0}$, we need to find a new basis which satisfies the nonnegativity constraints.
2. **The original basis is no longer optimal.** Recall that a basis is optimal if the reduced cost of every other decision variable is nonnegative. If, in the new problem, some variable has a strictly negative reduced cost, the original basis is not optimal. The simplex method should continue by having that variable with a negative reduced cost enter the basis.

As you might suspect, these two cases correspond exactly to changes in the requirement vector, and to changes in the cost vector.

7.1 Changes to the requirement vector \mathbf{b}

Assume that the j -th component of \mathbf{b} changes by δ , so $\mathbf{b}' = \mathbf{b} + \delta \mathbf{e}_j$. The reduced costs are unaffected by this change: they depend on \mathbf{c} , \mathbf{B} , and \mathbf{A} , but *not* the requirement vector \mathbf{b} . Therefore, as long as this basis is feasible, it will remain optimal.

The question, then, is what range of δ maintains feasibility of the original basis B . We have an

explicit formula for the values of the decision variables: $(\mathbf{x}^{\mathbf{B}})' = \mathbf{B}^{-1}(\mathbf{b} + \delta \mathbf{e}_j) = \mathbf{x}^{\mathbf{B}} + \delta \mathbf{B}^{-1} \mathbf{e}_j$, and each of these needs to be nonnegative. Furthermore, note that $\mathbf{B}^{-1} \mathbf{e}_j$ is simply the j -th column of the inverse of the basis matrix. If we have an optimal tableau, this is easy to find as long as one of the columns of \mathbf{A} is \mathbf{e}_j (which will always be true if we use the big- M method, or are given a problem in the “easy” form of Example 2), because if $\mathbf{A}^{\mathbf{k}} = \mathbf{e}_j$, the k -th column of the simplex tableau contains the vector $\mathbf{B}^{-1} \mathbf{A}^{\mathbf{k}} = \mathbf{B}^{-1} \mathbf{e}_j$.

Thus, each one of the basic variables x_i provides a constraint of the form

$$x_i + \delta \mathcal{T}_{ik} \geq 0$$

If $\mathcal{T}_{ik} > 0$, then this equation simplifies to

$$\delta \geq -x_i / \mathcal{T}_{ik}$$

so certainly

$$\delta \geq \max_{i: \mathcal{T}_{ik} > 0} -x_i / \mathcal{T}_{ik}$$

Likewise, if $\mathcal{T}_{ik} < 0$, then we have

$$\delta \leq -x_i / \mathcal{T}_{ik}$$

so

$$\delta \leq \min_{i: \mathcal{T}_{ik} < 0} -x_i / \mathcal{T}_{ik}$$

Putting this together, the current basis remains optimal if and only if

$$\max_{i: \mathcal{T}_{ik} > 0} -\frac{x_i}{\mathcal{T}_{ik}} \leq \delta \leq \min_{i: \mathcal{T}_{ik} < 0} -\frac{x_i}{\mathcal{T}_{ik}}$$

Example 4. Consider the linear programming problem

$$\begin{array}{llll} \min_{x_1, x_2, x_3, x_4} & -5x_1 - x_2 + 12x_3 & & \\ \text{s.t.} & 3x_1 + 2x_2 + x_3 & = & 10 \\ & 5x_1 + 3x_2 + x_4 & = & 16 \\ & x_1, x_2, x_3, x_4 & \geq & 0 \end{array}$$

for which

12	0	0	2	7
2	1	0	-3	2
2	0	1	5	-3

is the tableau for the optimal solution. How much can the right-hand side of the first constraint change before the current basis is no longer feasible?

Solution. The optimal basis for the problem is $\{x_1, x_2\}$, and the optimal solution is $(2, 2, 0, 0)$. (Why?) For this problem, the requirement vector is changing from $[10 \ 16]$ to $[10 + \delta \ 16]$, that is, to $[10 \ 16] + \delta [0 \ 1] = [10 \ 16] + \delta \mathbf{e}_1$. Because the third column of \mathbf{A} is \mathbf{e}_1 , the third column of the optimal tableau gives us $\mathbf{B}^{-1} \mathbf{e}_1 = [-3 \ 5]$. Thus, the lower bound on δ is given by $-x_2 / \mathcal{T}_{23} = -2/5$, while the upper bound is given by $-x_1 / \mathcal{T}_{13} = 3/5$. Therefore, the right-hand side of the first constraint must either increase by $3/5$, or decrease by $-2/5$ before the current basis is infeasible. If we are confident that the true value of the right-hand side is within this tolerance, we can be assured that this basis will be optimal no matter what. ■

7.2 Changes to the cost vector \mathbf{c}

The case of a change to \mathbf{c} is in many ways the opposite of a change to \mathbf{b} . When \mathbf{b} changed, the optimality condition (nonnegative reduced costs) never changed; we only had to worry about the basis becoming infeasible. On the other hand, when the cost vector changes, we never have to worry about feasibility, because $\mathbf{x}^{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{b}$ will never change. On the other hand, with the new cost vector, it may be worthwhile to have a new variable enter the basis in favor of one of the current basic variables.

By analogy, assume that the cost vector changes to $\mathbf{c}' = \mathbf{c} + \delta \mathbf{e}_i$. The current basis remains optimal if and only if $\bar{\mathbf{c}}' = \mathbf{c}' - \mathbf{c}^{\mathbf{B}'}\mathbf{B}^{-1}\mathbf{A} \geq 0$. If x_i (the decision variable corresponding to the changed element in the cost vector) is nonbasic, $\mathbf{c}^{\mathbf{B}'} = \mathbf{c}^{\mathbf{B}}$, and we have $\mathbf{c}^{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A} \leq \mathbf{c} + \delta \mathbf{e}_i$ as the condition which needs to be satisfied.

This is a vector equation, but all of the components except the i -th are the same as before. Therefore, the only condition we have on δ is

$$\mathbf{c}^{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A}^i \leq c^i + \delta$$

or

$$\delta \geq -\bar{c}_i = -\mathcal{T}_{0i}$$

which can be easily obtained from the final tableau.

On the other hand, if x_i is basic (say, the ℓ -th basic variable), we must satisfy $\mathbf{c}^{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A} + \delta \mathbf{e}_\ell \mathbf{B}^{-1}\mathbf{A} \leq \mathbf{c} + \mathbf{e}_i$. Decomposing this vector equation by components, we must have

$$\delta[\mathbf{B}^{-1}\mathbf{A}^j]_\ell \leq c_j - \mathbf{c}^{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A} = \bar{c}_j$$

for $j \neq i$ (because x_i is basic, its reduced cost is always zero so we don't have to consider that case.) Thus

$$\mathcal{T}_{\ell i} \delta \leq \bar{c}_j$$

for all j , and, as before, we find the δ must lie in the interval

$$\max_{j \neq i: \mathcal{T}_{\ell j} < 0} \left\{ \frac{\bar{c}_j}{\mathcal{T}_{\ell j}} \right\} \leq \delta \leq \min_{j \neq i: \mathcal{T}_{\ell j} > 0} \left\{ \frac{\bar{c}_j}{\mathcal{T}_{\ell j}} \right\}$$

Example 5. Continuing with the previous example, how much could the cost coefficients change for x_1 and x_3 before the current basis is suboptimal?

Solution. x_1 is the first basic variable, so $[\mathcal{T}_{11} \ \mathcal{T}_{12} \ \mathcal{T}_{13} \ \mathcal{T}_{14}] = [1 \ 0 \ -3 \ 2]$. Thus, the lower bound on δ is provided by the third column, and is $\bar{c}_3/\mathcal{T}_{13} = -2/3$, while the upper bound is provided by the fourth column: $\bar{c}_4/\mathcal{T}_{14} = 7/2$. Thus, if x_1 changes by δ , the current basis is optimal if and only if $-2/3 \leq \delta < 7/2$.

x_3 is not a basic variable, so we simply need $\delta \geq -\bar{c}_3 = -2$ for the current basis to remain optimal.

■