Linear Programming: Penn State Math 484 Lecture Notes

Version 1.8.3.1

Christopher Griffin

Ⓢ 2009-2014

With Contributions By:

Bob Pakzad-Hurson

Greg Ference

Veselka Kafedzhieva

Michael Cline

Akinwale Akinbiyi

Ethan Wright

Richard Benjamin

Douglas Mercer

Contents

List of Figures

Preface

This is a set of lecture notes. It is not a book. You should probably go away and come back when you have a real textbook on Linear Programming. Okay, do you have a book? Alright, let's move on then. This is a set of lecture notes for Math 484–Penn State's undergraduate Linear Programming course. Since I use these notes while I teach, there may be typographical errors that I noticed in class, but did not fix in the notes. If you see a typo, send me an e-mail and I'll add an acknowledgement. There may be many typos, that's why you should have a real textbook.

The lecture notes are (roughly) based on the first 6 chapters of Bazaraa et al.'s *Linear Programming and Network Flows* book. This is a reasonably good book, written primarily by and for Industrial Engineers. The only problem I have with the book is that it does not present major results in the standard *theorem-proof* style common to mathematical discourse. This set of notes corrects this problem by presenting the material in a format for presentation to a mathematics class. Many of the proofs in this set of notes are adapted from the textbook with some minor additions. (For example, each time Bazaraa et al. ask, "Why?" in a proof, I provide this information.) Additionally, I prefer to present maximization problems, while *Linear Programming and Network Flows* prefers the minimization format. I've modified all the proofs to operate on maximization problems. When used with the book, the student can obtain a complete set of proofs for elementary Linear Programming.

In order to use these notes successfully, you should have taken courses in:

(1) Vector calculus (Math 230/231 at Penn State)
(2) Matrix algebra (Math 220 at Penn State)

I review a substantial amount of the material you will need, but it's always good to have covered prerequisites before you get to a class. That being said, I hope you enjoy using these notes!

CHAPTER 1

Introduction to Optimization

Linear Programming is a sub-field of optimization theory, which is itself a sub-field of Applied Mathematics. Applied Mathematics is a very general area of study that could arguably encompass half of the engineering disciplines–if you feel like getting into an argument with an engineer. Put simply, applied mathematics is all about applying mathematical techniques to understand or do something practical.

Optimization is an exciting sub-discipline within applied mathematics! Optimization is all about making things better; this could mean helping a company make better decisions to maximize profit; helping a factory make products with less environmental impact; or helping a zoologist improve the diet of an animal. When we talk about optimization, we often use terms like *better* or *improvement*. It's important to remember that words like better can mean *more of something* (as in the case of profit) or *less of something* as in the case of waste. As we study linear programming, we'll quantify these terms in a mathematically precise way. For the time being, let's agree that when we optimize something we are trying to make some decisions that will make it better.

EXAMPLE 1.1. Let's recall a simple optimization problem from differential calculus (Math 140): Goats are an environmentally friendly and inexpensive way to control a lawn when there are lots of rocks or lots of hills. (Seriously, both Google and some U.S. Navy bases use goats on rocky hills instead of paying lawn mowers!)

Suppose I wish to build a pen to keep some goats. I have 100 meters of fencing and I wish to build the pen in a rectangle with the largest possible area. How long should the sides of the rectangle be? In this case, making the pen *better* means making it have the largest possible area.

The problem is illustrated in Figure 1.1. Clearly, we know that:

$$(1.1) \qquad 2x + 2y = 100$$

because $2x + 2y$ is the perimeter of the pen and I have 100 meters of fencing to build my pen. The area of the pen is $A(x, y) = xy$. We can use Equation 1.1 to solve for x in terms of y. Thus we have:

$$(1.2) \qquad y = 50 - x$$

and $A(x) = x(50 - x)$. To maximize $A(x)$, recall we take the first derivative of $A(x)$ with respect to x, set this derivative to zero and solve for x:

$$(1.3) \qquad \frac{dA}{dx} = 50 - 2x = 0;$$
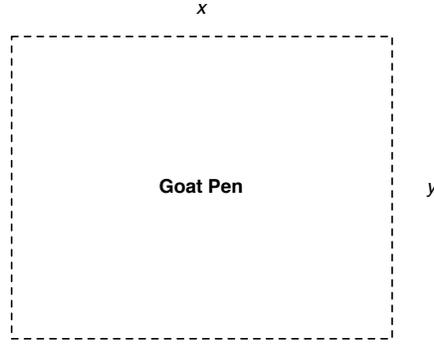
Figure 1.1. Goat pen with unknown side lengths. The objective is to identify the values of x and y that maximize the area of the pen (and thus the number of goats that can be kept).

Thus, $x = 25$ and $y = 50 - x = 25$. We further recall from basic calculus how to confirm that this is a maximum; note:

$$(1.4) \qquad \left.\frac{d^2 A}{dx^2}\right|_{x=25} = -2 < 0$$

Which implies that $x = 25$ is a *local maximum* for this function. Another way of seeing this is to note that $A(x) = 50x - x^2$ is an "upside-down" parabola. As we could have guessed, a square will maximize the area available for holding goats.

EXERCISE 1. A canning company is producing canned corn for the holidays. They have determined that each family prefers to purchase their corn in units of 12 fluid ounces. Assuming that metal costs 1 cent per square inch and 1 fluid ounce is about 1.8 cubic inches, compute the ideal height and radius for a can of corn assuming that cost is to be minimized. [Hint: Suppose that our can has radius r and height h. The formula for the surface area of a can is $2\pi rh + 2\pi r^2$. Since metal is priced by the square inch, the cost is a function of the surface area. The volume of the can is $\pi r^2 h$ and is constrained. Use the same trick we did in the example to find the values of r and h that minimize cost.

1. A General Maximization Formulation

Let's take a more general look at the goat pen example. The area function is a mapping from $\mathbb{R}^2$ to $\mathbb{R}$, written $A : \mathbb{R}^2 \to \mathbb{R}$. The domain of A is the two dimensional space $\mathbb{R}^2$ and its range is $\mathbb{R}$.

Our objective in Example 1.1 is to maximize the function A by choosing values for x and y. In optimization theory, the function we are trying to maximize (or minimize) is called the *objective function*. In general, an objective function is a mapping $z : D \subseteq \mathbb{R}^n \to \mathbb{R}$. Here D is the domain of the function z.

DEFINITION 1.2. Let $z : D \subseteq \mathbb{R}^n \to \mathbb{R}$. The point $\mathbf{x}^*$ is a *global maximum* for z if for all $\mathbf{x} \in D$, $z(\mathbf{x}^*) \geq z(\mathbf{x})$. A point $\mathbf{x}^* \in D$ is a *local maximum* for z if there is a neighborhood $S \subseteq D$ of $\mathbf{x}^*$ (i.e., $\mathbf{x}^* \in S$) so that for all $\mathbf{x} \in S$, $z(\mathbf{x}^*) \geq z(\mathbf{x})$.

REMARK 1.3. Clearly Definition 1.2 is valid only for domains and functions where the concept of a neighborhood is defined and understood. In general, S must be a topologically

connected set (as it is in a neighborhood in $\mathbb{R}^n$) in order for this definition to be used or at least we must be able to define the concept of *neighborhood* on the set[1].

EXERCISE 2. Using analogous reasoning write a definition for a global and local minimum. [Hint: Think about what a minimum means and find the correct direction for the $\geq$ sign in the definition above.]

In Example 1.1, we are constrained in our choice of x and y by the fact that $2x+2y = 100$. This is called a *constraint* of the optimization problem. More specifically, it's called an *equality constraint*. If we did not need to use all the fencing, then we could write the constraint as $2x+2y \leq 100$, which is called an *inequality constraint*. In complex optimization problems, we can have many constraints. The set of all points in $\mathbb{R}^n$ for which the constraints are true is called the *feasible set* (or feasible region). Our problem is to *decide* the best values of x and y to maximize the area $A(x,y)$. The variables x and y are called *decision variables*.

Let $z : D \subseteq \mathbb{R}^n \to \mathbb{R}$; for $i = 1, \ldots, m$, $g_i : D \subseteq \mathbb{R}^n \to \mathbb{R}$; and for $j = 1, \ldots, l$ $h_j : D \subseteq \mathbb{R}^n \to \mathbb{R}$ be functions. Then the general maximization problem with objective function $z(x_1, \ldots, x_n)$ and *inequality constraints* $g_i(x_1, \ldots, x_n) \leq b_i$ $(i = 1, \ldots, m)$ and *equality constraints* $h_j(x_1, \ldots, x_n) = r_j$ is written as:

(1.5)
$$\begin{cases} \max \; z(x_1, \ldots, x_n) \\ \text{s.t.} \;\; g_1(x_1, \ldots, x_n) \leq b_1 \\ \qquad \vdots \\ \qquad g_m(x_1, \ldots, x_n) \leq b_m \\ \qquad h_1(x_1, \ldots, x_n) = r_1 \\ \qquad \vdots \\ \qquad h_l(x_1, \ldots, x_n) = r_l \end{cases}$$

Expression 1.5 is also called a *mathematical programming problem*. Naturally when constraints are involved we define the global and local maxima for the objective function $z(x_1, \ldots, x_n)$ in terms of the feasible region instead of the entire domain of z, since we are only concerned with values of $x_1, \ldots, x_n$ that satisfy our constraints.

EXAMPLE 1.4 (Continuation of Example 1.1). We can re-write the problem in Example 1.1:

(1.6)
$$\begin{cases} \max \; A(x,y) = xy \\ \text{s.t.} \;\; 2x + 2y = 100 \\ \qquad x \geq 0 \\ \qquad y \geq 0 \end{cases}$$

Note we've added two inequality constraints $x \geq 0$ and $y \geq 0$ because it doesn't really make any sense to have negative lengths. We can re-write these constraints as $-x \leq 0$ and $-y \leq 0$ where $g_1(x,y) = -x$ and $g_2(x,y) = -y$ to make Expression 1.6 look like Expression 1.5.

---

[1]Thanks to Bob Pakzad-Hurson who suggested this remark for versions after 1.1.

We have formulated the general *maximization* problem in Proble 1.5. Suppose that we are interested in finding a value that minimizes an objective function $z(x_1, \ldots, x_n)$ subject to certain constraints. Then we can write Problem 1.5 replacing max with min.

EXERCISE 3. Write the problem from Exercise 1 as a general minimization problem. Add any appropriate non-negativity constraints. [Hint: You must change max to min.]

An alternative way of dealing with minimization is to transform a minimization problem into a maximization problem. If we want to minimize $z(x_1, \ldots, x_n)$, we can maximize $-z(x_1, \ldots, x_n)$. In maximizing the negation of the objective function, we are actually finding a value that minimizes $z(x_1, \ldots, x_n)$.

EXERCISE 4. Prove the following statement: *Consider Problem 1.5 with the objective function $z(x_1, \ldots, x_n)$ replaced by $-z(x_1, \ldots, x_n)$. Then the solution to this new problem minimizes $z(x_1, \ldots, x_n)$ subject to the constraints of Problem 1.5.*[Hint: Use the definition of global maximum and a multiplication by -1. Be careful with the direction of the inequality when you multiply by -1.]

2. Some Geometry for Optimization

A critical part of optimization theory is understanding the geometry of Euclidean space. To that end, we're going to review some critical concepts from Vector Calculus (Math 230/231). I'll assume that you remember some basic definitions like *partial derivative* and *Euclidean space*. If you need a refresher, you might want to consult [**MT03, Ste07**].

We'll denote vectors in $\mathbb{R}^n$ in boldface. So $\mathbf{x} \in \mathbb{R}^n$ is an n-dimensional vector and we have $\mathbf{x} = (x_1, \ldots, x_n)$.

DEFINITION 1.5 (Dot Product). Recall that if $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ are two n-dimensional vectors, then the *dot product* (*scalar product*) is:

$$(1.7) \qquad \mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i$$

where x_i is the ith component of the vector $\mathbf{x}$.

An alternative and useful definition for the dot product is given by the following formula. Let θ be the angle between the vectors $\mathbf{x}$ and $\mathbf{y}$. Then the dot product of $\mathbf{x}$ and $\mathbf{y}$ may be alternatively written as:

$$(1.8) \qquad \mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}|| ||\mathbf{y}|| \cos \theta$$

This fact can be proved using the *law of cosines* from trigonometry. As a result, we have the following small lemma (which is proved as Theorem 1 of [**MT03**]):

LEMMA 1.6. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Then the following hold:*
  (1) *The angle between $\mathbf{x}$ and $\mathbf{y}$ is less than $\pi/2$ (i.e., acute) iff $\mathbf{x} \cdot \mathbf{y} > 0$.*
  (2) *The angle between $\mathbf{x}$ and $\mathbf{y}$ is exactly $\pi/2$ (i.e., the vectors are orthogonal) iff $\mathbf{x} \cdot \mathbf{y} = 0$.*
  (3) *The angle between $\mathbf{x}$ and $\mathbf{y}$ is greater than $\pi/2$ (i.e., obtuse) iff $\mathbf{x} \cdot \mathbf{y} < 0$.*

EXERCISE 5. Use the value of the cosine function and the fact that $\mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}|| ||\mathbf{y}|| \cos \theta$ to prove the lemma. [Hint: For what values of θ is $\cos \theta > 0$.]

DEFINITION 1.7 (Graph). Let $z : D \subseteq \mathbb{R}^n \to \mathbb{R}$ be function, then the *graph* of z is the set of $n+1$ tuples:

$$(1.9) \quad \{(\mathbf{x}, z(\mathbf{x})) \in \mathbb{R}^{n+1} | \mathbf{x} \in D\}$$

When $z : D \subseteq \mathbb{R} \to \mathbb{R}$, the graph is precisely what you'd expect. It's the set of pairs $(x, y) \in \mathbb{R}^2$ so that $y = z(x)$. This is the graph that you learned about back in Algebra 1.

DEFINITION 1.8 (Level Set). Let $z : \mathbb{R}^n \to \mathbb{R}$ be a function and let $c \in \mathbb{R}$. Then the *level set of value c for function z* is the set:

$$(1.10) \quad \{\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n | z(\mathbf{x}) = c\} \subseteq \mathbb{R}^n$$

EXAMPLE 1.9. Consider the function $z = x^2 + y^2$. The level set of z at 4 is the set of points $(x, y) \in \mathbb{R}^2$ such that:

$$(1.11) \quad x^2 + y^2 = 4$$

You will recognize this as the equation for a circle with radius 4. We illustrate this in the following two figures. Figure 1.2 shows the level sets of z as they sit on the 3D plot of the function, while Figure 1.3 shows the level sets of z in $\mathbb{R}^2$. The plot in Figure 1.3 is called a *contour plot*.
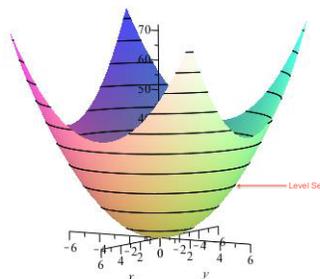
Figure 1.2. Plot with Level Sets Projected on the Graph of z. The level sets existing in $\mathbb{R}^2$ while the graph of z existing $\mathbb{R}^3$. The level sets have been projected onto their appropriate heights on the graph.

DEFINITION 1.10. (Line) Let $\mathbf{x}_0, \mathbf{v} \in \mathbb{R}^n$. Then the *line* defined by vectors $\mathbf{x}_0$ and $\mathbf{v}$ is the function $\mathbf{l}(t) = \mathbf{x}_0 + t\mathbf{v}$. Clearly $l : \mathbb{R} \to \mathbb{R}^n$. The vector $\mathbf{v}$ is called the direction of the line.

EXAMPLE 1.11. Let $\mathbf{x_0} = (2, 1)$ and let $\mathbf{v} = (2, 2)$. Then the line defined by $\mathbf{x}_0$ and $\mathbf{v}$ is shown in Figure 1.4. The set of points on this line is the set $L = \{(x, y) \in \mathbb{R}^2 : x = 2 + 2t, y = 1 + 2t, t \in \mathbb{R}\}$.

DEFINITION 1.12 (Directional Derivative). Let $z : \mathbb{R}^n \to \mathbb{R}$ and let $\mathbf{v} \in \mathbb{R}^n$ be a vector (direction) in n-dimensional space. Then the directional derivative of z at point $\mathbf{x}_0 \in \mathbb{R}^n$ in the direction of $\mathbf{v}$ is

$$(1.12) \quad \left. \frac{d}{dt} z(\mathbf{x}_0 + t\mathbf{v}) \right|_{t=0}$$

5
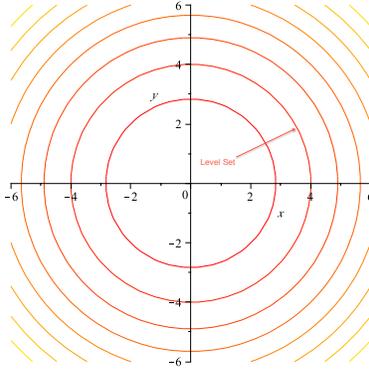
Figure 1.3. Contour Plot of $z = x^2 + y^2$. The circles in $\mathbb{R}^2$ are the level sets of the function. The lighter the circle hue, the higher the value of c that defines the level set.
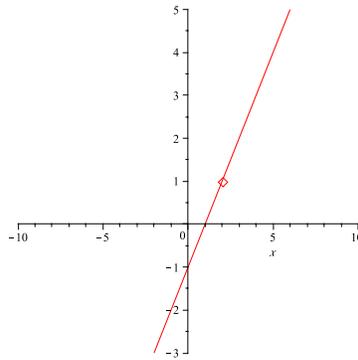
Figure 1.4. A Line Function: The points in the graph shown in this figure are in the set produced using the expression $\mathbf{x_0} + \mathbf{v}t$ where $\mathbf{x_0} = (2, 1)$ and let $\mathbf{v} = (2, 2)$.

when this derivative exists.

PROPOSITION 1.13. *The directional derivative of z at $\mathbf{x}_0$ in the direction $\mathbf{v}$ is equal to:*

$$(1.13) \quad \lim_{h \to 0} \frac{z(\mathbf{x}_0 + h\mathbf{v}) - z(\mathbf{x}_0)}{h}$$

EXERCISE 6. Prove Proposition 1.13. [Hint: Use the definition of derivative for a univariate function and apply it to the definition of directional derivative and evaluate $t = 0$.]

DEFINITION 1.14 (Gradient). Let $z : \mathbb{R}^n \to \mathbb{R}$ be a function and let $\mathbf{x}_0 \in \mathbb{R}^n$. Then the *gradient* of z at $\mathbf{x}_0$ is the vector in $\mathbb{R}^n$ given by:

$$(1.14) \quad \nabla z(\mathbf{x}_0) = \left(\frac{\partial z}{\partial x_1}(\mathbf{x}_0), \ldots, \frac{\partial z}{\partial x_n}(\mathbf{x}_0) \right)$$

Gradients are extremely important concepts in optimization (and vector calculus in general). Gradients have many useful properties that can be exploited. The relationship between the directional derivative and the gradient is of critical importance.

THEOREM 1.15. *If $z : \mathbb{R}^n \to \mathbb{R}$ is differentiable, then all directional derivatives exist. Furthermore, the directional derivative of z at $\mathbf{x}_0$ in the direction of $\mathbf{v}$ is given by:*

$$(1.15) \quad \nabla z(\mathbf{x}_0) \cdot \mathbf{v}$$

where $\cdot$ denotes the dot product of two vectors.

PROOF. Let $\mathbf{l}(t) = \mathbf{x}_0 + \mathbf{v}t$. Then $\mathbf{l}(t) = (l_1(t), \dots, l_n(t))$; that is, $\mathbf{l}(t)$ is a vector function whose i^{th} component is given by $l_i(t) = \mathbf{x}_{0_i} + \mathbf{v}_i t$.

Apply the chain rule:

$$(1.16) \quad \frac{dz(\mathbf{l}(t))}{dt} = \frac{\partial z}{\partial l_1}\frac{dl_1}{dt} + \cdots + \frac{\partial z}{\partial l_n}\frac{dl_n}{dt}$$

Thus:

$$(1.17) \quad \frac{d}{dt}z(\mathbf{l}(t)) = \nabla z \cdot \frac{d\mathbf{l}}{dt}$$

Clearly $d\mathbf{l}/dt = \mathbf{v}$. We have $\mathbf{l}(0) = \mathbf{x}_0$. Thus:

$$(1.18) \quad \left.\frac{d}{dt}z(\mathbf{x}_0 + t\mathbf{v})\right|_{t=0} = \nabla z(\mathbf{x}_0) \cdot \mathbf{v}$$

$\square$

We now come to the two most important results about gradients, (i) the fact that they always point in the direction of steepest ascent with respect to the level curves of a function and (ii) that they are perpendicular (normal) to the level curves of a function. We can exploit this fact as we seek to maximize (or minimize) functions.

THEOREM 1.16. *Let $z : \mathbb{R}^n \to \mathbb{R}$ be differentiable, $\mathbf{x}_0 \in \mathbb{R}^n$. If $\nabla z(\mathbf{x}_0) \neq 0$, then $\nabla z(\mathbf{x}_0)$ points in the direction in which z is increasing fastest.*

PROOF. Recall $\nabla z(\mathbf{x}_0) \cdot \mathbf{v}$ is the directional derivative of z in direction $\mathbf{v}$ at $\mathbf{x}_0$. Assume that $\mathbf{v}$ is a unit vector. We know that:

$$(1.19) \quad \nabla z(\mathbf{x}_0) \cdot \mathbf{v} = ||\nabla z(\mathbf{x}_0)|| \cos\theta$$

(because we assumed $\mathbf{v}$ was a unit vector) where θ is the angle between the vectors $\nabla z(\mathbf{x}_0)$ and $\mathbf{v}$. The function $\cos\theta$ is largest when $\theta = 0$, that is when $\mathbf{v}$ and $\nabla z(\mathbf{x}_0)$ are parallel vectors. (If $\nabla z(\mathbf{x}_0) = 0$, then the directional derivative is zero in all directions.) $\square$

THEOREM 1.17. *Let $z : \mathbb{R}^n \to \mathbb{R}$ be differentiable and let $\mathbf{x}_0$ lie in the level set S defined by $z(\mathbf{x}) = k$ for fixed $k \in \mathbb{R}$. Then $\nabla z(\mathbf{x}_0)$ is normal to the set S in the sense that if $\mathbf{v}$ is a tangent vector at $t = 0$ of a path $\mathbf{c}(t)$ contained entirely in S with $\mathbf{c}(0) = \mathbf{x}_0$, then $\nabla z(\mathbf{x}_0) \cdot \mathbf{v} = 0$.*

REMARK 1.18. Before giving the proof, we illustrate this theorem in Figure 1.5. The function is $z(x, y) = x^4 + y^2 + 2xy$ and $\mathbf{x}_0 = (1, 1)$. At this point $\nabla z(\mathbf{x}_0) = (6, 4)$. We include the tangent line to the level set at the point $(1,1)$ to illustrate the normality of the gradient to the level curve at the point.
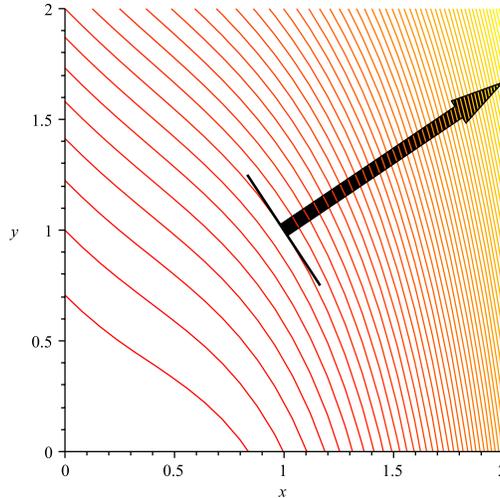
Figure 1.5. A Level Curve Plot with Gradient Vector: We've scaled the gradient vector in this case to make the picture understandable. Note that the gradient is perpendicular to the level set curve at the point $(1,1)$, where the gradient was evaluated. You can also note that the gradient is pointing in the direction of steepest ascent of $z(x,y)$.

PROOF. As stated, let $\mathbf{c}(t)$ be a curve in S. Then $\mathbf{c} : \mathbb{R} \to \mathbb{R}^n$ and $z(\mathbf{c}(t)) = k$ for all $t \in \mathbb{R}$. Let $\mathbf{v}$ be the tangent vector to $\mathbf{c}$ at $t = 0$; that is:

$$(1.20) \qquad \left.\frac{d\mathbf{c}(t)}{dt}\right|_{t=0} = \mathbf{v}$$

Differentiating $z(\mathbf{c}(t))$ with respect to t using the chain rule and evaluating at $t = 0$ yields:

$$(1.21) \qquad \left.\frac{d}{dt}z(\mathbf{c}(t))\right|_{t=0} = \nabla z(\mathbf{c}(0)) \cdot \mathbf{v} = \nabla z(\mathbf{x_0}) \cdot \mathbf{v} = 0$$

Thus $\nabla z(\mathbf{x_0})$ is perpendicular to $\mathbf{v}$ and thus normal to the set S as required. $\qquad \square$

REMARK 1.19. There's a simpler proof of this theorem in the case of a mapping $z : \mathbb{R}^2 \to \mathbb{R}$. For any such function $z(x,y)$, we know that a level set is an implicitly defined curve given by the expression

$$z(x, y) = k$$

where $k \in \mathbb{R}$. We can compute the slope of any tangent line to this curve at some point (x_0, y_0) with implicit differentiation. We have:

$$\left(\frac{d}{dx}\right) z(x, y) = k \left(\frac{d}{dx}\right)$$

yields:

$$\frac{\partial z}{\partial x} + \frac{\partial z}{\partial y}\frac{dy}{dx} = 0$$

Then the slope of the tangent line is given by:

$$\frac{dy}{dx} = \frac{-\partial z/\partial x}{\partial z/\partial y}$$

By $z_x(x_0, y_0)$ we mean $\partial z/\partial x$ evaluated at (x_0, y_0) and by $z_y(x_0, y_0)$ we mean $\partial z/\partial y$ evaluated at (x_0, y_0). Then the slope of the tangent line to the curve $z(x, y) = k$ at (x_0, y_0) is:

$$m = \frac{-z_x(x_0, y_0)}{z_y(x_0, y_0)}$$

An equation for the tangent line at this point is:

(1.22) $y - y_0 = m(x - x_0)$

We can compute a vector that is parallel to this line by taking two points on the line, (x_0, y_0) and (x_1, y_1) and computing the vector $(x_1 - x_0, y_1 - y_0)$. We know that:

$$y_1 - y_0 = m(x_1 - x_0)$$

because any pair (x_1, y_1) on the tangent line must satisfy Equation 1.22. Thus we have the vector $\mathbf{v} = (x_1 - x_0, m(x_1 - x_0))$ parallel to the tangent line. Now we compute the dot product of this vector with the gradient of the function:

$$\nabla z(x_0, y_0) = (z_x(x_0, y_0), z_y(x_0, y_0))$$

We obtain:

$$\nabla z(x_0, y_0) \cdot \mathbf{v} = z_x(x_0, y_0)\,(x_1 - x_0) + z_y(x_0, y_0)\,(m(x_1 - x_0)) =$$

$$z_x(x_0, y_0)\,(x_1 - x_0) + z_y(x_0, y_0)\left(\frac{-z_x(x_0, y_0)}{z_y(x_0, y_0)}(x_1 - x_0)\right) =$$

$$z_x(x_0, y_0)\,(x_1 - x_0) + (-z_x(x_0, y_0)(x_1 - x_0)) = 0$$

Thus, $\nabla z(x_0, y_0)$ is perpendicular to $\mathbf{v}$ as we expected from Theorem 1.17

EXAMPLE 1.20. Let's demonstrate the previous remark and Theorem 1.17. Consider the function $z(x, y) = x^4 + y^2 + 2xy$ with a point (x_0, y_0). Any level curve of the function is given by: $x^4 + y^2 + 2xy = k$. Taking the implicit derivative we obtain:

$$\left(\frac{d}{dx}\right)x^4 + y^2 + 2xy = k\left(\frac{d}{dx}\right) \implies 4x^3 + 2y\frac{dy}{dx} + 2y + 2x\frac{dy}{dx} = 0$$

Note that to properly differentiate $2xy$ implicitly, we needed to use the product rule from calculus. Now, we can solve for the slope of the tangent line to the curve at point (x_0, y_0) as:

$$m = \left(\frac{dy}{dx}\right) = \frac{-4x_0^3 - 2y_0}{2y_0 + 2x_0}$$

Our tangent line is then described the equation:

$$y - y_0 = m(x - x_0)$$

Using the same reasoning we did in the remark, a vector parallel to this line is given by $(x_1 - x_0, y_1 - y_0)$ where (x_1, y_1) is another point on the tangent line. Then we know that:

$$y_1 - y_0 = m(x_1 - x_0)$$

and thus our vector is $\mathbf{v} = (x_1 - x_0, m(x_1 - x_0))$. Now, computing the gradient of $z(x, y)$ at (x_0, y_0) is:

$$\nabla z(x_0, y_0) = (4x_0^3 + 2y_0, 2y_0 + 2x_0)$$

Lastly we compute:

$$\nabla z(x_0, y_0) \cdot \mathbf{v} = \left(4x_0^3 + 2y_0\right)(x_1 - x_0) + (2y_0 + 2x_0)(m(x_1 - x_0)) =$$

$$\left(4x_0^3 + 2y_0\right)(x_1 - x_0) + (2y_0 + 2x_0)\left(\frac{-4x_0^3 - 2y_0}{2y_0 + 2x_0}(x_1 - x_0)\right) =$$

$$\left(4x_0^3 + 2y_0\right)(x_1 - x_0) + \left(-4x_0^3 - 2y_0\right)(x_1 - x_0) = 0$$

Thus, for any point (x_0, y_0) on a level curve of $z(x, y) = x^4 + y^2 + 2xy$ we know that the gradient at that point is perpendicular to a tangent line (vector) to the curve at the point (x_0, y_0).

It is interesting to note that one can compute the slope of the tangent line (and its equation) in Figure 1.5. Here $(x_0, y_0) = (1, 1)$, thus the slope of the tangent line is:

$$m = \frac{-4x_0^3 - 2y_0}{2y_0 + 2x_0} = \frac{-6}{4} = \frac{-3}{2}$$

The equation for the line displayed in Figure 1.5 is:

$$y - 1 = \frac{-3}{2}(x - 1)$$

EXERCISE 7. In this exercise you will use elementary calculus (and a little bit of vector algebra) to show that the gradient of a simple function is perpendicular to its level sets:

(a): Plot the level sets of $z(x, y) = x^2 + y^2$. Draw the gradient at the point $(x, y) = (2, 0)$. Convince yourself that it is normal to the level set $x^2 + y^2 = 4$.

(b): Now, choose any level set $x^2 + y^2 = k$. Use implicit differentiation to find dy/dx. This is the slope of a tangent line to the circle $x^2 + y^2 = k$. Let (x_0, y_0) be a point on this circle.

(c): Find an expression for a vector parallel to the tangent line at (x_0, y_0) [Hint: you can use the slope you just found.]

(d): Compute the gradient of z at (x_0, y_0) and use it and the vector expression you just computed to show that two vectors are perpendicular. [Hint: use the dot product.]

3. Gradients, Constraints and Optimization

Since we're talking about optimization (i.e., minimizing or maximizing a certain function subject to some constraints), it follows that we should be interested in the gradient, which indicates the direction of greatest increase in a function. This information will be used in maximizing a function. Logically, the negation of the gradient will point in the direction of greatest decrease and can be used in minimization. We'll formalize these notions in the study of linear programming. We make one more definition:

DEFINITION 1.21 (Binding Constraint). Let $g(\mathbf{x}) \leq b$ be a constraint in an optimization problem. If at point $\mathbf{x}_0 \in \mathbb{R}^n$ we have $g(\mathbf{x}_0) = b$, then the constraint is said to be *binding*. Clearly equality constraints $h(\mathbf{x}) = r$ are always binding.

EXAMPLE 1.22 (Continuation of Example 1.1). Let's look at the level curves of the objective function and their relationship to the constraints at the point of optimality $(x, y) = (25, 25)$. In Figure 1.6 we see the level curves of the objective function (the hyperbolas) and the feasible region shown as shaded. The elements in the feasible regions are all values for x and y for which $2x + 2y \leq 100$ and $x, y \geq 0$. You'll note that at the point of optimality the level curve $xy = 625$ is tangent to the equation $2x + 2y = 100$; i.e., the level curve of the objective function is tangent to the binding constraint.
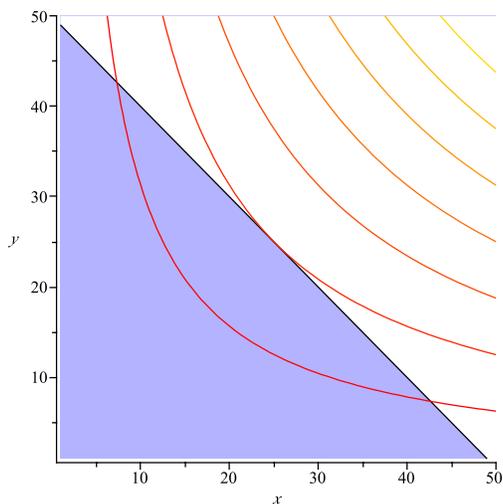
Figure 1.6. Level Curves and Feasible Region: At optimality the level curve of the objective function is tangent to the binding constraints.

If you look at the gradient of $A(x, y)$ at this point it has value $(25, 25)$. We see that it is pointing in the direction of increase for the function $A(x, y)$ (as should be expected) but more importantly let's look at the gradient of the function $2x + 2y$. It's gradient is $(2, 2)$, which is just a scaled version of the gradient of the objective function. Thus the gradient of the objective function is just a dilation of gradient of the binding constraint. This is illustrated in Figure 1.7.

The elements illustrated in the previous example are true in general. You may have discussed a simple example of these when you talked about *Lagrange Multipliers* in Vector Calculus (Math 230/231). We'll revisit these concepts later when we talk about *duality theory* for linear programs. We'll also discuss the gradients of the binding constraints with respect to optimality when we discuss linear programming.

EXERCISE 8. Plot the level sets of the objective function and the feasible region in Exercise 1. At the point of optimality you identified, show that the gradient of the objective function is a scaled version of the gradient (linear combination) of the binding constraints.
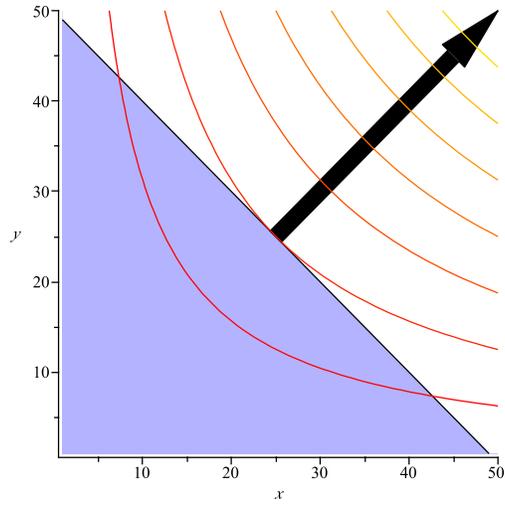
Figure 1.7. Gradients of the Binding Constraint and Objective: At optimality the gradient of the binding constraints and the objective function are *scaled versions of each other.*

CHAPTER 2

Simple Linear Programming Problems

When both the objective and all the constraints in Expression 1.5 are linear functions, then the optimization problem is called a *linear programming problem*. This has the general form:

$$(2.1) \quad \begin{cases} \max \ z(x_1, \ldots, x_n) = c_1 x_1 + \cdots + c_n x_n \\ s.t. \ a_{11} x_1 + \cdots + a_{1n} x_n \leq b_1 \\ \qquad \vdots \\ \qquad a_{m1} x_1 + \cdots + a_{mn} x_n \leq b_m \\ \qquad h_{11} x_1 + \cdots + h_{n1} x_n = r_1 \\ \qquad \vdots \\ \qquad h_{l1} x_1 + \cdots + h_{ln} x_n = r_l \end{cases}$$

DEFINITION 2.1 (Linear Function). A function $z : \mathbb{R}^n \to \mathbb{R}$ is *linear* if there are constants $c_1, \ldots, c_n \in \mathbb{R}$ so that:

$$(2.2) \quad z(x_1, \ldots, x_n) = c_1 x_1 + \cdots + c_n x_n$$

LEMMA 2.2 (Linear Function). *If $z : \mathbb{R}^n \to \mathbb{R}$ is linear then for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and for all scalar constants $\alpha \in \mathbb{R}$ we have:*

$$(2.3) \quad z(\mathbf{x}_1 + \mathbf{x}_2) = z(\mathbf{x}_1) + z(\mathbf{x}_2)$$
$$(2.4) \quad z(\alpha \mathbf{x}_1) = \alpha z(\mathbf{x}_1)$$

EXERCISE 9. Prove Lemma 2.2.

For the time being, we will eschew the general form and focus exclusively on linear programming problems with two variables. Using this limited case, we will develop a graphical method for identifying optimal solutions, which we will generalize later to problems with arbitrary numbers of variables.

EXAMPLE 2.3. Consider the problem of a toy company that produces toy planes and toy boats. The toy company can sell its planes for $10 and its boats for $8 dollars. It costs $3 in raw materials to make a plane and $2 in raw materials to make a boat. A plane requires 3 hours to make and 1 hour to finish while a boat requires 1 hour to make and 2 hours to finish. The toy company knows it will not sell anymore than 35 planes per week. Further, given the number of workers, the company cannot spend anymore than 160 hours per week

finishing toys and 120 hours per week making toys. The company wishes to maximize the profit it makes by choosing how much of each toy to produce.

We can represent the profit maximization problem of the company as a linear programming problem. Let x_1 be the number of planes the company will produce and let x_2 be the number of boats the company will produce. The profit for each plane is $\$10 - \$3 = \$7$ per plane and the profit for each boat is $\$8 - \$2 = \$6$ per boat. Thus the total profit the company will make is:

$$(2.5) \quad z(x_1, x_2) = 7x_1 + 6x_2$$

The company can spend no more than 120 hours per week making toys and since a plane takes 3 hours to make and a boat takes 1 hour to make we have:

$$(2.6) \quad 3x_1 + x_2 \leq 120$$

Likewise, the company can spend no more than 160 hours per week finishing toys and since it takes 1 hour to finish a plane and 2 hour to finish a boat we have:

$$(2.7) \quad x_1 + 2x_2 \leq 160$$

Finally, we know that $x_1 \leq 35$, since the company will make no more than 35 planes per week. Thus the complete linear programming problem is given as:

$$(2.8) \quad \begin{cases} \max \ z(x_1, x_2) = 7x_1 + 6x_2 \\ \text{s.t. } 3x_1 + x_2 \leq 120 \\ \qquad x_1 + 2x_2 \leq 160 \\ \qquad x_1 \leq 35 \\ \qquad x_1 \geq 0 \\ \qquad x_2 \geq 0 \end{cases}$$

EXERCISE 10. A chemical manufacturer produces three chemicals: A, B and C. These chemical are produced by two processes: 1 and 2. Running process 1 for 1 hour costs $\$4$ and yields 3 units of chemical A, 1 unit of chemical B and 1 unit of chemical C. Running process 2 for 1 hour costs $\$1$ and produces 1 units of chemical A, and 1 unit of chemical B (but none of Chemical C). To meet customer demand, at least 10 units of chemical A, 5 units of chemical B and 3 units of chemical C must be produced daily. Assume that the chemical manufacturer wants to minimize the cost of production. Develop a linear programming problem describing the constraints and objectives of the chemical manufacturer. [Hint: Let x_1 be the amount of time Process 1 is executed and let x_2 be amount of time Process 2 is executed. Use the coefficients above to express the cost of running Process 1 for x_1 time and Process 2 for x_2 time. Do the same to compute the amount of chemicals A, B, and C that are produced.]

1. Modeling Assumptions in Linear Programming

Inspecting Example 2.3 (or the more general Problem 2.1) we can see there are several assumptions that must be satisfied when using a linear programming model. We enumerate these below:

Proportionality Assumption: A problem can be phrased as a linear program only if the contribution to the objective function *and* the left-hand-side of each constraint by each decision variable $(x_1, \ldots, x_n)$ is proportional to the value of the decision variable.

Additivity Assumption: A problem can be phrased as a linear programming problem only if the contribution to the objective function *and* the left-hand-side of each constraint by any decision variable x_i $(i = 1, \ldots, n)$ is completely independent of any other decision variable x_j $(j \neq i)$ and additive.

Divisibility Assumption: A problem can be phrased as a linear programming problem only if the quantities represented by each decision variable are infinitely divisible (i.e., fractional answers make sense).

Certainty Assumption: A problem can be phrased as a linear programming problem only if the coefficients in the objective function and constraints are known with certainty.

The first two assumptions simply assert (in English) that both the objective function and functions on the left-hand-side of the (in)equalities in the constraints are linear functions of the variables $x_1, \ldots, x_n$.

The third assumption asserts that a valid optimal answer could contain fractional values for decision variables. It's important to understand how this assumption comes into play–even in the toy making example. Many quantities can be divided into non-integer values (ounces, pounds etc.) but many other quantities cannot be divided. For instance, can we really expect that it's reasonable to make 1/2 a plane in the toy making example? When values must be constrained to true integer values, the linear programming problem is called an *integer programming problem*. These problems are outside the scope of this course, but there is a *vast* literature dealing with them [**PS98**, **WN99**]. For many problems, particularly when the values of the decision variables may become large, a fractional optimal answer could be obtained and then rounded to the nearest integer to obtain a reasonable answer. For example, if our toy problem were re-written so that the optimal answer was to make 1045.3 planes, then we could round down to 1045.

The final assumption asserts that the coefficients (e.g., profit per plane or boat) is known with absolute certainty. In traditional linear programming, there is no lack of knowledge about the make up of the objective function, the coefficients in the left-hand-side of the constraints or the bounds on the right-hand-sides of the constraints. There is a literature on *stochastic programming* [**KW94**, **BTN02**] that relaxes some of these assumptions, but this too is outside the scope of the course.

EXERCISE 11. In a short sentence or two, discuss whether the problem given in Example 2.3 meets all of the assumptions of a scenario that can be modeled by a linear programming problem. Do the same for Exercise 10. [Hint: Can you make 2/3 of a toy? Can you run a process for 1/3 of an hour?]

EXERCISE 12. Suppose the costs are not known with certainty but instead a probability distribution for each value of c_i $(i = 1, \ldots, n)$ is known. Suggest a way of constructing a linear program from the probability distributions. [Hint: Suppose I tell you that I'll give you a uniformly random amount of money between $1 and $2. How much money do you expect to receive? Use the same reasoning to answer the question.]

2. Graphically Solving Linear Programs Problems with Two Variables (Bounded Case)

Linear Programs (LP's) with two variables can be solved graphically by plotting the feasible region along with the level curves of the objective function. We will show that we can find a point in the feasible region that maximizes the objective function using the level curves of the objective function. We illustrate the method first using the problem from Example 2.3.

EXAMPLE 2.4 (Continuation of Example 2.3). Let's continue the example of the Toy Maker begin in Example 2.3. To solve the linear programming problem graphically, begin by drawing the feasible region. This is shown in the blue shaded region of Figure 2.1.
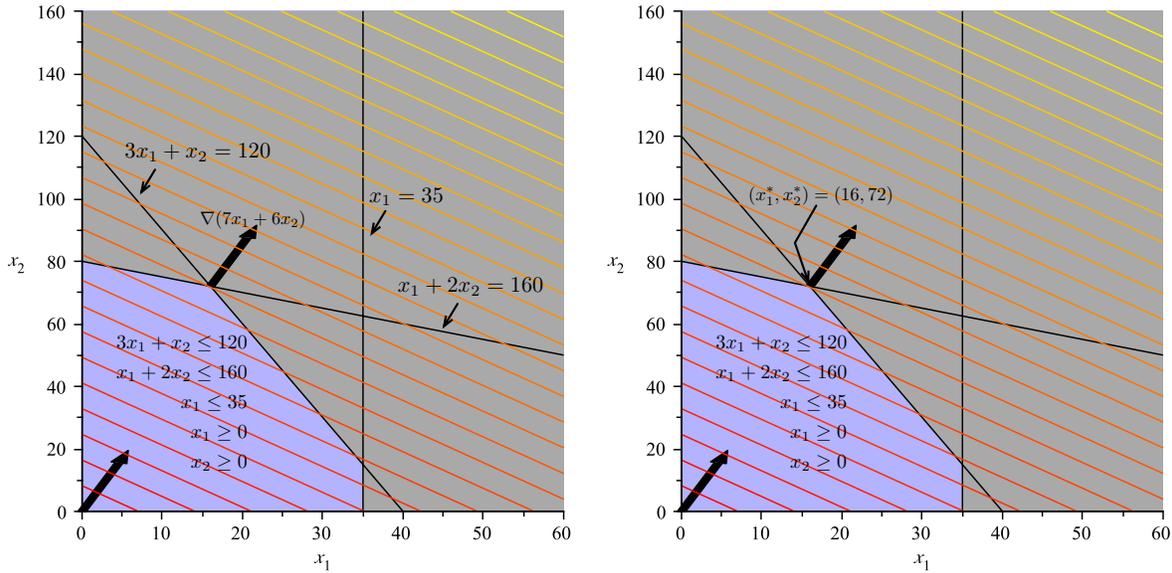
Figure 2.1. Feasible Region and Level Curves of the Objective Function: The shaded region in the plot is the feasible region and represents the intersection of the five inequalities constraining the values of x_1 and x_2. On the right, we see the optimal solution is the "last" point in the feasible region that intersects a level set as we move in the direction of increasing profit.

After plotting the feasible region, the next step is to plot the level curves of the objective function. In our problem, the level sets will have the form:

$$7x_1 + 6x_2 = c \implies x_2 = \frac{-7}{6}x_1 + \frac{c}{6}$$

This is a set of parallel lines with slope $-7/6$ and intercept $c/6$ where c can be varied as needed. The level curves for various values of c are parallel lines. In Figure 2.1 they are shown in colors ranging from red to yellow depending upon the value of c. Larger values of c are more yellow.

To solve the linear programming problem, follow the level sets along the gradient (shown as the black arrow) until the last level set (line) intersects the feasible region. If you are doing this by hand, you can draw a single line of the form $7x_1 + 6x_2 = c$ and then simply

draw parallel lines in the direction of the gradient $(7, 6)$. At some point, these lines will fail to intersect the feasible region. The last line to intersect the feasible region will do so at a point that maximizes the profit. In this case, the point that maximizes $z(x_1, x_2) = 7x_1 + 6x_2$, subject to the constraints given, is $(x_1^*, x_2^*) = (16, 72)$.

Note the point of optimality $(x_1^*, x_2^*) = (16, 72)$ is at a corner of the feasible region. This corner is formed by the intersection of the two lines: $3x_1 + x_2 = 120$ and $x_1 + 2x_2 = 160$. In this case, the constraints

$$3x_1 + x_2 \leq 120$$
$$x_1 + 2x_2 \leq 160$$

are both *binding*, while the other constraints are non-binding. In general, we will see that when an optimal solution to a linear programming problem exists, it will always be at the intersection of several binding constraints; that is, it will occur at a corner of a higher-dimensional polyhedron.

3. Formalizing The Graphical Method

In order to formalize the method we've shown above, we will require a few new definitions.

DEFINITION 2.5. Let $r \in \mathbb{R}$, $r \geq 0$ be a non-negative scalar and let $\mathbf{x}_0 \in \mathbb{R}^n$ be a point in $\mathbb{R}^n$. Then the set:

$$(2.9) \qquad B_r(\mathbf{x}_0) = \{\mathbf{x} \in \mathbb{R}^n | \ ||\mathbf{x} - \mathbf{x}_0|| \leq r\}$$

is called the *closed ball of radius r centered at point $\mathbf{x}_0$ in $\mathbb{R}^n$*.

In $\mathbb{R}^2$, a closed ball is just a disk and its circular boundary centered at $\mathbf{x}_0$ with radius r. In $\mathbb{R}^3$, a closed ball is a solid sphere and its spherical centered at $\mathbf{x}_0$ with radius r. Beyond three dimensions, it becomes difficult to visualize what a closed ball looks like.

We can use a closed ball to define the notion of boundedness of a feasible region:

DEFINITION 2.6. Let $S \subseteq \mathbb{R}^n$. Then the set S is *bounded* if there exists an $\mathbf{x}_0 \in \mathbb{R}^n$ and finite $r \geq 0$ such that S is totally contained in $B_r(\mathbf{x}_0)$; that is, $S \subset B_r(\mathbf{x}_0)$.

Definition 2.6 is illustrated in Figure 2.2. The set S is shown in blue while the ball of radius r centered at $\mathbf{x}_0$ is shown in gray.

We can now define an algorithm for identifying the solution to a linear programing problem in two variables with a *bounded* feasible region (see Algorithm 1):

The example linear programming problem presented in the previous section has a single optimal solution. In general, the following outcomes can occur in solving a linear programming problem:

(1) The linear programming problem has a unique solution. (We've already seen this.)
(2) There are infinitely many alternative optimal solutions.
(3) There is no solution and the problem's objective function can grow to positive infinity for maximization problems (or negative infinity for minimization problems).
(4) There is no solution to the problem at all.

Case 3 above can only occur when the feasible region is unbounded; that is, it cannot be surrounded by a ball with finite radius. We will illustrate each of these possible outcomes in the next four sections. We will prove that this is true in a later chapter.
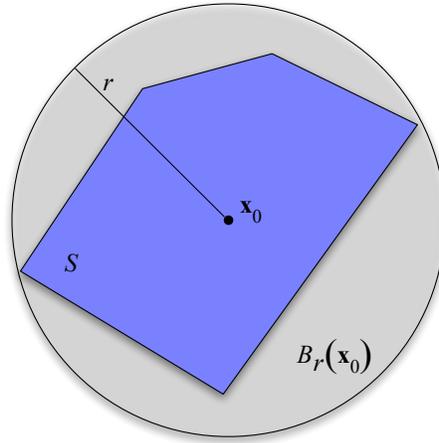
Figure 2.2. A Bounded Set: The set S (in blue) is bounded because it can be entirely contained inside a ball of a finite radius r and centered at some point $\mathbf{x}_0$. In this example, the set S is in $\mathbb{R}^2$. This figure also illustrates the fact that a ball in $\mathbb{R}^2$ is just a disk and its boundary.

Algorithm for Solving a Linear Programming Problem Graphically

Bounded Feasible Region, Unique Solution

(1) Plot the feasible region defined by the constraints.
(2) Plot the level sets of the objective function.
(3) For a maximization problem, identify the level set corresponding the greatest (least, for minimization) objective function value that intersects the feasible region. This point will be at a corner.
(4) The point on the corner intersecting the greatest (least) level set is a solution to the linear programming problem.

Algorithm 1. Algorithm for Solving a Two Variable Linear Programming Problem Graphically–Bounded Feasible Region, Unique Solution Case

EXERCISE 13. Use the graphical method for solving linear programming problems to solve the linear programming problem you defined in Exercise 10.

4. Problems with Alternative Optimal Solutions

We'll study a specific linear programming problem with an infinite number of solutions by modifying the objective function in Example 2.3.

EXAMPLE 2.7. Suppose the toy maker in Example 2.3 finds that it can sell planes for a profit of $18 each instead of $7 each. The new linear programming problem becomes:

$$(2.10) \quad \begin{cases} \max \ z(x_1, x_2) = 18x_1 + 6x_2 \\ s.t. \ 3x_1 + x_2 \leq 120 \\ \quad \ x_1 + 2x_2 \leq 160 \\ \quad \ x_1 \leq 35 \\ \quad \ x_1 \geq 0 \\ \quad \ x_2 \geq 0 \end{cases}$$

Applying our graphical method for finding optimal solutions to linear programming problems yields the plot shown in Figure 2.3. The level curves for the function $z(x_1, x_2) = 18x_1 + 6x_2$ are *parallel* to one face of the polygon boundary of the feasible region. Hence, as we move further up and to the right in the direction of the gradient (corresponding to larger and larger values of $z(x_1, x_2)$) we see that there is not *one* point on the boundary of the feasible region that intersects that level set with greatest value, but instead a side of the polygon boundary described by the line $3x_1 + x_2 = 120$ where $x_1 \in [16, 35]$. Let:

$$S = \{(x_1, x_2) | 3x_1 + x_2 \leq 120, \ x_1 + 2x_2 \leq 160, \ x_1 \leq 35, \ x_1, x_2 \geq 0\}$$

that is, S is the feasible region of the problem. Then for any value of $x_1^* \in [16, 35]$ and any value x_2^* so that $3x_1^* + x_2^* = 120$, we will have $z(x_1^*, x_2^*) \geq z(x_1, x_2)$ for all $(x_1, x_2) \in S$. Since there are infinitely many values that x_1 and x_2 may take on, we see this problem has an infinite number of alternative optimal solutions.

Based on the example in this section, we can modify our algorithm for finding the solution to a linear programming problem graphically to deal with situations with an infinite set of alternative optimal solutions (see Algorithm 2):

Algorithm for Solving a Linear Programming Problem Graphically
Bounded Feasible Region

(1) Plot the feasible region defined by the constraints.
(2) Plot the level sets of the objective function.
(3) For a maximization problem, identify the level set corresponding the greatest (least, for minimization) objective function value that intersects the feasible region. This point will be at a corner.
(4) The point on the corner intersecting the greatest (least) level set is a solution to the linear programming problem.
(5) **If the level set corresponding to the greatest (least) objective function value is parallel to a side of the polygon boundary next to the corner identified, then there are infinitely many alternative optimal solutions and any point on this side may be chosen as an optimal solution.**

Algorithm 2. Algorithm for Solving a Two Variable Linear Programming Problem Graphically–Bounded Feasible Region Case

EXERCISE 14. Modify the linear programming problem from Exercise 10 to obtain a linear programming problem with an infinite number of alternative optimal solutions. Solve
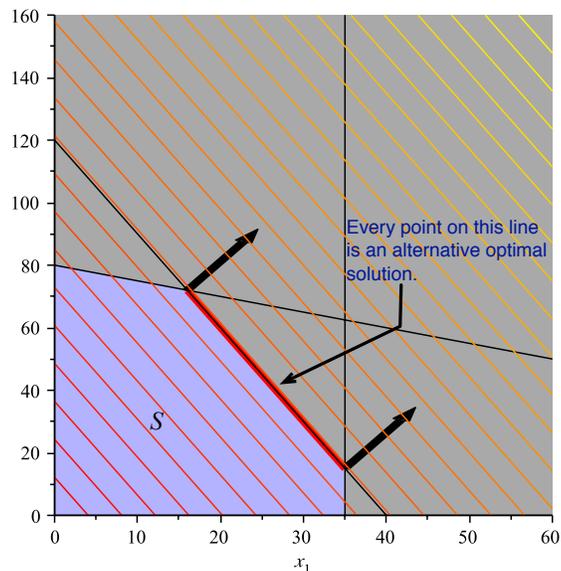
Figure 2.3. An example of infinitely many alternative optimal solutions in a linear programming problem. The level curves for $z(x_1, x_2) = 18x_1 + 6x_2$ are *parallel* to one face of the polygon boundary of the feasible region. Moreover, this side contains the points of greatest value for $z(x_1, x_2)$ inside the feasible region. Any combination of (x_1, x_2) on the line $3x_1 + x_2 = 120$ for $x_1 \in [16, 35]$ will provide the largest possible value $z(x_1, x_2)$ can take in the feasible region S.

the new problem and obtain a description for the set of alternative optimal solutions. [Hint: Just as in the example, x_1 will be bound between two value corresponding to a side of the polygon. Find those values and the constraint that is binding. This will provide you with a description of the form for any $x_1^* \in [a, b]$ and x_2^* is chosen so that $cx_1^* + dx_2^* = v$, the point (x_1^*, x_2^*) is an alternative optimal solution to the problem. Now you fill in values for a, b, c, d and v.]

5. Problems with No Solution

Recall for *any* mathematical programming problem, the feasible set or region is simply a subset of $\mathbb{R}^n$. If this region is empty, then there is no solution to the mathematical programming problem and the problem is said to be *over constrained*. We illustrate this case for linear programming problems with the following example.

EXAMPLE 2.8. Consider the following linear programming problem:

(2.11)
$$\begin{cases} \max \ z(x_1, x_2) = 3x_1 + 2x_2 \\ s.t. \ \dfrac{1}{40}x_1 + \dfrac{1}{60}x_2 \leq 1 \\ \quad \dfrac{1}{50}x_1 + \dfrac{1}{50}x_2 \leq 1 \\ \quad x_1 \geq 30 \\ \quad x_2 \geq 20 \end{cases}$$

The level sets of the objective and the constraints are shown in Figure 2.4.
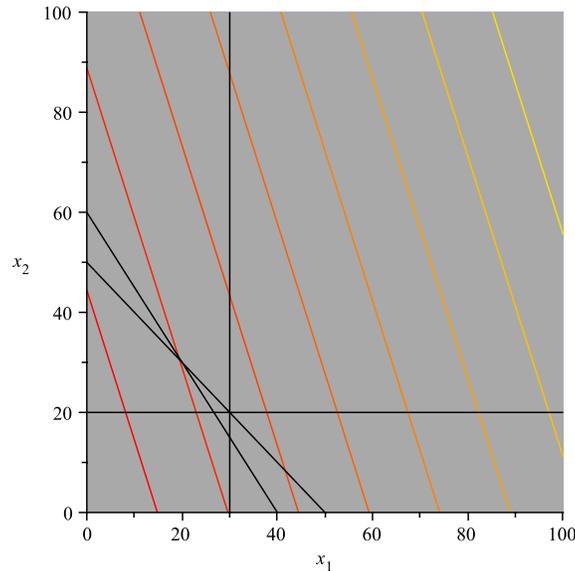
Figure 2.4. A Linear Programming Problem with no solution. The feasible region of the linear programming problem is empty; that is, there are no values for x_1 and x_2 that can simultaneously satisfy all the constraints. Thus, no solution exists.

The fact that the feasible region is empty is shown by the fact that in Figure 2.4 there is no blue region–i.e., all the regions are gray indicating that the constraints are not satisfiable.

Based on this example, we can modify our previous algorithm for finding the solution to linear programming problems graphically (see Algorithm 3):

Algorithm for Solving a Linear Programming Problem Graphically

Bounded Feasible Region

(1) Plot the feasible region defined by the constraints.
(2) **If the feasible region is empty, then no solution exists.**
(3) Plot the level sets of the objective function.
(4) For a maximization problem, identify the level set corresponding the greatest (least, for minimization) objective function value that intersects the feasible region. This point will be at a corner.
(5) The point on the corner intersecting the greatest (least) level set is a solution to the linear programming problem.
(6) **If the level set corresponding to the greatest (least) objective function value is parallel to a side of the polygon boundary next to the corner identified, then there are infinitely many alternative optimal solutions and any point on this side may be chosen as an optimal solution.**

Algorithm 3. Algorithm for Solving a Two Variable Linear Programming Problem Graphically–Bounded Feasible Region Case

6. Problems with Unbounded Feasible Regions

Again, we'll tackle the issue of linear programming problems with unbounded feasible regions by illustrating the possible outcomes using examples.

EXAMPLE 2.9. Consider the linear programming problem below:

$$(2.12) \quad \begin{cases} \max \ z(x_1, x_2) = 2x_1 - x_2 \\ s.t. \ x_1 - x_2 \leq 1 \\ \quad\quad 2x_1 + x_2 \geq 6 \\ \quad\quad x_1, x_2 \geq 0 \end{cases}$$

The feasible region and level curves of the objective function are shown in Figure 2.5. The
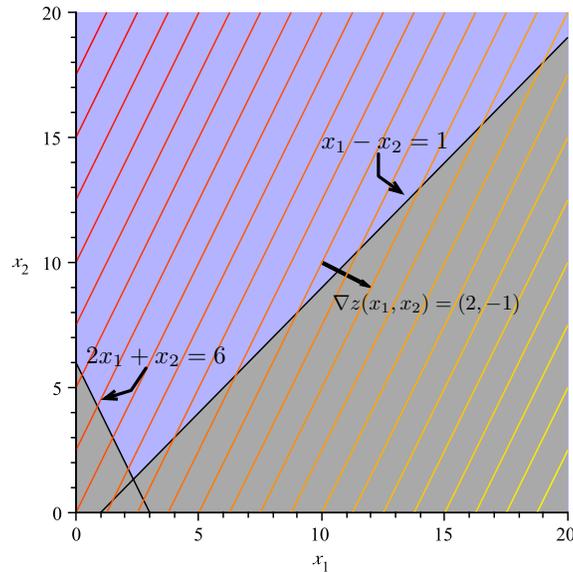
Figure 2.5. A Linear Programming Problem with Unbounded Feasible Region: Note that we can continue to make level curves of $z(x_1, x_2)$ corresponding to larger and larger values as we move down and to the right. These curves will continue to intersect the feasible region for any value of $v = z(x_1, x_2)$ we choose. Thus, we can make $z(x_1, x_2)$ as large as we want and still find a point in the feasible region that will provide this value. Hence, the optimal value of $z(x_1, x_2)$ subject to the constraints $+\infty$. That is, the problem is unbounded.

feasible region in Figure 2.5 is clearly unbounded since it stretches upward along the x_2 axis infinitely far and also stretches rightward along the x_1 axis infinitely far, bounded below by the line $x_1 - x_2 = 1$. There is no way to enclose this region by a disk of finite radius, hence the feasible region is not bounded.

We can draw more level curves of $z(x_1, x_2)$ in the direction of increase (down and to the right) as long as we wish. There will always be an intersection point with the feasible region because it is infinite. That is, these curves will continue to intersect the feasible region for any value of $v = z(x_1, x_2)$ we choose. Thus, we can make $z(x_1, x_2)$ as large as we want and still find a point in the feasible region that will provide this value. Hence, the largest value

$z(x_1, x_2)$ can take when (x_1, x_2) are in the feasible region is $+\infty$. That is, the problem is unbounded.

Just because a linear programming problem has an unbounded feasible region does not imply that there is not a finite solution. We illustrate this case by modifying example 2.9.

EXAMPLE 2.10 (Continuation of Example 2.9). Consider the linear programming problem from Example 2.9 with the new objective function: $z(x_1, x_2) = (1/2)x_1 - x_2$. Then we have the new problem:

$$(2.13) \quad \begin{cases} \max \ z(x_1, x_2) = \dfrac{1}{2}x_1 - x_2 \\[2mm] s.t. \ x_1 - x_2 \leq 1 \\[1mm] \qquad 2x_1 + x_2 \geq 6 \\[1mm] \qquad x_1, x_2 \geq 0 \end{cases}$$

The feasible region, level sets of $z(x_1, x_2)$ and gradients are shown in Figure 2.6. In this case note, that the direction of increase of the objective function is *away* from the direction in which the feasible region is unbounded (i.e., downward). As a result, the point in the feasible region with the largest $z(x_1, x_2)$ value is $(7/3, 4/3)$. Again this is a vertex: the binding constraints are $x_1 - x_2 = 1$ and $2x_1 + x_2 = 6$ and the solution occurs at the point these two lines intersect.
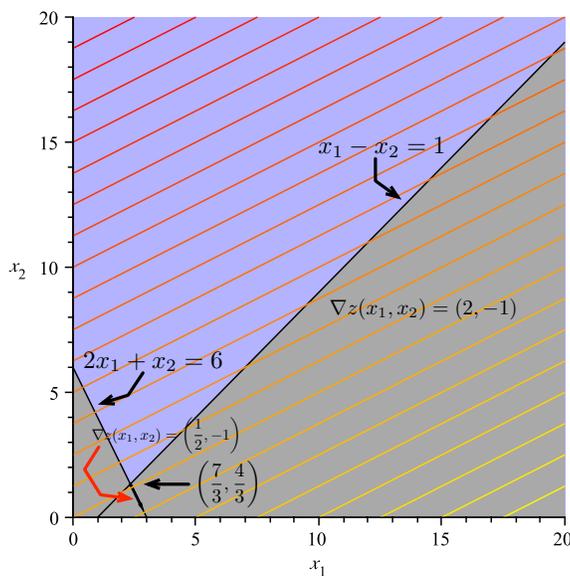
Figure 2.6. A Linear Programming Problem with Unbounded Feasible Region and Finite Solution: In this problem, the level curves of $z(x_1, x_2)$ increase in a more "southernly" direction that in Example 2.10–that is, *away* from the direction in which the feasible region increases without bound. The point in the feasible region with largest $z(x_1, x_2)$ value is $(7/3, 4/3)$. Note again, this is a vertex.

Based on these two examples, we can modify our algorithm for graphically solving a two variable linear programming problems to deal with the case when the feasible region is unbounded.

Algorithm for Solving a Two Variable Linear Programming Problem Graphically

   (1) Plot the feasible region defined by the constraints.
   (2) If the feasible region is empty, then no solution exists.
   (3) If the feasible region is unbounded goto Line 8. Otherwise, Goto Line 4.
   (4) Plot the level sets of the objective function.
   (5) For a maximization problem, identify the level set corresponding the greatest (least, for minimization) objective function value that intersects the feasible region. This point will be at a corner.
   (6) The point on the corner intersecting the greatest (least) level set is a solution to the linear programming problem.
   (7) **If the level set corresponding to the greatest (least) objective function value is parallel to a side of the polygon boundary next to the corner identified, then there are infinitely many alternative optimal solutions and any point on this side may be chosen as an optimal solution.**
   (8) (The feasible region is unbounded): Plot the level sets of the objective function.
   (9) If the level sets intersect the feasible region at larger and larger (smaller and smaller for a minimization problem), then the problem is unbounded and the solution is $+\infty$ ($-\infty$ for minimization problems).
  (10) Otherwise, identify the level set corresponding the greatest (least, for minimization) objective function value that intersects the feasible region. This point will be at a corner.
  (11) The point on the corner intersecting the greatest (least) level set is a solution to the linear programming problem. **If the level set corresponding to the greatest (least) objective function value is parallel to a side of the polygon boundary next to the corner identified, then there are infinitely many alternative optimal solutions and any point on this side may be chosen as an optimal solution.**

Algorithm 4. Algorithm for Solving a Linear Programming Problem Graphically– Bounded and Unbounded Case

EXERCISE 15. Does the following problem have a bounded solution? Why?

$$(2.14) \quad \begin{cases} \textbf{min} \;\; z(x_1, x_2) = 2x_1 - x_2 \\ \;\; s.t. \;\; x_1 - x_2 \leq 1 \\ \qquad\;\; 2x_1 + x_2 \geq 6 \\ \qquad\;\; x_1, x_2 \geq 0 \end{cases}$$

[Hint: Use Figure 2.6 and Algorithm 4.]

EXERCISE 16. Modify the objective function in Example 2.9 or Example 2.10 to produce a problem with an infinite number of solutions.

EXERCISE 17. Modify the objective function in Exercise 15 to produce a **minimization** problem that has a finite solution. Draw the feasible region and level curves of the objective

to "prove" your example works. [Hint: Think about what direction of increase is required for the level sets of $z(x_1, x_2)$ (or find a trick using Exercise 4).]

CHAPTER 3

Matrices, Linear Algebra and Linear Programming

In this section, we will review matrix concepts critical for the general understanding of general linear programming algorithms.

Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors in $\mathbb{R}^n$. Recall we denote the dot product of the two vectors as $\mathbf{x} \cdot \mathbf{y}$.

1. Matrices

Recall an $m \times n$ matrix is a rectangular array of numbers, usually drawn from a field such as $\mathbb{R}$. We write an $m \times n$ matrix with values in $\mathbb{R}$ as $\mathbf{A} \in \mathbb{R}^{m \times n}$. The matrix consists of m rows and n columns. The element in the ith row and j^{th} column of $\mathbf{A}$ is written as $\mathbf{A}_{ij}$. The jth column of $\mathbf{A}$ can be written as $\mathbf{A}_{\cdot j}$, where the $\cdot$ is interpreted as ranging over every value of i (from 1 to m). Similarly, the ith row of $\mathbf{A}$ can be written as $\mathbf{A}_{i\cdot}$. When $m = n$, then the matrix $\mathbf{A}$ is called *square*.

DEFINITION 3.1 (Matrix Addition). If $\mathbf{A}$ and $\mathbf{B}$ are both in $\mathbb{R}^{m \times n}$, then $\mathbf{C} = \mathbf{A} + \mathbf{B}$ is the matrix sum of $\mathbf{A}$ and $\mathbf{B}$ and

$$(3.1) \quad \mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{B}_{ij} \ \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n$$

EXAMPLE 3.2.

$$(3.2) \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

DEFINITION 3.3 (Row/Column Vector). A $1 \times n$ matrix is called a *row vector*, and a $m \times 1$ matrix is called a *column vector*. For the remainder of these notes, every vector will be thought of **column vector** unless otherwise noted.

It should be clear that any row of matrix $\mathbf{A}$ could be considered a row vector in $\mathbb{R}^n$ and any column of $\mathbf{A}$ could be considered a column vector in $\mathbb{R}^m$.

DEFINITION 3.4 (Matrix Multiplication). If $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$, then $\mathbf{C} = \mathbf{AB}$ is the *matrix product* of $\mathbf{A}$ and $\mathbf{B}$ and

$$(3.3) \quad \mathbf{C}_{ij} = \mathbf{A}_{i\cdot} \cdot \mathbf{B}_{\cdot j}$$

Note, $\mathbf{A}_{i\cdot} \in \mathbb{R}^{1 \times n}$ (an n-dimensional vector) and $\mathbf{B}_{\cdot j} \in \mathbb{R}^{n \times 1}$ (another n-dimensional vector), thus making the dot product meaningful.

EXAMPLE 3.5.

$$(3.4) \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1(5)+2(7) & 1(6)+2(8) \\ 3(5)+4(7) & 3(6)+4(8) \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

DEFINITION 3.6 (Matrix Transpose). If $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a $m \times n$ matrix, then the *transpose* of $\mathbf{A}$ dented $\mathbf{A}^T$ is an $m \times n$ matrix defined as:

$$(3.5) \qquad \mathbf{A}_{ij}^T = \mathbf{A}_{ji}$$

EXAMPLE 3.7.

$$(3.6) \qquad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

The matrix transpose is a particularly useful operation and makes it easy to transform column vectors into row vectors, which enables multiplication. For example, suppose $\mathbf{x}$ is an $n \times 1$ column vector (i.e., $\mathbf{x}$ is a vector in $\mathbb{R}^n$) and suppose $\mathbf{y}$ is an $n \times 1$ column vector. Then:

$$(3.7) \qquad \mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y}$$

EXERCISE 18. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$. Use the definitions of matrix addition and transpose to prove that:

$$(3.8) \qquad (\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$$

[Hint: If $\mathbf{C} = \mathbf{A} + \mathbf{B}$, then $\mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{B}_{ij}$, the element in the (i, j) position of matrix $\mathbf{C}$. This element moves to the (j, i) position in the transpose. The (j, i) position of $\mathbf{A}^T + \mathbf{B}^T$ is $\mathbf{A}_{ji}^T + \mathbf{B}_{ji}^T$, but $\mathbf{A}_{ji}^T = \mathbf{A}_{ij}$. Reason from this point.]

EXERCISE 19. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$. Prove by example that $\mathbf{AB} \neq \mathbf{BA}$; that is, matrix multiplication is *not commutative*. [Hint: Almost any pair of matrices you pick (that can be multiplied) will not commute.]

EXERCISE 20. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and let, $\mathbf{B} \in \mathbb{R}^{n \times p}$. Use the definitions of matrix multiplication and transpose to prove that:

$$(3.9) \qquad (\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$$

[Hint: Use similar reasoning to the hint in Exercise 18. But this time, note that $\mathbf{C}_{ij} = \mathbf{A}_{i\cdot} \cdot \mathbf{B}_{\cdot j}$, which moves to the (j, i) position. Now figure out what is in the (j, i) position of $\mathbf{B}^T \mathbf{A}^T$.]

Let $\mathbf{A}$ and $\mathbf{B}$ be two matrices with the same number of rows (so $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{m \times p}$). Then the augmented matrix $[\mathbf{A}|\mathbf{B}]$ is:

$$(3.10) \qquad \left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \dots & a_{1n} & b_{11} & b_{12} & \dots & b_{1p} \\ a_{21} & a_{22} & \dots & a_{2n} & b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_{m1} & b_{m2} & \dots & b_{mp} \end{array} \right]$$

Thus, $[\mathbf{A}|\mathbf{B}]$ is a matrix in $\mathbb{R}^{m \times (n+p)}$.

EXAMPLE 3.8. Consider the following matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Then $[\mathbf{A}|\mathbf{B}]$ is:

$$[\mathbf{A}|\mathbf{B}] = \left[\begin{array}{cc|c} 1 & 2 & 7 \\ 3 & 4 & 8 \end{array} \right]$$

EXERCISE 21. By analogy define the augmented matrix $\left[\frac{\mathbf{A}}{\mathbf{B}}\right]$. Note, this is **not** a fraction. In your definition, identify the appropriate requirements on the relationship between the number of rows and columns that the matrices must have. [Hint: Unlike $[\mathbf{A}|\mathbf{B}]$, the number of rows don't have to be the same, since your concatenating on the rows, not columns. There should be a relation between the numbers of columns though.]

2. Special Matrices and Vectors

DEFINITION 3.9 (Identify Matrix). The $n \times n$ *identify matrix* is:

$$(3.11) \quad \mathbf{I}_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

When it is clear from context, we may simply write $\mathbf{I}$ and omit the subscript n.

EXERCISE 22. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$. Show that $\mathbf{A}\mathbf{I}_n = \mathbf{I}_n\mathbf{A} = \mathbf{A}$. Hence, $\mathbf{I}$ is an identify for the matrix multiplication operation on square matrices. [Hint: Do the multiplication out long hand.]

DEFINITION 3.10 (Standard Basis Vector). The standard basis vector $\mathbf{e}_i \in \mathbb{R}^n$ is:

$$\mathbf{e}_i = \left(\underbrace{0, 0, \dots}_{i-1}, 1, \underbrace{0, \dots, 0}_{n-i-1} \right)$$

Note, this definition is only valid for $n \geq i$. Further the standard basis vector $\mathbf{e}_i$ is also the ith row or column of $\mathbf{I}_n$.

DEFINITION 3.11 (Unit and Zero Vectors). The vector $\mathbf{e} \in \mathbb{R}^n$ is the *one vector* $\mathbf{e} = (1, 1, \dots, 1)$. Similarly, the *zero vector* $\mathbf{0} = (0, 0, \dots, 0) \in \mathbb{R}^n$. We assume that the length of $\mathbf{e}$ and $\mathbf{0}$ will be determined from context.

EXERCISE 23. Let $\mathbf{x} \in \mathbb{R}^n$, considered as a column vector (our standard assumption). Define:

$$\mathbf{y} = \frac{\mathbf{x}}{\mathbf{e}^T \mathbf{x}}$$

Show that $\mathbf{e}^T\mathbf{y} = \mathbf{y}^T\mathbf{e} = 1$. [Hint: First remember that $\mathbf{e}^T\mathbf{x}$ is a scalar value (it's $\mathbf{e} \cdot \mathbf{x}$. Second, remember that a scalar times a vector is just a new vector with each term multiplied by the scalar. Last, use these two pieces of information to write the product $\mathbf{e}^T\mathbf{y}$ as a sum of fractions.]

3. Matrices and Linear Programming Expression

You will recall from your matrices class (Math 220) that matrices can be used as a short hand way to represent linear equations. Consider the following system of equations:

$$(3.12) \quad \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \qquad\qquad\qquad\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

Then we can write this in matrix notation as:

$$(3.13) \quad \mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}_{ij} = a_{ij}$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$ and $\mathbf{x}$ is a column vector in $\mathbb{R}^n$ with entries x_j $(j = 1, \ldots, n)$ and $\mathbf{b}$ is a column vector in $\mathbb{R}^m$ with entries b_i $(i = 1 \ldots, m)$. Obviously, if we replace the equalities in Expression 3.12 with inequalities, we can also express systems of inequalities in the form:

$$(3.14) \quad \mathbf{Ax} \le \mathbf{b}$$

Using this representation, we can write our general linear programming problem using matrix and vector notation. Expression 2.1 can be written as:

$$(3.15) \quad \begin{cases} \max\ z(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ \quad s.t.\ \mathbf{Ax} \le \mathbf{b} \\ \qquad\quad \mathbf{Hx} = \mathbf{r} \end{cases}$$

For historical reasons, linear programs are not written in the general form of Expression 3.15.

DEFINITION 3.12 (Canonical Form). A maximization linear programming problem is in *canonical form* if it is written as:

$$(3.16) \quad \begin{cases} \max\ z(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ \quad s.t.\ \mathbf{Ax} \le \mathbf{b} \\ \qquad\quad \mathbf{x} \ge \mathbf{0} \end{cases}$$

A minimization linear programming problem is in *canonical form* if it is written as:

$$(3.17) \quad \begin{cases} \min\ z(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ \quad s.t.\ \mathbf{Ax} \ge \mathbf{b} \\ \qquad\quad \mathbf{x} \ge \mathbf{0} \end{cases}$$

DEFINITION 3.13 (Standard Form (Max Problem)). A maximization linear programming problem is in *standard form* if it is written as:

$$(3.18) \quad \begin{cases} \max\ z(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ \quad s.t.\ \mathbf{Ax} = \mathbf{b} \\ \qquad\quad \mathbf{x} \ge \mathbf{0} \end{cases}$$

REMARK 3.14. In the previous definition, a problem is in standard form as long as its constraints have form $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$. The problem can be either a maximization or minimization problem.

THEOREM 3.15. *Every linear programming problem in canonical form can be put into standard form.*

PROOF. Consider the constraint corresponding to the first row of the matrix $\mathbf{A}$:

$$(3.19) \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1$$

Then we can add a new *slack* variable s_1 so that $s_1 \geq 0$ and we have:

$$(3.20) \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + s_1 = b_1$$

This act can be repeated for each row of $\mathbf{A}$ (constraint) yielding m new variables $s_1, \ldots, s_m$, which we can express as a row $\mathbf{s}$. Then the new linear programming problem can be expressed as:

$$\begin{cases} \max \quad z(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ \quad s.t. \quad \mathbf{Ax} + \mathbf{I}_m \mathbf{s} = \mathbf{b} \\ \qquad \quad \mathbf{x}, \mathbf{s} \geq \mathbf{0} \end{cases}$$

Using augmented matrices, we can express this as:

$$\begin{cases} \max \quad z(\mathbf{x}) = \begin{bmatrix} \mathbf{c} \\ \hline \mathbf{0} \end{bmatrix}^T \begin{bmatrix} \mathbf{x} \\ \hline \mathbf{s} \end{bmatrix} \\ \quad s.t. \quad [\mathbf{A}|\mathbf{I}_m] \begin{bmatrix} \mathbf{x} \\ \hline \mathbf{s} \end{bmatrix} = \mathbf{b} \\ \qquad \qquad \begin{bmatrix} \mathbf{x} \\ \hline \mathbf{s} \end{bmatrix} \geq \mathbf{0} \end{cases}$$

Clearly, this new linear programming problem is in standard form and any solution maximizing the original problem will necessarily maximize this one. $\square$

EXAMPLE 3.16. Consider the Toy Maker problem from Example 2.3. The problem in canonical form is:

$$\begin{cases} \max \quad z(x_1, x_2) = 7x_1 + 6x_2 \\ \quad s.t. \quad 3x_1 + x_2 \leq 120 \\ \qquad \quad x_1 + 2x_2 \leq 160 \\ \qquad \quad x_1 \leq 35 \\ \qquad \quad x_1 \geq 0 \\ \qquad \quad x_2 \geq 0 \end{cases}$$

We can introduce slack variables s_1, s_2 and s_3 into the constraints (one for each constraint) and re-write the problem as:

$$\begin{cases} \max \ z(x_1, x_2) = 7x_1 + 6x_2 \\ \quad s.t. \ 3x_1 + x_2 + s_1 = 120 \\ \qquad x_1 + 2x_2 + s_2 = 160 \\ \qquad x_1 + s_3 = 35 \\ \qquad x_1 \geq 0 \\ \qquad x_2 \geq 0 \end{cases}$$

REMARK 3.17. We can deal with constraints of the form:

$$(3.21) \quad a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i$$

in a similar way. In this case we subtract a surplus variable s_i to obtain:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - s_i = b_i$$

Again, we must have $s_i \geq 0$.

THEOREM 3.18. *Every linear programming problem in standard form can be put into canonical form.*

PROOF. Recall that $\mathbf{Ax} = \mathbf{b}$ if and only if $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{Ax} \geq \mathbf{b}$. The second inequality can be written as $-\mathbf{Ax} \leq -\mathbf{b}$. This yields the linear programming problem:

$$(3.22) \quad \begin{cases} \max \ z(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ \quad s.t. \ \mathbf{Ax} \leq \mathbf{b} \\ \qquad -\mathbf{Ax} \leq -\mathbf{b} \\ \qquad \mathbf{x} \geq \mathbf{0} \end{cases}$$

Defining the appropriate augmented matrices allows us to convert this linear programming problem into canonical form. $\qquad \square$

EXERCISE 24. Complete the "pedantic" proof of the preceding theorem by defining the correct augmented matrices to show that the linear program in Expression 3.22 is in canonical form.

The standard solution method for linear programming models (the Simplex Algorithm) assumes that all variables are non-negative. Though this assumption can be easily relaxed, the first implementation we will study imposes this restriction. The general linear programming problem we posed in Expression 3.15 does not (necessarily) impose any sign restriction on the variables. We will show that we can transform a problem in which x_i is unrestricted into a new problem in which all variables are positive. Clearly, if $x_i \leq 0$, then we simply replace x_i by $-y_i$ in every expression and then $y_i \geq 0$. On the other hand, if we have the constraint $x_i \geq l_i$, then clearly we can write $y_i = x_i - l_i$ and $y_i \geq 0$. We can then replace x_i by $y_i + l_i$ in every equation or inequality where x_i appears. Finally, if $x_i \leq u_i$, but x_i may be negative, then we may write $y_i = u_i - x_i$. Clearly, $y_i \geq 0$ and we can replace x_i by $u_i - y_i$ in every equation or inequality where x_i appears.

If x_i is unrestricted in sign and has no upper or lower bounds, then let $x_i = y_i - z_i$ where $y_i, z_i \geq 0$ and replace x_i by $(y_i - z_i)$ in the objective, equations and inequalities of a general linear programming problem. Since $y_i, z_i \geq 0$ and may be given any values as a part of the solution, clearly x_i may take any value in $\mathbb{R}$.

EXERCISE 25. Convince yourself that the general linear programming problem shown in Expression 3.15 can be converted into canonical (or standard) form using the following steps:

(1) Every constraint of the form $x_i \leq u_i$ can be dealt with by substituting $y_i = u_i - x_i$, $y_i \geq 0$.
(2) Every constraint of the form $l_i \leq x_i$ can be dealt with by substituting $y_i = x_i - l_i$, $y_i \geq 0$.
(3) If x_i is unrestricted in any way, then we can variables y_i and z_i so that $x_i = y_i - z_i$ where $y_i, z_i \geq 0$.
(4) Any equality constraints $\mathbf{Hx} = \mathbf{r}$ can be transformed into inequality constraints.

Thus, Expression 3.15 can be transformed to standard form. [Hint: No hint, the hint is in the problem.]

4. Gauss-Jordan Elimination and Solution to Linear Equations

In this sub-section, we'll review Gauss-Jordan Elimination as a solution method for linear equations. We'll use Gauss-Jordan Elimination *extensively* in the coming chapters.

DEFINITION 3.19 (Elementary Row Operation). Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a matrix. Recall $\mathbf{A}_{i\cdot}$ is the ith row of $\mathbf{A}$. There are three *elementary row operations*:

(1) (Scalar Multiplication of a Row) Row $\mathbf{A}_{i\cdot}$ is replaced by $\alpha \mathbf{A}_{i\cdot}$, where $\alpha \in \mathbb{R}$ and $\alpha \neq 0$.
(2) (Row Swap) Row $\mathbf{A}_{i\cdot}$ is swapped with Row $\mathbf{A}_{j\cdot}$ for $i \neq j$.
(3) (Scalar Multiplication and Addition) Row $\mathbf{A}_{j\cdot}$ is replaced by $\alpha \mathbf{A}_{i\cdot} + \mathbf{A}_{j\cdot}$ for $\alpha \in \mathbb{R}$ and $i \neq j$.

EXAMPLE 3.20. Consider the matrix:
$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$
In an example of scalar multiplication of a row by a constant, we can multiply the second row by $1/3$ to obtain:
$$\mathbf{B} = \begin{bmatrix} 1 & 2 \\ 1 & \frac{4}{3} \end{bmatrix}$$
As an example of scalar multiplication and addition, we can multiply the second row by (-1) and add the result to the first row to obtain:
$$\mathbf{C} = \begin{bmatrix} 0 & 2 - \frac{4}{3} \\ 1 & \frac{4}{3} \end{bmatrix} = \begin{bmatrix} 0 & \frac{2}{3} \\ 1 & \frac{4}{3} \end{bmatrix}$$
We can then use scalar multiplication and multiply the first row by $(3/2)$ to obtain:
$$\mathbf{D} = \begin{bmatrix} 0 & 1 \\ 1 & \frac{4}{3} \end{bmatrix}$$

33

We can then use scalar multiplication and addition to multiply the first row by $(-4/3)$ add it to the second row to obtain:

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

Finally, we can swap row 2 and row 1 to obtain:

$$\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Thus using elementary row operations, we have transformed the matrix $\mathbf{A}$ into the matrix $\mathbf{I}_2$.

THEOREM 3.21. *Each elementary row operation can be accomplished by a matrix multiplication.*

PROOF. We'll show that scalar multiplication and row addition can be accomplished by a matrix multiplication. In Exercise 26, you'll be asked to complete the proof for the other two elementary row operations.

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. Without loss of generality, suppose we wish to multiply row 1 by α and add it to row 2, replacing row 2 with the result. Let:

$$(3.23) \quad \mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ \alpha & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

This is simply the identity I_m with an α in the $(2,1)$ position instead of 0. Now consider $\mathbf{EA}$. Let $\mathbf{A}_{\cdot j} = [a_{1j}, a_{2j}, \ldots, a_{mj}]^T$ be the jth column of $\mathbf{A}$. Then :

$$(3.24) \quad \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ \alpha & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix} = \begin{bmatrix} a_{1j} \\ \alpha(a_{1j}) + a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}$$

That is, we have taken the first element of $\mathbf{A}_{\cdot j}$ and multiplied it by α and added it to the second element of $\mathbf{A}_{\cdot j}$ to obtain the new second element of the product. All other elements of $\mathbf{A}_{\cdot j}$ are unchanged. Since we chose an arbitrary column of $\mathbf{A}$, it's clear this will occur in each case. Thus $\mathbf{EA}$ will be the new matrix with rows the same as $\mathbf{A}$ *except* for the second row, which will be replaced by the first row of $\mathbf{A}$ multiplied by the constant α and added to the second row of $\mathbf{A}$. To multiply the ith row of $\mathbf{A}$ and add it to the jth row, we would simply make a matrix E by starting with $\mathbf{I}_m$ and replacing the ith element of row j with α. $\square$

EXERCISE 26. Complete the proof by showing that scalar multiplication and row swapping can be accomplished by a matrix multiplication. [Hint: Scalar multiplication should be easy, given the proof above. For row swap, try multiplying matrix $\mathbf{A}$ from Example 3.20 by:

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

and see what comes out. Can you generalize this idea for arbitrary row swaps?]

Matrices of the kind we've just discussed are called *elementary matrices*. Theorem 3.21 will be important when we study efficient methods for solving linear programming problems. It tells us that *any* set of elementary row operations can be performed by finding the right matrix. That is, suppose I list 4 elementary row operations to perform on matrix A. These elementary row operations correspond to for matrices $\mathbf{E}_1, \ldots, \mathbf{E}_4$. Thus the transformation of $\mathbf{A}$ under these row operations can be written using only matrix multiplication as $\mathbf{B} = \mathbf{E}_4 \cdots \mathbf{E}_1 \mathbf{A}$. This representation is *much* simpler for a computer to keep track of in algorithms that require the transformation of matrices by elementary row operations.

DEFINITION 3.22 (Row Equivalence). Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and let $\mathbf{B} \in \mathbb{R}^{m \times n}$. If there is a sequence of elementary matrices $\mathbf{E}_1, \ldots, \mathbf{E}_k$ so that:

$$\mathbf{B} = \mathbf{E}_k \cdots \mathbf{E}_1 \mathbf{A}$$

then $\mathbf{A}$ and $\mathbf{B}$ are said to be *row equivalent*.

5. Matrix Inverse

DEFINITION 3.23 (Invertible Matrix). Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a square matrix. If there is a matrix $\mathbf{A}^{-1}$ such that

$$(3.25) \quad \mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n$$

then matrix $\mathbf{A}$ is said to be *invertible* (or *nonsingular*) and $\mathbf{A}^{-1}$ is called its *inverse*. If $\mathbf{A}$ is not invertible, it is called a *singular* matrix.

EXERCISE 27. Find the equivalent elementary row operation matrices for Example 3.20. There should be five matrices $\mathbf{E}_1, \ldots, \mathbf{E}_5$ corresponding to the five steps shown. Show that the product of these matrices (in the correct order) yields the identity matrix. Now compute the product $\mathbf{B} = \mathbf{E}_5 \cdots \mathbf{E}_1$. Show that $\mathbf{B} = \mathbf{A}^{-1}$ [Hint: You've done most of the work.]

The proof of the following theorem is beyond the scope of this class. Proofs can be found in [**Lan87**] (Chapter 2) and [**Cul72**] (Chapter 1) and should be covered in a Linear Algebra course (Math 436).

THEOREM 3.24. *If* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *is a square matrix and* $\mathbf{X} \in \mathbb{R}^{n \times n}$ *so that* $\mathbf{X}\mathbf{A} = \mathbf{I}_n$, *then:*
(1) $\mathbf{A}\mathbf{X} = \mathbf{I}_n$
(2) $\mathbf{X} = \mathbf{A}^{-1}$
(3) $\mathbf{A}$ *and* $\mathbf{A}^{-1}$ *can be written as a product of elementary matrices.*

The process we've illustrated in Example 3.20 is an instance of Gauss-Jordan elimination and can be used to find the inverse of any matrix (or to solve systems of linear equations). This process is summarized in Algorithm 5.

DEFINITION 3.25 (Pivoting). In Algorithm 5 when $\mathbf{A}_{ii} \neq 0$, the process performed in Steps 4 and 5 is called *pivoting* on element (i, i).

We illustrate this algorithm in the following example.

EXAMPLE 3.26. Again consider the matrix $\mathbf{A}$ from Example 3.20. We can follow the steps in Algorithm 5 to compute $\mathbf{A}^{-1}$.

Algorithm 5. Gauss-Jordan Elimination for Matrix Inversion

Step 1:
$$\mathbf{X} := \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{array} \right]$$

Step 2: $i := 1$

Step 3 and 4 ($i = 1$): $\mathbf{A}_{11} = 1$, so no swapping is required. Furthermore, replacing $\mathbf{X}_{1\cdot}$ by $(1/1)\mathbf{X}_{1\cdot}$ will not change $\mathbf{X}$.

Step 5 ($i = 1$): We multiply row 1 of $\mathbf{X}$ by -3 and add the result to row 2 of $\mathbf{X}$ to obtain:
$$\mathbf{X} := \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & -2 & -3 & 1 \end{array} \right]$$

Step 6: $i := 1 + 1 = 2$ and $i = n$ so we return to Step 3.

Steps 3 ($i = 2$): The new element $\mathbf{A}_{22} = -2 \neq 0$. Therefore, no swapping is required.

Step 4 ($i = 2$): We replace row 2 of $\mathbf{X}$ with row 2 of $\mathbf{X}$ multiplied by $-1/2$.
$$\mathbf{X} := \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 1 & \frac{3}{2} & -\frac{1}{2} \end{array} \right]$$

Step 5 ($i = 2$): We multiply row 2 of $\mathbf{X}$ by -2 and add the result to row 1 of $\mathbf{X}$ to obtain:
$$\mathbf{X} := \left[\begin{array}{cc|cc} 1 & 0 & -2 & 1 \\ 0 & 1 & \frac{3}{2} & -\frac{1}{2} \end{array} \right]$$

Step 6 ($i = 2$): $i := 2 + 1 = 3$. We now have $i > n$ and the algorithm terminates.

Thus using Algorithm 5 we have computed:
$$\mathbf{A}^{-1} = \left[\begin{array}{cc} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{array} \right]$$

This value should be the result you obtained in Exercise 27.

EXERCISE 28. Does the matrix:
$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

have an inverse? [Hint: Use Gauss-Jordan elimination to find the answer.]

Exercise 29 (**Bonus**). Implement Gauss-Jordan elimination in the programming language of your choice. Illustrate your implementation by using it to solve the previous exercise. [Hint: Implement sub-routines for each matrix operation. You don't have to write them as matrix multiplication, though in Matlab, it might speed up your execution. Then use these subroutines to implement each step of the Gauss-Jordan elimination.]

6. Solution of Linear Equations

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$. Consider the problem:

$$(3.26) \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

There are three possible scenarios:

(1) There is a unique solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.
(2) There are no solutions; i.e., there is no vector $\mathbf{x}$ so that $\mathbf{A}\mathbf{x} = \mathbf{b}$.
(3) There are infinitely many solutions; i.e., there is an infinite set $\mathcal{X} \subseteq \mathbb{R}^n$ such that for all $\mathbf{x} \in \mathcal{X}$, $\mathbf{A}\mathbf{x} = \mathbf{b}$.

We can use Gauss-Jordan elimination to find a solution $\mathbf{x}$ to Equation 3.26 when it occurs. Instead of operating on the augmented $\mathbf{X} = [\mathbf{A}|\mathbf{I}_n]$ in Algorithm 5, we use the augmented matrix $\mathbf{X} := [\mathbf{A}|\mathbf{b}]$. If the algorithm terminates with $\mathbf{A}$ replaced by $\mathbf{I}_n$, then the solution vector resides where $\mathbf{b}$ began. That is, $\mathbf{X}$ is transformed to $\mathbf{X} := [\mathbf{I}_n|\mathbf{A}^{-1}\mathbf{b}]$.

If the algorithm *does not* terminate with $\mathbf{X} := [\mathbf{I}_n|\mathbf{A}^{-1}\mathbf{b}]$, then suppose the algorithm terminates with $\mathbf{X} := [\mathbf{A}'|\mathbf{b}']$. There is *at least* one row in $\mathbf{A}'$ with all zeros. That is $\mathbf{A}'$ has the form:

$$(3.27) \quad \mathbf{A}' = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

In this case, there are two possibilities:

(1) For every zero row in $\mathbf{A}'$, the corresponding element in $\mathbf{b}'$ is 0. In this case, there are an infinite number of alternative solutions to the problem expressed by Equation 3.26.
(2) There is at least one zero row in $\mathbf{A}'$ whose corresponding element in $\mathbf{b}'$ is not zero. In this case, there are no solutions to Equation 3.26.

We illustrate these two possibilities in the following example:

Example 3.27. Consider the system of equations:

$$x_1 + 2x_2 + 3x_3 = 7$$
$$4x_1 + 5x_2 + 6x_3 = 8$$
$$7x_1 + 8x_2 + 9x_3 = 9$$

This yields matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

and right hand side vector $\mathbf{b} = [7, 8, 9]^T$. Applying Gauss-Jordan elimination in this case yields:

$$(3.28) \quad \mathbf{X} := \begin{bmatrix} 1 & 0 & -1 & -\frac{19}{3} \\ 0 & 1 & 2 & \frac{20}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Since the third row is all zeros, there are an infinite number of solutions. An easy way to solve for this set of equations is to let $x_3 = t$, where t may take on any value in $\mathbb{R}$. Then, row 2 of Expression 3.28 tells us that:

$$(3.29) \quad x_2 + 2x_3 = \frac{20}{3} \implies x_2 + 2t = \frac{20}{3} \implies x_2 = \frac{20}{3} - 2t$$

We then solve for x_1 in terms of t. From row 1 of Expression 3.28 we have:

$$(3.30) \quad x_1 - x_3 = -\frac{19}{3} \implies x_1 - t = -\frac{19}{3} \implies x_1 = t - \frac{19}{3}$$

Thus every vector in the set:

$$(3.31) \quad \mathcal{X} = \left\{ \left(t - \frac{19}{3}, \frac{20}{3} - 2t, t \right) : t \in \mathbb{R} \right\}$$

is a solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Conversely, suppose we have the problem:

$$x_1 + 2x_2 + 3x_3 = 7$$
$$4x_1 + 5x_2 + 6x_3 = 8$$
$$7x_1 + 8x_2 + 9x_3 = 10$$

The new right hand side vector is $\mathbf{b} = [7, 8, 20]^T$. Applying Gauss-Jordan elimination in this case yields:

$$(3.32) \quad \mathbf{X} := \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since row 3 of $\mathbf{X}$ has a non-zero element in the $\mathbf{b}'$ column, we know this problem has no solution, since there is no way that we can find values for x_1, x_2 and x_3 satisfying:

$$(3.33) \quad 0x_1 + 0x_2 + 0x_3 = 1$$

EXERCISE 30. Solve the problem

$$x_1 + 2x_2 = 7$$
$$3x_1 + 4x_2 = 8$$

using Gauss-Jordan elimination.

7. Linear Combinations, Span, Linear Independence

DEFINITION 3.28. Let $\mathbf{x}_1, \ldots, \mathbf{x}_m$ be vectors in $\in \mathbb{R}^n$ and let $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ be scalars. Then

$$(3.34) \quad \alpha_1 \mathbf{x}_1 + \cdots + \alpha_m \mathbf{x}_m$$

is a *linear combination* of the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

Clearly, any linear combination of vectors in $\mathbb{R}^n$ is also a vector in $\mathbb{R}^n$.

DEFINITION 3.29 (Span). Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a set of vectors in $\in \mathbb{R}^n$, then the span of $\mathcal{X}$ is the set:

$$(3.35) \quad \operatorname{span}(\mathcal{X}) = \{\mathbf{y} \in \mathbb{R}^n | \mathbf{y} \text{ is a linear combination of vectors in } \mathcal{X}\}$$

DEFINITION 3.30 (Linear Independence). Let $\mathbf{x}_1, \ldots, \mathbf{x}_m$ be vectors in $\in \mathbb{R}^n$. The vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are *linearly dependent* if there exists $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$, not all zero, such that

$$(3.36) \quad \alpha_1 \mathbf{x}_1 + \cdots + \alpha_m \mathbf{x}_m = \mathbf{0}$$

If the set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$ is not linearly dependent, then they are *linearly independent* and Equation 3.36 holds just in case $\alpha_i = 0$ for all $i = 1, \ldots, n$.

EXERCISE 31. Consider the vectors $\mathbf{x}_1 = [0, 0]^T$ and $\mathbf{x}_2 = [1, 0]^T$. Are these vectors linearly independent? Explain why or why not.

EXAMPLE 3.31. In $\mathbb{R}^3$, consider the vectors:

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \ \mathbf{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \ \mathbf{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

We can show these vectors are linearly independent: Suppose there are values $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}$ such that

$$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \alpha_3 \mathbf{x}_3 = 0$$

Then:

$$\begin{bmatrix} \alpha_1 \\ \alpha_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \alpha_2 \\ 0 \\ \alpha_2 \end{bmatrix} \begin{bmatrix} 0 \\ \alpha_3 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_1 + \alpha_2 \\ \alpha_1 + \alpha_3 \\ \alpha_2 + \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Thus we have the system of linear equations:

$$\begin{aligned} \alpha_1 + \alpha_2 &= 0 \\ \alpha_1 \quad\quad + \alpha_3 &= 0 \\ \alpha_2 + \alpha_3 &= 0 \end{aligned}$$

which can be written as the matrix expression:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

This is just a simple matrix equation, but note that the three vectors we are focused on: $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, have become the *columns* of the matrix on the left-hand-side. We can use

Gauss-Jordan elimination to solve this matrix equation yielding: $\alpha_1 = \alpha_2 = \alpha_3 = 0$. Thus these vectors are linearly independent.

REMARK 3.32. It is worthwhile to note that the zero vector $\mathbf{0}$ makes any set of vectors a linearly dependent set.

EXERCISE 32. Prove the remark above.

EXERCISE 33. Show that the vectors

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \ \mathbf{x}_2 = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}, \ \mathbf{x}_3 = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

are *not* linearly independent. [Hint: Following the example, create a matrix whose columns are the vectors in question and solve a matrix equation with right-hand-side equal to zero. Using Gauss-Jordan elimination, show that a zero row results and thus find the infinite set of values solving the system.]

REMARK 3.33. So far we have only given examples and exercises in which the number of vectors was equal to the dimension of the space they occupied. Clearly, we could have, for example, 3 linearly independent vectors in 4 dimensional space. We illustrate this case in the following example.

EXAMPLE 3.34. Consider the vectors:

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \ \mathbf{x}_2 = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Determining linear independence requires us to solve the matrix equation:

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The augmented matrix:

$$\left[\begin{array}{cc|c} 1 & 4 & 0 \\ 2 & 5 & 0 \\ 4 & 6 & 0 \end{array} \right]$$

represents the matrix equation. Using Gauss-Jordan elimination yields:

$$\left[\begin{array}{cc|c} 1 & 4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array} \right]$$

This implies that the following system of equations:

$$\begin{aligned} \alpha_1 + 4\alpha_2 &= 0 \\ \alpha_2 &= 0 \\ 0\alpha_1 + 0\alpha_2 &= 0 \end{aligned}$$

The last equation is tautological (true regardless of the values of α_1 and α_2). The second equation implies $\alpha_2 = 0$. Using this value in first equation implies that $\alpha_1 = 0$. This is the unique solution to the problem and thus the vectors are linearly independent.

The following theorem is related to the example above. It's proof is outside the scope of the course. It should be taught in a Linear Algebra course (Math 436). Proofs can be found in most Linear Algebra textbooks. Again, see [**Lan87**] (Theorem 3.1) for a proof using vector spaces.

THEOREM 3.35. *Let $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$. If $m > n$, then the vectors are linearly dependent.*

8. Basis

DEFINITION 3.36 (Basis). Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a set of vectors in $\mathbb{R}^n$. The set $\mathcal{X}$ is called a *basis* of $\mathbb{R}^n$ if $\mathcal{X}$ is a linearly independent set of vectors and every vector in $\mathbb{R}^n$ is in the span of $\mathcal{X}$. That is, for any vector $\mathbf{w} \in \mathbb{R}^n$ we can find scalar values $\alpha_1, \ldots, \alpha_m$ such that

$$(3.37) \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i$$

EXAMPLE 3.37. We can show that the vectors:

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \ \mathbf{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \ \mathbf{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

form a basis of $\mathbb{R}^3$. We already know that the vectors are linearly independent. To show that $\mathbb{R}^3$ is in their span, chose an arbitrary vector in $\mathbb{R}^m$: $[a, b, c]^T$. Then we hope to find coefficients α_1, α_2 and α_3 so that:

$$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \alpha_3 \mathbf{x}_3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Expanding this, we must find α_1, α_2 and α_3 so that:

$$\begin{bmatrix} \alpha_1 \\ \alpha_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \alpha_2 \\ 0 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha_3 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Just as in Example 3.31, this can be written as an augmented matrix representing a set of linear equations:

$$(3.38) \quad \left[\begin{array}{ccc|c} 1 & 1 & 0 & a \\ 1 & 0 & 1 & b \\ 0 & 1 & 1 & c \end{array} \right]$$

Applying Gauss-Jordan elimination to the augmented matrix yields:

$$(3.39) \quad \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1/2\,a + 1/2\,b - 1/2\,c \\ 0 & 1 & 0 & -1/2\,b + 1/2\,a + 1/2\,c \\ 0 & 0 & 1 & 1/2\,c + 1/2\,b - 1/2\,a \end{array} \right]$$

which clearly has a solution for all a, b, and c. Another way of seeing this is to note that the matrix:

$$(3.40) \quad \mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

is invertible.

The following theorem on the size of a basis in $\mathbb{R}^n$ is outside the scope of this course. A proof can be found in [**Lan87**].

THEOREM 3.38. *If $\mathcal{X}$ is a basis of $\mathbb{R}^n$, then $\mathcal{X}$ contains precisely n vectors.*

EXERCISE 34. Show that the vectors

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \; \mathbf{x}_2 = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}, \; \mathbf{x}_3 = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

are not a basis for $\mathbb{R}^3$. [Hint: See exercise 33.]

We will use the following lemma, which is related to the notion of the basis of $\mathbb{R}^n$ when we come to our formal method for solving linear programming problems.

LEMMA 3.39. *Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_{m+1}\}$ be a linearly dependent set of vectors in $\mathbb{R}^n$ and let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a linearly independent set. Further assume that $\mathbf{x}_{m+1} \neq \mathbf{0}$. Assume $\alpha_1, \ldots, \alpha_{m+1}$ are a set of scalars, not all zero, so that*

$$(3.41) \quad \sum_{i=1}^{m+1} \alpha_i \mathbf{x}_i = \mathbf{0}$$

For any $j \in \{1, \ldots, m\}$ such that $\alpha_j \neq 0$, if we replace $\mathbf{x}_j$ in the set $\mathcal{X}$ with $\mathbf{x}_{m+1}$, then this new set of vectors is linearly independent.

PROOF. Clearly α_{m+1} cannot be zero, since we assumed that $\mathcal{X}$ is linearly independent. Since $\mathbf{x}_{m+1} \neq \mathbf{0}$, we know there is at least one other α_i $(i = 1, \ldots, m)$ not zero. Without loss of generality, assume that $\alpha_m \neq 0$ (if not, rearrange the vectors to make this true).

We can solve for $\mathbf{x}_{m+1}$ using this equation to obtain:

$$(3.42) \quad \mathbf{x}_{m+1} = \sum_{i=1}^{m} -\frac{\alpha_i}{\alpha_{m+1}} \mathbf{x}_i$$

Suppose, without loss of generality, we replace $\mathbf{x}_m$ by $\mathbf{x}_{m+1}$ in $\mathcal{X}$. We now proceed by contradiction. Assume this new set is linearly dependent. There there exists constants $\beta_1, \ldots, \beta_{m-1}, \beta_{m+1}$, not all zero, such that:

$$(3.43) \quad \beta_1 \mathbf{x}_1 + \cdots + \beta_{m-1} \mathbf{x}_{m-1} + \beta_{m+1} \mathbf{x}_{m+1} = \mathbf{0}.$$

Again, we know that $\beta_{m+1} \neq 0$ since the set $\{\mathbf{x}_1, \ldots, \mathbf{x}_{m-1}\}$ is linearly independent because $\mathcal{X}$ is linearly independent. Then using Equation 3.42 we see that:

$$(3.44) \quad \beta_1 \mathbf{x}_1 + \cdots + \beta_{m-1} \mathbf{x}_{m-1} + \beta_{m+1} \left(\sum_{i=1}^{m} -\frac{\alpha_i}{\alpha_{m+1}} \mathbf{x}_i \right) = \mathbf{0}.$$

We can rearrange the terms in this sum as:

$$(3.45) \quad \left(\beta_1 - \frac{\beta_{m+1}\alpha_1}{\alpha_{m+1}}\right)\mathbf{x}_1 + \cdots + \left(\beta_{m-1} - \frac{\beta_{m+1}\alpha_{m-1}}{\alpha_{m+1}}\right)\mathbf{x}_{m-1} - \frac{\alpha_m}{\alpha_{m+1}}\mathbf{x}_m = \mathbf{0}$$

The fact that $\alpha_m \neq 0$ and $\beta_{m+1} \neq 0$ and $\alpha_{m+1} \neq 0$ means we have found $\gamma_1, \ldots, \gamma_m$, not all zero, such that $\gamma_1\mathbf{x}_1 + \cdots + \gamma_m\mathbf{x}_m = 0$, contradicting our assumption that $\mathcal{X}$ was linearly independent. This contradiction completes the proof. $\qquad\square$

REMARK 3.40. This lemma proves an interesting result. If $\mathcal{X}$ is a basis of $\mathbb{R}^m$ and $\mathbf{x}_{m+1}$ is another, non-zero, vector in $\mathbb{R}^m$, we can swap $\mathbf{x}_{m+1}$ for any vector $\mathbf{x}_j$ in $\mathcal{X}$ as long as when we express $\mathbf{x}_{m+1}$ as a linear combination of vectors in $\mathcal{X}$ the coefficient of $\mathbf{x}_j$ is not zero. That is, since $\mathcal{X}$ is a basis of $\mathbb{R}^m$ we can express:

$$\mathbf{x}_{m+1} = \sum_{i=1}^m \alpha_i\mathbf{x}_i$$

As long as $\alpha_j \neq 0$, then we can replace $\mathbf{x}_j$ with $\mathbf{x}_{m+1}$ and still have a basis of $\mathbb{R}^m$.

EXERCISE 35. Prove the following theorem: *In $\mathbb{R}^n$ every set of n linearly independent vectors is a basis.* [Hint: Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the set. Use the fact that $\alpha_1\mathbf{x}_1 + \cdots + \alpha_n\mathbf{x}_n = \mathbf{0}$ has exactly one solution.]

9. Rank

DEFINITION 3.41 (Row Rank). Let $\mathbf{A} \in \mathbb{R}^{m\times n}$. The *row rank* of $\mathbf{A}$ is the size of the largest set of row (vectors) from $\mathbf{A}$ that are linearly independent.

EXERCISE 36. By analogy define the *column rank* of a matrix. [Hint: You don't need a hint.]

THEOREM 3.42. *If $\mathbf{A} \in \mathbb{R}^{m\times n}$ is a matrix, then elementary row operations on A do not change the row rank.*

PROOF. Denote the rows of $\mathbf{A}$ by $R = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$; i.e., $\mathbf{a}_i = \mathbf{A}_{i\cdot}$. Suppose we apply elementary row operations on these rows to obtain a new matrix $\mathbf{A}'$. First, let us consider row swapping. Obviously if the size of the largest subset of R that is linearly independent has value $k \leq \min\{m, n\}$ then swapping two rows will not change this. Let us now consider the effect of multiplying $\mathbf{a}_i$ by α and adding it to $\mathbf{a}_j$ (where neither row is the zero row). Then we replace $\mathbf{a}_j$ with $\alpha\mathbf{a}_i + \mathbf{a}_j$. There are two possibilities:

Case 1: There are no $\alpha_1, \ldots, \alpha_m$ (not all zero) so that:

$$\alpha_1\mathbf{a}_1 + \cdots + \alpha_m\mathbf{a}_m = \mathbf{0}$$

Suppose there is some $\beta_1, \ldots, \beta_m$ not all zero so that:

$$\beta_1\mathbf{a}_1 + \cdots + \beta_{j-1}\mathbf{a}_{j-1} + \beta_j(\alpha\mathbf{a}_i + \mathbf{a}_j) + \cdots + \beta_m\mathbf{a}_m = \mathbf{0}$$

In this case, we see immediately that:

$$\beta_1\mathbf{a}_1 + \cdots + \beta_{j-1}\mathbf{a}_{j-1} + \beta_j\alpha\mathbf{a}_i + \beta_j\mathbf{a}_j + \cdots + \beta_m\mathbf{a}_m = \mathbf{0}$$

and thus if we have $\alpha_k = \beta_k$ for $k \neq i$ and $\alpha_i = \beta_i + \beta_j\alpha$ then we have an $\alpha_1, \ldots, \alpha_m$ (not all zero) so that:

$$\alpha_1\mathbf{a}_1 + \cdots + \alpha_m\mathbf{a}_m = \mathbf{0}$$

which is a contradiction.

Case 2: Suppose that the size of the largest set of linearly independent rows is k. Denote such a set by $S = \{\mathbf{a}_{s_1}, \ldots, \mathbf{a}_{s_k}\}$. There are several possibilities: (i) Both $\mathbf{a}_i$ and $\mathbf{a}_j$ are in this set. In this case, we simply replace our argument above with constants α_1 through α_k and the result is the same.

(ii) $\mathbf{a}_i$ and $\mathbf{a}_j$ are not in this set, in which case we know that there are $\alpha_{s_1}, \ldots, \alpha_{s_k}$ and $\beta_{s_1}, \ldots, \beta_{s_k}$ so that:

$$\alpha_{s_1}\mathbf{a}_1 + \cdots + \alpha_{s_k}\mathbf{a}_k = \mathbf{a}_i$$
$$\beta_{s_1}\mathbf{a}_1 + \cdots + \beta_{s_k}\mathbf{a}_k = \mathbf{a}_j$$

But this implies that $\alpha\mathbf{a}_i + \mathbf{a}_j$ can also be written as a linear combination of the elements of S and thus the rank of $\mathbf{A}'$ is no larger than the rank of $\mathbf{A}$.

(iii) Now suppose that $\mathbf{a}_i$ in the set S. Then there are constants $\alpha_{s_1}, \ldots, \alpha_{s_k}$ so that:

$$\alpha_{s_1}\mathbf{a}_1 + \cdots + \alpha_{s_k}\mathbf{a}_k = \mathbf{a}_j$$

Without loss of generality, suppose that $\mathbf{a}_i = \mathbf{a}_{s_1}$ then:

$$\alpha\mathbf{a}_i + \alpha_{s_1}\mathbf{a}_i + \alpha_{s_2}\mathbf{a}_{s_2} + \cdots + \alpha_{s_k}\mathbf{a}_k = \alpha\mathbf{a}_i + \mathbf{a}_j$$

Again, this implies that $\alpha\mathbf{a}_i + \mathbf{a}_j$ is still a linear combination of the elements of S and so we cannot have increased the size of the largest linearly independent set of vectors, nor could we have decreased it.

(iv) Finally, suppose that $\mathbf{a}_j \in S$. Again let $\mathbf{a}_j = \mathbf{a}_{s_1}$. Then there are constants $\alpha_{s_1}, \ldots, \alpha_{s_k}$

$$\alpha_{s_1}\mathbf{a}_J + \alpha_{s_2}\mathbf{a}_{s_2} + \cdots + \alpha_{s_k}\mathbf{a}_{s_k} = \mathbf{a}_i$$

Apply Lemma 3.39 to replace $\mathbf{a}_j$ in S with some other row vector $\mathbf{a}_l$. If $l = i$, then we reduce to sub-case (iii). If $l \neq i$, then we reduce to sub-case (ii).

Finally, suppose we multiply a row by α. This is reduces to the case of multiplying row i by $\alpha - 1$ and adding it to row i, which is covered in the above analysis. This completes the proof. $\square$

We have the following theorem, whose proof is again outside the scope of this course. There are very nice proofs available in [**Lan87**].

THEOREM 3.43. *If $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a matrix, then the row rank of $\mathbf{A}$ is equal to the column rank of $\mathbf{A}$. Further, $\mathrm{rank}(\mathbf{A}) \leq \min\{m, n\}$.*

Lastly, we will not prove the following theorem, but it should be clear from all the work we have done up to this point.

THEOREM 3.44. *If $\mathbf{A} \in \mathbb{R}^{m \times m}$ (i.e., $\mathbf{A}$ is a square matrix) and $\mathrm{rank}(\mathbf{A}) = m$, then $\mathbf{A}$ is invertible.*

DEFINITION 3.45. Suppose that $\mathbf{A} \in \mathbb{R}^{m \times n}$ and let $m \leq n$. Then $\mathbf{A}$ has *full row rank* if $\mathrm{rank}(\mathbf{A}) = m$.

EXAMPLE 3.46. Again consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

By now you should suspect that it does not have full row rank. Recall that the application of Gauss-Jordan elimination transforms $\mathbf{A}$ into the matrix

$$\mathbf{A}' = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

No further transformation is possible. It's easy to see that the first two rows of $\mathbf{A}'$ are linearly independent. (Note that the first row vector has a non-zero element in its first position and zero in it's second position, while the second row vector has a non-zero element in the second position and a zero element in the first position. Because of this, it's impossible to find any non-zero linear combination of those vectors that leads to zero.) Thus we conclude the matrix $\mathbf{A}$ has the same rank as matrix $\mathbf{A}'$ which is 2.

EXERCISE 37. Change one number in matrix $\mathbf{A}$ in the preceding example to create a new matrix $\mathbf{B}$ that as full row rank. Show that your matrix has rank 3 using Gauss-Jordan elimination.

10. Solving Systems with More Variables than Equations

Suppose now that $\mathbf{A} \in \mathbb{R}^{m \times n}$ where $m \leq n$. Let $\mathbf{b} \in \mathbb{R}^m$. Then the equation:

$$(3.46) \quad \mathbf{Ax} = \mathbf{b}$$

has more variables than equations and is underdetermined and if $\mathbf{A}$ has full row rank then the system will have an infinite number of solutions. We can formulate an expression to describe this infinite set of solutions.

Sine $\mathbf{A}$ has full row rank, we may choose any m linearly independent columns of $\mathbf{A}$ corresponding to a subset of the variables, say $x_{i_1}, \ldots, x_{i_m}$. We can use these to form the matrix

$$(3.47) \quad \mathbf{B} = [\mathbf{A}_{\cdot i_1} \cdots \mathbf{A}_{\cdot i_m}]$$

from the columns $\mathbf{A}_{\cdot i_1}, \ldots, \mathbf{A}_{\cdot i_m}$ of $\mathbf{A}$, so that B is invertible. It should be clear at this point that B will be invertible precisely because we've chosen m linearly independent column vectors. We can then use elementary column operations to write the matrix $\mathbf{A}$ as:

$$(3.48) \quad \mathbf{A} = [\mathbf{B}|\mathbf{N}]$$

The matrix $\mathbf{N}$ is composed of the $n - m$ other columns of $\mathbf{A}$ not in $\mathbf{B}$. We can similarly sub-divide the column vector $\mathbf{x}$ and write:

$$(3.49) \quad [\mathbf{B}|\mathbf{N}] \begin{bmatrix} \mathbf{x_B} \\ \mathbf{x_N} \end{bmatrix} = \mathbf{b}$$

where the vector $\mathbf{x_B}$ are the variables corresponding to the columns in $\mathbf{B}$ and the vector $\mathbf{x_N}$ are the variables corresponding to the columns of the matrix $\mathbf{N}$.

DEFINITION 3.47 (Basic Variables). For historical reasons, the variables in the vector $\mathbf{x_B}$ are called the *basic variables* and the variables in the vector $\mathbf{x_N}$ are called the *non-basic* variables.

We can use matrix multiplication to expand the left hand side of this expression as:

$$(3.50) \quad \mathbf{Bx_B} + \mathbf{Nx_N} = \mathbf{b}$$

The fact that $\mathbf{B}$ is composed of all linearly independent columns implies that applying Gauss-Jordan elimination to it will yield an $m \times m$ identity and thus that $\mathbf{B}$ is invertible. We can solve for basic variables $\mathbf{x_B}$ in terms of the non-basic variables:

$$(3.51) \quad \mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{Nx_N}$$

We can find an arbitrary solution to the system of linear equations by choosing values for the variables the non-basic variables and solving for the basic variable values using Equation 3.51.

DEFINITION 3.48. (Basic Solution) When we assign $\mathbf{x_N} = 0$, the resulting solution for $\mathbf{x}$ is called a *basic solution* and

$$(3.52) \quad \mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b}$$

EXAMPLE 3.49. Consider the problem:

$$(3.53) \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Then we can let $x_3 = 0$ and:

$$(3.54) \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$$

We then solve[1]:

$$(3.55) \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{B}^{-1} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} \frac{-19}{3} \\ \frac{20}{3} \end{bmatrix}$$

Other basic solutions could be formed by creating $\mathbf{B}$ out of columns 1 and 3 or columns 2 and 3.

EXERCISE 38. Find the two other basic solutions in Example 3.49 corresponding to

$$\mathbf{B} = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} 1 & 3 \\ 4 & 6 \end{bmatrix}$$

In each case, determine what the matrix $\mathbf{N}$ is. [Hint: Find the solutions any way you like. Make sure you record exactly which x_i ($i \in \{1, 2, 3\}$) is equal to zero in each case.]

---

[1]Thanks to Doug Mercer, who found a typo below that was fixed.

11. Solving Linear Programs with Matlab

In this section, we'll show how to solve Linear Programs using Matlab. Matlab assumes that all linear programs are input in the following form:

$$(3.56) \quad \begin{cases} \min \quad z(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ \quad s.t. \ \ \mathbf{Ax} \leq \mathbf{b} \\ \qquad \ \ \mathbf{Hx} = \mathbf{r} \\ \qquad \ \ \mathbf{x} \geq \mathbf{l} \\ \qquad \ \ \mathbf{x} \leq \mathbf{u} \end{cases}$$

Here $\mathbf{c} \in \mathbb{R}^{n \times 1}$, so there are n variables in the vector $\mathbf{x}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^{m \times 1}$, $\mathbf{H} \in \mathbb{R}^{l \times n}$ and $\mathbf{r} \in \mathbb{R}^{l \times 1}$. The vectors $\mathbf{l}$ and $\mathbf{u}$ are lower and upper bounds respectively on the decision variables in the vector $\mathbf{x}$.

The Matlab command for solving linear programs is `linprog` and it takes the parameters:

(1) $\mathbf{c}$,
(2) $\mathbf{A}$,
(3) $\mathbf{b}$,
(4) $\mathbf{H}$,
(5) $\mathbf{r}$,
(6) $\mathbf{l}$,
(7) $\mathbf{u}$

If there are no inequality constraints, then we set $\mathbf{A} = [\,]$ and $\mathbf{b} = [\,]$ in Matlab; i.e., $\mathbf{A}$ and $\mathbf{b}$ are set as the *empty matrices*. A similar requirement holds on $\mathbf{H}$ and $\mathbf{r}$ if there are no equality constraints. If some decision variables have lower bounds and others don't, the term `-inf` can be used to set a lower bound at $-\infty$ (in $\mathbf{l}$). Similarly, the term `inf` can be used if the upper bound on a variable (in $\mathbf{u}$) is infinity. The easiest way to understand how to use Matlab is to use it on an example.

EXAMPLE 3.50. Suppose I wish to design a diet consisting of Raman noodles and ice cream. I'm interested in spending as little money as possible but I want to ensure that I eat at least 1200 calories per day and that I get at least 20 grams of protein per day. Assume that each serving of Raman costs \$1 and contains 100 calories and 2 grams of protein. Assume that each serving of ice cream costs \$1.50 and contains 200 calories and 3 grams of protein.

We can construct a linear programming problem out of this scenario. Let x_1 be the amount of Raman we consume and let x_2 be the amount of ice cream we consume. Our objective function is our cost:

$$(3.57) \quad x_1 + 1.5x_2$$

Our constraints describe our protein requirements:

$$(3.58) \quad 2x_1 + 3x_2 \geq 20$$

and our calorie requirements (expressed in terms of 100's of calories):

$$(3.59) \quad x_1 + 2x_2 \geq 12$$

This leads to the following linear programming problem:

$$
(3.60) \quad
\begin{cases}
\min \ x_1 + 1.5x_2 \\
\text{s.t.} \ 2x_1 + 3x_2 \geq 20 \\
\qquad x_1 + 2x_2 \geq 12 \\
\qquad x_1, x_2 \geq 0
\end{cases}
$$

Before turning to Matlab, let's investigate this problem in standard form. To transform the problem to standard form, we introduce *surplus* variables s_1 and s_2 and our problem becomes:

$$
(3.61) \quad
\begin{cases}
\min \ x_1 + 1.5x_2 \\
\text{s.t.} \ 2x_1 + 3x_2 - s_1 = 20 \\
\qquad x_1 + 2x_2 - s_2 = 12 \\
\qquad x_1, x_2, s_1, s_2 \geq 0
\end{cases}
$$

This leads to a set of two linear equations with four variables:

$$
2x_1 + 3x_2 - s_1 = 20x_1 + 2x_2 - s_2 = 12
$$

We can look at the various results from using Expression 3.51 and Definition 3.48. Let:

$$
(3.62) \quad \mathbf{A} = \begin{bmatrix} 2 & 3 & -1 & 0 \\ 1 & 2 & 0 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 20 \\ 12 \end{bmatrix}
$$

Then our vector of decision variables is:

$$
(3.63) \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \end{bmatrix}
$$

We can use Gauss-Jordan elimination on the augmented matrix:

$$
\left[\begin{array}{cccc|c} 2 & 3 & -1 & 0 & 20 \\ 1 & 2 & 0 & -1 & 12 \end{array} \right]
$$

Suppose we wish to have $\mathbf{x_B} = [s_1 \ s_2]^T$. Then we would transform the matrix as:

$$
\left[\begin{array}{cccc|c} -2 & -3 & 1 & 0 & -20 \\ -1 & -2 & 0 & 1 & -12 \end{array} \right]
$$

This would result in $x_1 = 0$, $x_2 = 0$ (because $\mathbf{x_N} = [x_1 \ x_2]^T$ and $s_1 = -20$ and $s_2 = -12$. Unfortunately, this is not a *feasible* solution to our linear programming problem because we require $s_1, s_2 \geq 0$. Alternatively we could look at the case when $\mathbf{x_B} = [x_1 \ x_2]^T$ and $\mathbf{x_N} = [s_1 \ s_2]^T$. Then we would perform Gauss-Jordan elimination on the augmented matrix to obtain:

$$
\left[\begin{array}{cccc|c} 1 & 0 & -2 & 3 & 4 \\ 0 & 1 & 1 & -2 & 4 \end{array} \right]
$$

That is, $x_1 = 4, x_2 = 4$ and of course $s_1 = s_2 = 0$. Notice something interesting:

$$
\begin{bmatrix} -2 & 3 \\ 1 & -2 \end{bmatrix} = - \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}^{-1}
$$
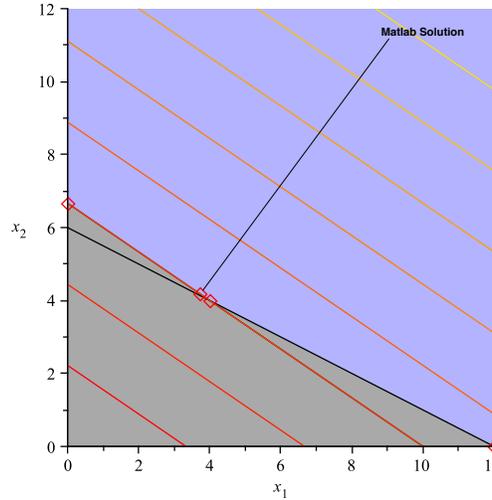
Figure 3.1. The feasible region for the diet problem is unbounded and there are alternative optimal solutions, since we are seeking a minimum, we travel in the opposite direction of the gradient, so toward the origin to reduce the objective function value. Notice that the level curves hit one side of the boundary of the feasible region.

This is not an accident, it's because we started with a negative identity matrix inside the augmented matrix. The point $x_1 = 4, x_2 = 4$ is a point of intersection, shown in Figure 3.1. It also happens to be one of the alternative optimal solutions of this problem. Notice in Figure 3.1 that the level curves of the objective function are parallel to one of the sides of the boundary of the feasible region. If we continued in this way, we could actually construct all the points of intersection that make up the boundary of the feasible region. We'll can do one more, suppose $\mathbf{x_B} = [x_1 \ \ s_1]^T$. Then we would use Gauss-Jordan elimination to obtain:

$$\left[\begin{array}{cccc|c} 1 & 2 & 0 & -1 & 12 \\ 0 & 1 & 1 & -2 & 4 \end{array} \right]$$

Notice there are now columns of the identity matrix in the columns corresponding to s_1 and x_1. That's how we know we're solving for s_1 and x_2. We have $x_1 = 12$ and $s_1 = 4$. By definition $x_1 = s_2 = 0$. This corresponds to the point $x_1 = 12, x_2 = 0$ shown in Figure 3.1.

Let's use Matlab to solve this problem. Our original problem is:

$$\begin{cases} \min \ x_1 + 1.5x_2 \\ \ s.t. \ \ 2x_1 + 3x_2 \geq 20 \\ \qquad \ x_1 + 2x_2 \geq 12 \\ \qquad \ x_1, x_2 \geq 0 \end{cases}$$

49

This is not in a form Matlab likes, so we change it by multiplying the constraints by -1 on both sides to obtain:

$$\begin{cases} \min \ x_1 + 1.5x_2 \\ s.t. \ -2x_1 - 3x_2 \leq -20 \\ \quad -x_1 - 2x_2 \leq -12 \\ \quad x_1, x_2 \geq 0 \end{cases}$$

Then we have:

$$\mathbf{c} = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} -2 & -3 \\ -1 & -2 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -20 \\ -12 \end{bmatrix}$$

$$\mathbf{H} = \mathbf{r} = []$$

$$\mathbf{l} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \mathbf{u} = []$$

The Matlab code to solve this problem is shown in Figure 3.2 The solution Matlab returns in

```
%%Solve the Diet Linear Programming Problem
c = [1 1.5]';
A = [[-2 -3];...
     [-1 -2]];
b = [-20 -12]';
H = [];
r = [];
l = [0 0]';
u = [];
[x obj] = linprog(c,A,b,H,r,l,u);
```

Figure 3.2. Matlab input for solving the diet problem. Note that we are solving a *minimization* problem. Matlab assumes all problems are *mnimization* problems, so we don't need to multiply the objective by -1 like we would if we started with a maximization problem.

the $\mathbf{x}$ variable is $x_1 = 3.7184$ and $x_2 = 4.1877$, note this is on the line of alternative optimal solutions, but it is not at either end of the line. I prefer to have a little less ice cream, so I'd rather have the alternative optimal solution $x_1 = x_2 = 4$.

EXERCISE 39. In previous example, you could also have just used the problem in standard form with the surplus variables and had $\mathbf{A} = \mathbf{b} = []$ and defined $\mathbf{H}$ and $\mathbf{r}$ instead. Use Matlab to solve the diet problem in *standard form*. Compare your results to Example 3.50

CHAPTER 4

Convex Sets, Functions and Cones and Polyhedral Theory

In this chapter, we will cover all of the geometric prerequisites for understanding the theory of linear programming. We will use the results in this section to prove theorems about the Simplex Method in other sections.

1. Convex Sets

DEFINITION 4.1 (Convex Set). Let $X \subseteq \mathbb{R}^n$. Then the set X is convex if and only if for all pairs $\mathbf{x}_1, \mathbf{x}_2 \in X$ we have $\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in X$ for all $\lambda \in [0, 1]$.

The definition of convexity seems complex, but it is easy to understand. First recall that if $\lambda \in [0, 1]$, then the point $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ is on the line segment connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathbb{R}^n$. For example, when $\lambda = 1/2$, then the point $\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2$ is the midpoint between $\mathbf{x}_1$ and $\mathbf{x}_2$. In fact, for every point $\mathbf{x}$ on the line connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ we can find a value $\lambda \in [0, 1]$ so that $\mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2$. Then we can see that, convexity asserts that if $\mathbf{x}_1, \mathbf{x}_2 \in X$, then every point on the line connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ is also in the set X.

DEFINITION 4.2 (Positive Combination). Let $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$. If $\lambda_1, \ldots, \lambda_m > 0$ and then

$$(4.1) \quad \mathbf{x} = \sum_{i=1}^{m} \lambda_i \mathbf{x}_i$$

is called a *positive combination* of $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

DEFINITION 4.3 (Convex Combination). Let $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$. If $\lambda_1, \ldots, \lambda_m \in [0, 1]$ and

$$\sum_{i=1}^{m} \lambda_i = 1$$

then

$$(4.2) \quad \mathbf{x} = \sum_{i=1}^{m} \lambda_i \mathbf{x}_i$$

is called a *convex combination* of $\mathbf{x}_1, \ldots, \mathbf{x}_m$. If $\lambda_i < 1$ for all $i = 1, \ldots, m$, then Equation 4.2 is called a *strict convex combination*.

REMARK 4.4. If you recall the definition of linear combination, we can see that we move from the very general to the very specific as we go from linear combinations to positive combinations to convex combinations. A linear combination of points or vectors allowed us to choose any real values for the coefficients. A positive combination restricts us to positive values, while a convex combination asserts that those values must be non-negative and sum to 1.

EXAMPLE 4.5. Figure 4.1 illustrates a convex and non-convex set. Non-convex sets have
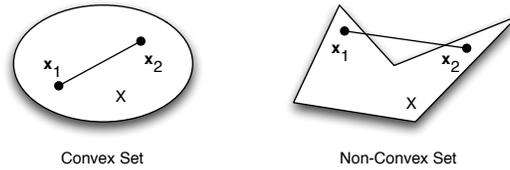
Figure 4.1. Examples of Convex Sets: The set on the left (an ellipse and its interior) is a convex set; every pair of points inside the ellipse can be connected by a line contained entirely in the ellipse. The set on the right is clearly not convex as we've illustrated two points whose connecting line is not contained inside the set.

some resemblance to crescent shapes or have components that look like crescents.

THEOREM 4.6. *The intersection of a finite number of convex sets in $\mathbb{R}^n$ is convex.*

PROOF. Let $C_1, \ldots, C_n \subseteq \mathbb{R}^n$ be a finite collection of convex sets. Let

$$(4.3) \quad C = \bigcap_{i=1}^{n} C_i$$

be the set formed from the intersection of these sets. Choose $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $\lambda \in [0,1]$. Consider $\mathbf{x} = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$. We know that $\mathbf{x}_1, \mathbf{x}_2 \in C_1, \ldots, C_n$ by definition of C. By convexity, we know that $\mathbf{x} \in C_1, \ldots, C_n$ by convexity of each set. Therefore, $\mathbf{x} \in C$. Thus C is a convex set. $\qquad\square$

2. Convex and Concave Functions

DEFINITION 4.7 (Convex Function). A function $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function if it satisfies:

$$(4.4) \quad f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda)f(\mathbf{x}_2)$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and for all $\lambda \in [0,1]$.

This definition is illustrated in Figure 4.2. When f is a univariate function, this definition can be shown to be equivalent to the definition you learned in Calculus I (Math 140) using first and second derivatives.

DEFINITION 4.8 (Concave Function). A function $f : \mathbb{R}^n \to \mathbb{R}$ is a concave function if it satisfies:

$$(4.5) \quad f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \geq \lambda f(\mathbf{x}_1) + (1-\lambda)f(\mathbf{x}_2)$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and for all $\lambda \in [0,1]$ [1].

To visualize this definition, simply flip Figure 4.2 upside down. The following theorem is a powerful tool that can be used to show sets are convex. It's proof is outside the scope of the class, but relatively easy.

---

[1]Thanks to Greg Ference and Veselka Kafedzhieva for catching a typo in this definition.
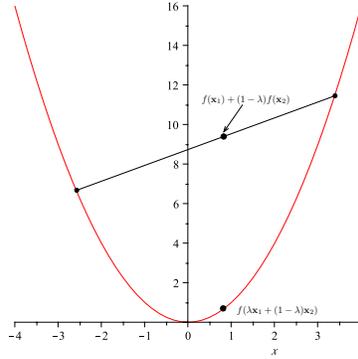
Figure 4.2. A convex function: A convex function satisfies the expression $f(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda)f(\mathbf{x}_2)$ for all $\mathbf{x}_1$ and $\mathbf{x}_2$ and $\lambda \in [0, 1]$.

THEOREM 4.9. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a convex function. Then the set* $C = \{\mathbf{x} \in \mathbb{R}^n : f(x) \leq c\}$*, where* $c \in \mathbb{R}$*, is a convex set.*

EXERCISE 40. Prove the Theorem 4.9. [Hint: Skip ahead and read the proof of Lemma 4.15. Follow the steps in that proof, but apply them to f.]

3. Polyhedral Sets

Important examples of convex sets are polyhedral sets, the multi-dimensional analogs of polygons in the plane. In order to understand these structures, we must first understand hyperplanes and half-spaces.

DEFINITION 4.10 (Hyperplane). Let $\mathbf{a} \in \mathbb{R}^n$ be a constant vector in n-dimensional space and let $b \in \mathbb{R}$ be a constant scalar. The set of points

$$(4.6) \qquad H = \left\{\mathbf{x} \in \mathbb{R}^n | \mathbf{a}^T\mathbf{x} = b\right\}$$

is a *hyperplane* in n-dimensional space. *Note the use of column vectors for* $\mathbf{a}$ *and* $\mathbf{x}$ *in this definition.*

EXAMPLE 4.11. Consider the hyper-plane $2x_1 + 3x_2 + x_3 = 5$. This is shown in Figure 4.3. This hyperplane is composed of the set of points $(x_1, x_2, x_3) \in \mathbb{R}^3$ satisfying $2x_1 + 3x_2 + x_3 = 5$. This can be plotted implicitly or explicitly by solving for one of the variables, say x_3. We can write x_3 as a function of the other two variables as:

$$(4.7) \qquad x_3 = 5 - 2x_1 - 3x_2$$

DEFINITION 4.12 (Half-Space). Let $\mathbf{a} \in \mathbb{R}^n$ be a constant vector in n-dimensional space and let $b \in \mathbb{R}$ be a constant scalar. The sets of points

$$(4.8) \qquad H_l = \left\{\mathbf{x} \in \mathbb{R}^n | \mathbf{a}^T\mathbf{x} \leq b\right\}$$

$$(4.9) \qquad H_u = \left\{\mathbf{x} \in \mathbb{R}^n | \mathbf{a}^T\mathbf{x} \geq b\right\}$$

are the half-spaces defined by the hyperplane $\mathbf{a}^T\mathbf{x} = b$.

EXAMPLE 4.13. Consider the two dimensional hyperplane (line) $x_1 + x_2 = 1$. Then the two half-spaces associated with this hyper-plane are shown in Figure 4.4. A half-space is
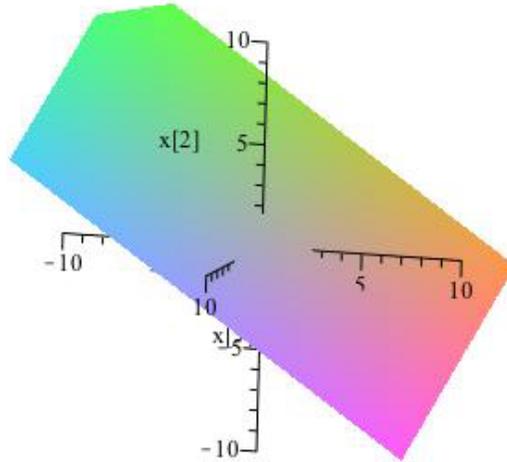
Figure 4.3. A hyperplane in 3 dimensional space: A hyperplane is the set of points satisfying an equation $\mathbf{a}^T\mathbf{x} = b$, where k is a constant in $\mathbb{R}$ and $\mathbf{a}$ is a constant vector in $\mathbb{R}^n$ and $\mathbf{x}$ is a variable vector in $\mathbb{R}^n$. The equation is written as a matrix multiplication using our assumption that all vectors are column vectors.

(a) H_l (b) H_u
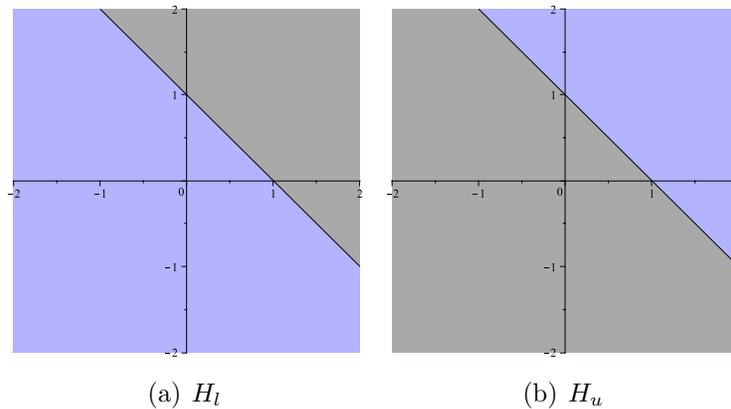
Figure 4.4. Two half-spaces defined by a hyper-plane: A half-space is so named because any hyper-plane divides $\mathbb{R}^n$ (the space in which it resides) into two halves, the side "on top" and the side "on the bottom."

so named because the hyperplane $\mathbf{a}^T\mathbf{x} = b$ literally separates $\mathbb{R}^n$ into two halves: the half above the hyperplane and the half below the hyperplane.

LEMMA 4.14. *Every hyper-plane is convex.*

PROOF. Let $\mathbf{a} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ and let H be the hyperplane defined by $\mathbf{a}$ and b. Choose $\mathbf{x}_1, \mathbf{x}_2 \in H$ and $\lambda \in [0, 1]$. Let $\mathbf{x} = \lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2$. By definition we know that:

$$\mathbf{a}^T\mathbf{x}_1 = b$$
$$\mathbf{a}^T\mathbf{x}_2 = b$$

Then we have:

$$(4.10) \quad \mathbf{a}^T\mathbf{x} = \mathbf{a}^T\left[\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2\right] = \lambda\mathbf{a}^T\mathbf{x}_1 + (1-\lambda)\mathbf{a}^T\mathbf{x}_2 = \lambda b + (1-\lambda)b = b$$

Thus, $\mathbf{x} \in H$ and we see that H is convex. This completes the proof. $\qquad \square$

LEMMA 4.15. *Every half-space is convex.*

PROOF. Let $\mathbf{a} \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Without loss of generality, consider the half-space H_l defined by $\mathbf{a}$ and b. For arbitrary $\mathbf{x}_1$ and $\mathbf{x}_2$ in H_l we have:

$$\mathbf{a}^T\mathbf{x}_1 \leq b$$
$$\mathbf{a}^T\mathbf{x}_2 \leq b$$

Suppose that $\mathbf{a}^T\mathbf{x}_1 = b_1 \leq b$ and $\mathbf{a}^T\mathbf{x}_2 = b_2 \leq b$. Again let $\mathbf{x} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$. Then:

$$(4.11) \quad \mathbf{a}^T\mathbf{x} = \mathbf{a}^T\left[\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2\right] = \lambda\mathbf{a}^T\mathbf{x}_1 + (1-\lambda)\mathbf{a}^T\mathbf{x}_2 = \lambda b_1 + (1-\lambda)b_2$$

Since $\lambda \leq 1$ and $1 - \lambda \leq 1$ and $\lambda \geq 0$ we know that $\lambda b_1 \leq \lambda b$, since $b_1 \leq b$. Similarly we know that $(1-\lambda)b_2 \leq (1-\lambda)b$, since $b_2 \leq b$. Thus:

$$(4.12) \quad \lambda b_1 + (1-\lambda)b_2 \leq \lambda b + (1-\lambda)b = b$$

Thus we have shown that $\mathbf{a}^T\mathbf{x} \leq b$. The case for H_u is identical with the signs of the inequalities reversed. This completes the proof. $\qquad \square$

Using these definitions, we are now in a position to define polyhedral sets, which will be the subject of our study for most of the remainder of this chapter.

DEFINITION 4.16 (Polyhedral Set). If $P \subseteq \mathbb{R}^n$ is the intersection of a finite number of half-spaces, then P is a *polyhedral set*. Formally, let $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^n$ be a finite set of constant vectors and let $b_1, \ldots, b_m \in \mathbb{R}$ be constants. Consider the set of half-spaces:

$$H_i = \{\mathbf{x}|\mathbf{a}_i^T\mathbf{x} \leq b_i\}$$

Then the set:

$$(4.13) \quad P = \bigcap_{i=1}^{m} H_i$$

is a *polyhedral set*.

It should be clear that we can represent any polyhedral set using a matrix inequality. The set P is defined by the set of vectors $\mathbf{x}$ satisfying:

$$(4.14) \quad \mathbf{A}\mathbf{x} \leq \mathbf{b},$$

where the *rows* of $\mathbf{A} \in \mathbb{R}^{m \times n}$ are made up of the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ and $\mathbf{b} \in \mathbb{R}^m$ is a column vector composed of elements $b_1, \ldots, b_m$.

THEOREM 4.17. *Every polyhedral set is convex.*

EXERCISE 41. Prove Theorem 4.17. [Hint: You can prove this by brute force, verifying convexity. You can also be clever and use two results that we've proved in the notes.]

4. Rays and Directions

Recall the definition of a *line* (Definition 1.10 from Chapter 1. A ray is a one sided line.

DEFINITION 4.18 (Ray). Let $\mathbf{x_0} \in \mathbb{R}^n$ be a point and and let $\mathbf{d} \in \mathbb{R}^n$ be a vector called the *direction*. Then the *ray* with vertex $\mathbf{x_0}$ and direction $\mathbf{d}$ is the collection of points $\{\mathbf{x} | \mathbf{x} = \mathbf{x_0} + \lambda \mathbf{d}, \ \lambda \geq 0\}$.

EXAMPLE 4.19. We will use the same point and direction as we did for a line in Chapter 1. Let $\mathbf{x_0} = [2, 1]^T$ and let $\mathbf{d} = [2, 2]^T$. Then the ray defined by $\mathbf{x_0}$ and $\mathbf{d}$ is shown in Figure 1.4. The set of points is $R = \{(x, y) \in \mathbb{R}^2 : x = 2 + 2\lambda, y = 1 + 2\lambda, \lambda \geq 0\}$.
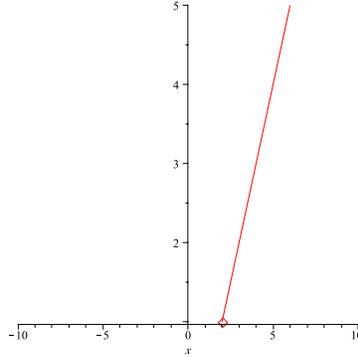
Figure 4.5. A Ray: The points in the graph shown in this figure are in the set produced using the expression $\mathbf{x_0} + \mathbf{d}\lambda$ where $\mathbf{x_0} = [2, 1]^T$ and $\mathbf{d} = [2, 2]^T$ and $\lambda \geq 0$.

Rays are critical for understanding unbounded convex sets. Specifically, a set is unbounded, in a sense, only if you can show that it contains a ray. An interesting class of unbounded convex sets are convex cones:

DEFINITION 4.20 (Convex Cone). Let $C \subseteq \mathbb{R}^n$ be a convex set. Then C is a *convex cone* if for all $\mathbf{x} \in C$ and for all $\lambda \in \mathbb{R}$ with $\lambda \geq 0$ we have $\lambda \mathbf{x} \in C$.

LEMMA 4.21. *Every convex cone contains the origin.*

EXERCISE 42. Prove the previous lemma.

The fact that every convex cone contains the origin by Lemma 4.21 along with the fact that for every point $\mathbf{x} \in C$ we have $\lambda \mathbf{x} \in C$ ($\lambda \geq 0$) implies that the ray $\mathbf{0} + \lambda \mathbf{x} \subseteq C$. Thus, since every point $\mathbf{x} \in C$ must be on a ray, it follows that a convex cone is just made up of rays beginning at the origin.

Another key element to understanding unbounded convex sets is the notion of direction. A direction can be thought of as a "direction of travel" from a starting point inside an unbounded convex set so that you (the traveler) can continue moving forever and never leave the set.

DEFINITION 4.22 (Direction of a Convex Set). Let C be a convex set. Then $\mathbf{d} \neq \mathbf{0}$ is a *(recession) direction of the convex set* if for all $\mathbf{x_0} \in C$ the ray with vertex $\mathbf{x_0}$ and direction $\mathbf{d}$ is contained entirely in C. Formally, for all $\mathbf{x_0} \in C$ we have:

$$(4.15) \quad \{\mathbf{x} : \mathbf{x} = \mathbf{x_0} + \lambda \mathbf{d}, \ \lambda \geq 0\} \subseteq C$$

EXAMPLE 4.23. Consider the unbounded convex set shown in Figure 4.6. This set has direction $[1, 0]^T$. To see this note that for any positive scaling parameter λ and for any vertex
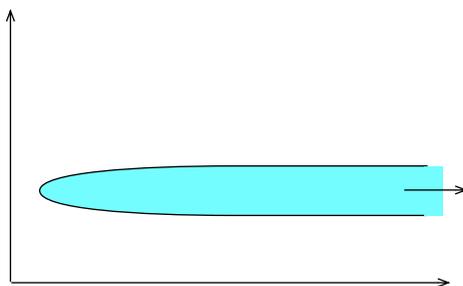
Figure 4.6. Convex Direction: Clearly every point in the convex set (shown in blue) can be the vertex for a ray with direction $[1, 0]^T$ contained entirely in the convex set. Thus $[1, 0]^T$ is a direction of this convex set.

point $\mathbf{x}_0$, we can draw an arrow pointing to the right (in the direction of $[1, 0]^T$) with vertex at $\mathbf{x}_0$ scaled by λ that is entirely contained in the convex set.

EXERCISE 43. Prove the following: Let $C \subseteq \mathbb{R}^n$ be a convex cone and let $\mathbf{x}_1, \mathbf{x}_2 \in C$. If $\alpha, \beta \in \mathbb{R}$ and $\alpha, \beta \geq 0$, then $\alpha \mathbf{x}_1 + \beta \mathbf{x}_2 \in C$. [Hint: Use the definition of convex cone and the definition of convexity with $\lambda = 1/2$, then multiply by 2.]

EXERCISE 44. Use Exercise 43 to prove that if $C \subseteq \mathbb{R}^n$ is a convex cone, then every element $\mathbf{x} \in C$ (except the origin) is also a direction of C.

5. Directions of Polyhedral Sets

There is a unique relationship between the defining matrix $\mathbf{A}$ of a polyhedral set P and a direction of this set that is particularly useful when we assume that P is located in the positive orthant of $\mathbb{R}^n$ (i.e., $\mathbf{x} \geq 0$ are defining constraints of P).

THEOREM 4.24. *Suppose that $P \subseteq \mathbb{R}^n$ is a polyhedral set defined by:*

(4.16) $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\}$

If $\mathbf{d}$ is a direction of P, then the following hold:

(4.17) $\mathbf{A}\mathbf{d} \leq \mathbf{0}, \ \mathbf{d} \geq \mathbf{0}, \ \mathbf{d} \neq \mathbf{0}$.

PROOF. The fact that $\mathbf{d} \neq \mathbf{0}$ is clear from the definition of direction of a convex set. Furthermore, $\mathbf{d}$ is a direction if and only if

(4.18) $\mathbf{A}(\mathbf{x} + \lambda \mathbf{d}) \leq \mathbf{b}$

(4.19) $\mathbf{x} + \lambda \mathbf{d} \geq \mathbf{0}$

for all $\lambda > 0$ and for all $\mathbf{x} \in P$ (which is to say $\mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$). But then

$\mathbf{A}\mathbf{x} + \lambda \mathbf{A}\mathbf{d} \leq \mathbf{b}$

for all $\lambda > 0$. This can only be true if $\mathbf{Ad} \leq \mathbf{0}$. Likewise:$\mathbf{x} + \lambda\mathbf{d} \geq \mathbf{0}$ holds for all $\lambda > 0$ if and only if $\mathbf{d} \geq \mathbf{0}$. This completes the proof. $\qquad\square$

COROLLARY 4.25. *If*

$$(4.20) \quad P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

and $\mathbf{d}$ *is a direction of* P, *then* $\mathbf{d}$ *must satisfy:*

$$(4.21) \quad \mathbf{Ad} = \mathbf{0},\ \mathbf{d} \geq \mathbf{0},\ \mathbf{d} \neq \mathbf{0}.$$

EXERCISE 45. Prove the corollary above.

EXAMPLE 4.26. Consider the polyhedral set defined by the equations:

$$x_1 - x_2 \leq 1$$
$$2x_1 + x_2 \geq 6$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

This set is clearly unbounded as we showed in class and it has at least one direction. The direction $\mathbf{d} = [0,1]^T$ pointing directly up is a direction of this set. This is illustrated in Figure 4.7. In this example, we have:
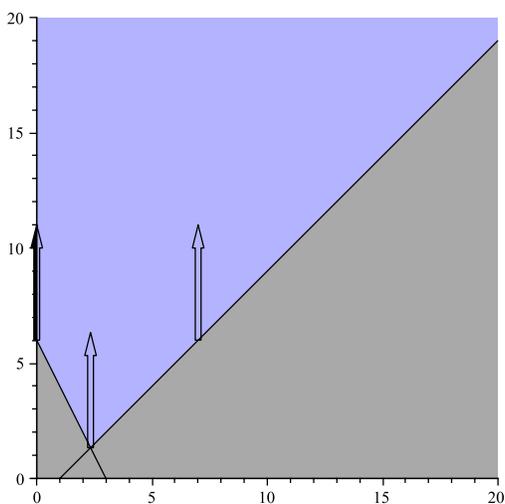
Figure 4.7. An Unbounded Polyhedral Set: This unbounded polyhedral set has many directions. One direction is $[0,1]^T$.

$$(4.22) \quad \mathbf{A} = \begin{bmatrix} 1 & -1 \\ -2 & -1 \end{bmatrix}$$

Note, the second inequality constraint was a greater-than constraint. We reversed it to a less-than inequality constraint $-2x_1 - x_2 \leq -6$ by multiplying by -1. For our chosen direction $\mathbf{d} = [0,1]^T$, we can see that:

$$(4.23) \quad \mathbf{Ad} = \begin{bmatrix} 1 & -1 \\ -2 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \leq \mathbf{0}$$

Clearly $\mathbf{d} \geq \mathbf{0}$ and $\mathbf{d} \neq \mathbf{0}$.

6. Extreme Points

DEFINITION 4.27 (Extreme Point of a Convex Set). Let C be a convex set. A point $\mathbf{x}_0 \in C$ is a *extreme point* of C if there are *no points* $\mathbf{x}_1$ and $\mathbf{x}_2$ ($\mathbf{x}_1 \neq \mathbf{x}_0$ or $\mathbf{x}_2 \neq \mathbf{x}_0$) so that $\mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2$ for some $\lambda \in (0, 1)$.[2]

An extreme point is simply a point in a convex set C that cannot be expressed as a strict convex combination of any other pair of points in C. We will see that extreme points must be located in specific locations in convex sets.

DEFINITION 4.28 (Boundary of a set). Let $C \subseteq \mathbb{R}^n$ be (convex) set. A point $\mathbf{x}_0 \in C$ is on the *boundary* of C if for all $\epsilon > 0$,

$$B_\epsilon(\mathbf{x}_0) \cap C \neq \emptyset \text{ and}$$
$$B_\epsilon(\mathbf{x}_0) \cap \mathbb{R}^n \setminus C \neq \emptyset$$

EXAMPLE 4.29. A convex set, its boundary and a boundary point are illustrated in Figure 4.8.
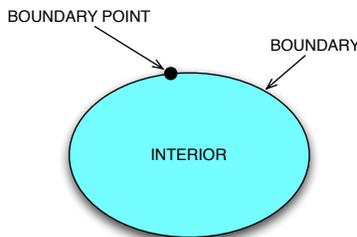
Figure 4.8. Boundary Point: A boundary point of a (convex) set C is a point in the set so that for *every* ball of any radius centered at the point contains some points inside C and some points outside C.

LEMMA 4.30. *Suppose C is a convex set. If $\mathbf{x}$ is an extreme point of C, then $\mathbf{x}$ is on the boundary of C.*

PROOF. Suppose not, then $\mathbf{x}$ is not on the boundary and thus there is some $\epsilon > 0$ so that $B_\epsilon(\mathbf{x}_0) \subset C$. Since $B_\epsilon(\mathbf{x}_0)$ is a hypersphere, we can choose two points $\mathbf{x}_1$ and $\mathbf{x}_2$ on the boundary of $B_\epsilon(\mathbf{x}_0)$ so that the line segment between these points passes through the center of $B_\epsilon(\mathbf{x}_0)$. But this center point is $\mathbf{x}_0$. Therefore $\mathbf{x}_0$ is the mid-point of $\mathbf{x}_1$ and $\mathbf{x}_2$ and since $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 = \mathbf{x}_0$ with $\lambda = 1/2$ it follows that $\mathbf{x}_0$ cannot be an extreme point, since it is a strict convex combination of $\mathbf{x}_1$ and $\mathbf{x}_2$. This completes the proof. $\square$

Most important in our discussion of linear programming will be the extreme points of polyhedral sets that appear in linear programming problems. The following theorem establishes the relationship between extreme points in a polyhedral set and the intersection of hyperplanes in such a set.

---

[2]Thanks to Bob Pakzad-Hurson who fixed a typo in this definition in Version ≤ 1.4.

THEOREM 4.31. *Let $P \subseteq \mathbb{R}^n$ be a polyhedral set and suppose P is defined as:*

(4.24) $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. A point $\mathbf{x}_0 \in P$ is an extreme point of P if and only if $\mathbf{x}_0$ is the intersection of n linearly independent hyperplanes from the set defining P.

REMARK 4.32. The easiest way to see this as relevant to linear programming is to assume that

(4.25) $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\}$

In this case, we could have $m < n$. In that case, P is composed of the intersection of $n + m$ half-spaces. The first m are for the rows of $\mathbf{A}$ and the second n are for the non-negativity constraints. An extreme point comes from the intersection of n of the hyperplanes defining these half-spaces. We might have m come from the constraints $\mathbf{Ax} \leq \mathbf{b}$ and the other $n - m$ from $\mathbf{x} \geq \mathbf{0}$.

PROOF. ($\Leftarrow$) Suppose that $\mathbf{x}_0$ is the intersection of n hyperplanes. Then $\mathbf{x}_0$ lies on n hyperplanes. By way of contradiction of suppose that $\mathbf{x}_0$ is not an extreme point. Then there are two points $\overline{\mathbf{x}}, \hat{\mathbf{x}} \in P$ and a scalar $\lambda \in (0, 1)$ so that

$$\mathbf{x}_0 = \lambda \overline{\mathbf{x}} + (1 - \lambda)\hat{\mathbf{x}}$$

If this is true, then for some $\mathbf{G} \in \mathbb{R}^{n \times n}$ whose rows are drawn from $\mathbf{A}$ and a vector $\mathbf{g}$ whose entries are drawn from the vector $\mathbf{b}$, so that $\mathbf{Gx}_0 = \mathbf{g}$. But then we have:

(4.26) $\mathbf{g} = \mathbf{Gx}_0 = \lambda \mathbf{G}\overline{\mathbf{x}} + (1 - \lambda)\mathbf{G}\hat{\mathbf{x}}$

and $\mathbf{G}\overline{\mathbf{x}} \leq \mathbf{g}$ and $\mathbf{G}\hat{\mathbf{x}} \leq \mathbf{g}$ (since $\overline{\mathbf{x}}, \hat{\mathbf{x}} \in P$). But the only way for Equation 4.26 to hold is if

(1) $\mathbf{G}\overline{\mathbf{x}} = \mathbf{g}$ and
(2) $\mathbf{G}\hat{\mathbf{x}} = \mathbf{g}$

The fact that the hyper-planes defining $\mathbf{x}_0$ are linearly independent implies that the solution to $\mathbf{Gx}_0 = \mathbf{g}$ is unique. (That is, we have chosen n equations in n unknowns and $\mathbf{x}_0$ is the solution to these n equations.) Therefore, it follows that $\mathbf{x}_0 = \overline{\mathbf{x}} = \hat{\mathbf{x}}$ and thus $\mathbf{x}_0$ is an extreme point since it cannot be expressed as a convex combination of other points in P.

($\Rightarrow$) By Lemma 4.30, we know that any extreme point $\mathbf{x}_0$ lies on the boundary of P and therefore there is at least one row $\mathbf{A}_{i\cdot}$ such that $\mathbf{A}_{i\cdot}\mathbf{x}_0 = b_i$ (otherwise, clearly $\mathbf{x}_0$ does not lie on the boundary of P). By way of contradiction, suppose that $\mathbf{x}_0$ is the intersection of $r < n$ linearly independent hyperplanes (that is, only these r constraints are binding). Then there is a matrix $\mathbf{G} \in \mathbb{R}^{r \times n}$ whose rows are drawn from $\mathbf{A}$ and a vector $\mathbf{g}$ whose entries are drawn from the vector $\mathbf{b}$, so that $\mathbf{Gx}_0 = \mathbf{g}$. Linear independence of the hyperplanes implies that the rows of $\mathbf{G}$ are linearly independent and therefore there is a non-zero solution to the equation $\mathbf{Gd} = \mathbf{0}$. To see this, apply Expression 3.51 and choose solution in which $\mathbf{d}$ is non-zero. Then we can find an $\epsilon > 0$ such that:

(1) If $\overline{\mathbf{x}} = \mathbf{x}_0 + \epsilon\mathbf{d}$, then $\mathbf{G}\overline{\mathbf{x}} = \mathbf{g}$ *and* all non-binding constraints at $\mathbf{x}_0$ remain non-binding at $\overline{\mathbf{x}}$.
(2) If $\hat{\mathbf{x}} = \mathbf{x}_0 - \epsilon\mathbf{d}$, then $\mathbf{G}\hat{\mathbf{x}} = \mathbf{g}$ *and* all non-binding constraints at $\mathbf{x}_0$ remain non-binding at $\hat{\mathbf{x}}$.

These two facts hold since $\mathbf{G}\mathbf{d} = \mathbf{0}$ and if $\mathbf{A}_{i\cdot}$ is a row of $\mathbf{A}$ with $\mathbf{A}_{i\cdot}\mathbf{x}_0 < b_i$ (or $\mathbf{x} > 0$), then there is at least one non-zero ϵ so that $\mathbf{A}_{i\cdot}(\mathbf{x}_0 \pm \epsilon\mathbf{d}) < b_i$ (or $\mathbf{x}_0 \pm \epsilon\mathbf{d} > \mathbf{0}$) still holds and therefore $(\mathbf{x}_0 \pm \epsilon\mathbf{d}) \in P$. Since we have a finite number of constraints that are non-binding, we may choose ϵ to be the smallest value so that the previous statements hold for all of them. Finally we can choose $\lambda = 1/2$ and see that $\mathbf{x}_0 = \lambda\overline{\mathbf{x}} + (1 - \lambda)\hat{\mathbf{x}}$ and $\overline{\mathbf{x}}, \hat{\mathbf{x}} \in P$. Thus $\mathbf{x}_0$ cannot have been an extreme point, contradicting our assumption. This completes the proof. $\square$

DEFINITION 4.33. Let P be the polyhedral set from Theorem 4.31. If $\mathbf{x}_0$ is an extreme point of P and *more* than n hyperplanes are binding at $\mathbf{x}_0$, then $\mathbf{x}_0$ is called a *degenerate* extreme point.

DEFINITION 4.34 (Face). Let P be a polyhedral set defined by

$$P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. If $X \subseteq P$ is defined by a non-empty set of binding linearly independent hyperplanes, then X is a *face* of P.

That is, there is some set of linearly independent rows $A_{i_1\cdot}, \ldots A_{i_l\cdot}$ with $i_l < m$ so that when $\mathbf{G}$ is the matrix made of these rows and $\mathbf{g}$ is the vector of $b_{i_1}, \ldots, b_{i_l}$ then:

(4.27) $\quad X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{G}\mathbf{x} = \mathbf{g} \text{ and } \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$

In this case we say that X has *dimension* $n - l$.

REMARK 4.35. Based on this definition, we can easily see that an extreme point, which is the intersection n linearly independent hyperplanes is a face of dimension zero.

DEFINITION 4.36 (Edge and Adjacent Extreme Point). An edge of a polyhedral set P is any face of dimension 1. Two extreme points are called *adjacent* if they share $n - 1$ binding constraints. That is, they are connected by an edge of P.

EXAMPLE 4.37. Consider the polyhedral set defined by the system of inequalities:

$$3x_1 + x_2 \leq 120$$
$$x_1 + 2x_2 \leq 160$$
$$\frac{28}{16}x_1 + x_2 \leq 100$$
$$x_1 \leq 35$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

The polyhedral set is shown in Figure 4.9. The extreme points of the polyhedral set are shown as large diamonds and correspond to intersections of binding constraints. Note the extreme point $(16, 72)$ is *degenerate* since it occurs at the intersection of three binding constraints $3x_1 + x_2 \leq 120$, $x_1 + 2x_2 \leq 160$ and $\frac{28}{16}x_1 + x_2 <= 100$. All the faces of the polyhedral set are shown in bold. They are locations where one constraint (or half-space) is binding. An example of a pair of adjacent extreme points is $(16, 72)$ and $(35, 15)$, as they are connected by the edge defined by the binding constraint $3x_1 + x_2 \leq 120$.
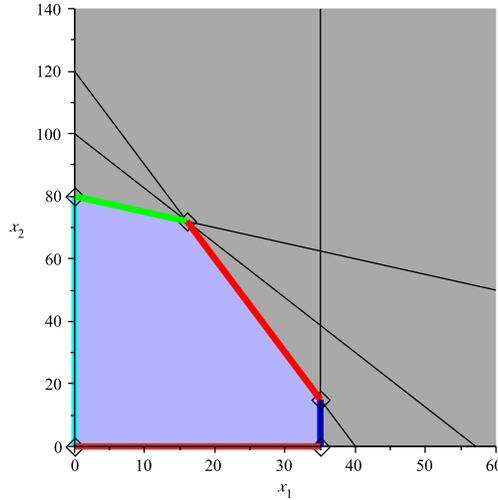
Figure 4.9. A Polyhedral Set: This polyhedral set is defined by five half-spaces and has a single degenerate extreme point located at the intersection of the binding constraints $3x_1 + x_2 \leq 120$, $x_1 + 2x_2 \leq 160$ and $\frac{28}{16}x_1 + x_2 <= 100$. All faces are shown in bold.

EXERCISE 46. Consider the polyhedral set defined by the system of inequalities:

$$4x_1 + x_2 \leq 120$$
$$x_1 + 8x_2 \leq 160$$
$$x_1 + x_2 \leq 30$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

Identify all extreme points and edges in this polyhedral set and their binding constraints. Are any extreme points degenerate? List all pairs of adjacent extreme points.

7. Extreme Directions

DEFINITION 4.38 (Extreme Direction). Let $C \subseteq \mathbb{R}^n$ be a convex set. Then a direction $\mathbf{d}$ of C is an *extreme direction* if there are no two other directions $\mathbf{d}_1$ and $\mathbf{d}_2$ of C ($\mathbf{d}_1 \neq \mathbf{d}$ and $\mathbf{d}_2 \neq \mathbf{d}$) and scalars $\lambda_1, \lambda_2 > 0$ so that $\mathbf{d} = \lambda_1 \mathbf{d}_1 + \lambda_2 \mathbf{d}_2$.

We have already seen by Theorem 4.24 that is P is a polyhedral set in the positive orthant of $\mathbb{R}^n$ with form:

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

then a direction $\mathbf{d}$ of P is characterized by the set of inequalities and equations

$$\mathbf{Ad} \leq \mathbf{0}, \ \mathbf{d} \geq \mathbf{0}, \ \mathbf{d} \neq \mathbf{0}.$$

Clearly two directions $\mathbf{d}_1$ and $\mathbf{d}_2$ with $\mathbf{d_1} = \lambda \mathbf{d}_2$ for some $\lambda \geq 0$ may both satisfy this system. To isolate a unique set of directions, we can normalize and construct the set:

$$(4.28) \quad D = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{Ad} \leq \mathbf{0}, \ \mathbf{d} \geq \mathbf{0}, \mathbf{e}^T\mathbf{d} = 1\}$$

here we are interested only in directions satisfying $\mathbf{e}^T\mathbf{d} = 1$. This is a normalizing constraint that will chose only vectors whose components sum to 1.

THEOREM 4.39. *A direction $\mathbf{d} \in D$ is an extreme direction of P if and only if $\mathbf{d}$ is an extreme point of D when D is taken as a polyhedral set.*

PROOF. ($\Rightarrow$)Suppose that $\mathbf{d}$ is an extreme point of D (as a polyhedral set) and not an extreme direction of P. Then there exist two directions $\mathbf{d}_1$ and $\mathbf{d}_2$ of P and two constants λ_1 and λ_2 with $\lambda_1, \lambda_2 \geq 0$ so that $\mathbf{d} = \lambda_1\mathbf{d}_1 + \lambda_2\mathbf{d}_2$. Without loss of generality, we may assume that $\mathbf{d}_1$ and $\mathbf{d}_2$ are vectors satisying $\mathbf{e}^T\mathbf{d}_i = 1$ ($i = 1, 2$). If not, then we can scale them so their components sum to 1 and adjust λ_1 and λ_2 accordingly. But this implies that:

$$1 = \mathbf{e}^T\mathbf{d} = \lambda_1\mathbf{e}^T\mathbf{d}_1 + \lambda_2\mathbf{e}^T\mathbf{d}_2 = \lambda_1 + \lambda_2$$

Further, the fact that $\mathbf{d}_1$ and $\mathbf{d}_2$ are directions of P implies they must be in D. Thus we have found a convex combination of element of D that equals $\mathbf{d}$, contradicting our assumption that $\mathbf{d}$ was an extreme point.

($\Leftarrow$)Conversely, suppose that $\mathbf{d}$ is an extreme direction of P whose components sum to 1 and not an extreme point of D. (Again, we could scale $\mathbf{d}$ if needed.) Then $\mathbf{d}$ cannot be recovered from a positive combination of directions $\mathbf{d}_1$ and $\mathbf{d}_2$. But if $\mathbf{d}$ is not an extreme point of D, then we know $\lambda_1\mathbf{d}_1 + \lambda_2\mathbf{d}_2 = \mathbf{d}$ for $\mathbf{d}_1, \mathbf{d}_2 \in D$ and $\lambda_1 + \lambda_2 = 1$ and $\lambda_1, \lambda_2 \in (0, 1)$. This is clearly contradictory. Every strict convex combination is a positive combination and therefore our assumption that $\mathbf{d}$ was an extreme direction was false. $\square$

EXAMPLE 4.40. Let's consider Example 4.26 again. The polyhedral set in this example was defined by the $\mathbf{A}$ matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ -2 & -1 \end{bmatrix}$$

and the $\mathbf{b}$ vector:

$$\mathbf{b} = \begin{bmatrix} 1 \\ -6 \end{bmatrix}$$

If we assume that $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \ \mathbf{x} \geq 0\}$, then the set of extreme directions of P is the same as the set of extreme points of the set

$$D = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{A}\mathbf{d} \leq \mathbf{0}, \ \mathbf{d} \geq \mathbf{0}, \mathbf{e}^T\mathbf{d} = 1\}$$

Then we have the set of directions $\mathbf{d} = [d_1, d_2]^T$ so that:

$$d_1 - d_2 \leq 0$$
$$-2d_1 - d_2 \leq 0$$
$$d_1 + d_2 = 1$$
$$d_1 \geq 0$$
$$d_2 \geq 0$$

The feasible region (which is really only the line $d_1 + d_2 = 1$) is shown in red in Figure 4.10. The critical part of this figure is the red line. It is the true set D. As a line, it has two extreme points: $(0, 1)$ and $(1/2, 1/2)$. Note that $(0, 1)$ as an extreme point is one of the direction $[0, 1]^T$ we illustrated in Example 4.26.
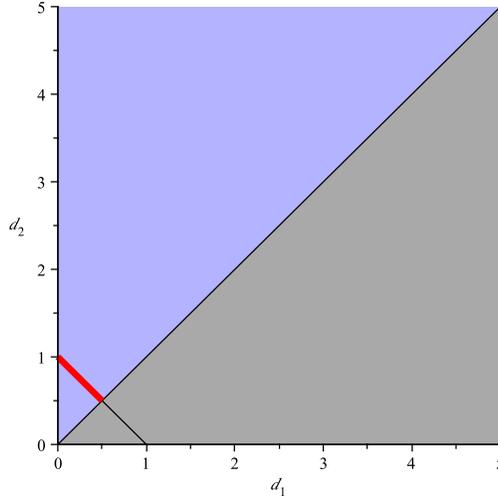
Figure 4.10. Visualization of the set D: This set really consists of the set of points on the red line. This is the line where $d_1 + d_2 = 1$ and all other constraints hold. This line has two extreme points $(0, 1)$ and $(1/2, 1/2)$.

EXERCISE 47. Show that $\mathbf{d} = [1/2, 1/2]^T$ is a direction of the polyhedral set P from Example 4.26. Now find a non-extreme direction (whose components sum to 1) using the feasible region illustrated in the previous example. Show that the direction you found is a direction of the polyhedral set. Create a figure like Figure 4.7 to illustrate *both* these directions.

8. Caratheodory Characterization Theorem

LEMMA 4.41. *The polyhedral set defined by:*

$$P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

has a finite, non-zero number of extreme points (assuming that $\mathbf{A}$ is not an empty matrix)[3].

PROOF. Let $\mathbf{x} \in P$. If $\mathbf{x}$ is an extreme point, then the theorem is proved. Suppose that $\mathbf{x}$ is not an extreme point. Then by Theorem 4.31, $\mathbf{x}$ lies at the intersection of $r < n$ binding constraints (where r could be zero). The fact that $\mathbf{x}$ is not an extreme point of P implies the existence of $\mathbf{y}_1, \mathbf{y}_2 \in P$ and a $\lambda > 0$ so that $\mathbf{x} = \lambda \mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2$. For this to hold, we know that the r constraints that bind at $\mathbf{x}$ must also be binding at $\mathbf{y}_1$ and $\mathbf{y}_2$.

Let $\mathbf{d} = \mathbf{y}_2 - \mathbf{y}_1$ be the direction from $\mathbf{y}_1$ to $\mathbf{y}_2$. We can see that:

$$(4.29) \quad \begin{aligned} \mathbf{y}_1 &= \mathbf{x} - (1 - \lambda)\mathbf{d} \\ \mathbf{y}_2 &= \mathbf{x} + \lambda \mathbf{d} \end{aligned}$$

The values $\mathbf{x} + \gamma \mathbf{d}$ and $\mathbf{x} - \gamma \mathbf{d}$ for $\gamma > 0$ correspond to motion from $\mathbf{x}$ along the direction of $\mathbf{d}$. From Expression 4.29, we can move in either the positive or negative direction of $\mathbf{d}$ and remain in P. Let γ be the *largest* value so that both $\mathbf{x} + \gamma \mathbf{d}$ or $\mathbf{x} - \gamma \mathbf{d}$ is in P. Clearly we cannot move in both directions infinitely far since $\mathbf{x} \geq \mathbf{0}$ and hence $\gamma < \infty$. Without loss of

---

[3]Thanks to Bob Pakzah-Hurson for the suggestion to improve the statement of this lemma.

generality, suppose that γ is determined by $\mathbf{x} - \gamma\mathbf{d}$. (That is, $\mathbf{x} - (\gamma + \epsilon)\mathbf{d} \notin P$ for $\epsilon > 0$). Let $\mathbf{x}_1 = \mathbf{x} - \gamma\mathbf{d}$. Since r hyperplanes are binding at $\mathbf{x}$ (and $\mathbf{y}_1$ and $\mathbf{y}_2$) it is clear that these same hyperplanes are binding at $\mathbf{x}_1$ and at least one more (because of how we selected γ). Thus there are at least $r + 1$ binding hyperplanes at $\mathbf{x}_1$. If $r + 1 = n$, then we have identified an extreme point. Otherwise, we may repeat the process until we find an extreme point.

To show that the number of extreme points is finite, we note that every extreme point is the intersection of n linearly independent hyperplanes defining P. There are $n + m$ hyperplanes defining P and therefore the number of possible extreme points is limited by $\binom{n+m}{n}$. This completes the proof. $\qquad\square$

LEMMA 4.42. *Let P be a non-empty polyhedral set. Then the set of directions of P is empty if and only if P is bounded.*

PROOF. Clearly if P is bounded then it cannot have a direction. If P were contained in a ball $B_r(\mathbf{x}_0)$ then we know that for every $\mathbf{x} \in P$ we have $|\mathbf{x} - \mathbf{x}_0| < r$. If $\mathbf{d}$ is a direction of P, then we have $\mathbf{x} + \lambda\mathbf{d}$ for all $\lambda > 0$. We simply need to choose λ large enough so that $|\mathbf{x} + \lambda\mathbf{d} - \mathbf{x}_0| > r$.

If P has no directions, then there is some absolute upper bound on the value of $|\mathbf{x}|$ for all $\mathbf{x} \in P$. Let r be this value. Then trivially, $B_{r+1}(\mathbf{0})$ contains P and so P is bounded. $\quad\square$

LEMMA 4.43. *Let P be a non-empty unbounded polyhedral set. Then the number extreme directions of P is finite and non-zero.*

PROOF. The result follows immediately from Theorem 4.39 and Lemma 4.41. $\qquad\square$

THEOREM 4.44. *Let P be a non-empty, unbounded polyhedral set defined by:*

$$P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

(where we assume $\mathbf{A}$ is not an empty matrix). Suppose that P has extreme points $\mathbf{x}_1, \ldots, \mathbf{x}_k$ and extreme directions $\mathbf{d}_1, \ldots, \mathbf{d}_l$. If $\mathbf{x} \in P$, then there exists constants $\lambda_1, \ldots, \lambda_k$ and $\mu_1, \ldots, \mu_l$ such that:

$$\mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{x}_i + \sum_{j=1}^{l} \mu_j \mathbf{d}_j$$

(4.30)
$$\sum_{i=1}^{k} \lambda_i = 1$$
$$\lambda_i \geq 0 \quad i = 1, \ldots, k$$
$$\mu_j \geq 0 \quad 1, \ldots, l$$

PROOF. Let $\mathbf{x} \in P$. We will show that we can identify $\lambda_1, \ldots, \lambda_k$ and $\mu_1, \ldots, \mu_l$ making Expression 4.30 true. Define the set:

(4.31) $\quad \overline{P} = P \cap \{\mathbf{x} \in \mathbb{R}^n : \mathbf{e}^T\mathbf{x} \leq M\}$

where M is a large constant so that $\mathbf{e}^T\mathbf{x}_i < M$ for $i = 1, \ldots, k$ and $\mathbf{e}^T\mathbf{x} < M$. That is, M is large enough so that the sum of the components of any extreme point is less than M and the sum of the components of $\mathbf{x}$ is less than M.

It is clear that $\overline{P}$ is bounded. In fact, if P is bounded, then $\overline{P} = P$. Furthermore $\overline{P}$ is a polyhedral set contained in P and therefore the extreme points of P are also the extreme points of $\overline{P}$. Define

$$\overline{E}_p = \{\mathbf{x}_1, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_{k+u}\}$$

as the extreme points of $\overline{P}$. By Theorem 4.41 we know that $0 \le u < \infty$. If $\mathbf{x} \in \overline{E}_p$, then $\mathbf{x}$ can be written as a convex combination of the elements of $\overline{E}_p$. Therefore, assume that $\mathbf{x} \notin \overline{E}_p$. Now, suppose that the system of equations $\mathbf{Gy} = \mathbf{g}$ represents the binding hyperplanes (constraints) of $\overline{P}$ that are active at $\mathbf{x}$. Clearly $\mathrm{rank}(\mathbf{G}) < n$ (otherwise, $\mathbf{x}$ would be an extreme point of $\overline{E}_p$).

Let $\mathbf{d} \ne 0$ be a solution to the problem $\mathbf{Gd} = 0$ and compute $\gamma_1 = \max\{\gamma : \mathbf{x} + \gamma \mathbf{d} \in \overline{X}\}$. Since $\overline{X}$ is bounded and $\mathbf{x}$ is not in $\overline{E}_p$, then $0 < \gamma_1 < \infty$. Let $\mathbf{y} = \mathbf{x} + \gamma_1 \mathbf{d}$. Just as in the proof of Lemma 4.41, at $\mathbf{y}$, we now have (at least one) additional linearly independent binding hyperplane of $\overline{P}$. If there are now n binding hyperplanes, then $\mathbf{y}$ is an extreme point of $\overline{P}$. Otherwise, we may repeat this process until we identify such an extreme point. Let $\mathbf{y}_1$ be this extreme point. Clearly $\mathbf{Gy}_1 = \mathbf{g}$. Now define:

$$\gamma_2 = \max\{\gamma : \mathbf{x} + \gamma(\mathbf{x} - \mathbf{y}_1) \in \overline{P}\}$$

The value value $\mathbf{x} - \mathbf{y}_1$ is the direction from $\mathbf{y}_1$ to $\mathbf{x}$ and γ_2 may be thought of as the size of a step that one can from $\mathbf{x}$ away from $\mathbf{y}_1$ along the line from $\mathbf{y}_1$ to $\mathbf{x}$. Let

$$(4.32) \quad \mathbf{y}_2 = \mathbf{x} + \gamma_2(\mathbf{x} - \mathbf{y}_1)$$

Again $\gamma_2 < \infty$ since $\overline{P}$ is bounded and further $\gamma_2 > 0$ since:

$$(4.33) \quad \mathbf{G}\left(\mathbf{x} + \gamma_2(\mathbf{x} - \mathbf{y}_1)\right) = \mathbf{g}$$

for all $\gamma \ge 0$ (as $\mathbf{Gx} = \mathbf{Gy}_1$). As we would expect, $\mathbf{Gy}_2 = \mathbf{g}$ and there is at least one additional hyperplane binding at $\mathbf{y}_2$ (as we saw in the proof of Lemma 4.41). Trivially, $\mathbf{x}$ is a convex combination of $\mathbf{y}_1$ and $\mathbf{y}_2$. Specifically, let

$$\mathbf{x} = \delta \mathbf{y}_1 + (1 - \delta)\mathbf{y}_2$$

with $\delta \in (0, 1)$ and $\delta = \gamma_2/(1 + \gamma_2)$. This follows from Equation 4.32, by solving for $\mathbf{x}$. Now if $\mathbf{y}_2 \in \overline{E}_p$, then we have written $\mathbf{x}$ as a convex combination of extreme points of $\overline{P}$. Otherwise, we can repeat the process we used to find $\mathbf{y}_2$ in terms of $\mathbf{y}_3$ and $\mathbf{y}_4$, at which an additional hyperplane (constraint) is binding. We may repeat this process until we identify elements of $\overline{E}_P$. Reversing this process, we can ultimately write $\mathbf{x}$ as a convex combination of these extreme points. Thus we have shown that we can express $\mathbf{x}$ as a convex combination of the extreme points of $\overline{E}_P$.

Based on our deduction, we may write:

$$\mathbf{x} = \sum_{i=1}^{k+u} \delta_i \mathbf{x}_i$$

$$(4.34) \quad 1 = \sum_{i=1}^{k+u} \delta_i$$

$$\delta_i \ge 0 \quad i = 1, \ldots, k + u$$

If $\delta_i = 0$ if $i > k$, then we have succeeded in expressing $\mathbf{x}$ as a convex combination of the extreme points of P. Suppose not. Consider $\mathbf{x}_v$ with $v > k$ (so $\mathbf{x}_v$ is an extreme point of $\overline{P}$ but not P). Then it follows that $\mathbf{e}^T\mathbf{x}_v = M$ must hold (otherwise, $\mathbf{x}_v$ is an extreme point of P). Since there are n binding hyperplanes at $\mathbf{x}_v$, there must be $n-1$ hyperplanes defining P binding at $\mathbf{x}_v$. Thus, there is an edge connecting $\mathbf{x}_v$ and some extreme point of P (i.e., there is an extreme point of P that shares $n-1$ binding hyperplanes with $\mathbf{x}_v$). Denote this extreme point as $\mathbf{x}_{i(v)}$; it is adjacent to $\mathbf{x}_v$. Consider the direction $\mathbf{x}_v - \mathbf{x}_{v(i)}$. This must be a recession direction of P since there is no other hyperplane that binds in this direction before $\mathbf{e}^T\mathbf{x} = M$. Define:

$$\theta_v = \mathbf{e}^T(\mathbf{x}_v - \mathbf{x}_{v(i)})$$

and let

$$\mathbf{d} = \frac{\mathbf{x}_v - \mathbf{x}_{v(i)}}{\theta_v}$$

then $\mathbf{e}^T\mathbf{d} = 1$ as we have normalized the direction elements and therefore $\mathbf{d} \in D$. Again, let $\mathbf{G}\mathbf{y} = \mathbf{g}$ be the system of $n-1$ binding linear hyperplanes shared by $\mathbf{x}_v$ and $\mathbf{x}_{v(i)}$. Then trivially:

$$\mathbf{G}(\mathbf{x}_v - \mathbf{x}_{v(i)}) = \mathbf{G}\mathbf{d} = 0$$

and therefore, there are $n-1$ linearly independent binding hyperplanes in the system $\mathbf{A}\mathbf{d} \leq 0$, $\mathbf{d} \geq 0$. At last we see that with $\mathbf{e}^T\mathbf{d} = 1$ that $\mathbf{d}$ must be an extreme point of D and therefore an extreme direction of P. Let $\mathbf{d}_{j(v)} = \mathbf{d}$ be this extreme direction. Thus we have:

$$\mathbf{x}_v = \mathbf{x}_{i(v)} + \theta_v\mathbf{d}_{j(v)}$$

At last we can see that by substituting this into Expression 4.34 for each such v and arbitrarily letting $i(v) = j(v) = 1$ if $\delta_v = 0$ (in which case it doesn't matter), we obtain:

$$(4.35) \quad \mathbf{x} = \sum_{i=1}^{k}\delta_j\mathbf{x}_j + \sum_{v=k+1}^{k+u}\delta_v\mathbf{x}_{i(v)} + \sum_{v=k+1}^{k+u}\delta_v\theta_v\mathbf{d}_{j(v)}$$

We have now expressed $\mathbf{x}$ as desired. This completes the proof. $\qquad\square$

EXAMPLE 4.45. The Cartheodory Characterization Theorem is illustrated for a bounded and unbounded polyhedral set in Figure 4.11. This example illustrates simply how one could construct an expression for an arbitrary point $\mathbf{x}$ inside a polyhedral set in terms of extreme points and extreme directions.

$$x = \mu x_5 + (1 - \mu)(\lambda x_2 + (1 - \lambda)x_3)$$

$$x = \lambda x_2 + (1 - \lambda)x_3 + \theta d_1$$
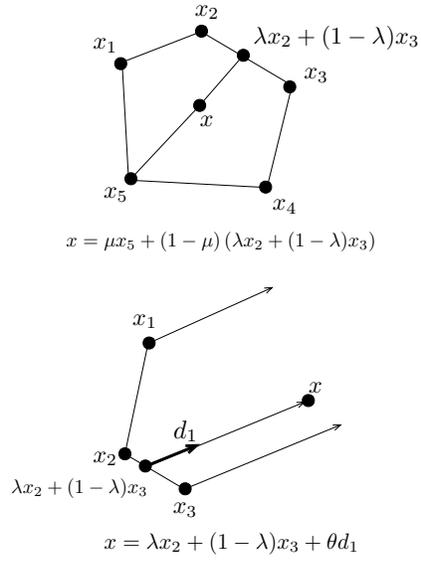
Figure 4.11. The Cartheodory Characterization Theorem: Extreme points and extreme directions are used to express points in a bounded and unbounded set.

CHAPTER 5

The Simplex Method

1. Linear Programming and Extreme Points

In this section we formalize the intuition we've obtained in all our work in two dimensional linear programming problems. Namely, we noted that if an optimal solution existed, then it occurred at an extreme point. For the remainder of this chapter, assume that $A \in \mathbb{R}^{m \times n}$ with full row rank and $b \in \mathbb{R}^m$ let

$$(5.1) \quad X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\}$$

be a polyhedral set over which we will maximize the objective function $z(x_1, \ldots, x_n) = \mathbf{c}^T \mathbf{x}$, where $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$. That is, we will focus on the linear programming problem:

$$(5.2) \quad P \begin{cases} \max \ \mathbf{c}^T \mathbf{x} \\ s.t. \ \mathbf{Ax} \leq \mathbf{b} \\ \quad \mathbf{x} \geq \mathbf{0} \end{cases}$$

THEOREM 5.1. *If Problem P has an optimal solution, then Problem P has an optimal extreme point solution.*

PROOF. Applying the Cartheodory Characterization theorem, we know that any point $\mathbf{x} \in X$ can be written as:

$$(5.3) \quad \mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{x}_i + \sum_{i=1}^{l} \mu_i \mathbf{d}_i$$

where $\mathbf{x}_1, \ldots \mathbf{x}_k$ are the extreme points of X and $\mathbf{d}_1, \ldots, \mathbf{d}_l$ are the extreme directions of X and we know that

$$(5.4) \quad \begin{aligned} &\sum_{i=1}^{k} \lambda_i = 1 \\ &\lambda_i, \mu_i \geq 0 \ \forall i \end{aligned}$$

We can rewrite problem P using this characterization as:

$$(5.5) \quad \begin{aligned} \max \ & \sum_{i=1}^{k} \lambda_i \mathbf{c}^T \mathbf{x}_i + \sum_{i=1}^{l} \mu_i \mathbf{c}^T \mathbf{d}_i \\ s.t. \ & \sum_{i=1}^{k} \lambda_i = 1 \\ & \lambda_i, \mu_i \geq 0 \ \forall i \end{aligned}$$

If there is some i such that $\mathbf{c}^T\mathbf{d}_i > 0$, then we can simply choose μ_i as large as we like, making the objective as large as we like, the problem will have no finite solution.

Therefore, assume that $\mathbf{c}^T\mathbf{d}_i \leq 0$ for all $i = 1, \ldots, l$ (in which case, we may simply choose $\mu_i = 0$, for all i). Since the set of extreme points $\mathbf{x}_1, \ldots \mathbf{x}_k$ is finite, we can simply set $\lambda_p = 1$ if $\mathbf{c}^T\mathbf{x}_p$ has the largest value among all possible values of $\mathbf{c}^T\mathbf{x}_i$, $i = 1, \ldots, k$. This is clearly the solution to the linear programming problem. Since $\mathbf{x}_p$ is an extreme point, we have shown that if P has a solution, it must have an extreme point solution. $\square$

COROLLARY 5.2. *Problem P has a finite solution if and only if $\mathbf{c}^T\mathbf{d}_i \leq 0$ for all $i = 1, \ldots l$ when $\mathbf{d}_1, \ldots, \mathbf{d}_l$ are the extreme directions of X.*

PROOF. This is implicit in the proof of the theorem. $\square$

COROLLARY 5.3. *Problem P has alternative optimal solutions if there are at least two extreme points $\mathbf{x}_p$ and $\mathbf{x}_q$ so that $\mathbf{c}^T\mathbf{x}_p = \mathbf{c}^T\mathbf{x}_q$ and so that $\mathbf{x}_p$ is the extreme point solution to the linear programming problem.*

PROOF. Suppose that $\mathbf{x}_p$ is the extreme point solution to P identified in the proof of the theorem. Suppose $\mathbf{x}_q$ is another extreme point solution with $\mathbf{c}^T\mathbf{x}_p = \mathbf{c}^T\mathbf{x}_q$. Then every convex combination of $\mathbf{x}_p$ and $\mathbf{x}_q$ is contained in X (since X is convex). Thus every $\mathbf{x}$ with form $\lambda\mathbf{x}_p + (1 - \lambda)\mathbf{x}_q$ and $\lambda \in [0, 1]$ has objective function value:

$$\lambda\mathbf{c}^T\mathbf{x}_p + (1 - \lambda)\mathbf{c}^T\mathbf{x}_q = \lambda\mathbf{c}^T\mathbf{x}_p + (1 - \lambda)\mathbf{c}^T\mathbf{x}_p = \mathbf{c}^T\mathbf{x}_p$$

which is the optimal objective function value, by assumption. $\square$

EXERCISE 48. Let $X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\}$ and suppose that $\mathbf{d}_1, \ldots \mathbf{d}_l$ are the extreme directions of X (assuming it has any). Show that the problem:

$$\begin{aligned}
& \min \ \mathbf{c}^T\mathbf{x} \\
(5.6) \quad & s.t. \ \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
& \quad \mathbf{x} \geq \mathbf{0}
\end{aligned}$$

has a finite optimal solution if (and only if) $\mathbf{c}^T\mathbf{d}_j \geq 0$ for $k = 1, \ldots, l$. [Hint: Modify the proof above using the Cartheodory characterization theorem.]

2. Algorithmic Characterization of Extreme Points

In the previous sections we showed that if a linear programming problem has a solution, then it must have an extreme point solution. The challenge now is to identify some simple way of identifying extreme points. To accomplish this, let us now assume that we write X as:

$$(5.7) \quad X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\}$$

Our work in the previous sections shows that this is possible. Recall we can separate $\mathbf{A}$ into an $m \times m$ matrix B and an $m \times (n - m)$ matrix N and we have the result:

$$(5.8) \quad \mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x_N}$$

We know that $\mathbf{B}$ is invertible since we assumed that $\mathbf{A}$ had full row rank. If we assume that $\mathbf{x_N} = \mathbf{0}$, then the solution

$$(5.9) \quad \mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b}$$

was called a *basic solution* (See Definition 3.48.) Clearly any basic solution satisfies the constraints $\mathbf{Ax} = \mathbf{b}$ but it may not satisfy the constraints $\mathbf{x} \geq \mathbf{0}$.

DEFINITION 5.4 (Basic Feasible Solution). If $\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b}$ and $\mathbf{x_N} = \mathbf{0}$ is a basic solution to $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x_B} \geq 0$, then the solution $(\mathbf{x_B}, \mathbf{x_N})$ is called *basic feasible solution.*

THEOREM 5.5. *Every basic feasible solution is an extreme point of X. Likewise, every extreme point is characterized by a basic feasible solution of $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$.*

PROOF. Since $\mathbf{Ax} = \mathbf{Bx_B} + \mathbf{Nx_N} = \mathbf{b}$ this represents the intersection of m linearly independent hyperplanes (since the rank of $\mathbf{A}$ is m). The fact that $\mathbf{x_N} = \mathbf{0}$ and $\mathbf{x_N}$ contains $n - m$ variables, then we have $n - m$ binding, linearly independent hyperplanes in $\mathbf{x_N} \geq 0$. Thus the point $(\mathbf{x_B}, \mathbf{x_N})$ is the intersection of $m + (n - m) = n$ linearly independent hyperplanes. By Theorem 4.31 we know that $(\mathbf{x_B}, \mathbf{x_N})$ must be an extreme point of X.

Conversely, let $\mathbf{x}$ be an extreme point of X. Clearly $\mathbf{x}$ is feasible and by Theorem 4.31 it must represent the intersection of n hyperplanes. The fact that $\mathbf{x}$ is feasible implies that $\mathbf{Ax} = \mathbf{b}$. This accounts for m of the intersecting linearly independent hyperplanes. The remaining $n - m$ hyperplanes must come from $\mathbf{x} \geq \mathbf{0}$. That is, $n - m$ variables are zero. Let $\mathbf{x_N} = \mathbf{0}$ be the variables for which $\mathbf{x} \geq \mathbf{0}$ are binding. Denote the remaining variables $\mathbf{x_B}$. We can see that $\mathbf{A} = [\mathbf{B}|\mathbf{N}]$ and that $\mathbf{Ax} = \mathbf{Bx_B} + \mathbf{Nx_N} = \mathbf{b}$. Clearly, $\mathbf{x_B}$ is the unique solution to $\mathbf{Bx_B} = \mathbf{b}$ and thus $(\mathbf{x_B}, \mathbf{x_N})$ is a basic feasible solution. $\square$

3. The Simplex Algorithm–Algebraic Form

In this section, we will develop the simplex algorithm algebraically. The idea behind the simplex algorithm is as follows:

(1) Convert the linear program to standard form.
(2) Obtain an initial basic feasible solution (if possible).
(3) Determine whether the basic feasible solution is optimal. If yes, stop.
(4) If the current basic feasible solution is not optimal, then determine which non-basic variable (zero valued variable) should become basic (become non-zero) and which basic variable (non-zero valued variable) should become non-basic (go to zero) to make the objective function value better.
(5) Determine whether the problem is unbounded. If yes, stop.
(6) If the problem doesn't seem to be unbounded at this stage, find a new basic feasible solution from the old basic feasible solution. Go back to Step 3.

Suppose we have a basic feasible solution $\mathbf{x} = (\mathbf{x_B}, \mathbf{x_N})$. We can divide the cost vector $\mathbf{c}$ into its basic and non-basic parts, so we have $\mathbf{c} = [\mathbf{c_B}|\mathbf{c_N}]^T$. Then the objective function becomes:

$$(5.10) \quad \mathbf{c}^T\mathbf{x} = \mathbf{c_B}^T\mathbf{x_B} + \mathbf{c_N}^T\mathbf{x_N}$$

We can substitute Equation 5.8 into Equation 5.10 to obtain:

$$(5.11) \quad \mathbf{c}^T\mathbf{x} = \mathbf{c_B}^T\left(\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{Nx_N}\right) + \mathbf{c_N}\mathbf{x_N} = \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{b} + \left(\mathbf{c_N}^T - \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{N}\right)\mathbf{x_N}$$

Let $\mathcal{J}$ be the set of indices of non-basic variables. Then we can write Equation 5.11 as:

$$(5.12) \quad z(x_1, \ldots, x_n) = \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{b} + \sum_{j \in \mathcal{J}} \left(\mathbf{c}_j - \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{A}_{\cdot j}\right) x_j$$

Consider now the fact $x_j = 0$ for all $j \in \mathcal{J}$. Further, we can see that:

$$(5.13) \quad \frac{\partial z}{\partial x_j} = \mathbf{c}_j - \mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1} \mathbf{A}_{\cdot j}$$

This means that if $\mathbf{c}_j - \mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1} \mathbf{A}_{\cdot j} > 0$ and we *increase* x_j from zero to some new value, then we will *increase* the value of the objective function. For historic reasons, we actually consider the value $\mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1} \mathbf{A}_{\cdot j} - \mathbf{c}_j$, called the *reduced cost* and denote it as:

$$(5.14) \quad -\frac{\partial z}{\partial x_j} = z_j - c_j = \mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1} \mathbf{A}_{\cdot j} - \mathbf{c}_j$$

In a maximization problem, we chose non-basic variables x_j with negative reduced cost to become basic because, in this case, $\partial z / \partial x_j$ is *positive*.

Assume we chose x_j, a non-basic variable to become non-zero (because $z_j - c_j < 0$). We wish to know which of the basic variables will become zero as we *increase* x_j away from zero. We must also be very careful that *none* of the variables become negative as we do this.

By Equation 5.8 we know that the only current basic variables will be affected by increasing x_j. Let us focus explicitly on Equation 5.8 where we include only variable x_j (since all other non-basic variables are kept zero). Then we have:

$$(5.15) \quad \mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{A}_{\cdot j} x_j$$

Let $\overline{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$ be an $m \times 1$ column vector and let that $\overline{\mathbf{a}}_j = \mathbf{B}^{-1}\mathbf{A}_{\cdot j}$ be another $m \times 1$ column. Then we can write:

$$(5.16) \quad \mathbf{x_B} = \overline{\mathbf{b}} - \overline{\mathbf{a}}_j x_j$$

Let $\overline{\mathbf{b}} = [\overline{b}_1, \ldots \overline{b}_m]^T$ and $\overline{\mathbf{a}}_j = [\overline{a}_{j_1}, \ldots, \overline{a}_{j_m}]$, then we have:

$$(5.17) \quad \begin{bmatrix} x_{B_1} \\ x_{B_2} \\ \vdots \\ x_{B_m} \end{bmatrix} = \begin{bmatrix} \overline{b}_1 \\ \overline{b}_2 \\ \vdots \\ \overline{b}_m \end{bmatrix} - \begin{bmatrix} \overline{a}_{j_1} \\ \overline{a}_{j_2} \\ \vdots \\ \overline{a}_{j_m} \end{bmatrix} x_j = \begin{bmatrix} \overline{b}_1 - \overline{a}_{j_1} x_j \\ \overline{b}_2 - \overline{a}_{j_2} x_j \\ \vdots \\ \overline{b}_m - \overline{a}_{j_m} x_j \end{bmatrix}$$

We know (a priori) that $\overline{b}_i \geq 0$ for $i = 1, \ldots, m$. If $\overline{a}_{j_i} \leq 0$, then as we increase x_j, $\overline{b}_i - \overline{a}_{j_i} \geq 0$ no matter how large we make x_j. On the other hand, if $\overline{a}_{j_i} > 0$, then as we increase x_j we know that $\overline{b}_i - \overline{a}_{j_i} x_j$ will get smaller and eventually hit zero. In order to ensure that *all* variables remain non-negative, we cannot increase x_j beyond a certain point.

For each i ($i = 1, \ldots, m$) such that $\overline{a}_{j_i} > 0$, the value of x_j that will make x_{B_i} goto 0 can be found by observing that:

$$(5.18) \quad x_{B_i} = \overline{b}_i - \overline{a}_{j_i} x_j$$

and if $x_{B_i} = 0$, then we can solve:

$$(5.19) \quad 0 = \overline{b}_i - \overline{a}_{j_i} x_j \implies x_j = \frac{\overline{b}_i}{\overline{a}_{j_i}}$$

Thus, the *largest possible value* we can assign x_j and ensure that all variables remain positive is:

$$(5.20) \quad \min \left\{ \frac{\overline{b}_i}{\overline{a}_{j_i}} : i = 1, \ldots, m \text{ and } a_{j_i} > 0 \right\}$$

Expression 5.20 is called the *minimum ratio test*. We are interested in which index i is the minimum ratio.

Suppose that in executing the minimum ratio test, we find that $x_j = \bar{b}_k/\bar{a}_{j_k}$. The variable x_j (which was non-basic) becomes basic and the variable $x_{\mathbf{B}_k}$ becomes non-basic. All other basic variables remain basic (and positive). In executing this procedure (of exchanging one basic variable and one non-basic variable) we have moved from one extreme point of X to another.

THEOREM 5.6. *If $z_j - c_j \geq 0$ for all $j \in \mathcal{J}$, then the current basic feasible solution is optimal.*

PROOF. We have already shown in Theorem 5.1 that if a linear programming problem has an optimal solution, then it occurs at an extreme point and we've shown in Theorem 5.5 that there is a one-to-one correspondence between extreme points and basic feasible solutions. If $z_j - c_j \geq 0$ for all $j \in \mathcal{J}$, then $\partial z/\partial x_j \leq 0$ for all non-basic variables x_j. That is, we cannot increase the value of the objective function by increasing the value of any non-basic variable. Thus, since moving to another basic feasible solution (extreme point) will not improve the objective function, it follows we must be at the optimal solution. $\square$

THEOREM 5.7. *In a maximization problem, if $\bar{a}_{j_i} \leq 0$ for all $i = 1, \ldots, m$, and $z_j - c_j < 0$, then the linear programming problem is unbounded.*

PROOF. The fact that $z_j - c_j < 0$ implies that increasing x_j will improve the value of the objective function. Since $\bar{a}_{j_i} < 0$ for all $i = 1, \ldots, m$, we can increase x_j indefinitely without violating feasibility (no basic variable will ever go to zero). Thus the objective function can be made as large as we like. $\square$

REMARK 5.8. We should note that in executing the exchange of one basic variable and one non-basic variable, we must be *very* careful to ensure that the resulting basis consist of m linearly independent columns of the original matrix $\mathbf{A}$. The conditions for this are provided in Lemma 3.39. Specifically, we must be able to write the column corresponding to x_j, the entering variable, as a linear combination of the columns of $\mathbf{B}$ so that:

$$(5.21) \quad \alpha_1 \mathbf{b}_1 + \ldots \alpha_m \mathbf{b}_m = \mathbf{A}_{\cdot j}$$

and further if we are exchanging x_j for $\mathbf{x}_{\mathbf{B}_i}$ $(i = 1, \ldots, m)$, then $\alpha_i \neq 0$.

We can see this from the fact that $\bar{\mathbf{a}}_j = \mathbf{B}^{-1}\mathbf{A}_{\cdot j}$ and therefore:

$$\mathbf{B}\bar{\mathbf{a}}_j = \mathbf{A}_{\cdot j}$$

and therefore we have:

$$\mathbf{A}_{\cdot j} = \mathbf{B}_{\cdot 1}\bar{\mathbf{a}}_{j_1} + \cdots + \mathbf{B}_{\cdot m}\bar{\mathbf{a}}_{j_m}$$

which shows how to write the column $\mathbf{A}_{\cdot j}$ as a linear combination of the columns of $\mathbf{B}$.

EXERCISE 49. Consider the linear programming problem given in Exercise 48. Under what conditions should a non-basic variable enter the basis? State and prove an analogous theorem to Theorem 5.6 using your observation. [Hint: Use the definition of reduced cost. Remember that it is $-\partial z/\partial x_j$.]

EXAMPLE 5.9. Consider the Toy Maker Problem (from Example 2.3). The linear programming problem given in Equation 2.8 is:

$$\begin{cases} \max \ z(x_1, x_2) = 7x_1 + 6x_2 \\ \ \ \text{s.t.} \ \ 3x_1 + x_2 \leq 120 \\ \ \ \ \ \ \ \ \ \ x_1 + 2x_2 \leq 160 \\ \ \ \ \ \ \ \ \ \ x_1 \leq 35 \\ \ \ \ \ \ \ \ \ \ x_1 \geq 0 \\ \ \ \ \ \ \ \ \ \ x_2 \geq 0 \end{cases}$$

We can convert this problem to standard form by introducing the slack variables s_1, s_2 and s_3:

$$\begin{cases} \max \ z(x_1, x_2) = 7x_1 + 6x_2 \\ \ \ \text{s.t.} \ \ 3x_1 + x_2 + s_1 = 120 \\ \ \ \ \ \ \ \ \ \ x_1 + 2x_2 + s_2 = 160 \\ \ \ \ \ \ \ \ \ \ x_1 + s_3 = 35 \\ \ \ \ \ \ \ x_1, x_2, s_1, s_2, s_3 \geq 0 \end{cases}$$

which yields the matrices

$$\mathbf{c} = \begin{bmatrix} 7 \\ 6 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 3 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 120 \\ 160 \\ 35 \end{bmatrix}$$

We can begin with the matrices:

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 1 & 0 \end{bmatrix}$$

In this case we have:

$$\mathbf{x_B} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \quad \mathbf{x_N} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \mathbf{c_B} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{c_N} = \begin{bmatrix} 7 \\ 6 \end{bmatrix}$$

and

$$\mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 120 \\ 160 \\ 35 \end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 1 & 0 \end{bmatrix}$$

Therefore:

$$\mathbf{c_B^T}\mathbf{B}^{-1}\mathbf{b} = 0 \quad \mathbf{c_B^T}\mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} 0 & 0 \end{bmatrix} \quad \mathbf{c_B^T}\mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N} = \begin{bmatrix} -7 & -6 \end{bmatrix}$$

Using this information, we can compute:

$$\mathbf{c_B^T B^{-1} A_{.1}} - \mathbf{c_1} = -7$$
$$\mathbf{c_B^T B^{-1} A_{.2}} - \mathbf{c_2} = -6$$

and therefore:

$$\frac{\partial z}{\partial x_1} = 7 \text{ and } \frac{\partial z}{\partial x_2} = 6$$

Based on this information, we could chose either x_1 or x_2 to enter the basis and the value of the objective function would increase. If we chose x_1 to enter the basis, then we must determine which variable will leave the basis. To do this, we must investigate the elements of $\mathbf{B^{-1} A_{.1}}$ and the current basic feasible solution $\mathbf{B^{-1} b}$. Since each element of $\mathbf{B^{-1} A_{.1}}$ is positive, we must perform the minimum ratio test on each element of $\mathbf{B^{-1} A_{.1}}$. We know that $\mathbf{B^{-1} A_{.1}}$ is just the first column of $\mathbf{B^{-1} N}$ which is:

$$\mathbf{B^{-1} A_{.1}} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

Performing the minimum ratio test, we see have:

$$\min \left\{ \frac{120}{3}, \frac{160}{1}, \frac{35}{1} \right\}$$

In this case, we see that index 3 (35/1) is the minimum ratio. Therefore, variable x_1 will enter the basis and variable s_3 will leave the basis. The new basic and non-basic variables will be:

$$\mathbf{x_B} = \begin{bmatrix} s_1 \\ s_2 \\ x_1 \end{bmatrix} \quad \mathbf{x_N} = \begin{bmatrix} s_3 \\ x_2 \end{bmatrix} \quad \mathbf{c_B} = \begin{bmatrix} 0 \\ 0 \\ 7 \end{bmatrix} \quad \mathbf{c_N} = \begin{bmatrix} 0 \\ 6 \end{bmatrix}$$

and the matrices become:

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 0 & 1 \\ 0 & 2 \\ 1 & 0 \end{bmatrix}$$

Note we have simply swapped the column corresponding to x_1 with the column corresponding to s_3 in the basis matrix $\mathbf{B}$ and the non-basic matrix $\mathbf{N}$. We will do this repeatedly in the example and we recommend the reader keep track of which variables are being exchanged and why certain columns in $\mathbf{B}$ are being swapped with those in $\mathbf{N}$.

Using the new $\mathbf{B}$ and $\mathbf{N}$ matrices, the derived matrices are then:

$$\mathbf{B^{-1} b} = \begin{bmatrix} 15 \\ 125 \\ 35 \end{bmatrix} \quad \mathbf{B^{-1} N} = \begin{bmatrix} -3 & 1 \\ -1 & 2 \\ 1 & 0 \end{bmatrix}$$

The cost information becomes:

$$\mathbf{c_B^T B^{-1} b} = 245 \quad \mathbf{c_B^T B^{-1} N} = \begin{bmatrix} 7 & 0 \end{bmatrix} \quad \mathbf{c_B^T B^{-1} N} - \mathbf{c_N} = \begin{bmatrix} 7 & -6 \end{bmatrix}$$

using this information, we can compute:

$$\mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1} \mathbf{A}_{.5} - \mathbf{c}_5 = 7$$
$$\mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1} \mathbf{A}_{.2} - \mathbf{c}_2 = -6$$

Based on this information, we can only choose x_2 to enter the basis to ensure that the value of the objective function increases. We can perform the minimum ration test to figure out which basic variable will leave the basis. We know that $\mathbf{B}^{-1}\mathbf{A}_{.2}$ is just the second column of $\mathbf{B}^{-1}\mathbf{N}$ which is:

$$\mathbf{B}^{-1}\mathbf{A}_{.2} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

Performing the minimum ratio test, we see have:

$$\min\left\{ \frac{15}{1}, \frac{125}{2} \right\}$$

In this case, we see that index 1 (15/1) is the minimum ratio. Therefore, variable x_2 will enter the basis and variable s_1 will leave the basis. The new basic and non-basic variables will be: The new basic and non-basic variables will be:

$$\mathbf{x_B} = \begin{bmatrix} x_2 \\ s_2 \\ x_1 \end{bmatrix} \quad \mathbf{x_N} = \begin{bmatrix} s_3 \\ s_1 \end{bmatrix} \quad \mathbf{c_B} = \begin{bmatrix} 6 \\ 0 \\ 7 \end{bmatrix} \quad \mathbf{c_N} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and the matrices become:

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 3 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

The derived matrices are then:

$$\mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 15 \\ 95 \\ 35 \end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} -3 & 1 \\ 5 & -2 \\ 1 & 0 \end{bmatrix}$$

The cost information becomes:

$$\mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1}\mathbf{b} = 335 \quad \mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} -11 & 6 \end{bmatrix} \quad \mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N} = \begin{bmatrix} -11 & 6 \end{bmatrix}$$

Based on this information, we can only choose s_3 to (re-enter) the basis to ensure that the value of the objective function increases. We can perform the minimum ration test to figure out which basic variable will leave the basis. We know that $\mathbf{B}^{-1}\mathbf{A}_{.5}$ is just the fifth column of $\mathbf{B}^{-1}\mathbf{N}$ which is:

$$\mathbf{B}^{-1}\mathbf{A}_{.5} = \begin{bmatrix} -3 \\ 5 \\ 1 \end{bmatrix}$$

Performing the minimum ratio test, we see have:

$$\min\left\{ \frac{95}{5}, \frac{35}{1} \right\}$$

In this case, we see that index 2 (95/5) is the minimum ratio. Therefore, variable s_3 will enter the basis and variable s_2 will leave the basis. The new basic and non-basic variables will be:

$$\mathbf{x_B} = \begin{bmatrix} x_2 \\ s_3 \\ x_1 \end{bmatrix} \quad \mathbf{x_N} = \begin{bmatrix} s_2 \\ s_1 \end{bmatrix} \quad \mathbf{c_B} = \begin{bmatrix} 6 \\ 0 \\ 7 \end{bmatrix} \quad \mathbf{c_N} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and the matrices become:

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 3 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$$

The derived matrices are then:

$$\mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 72 \\ 19 \\ 16 \end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} 6/10 & -1/5 \\ 1/5 & -2/5 \\ -1/5 & 2/5 \end{bmatrix}$$

The cost information becomes:

$$\mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{b} = 544 \quad \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} 11/5 & 8/5 \end{bmatrix} \quad \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N} = \begin{bmatrix} 11/5 & 8/5 \end{bmatrix}$$

Since the reduced costs are now positive, we can conclude that we've obtained an optimal solution because no improvement is possible. The final solution then is:

$$\mathbf{x_B}^* = \begin{bmatrix} x_2 \\ s_3 \\ x_1 \end{bmatrix} = \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 72 \\ 19 \\ 16 \end{bmatrix}$$

Simply, we have $x_1 = 16$ and $x_2 = 72$ as we obtained in Example 2.3. The path of extreme points we actually took in traversing the boundary of the polyhedral feasible region is shown in Figure 5.1.

EXERCISE 50. Assume that a leather company manufactures two types of belts: regular and deluxe. Each belt requires 1 square yard of leather. A regular belt requires 1 hour of skilled labor to produce, while a deluxe belt requires 2 hours of labor. The leather company receives 40 square yards of leather each week and a total of 60 hours of skilled labor is available. Each regular belt nets $3 in profit, while each deluxe belt nets $5 in profit. The company wishes to maximize profit.

(1) Ignoring the divisibility issues, construct a linear programming problem whose solution will determine the number of each type of belt the company should produce.
(2) Use the simplex algorithm to solve the problem you stated above remembering to convert the problem to *standard form* before you begin.
(3) Draw the feasible region and the level curves of the objective function. Verify that the optimal solution you obtained through the simplex method is the point at which the level curves no longer intersect the feasible region in the direction following the gradient of the objective function.
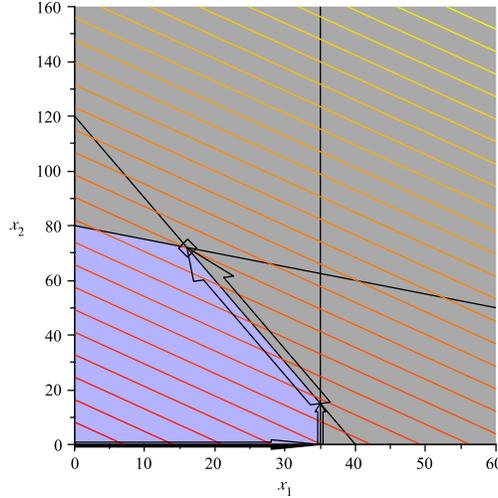
Figure 5.1. The Simplex Algorithm: The path around the feasible region is shown in the figure. Each exchange of a basic and non-basic variable moves us along an edge of the polygon in a direction that increases the value of the objective function.

4. Simplex Method–Tableau Form

No one executes the simplex algorithm in algebraic form. Instead, several representations (tableau representations) have been developed to lesson the amount of writing that needs to be done and to collect all pertinent information into a single table.

To see how a *Simplex Tableau* is derived, consider Problem P in standard form:

$$P \begin{cases} \max \ \mathbf{c}^T\mathbf{x} \\ s.t. \ \mathbf{A}\mathbf{x} = \mathbf{b} \\ \quad \mathbf{x} \geq \mathbf{0} \end{cases}$$

We can re-write P in an unusual way by introducing a new variable z and separating $\mathbf{A}$ into its basic and non-basic parts to obtain:

$$(5.22) \quad \begin{aligned} \max \ & z \\ s.t. \ & z - \mathbf{c_B}^T\mathbf{x_B} - \mathbf{c_N}^T\mathbf{x_N} = 0 \\ & \mathbf{B}\mathbf{x_B} + \mathbf{N}\mathbf{x_N} = \mathbf{b} \\ & \mathbf{x_B}, \mathbf{x_N} \geq 0 \end{aligned}$$

From the second equation, it's clear

$$(5.23) \quad \mathbf{x_B} + \mathbf{B}^{-1}\mathbf{N}\mathbf{x_N} = \mathbf{B}^{-1}\mathbf{b}$$

We can multiply this equation by $\mathbf{c_B^T}$ to obtain:

$$(5.24) \quad \mathbf{c_B}^T\mathbf{x_B} + \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{N}\mathbf{x_N} = \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{b}$$

If we add this equation to the equation $z - \mathbf{c_B}^T\mathbf{x_B} - \mathbf{c_N}^T\mathbf{x_N} = 0$ we obtain:

$$(5.25) \quad z + \mathbf{0}^T\mathbf{x_B} + \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{N}\mathbf{x_N} - \mathbf{c_N}^T\mathbf{x_N} = \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{b}$$

Here $\mathbf{0}$ is the vector of zeros of appropriate size. This equation can be written as:

$$(5.26) \quad z + \mathbf{0}^T \mathbf{x_B} + \left(\mathbf{c_B^T B^{-1} N} - \mathbf{c_N^T} \right) \mathbf{x_N} = \mathbf{c_B^T B^{-1} b}$$

We can now represent this set of equations as a large matrix (or tableau):

	z	$\mathbf{x_B}$	$\mathbf{x_N}$	RHS	
z	1	$\mathbf{0}$	$\mathbf{c_B^T B^{-1} N} - \mathbf{c_N^T}$	$\mathbf{c_B^T B^{-1} b}$	Row 0
$\mathbf{x_B}$	$\mathbf{0}$	1	$\mathbf{B^{-1} N}$	$\mathbf{B^{-1} b}$	Rows 1 through m

The augmented matrix shown within the table:

$$(5.27) \quad \begin{bmatrix} 1 & \mathbf{0} & \mathbf{c_B^T B^{-1} N} - \mathbf{c_N^T} & \mathbf{c_B^T B^{-1} b} \\ \mathbf{0} & 1 & \mathbf{B^{-1} N} & \mathbf{B^{-1} b} \end{bmatrix}$$

is simply the matrix representation of the simultaneous equations described by Equations 5.23 and 5.26. We can see that the first row consists of a row of the first row of the $(m+1) \times (m+1)$ identity matrix, the reduced costs of the non-basic variables and the current objective function values. The remainder of the rows consist of the rest of the $(m+1) \times (m+1)$ identity matrix, the matrix $\mathbf{B^{-1} N}$ and $\mathbf{B^{-1} b}$ the current non-zero part of the basic feasible solution.

This matrix representation (or tableau representation) contains all of the information we need to execute the simplex algorithm. An entering variable is chosen from among the columns containing the reduced costs and matrix $\mathbf{B^{-1} N}$. Naturally, a column with a negative reduced cost is chosen. We then chose a leaving variable by performing the minimum ratio test on the chosen column and the right-hand-side (RHS) column. We pivot on the element at the entering column and leaving row and this transforms the tableau into a new tableau that represents the new basic feasible solution.

EXAMPLE 5.10. Again, consider the toy maker problem. We will execute the simplex algorithm using the tableau method. Our problem in standard form is given as:

$$\begin{cases} \max & z(x_1, x_2) = 7x_1 + 6x_2 \\ \text{s.t.} & 3x_1 + x_2 + s_1 = 120 \\ & x_1 + 2x_2 + s_2 = 160 \\ & x_1 + s_3 = 35 \\ & x_1, x_2, s_1, s_2, s_3 \geq 0 \end{cases}$$

We can assume our initial basic feasible solution has s_1, s_2 and s_3 as basic variables and x_1 and x_2 as non-basic variables. Thus our initial tableau is simply:

$$(5.28) \quad \begin{array}{c} \\ z \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{bmatrix} z & x_1 & x_2 & s_1 & s_2 & s_3 & \text{RHS} \\ 1 & -7 & -6 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 1 & 0 & 0 & 120 \\ 0 & 1 & 2 & 0 & 1 & 0 & 160 \\ 0 & 1 & 0 & 0 & 0 & 1 & 35 \end{bmatrix}$$

Note that the columns have been swapped so that the identity matrix is divided and $\mathbf{B^{-1} N}$ is located in columns 2 and 3. This is because of our choice of basic variables. The reduced cost vector is in Row 0.

Using this information, we can see that either x_1 or x_2 can enter. We can compute the minimum ratio test (MRT) next to the RHS column. If we chose x_2 as the entering variable, then the MRT tells us s_2 will leave. We put a box around the element on which we will pivot:

$$(5.29) \quad \begin{array}{c|c|cccccc|c} & z & x_1 & x_2 & s_1 & s_2 & s_3 & \text{RHS} \\ \hline z & 1 & -7 & -6 & 0 & 0 & 0 & 0 \\ s_1 & 0 & 3 & 1 & 1 & 0 & 0 & 120 \\ s_2 & 0 & 1 & \boxed{2} & 0 & 1 & 0 & 160 \\ s_3 & 0 & 1 & 0 & 0 & 0 & 1 & 35 \end{array} \qquad \begin{array}{c} \underline{\text{MRT } (x_2)} \\ \\ 120 \\ 80 \\ - \end{array}$$

If we pivot on this element, then we transform the column corresponding to x_2 into the identity column:

$$(5.30) \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

This process will correctly compute the new reduced costs and $\mathbf{B}^{-1}$ matrix as well as the new cost information. The new tableau becomes:

$$(5.31) \quad \begin{array}{c|c|ccccc|c} & z & x_1 & x_2 & s_1 & s_2 & s_3 & \text{RHS} \\ \hline z & 1 & -4 & 0 & 0 & 3 & 0 & 480 \\ s_1 & 0 & 2.5 & 0 & 1 & -0.5 & 0 & 40 \\ x_2 & 0 & 0.5 & 1 & 0 & 0.5 & 0 & 80 \\ s_3 & 0 & 1 & 0 & 0 & 0 & 1 & 35 \end{array}$$

We can see that x_1 is a valid entering variable, as it has a negative reduced cost (-4). We can again place the minimum ratio test values on the right-hand-side of the matrix to obtain:

$$(5.32) \quad \begin{array}{c|c|ccccc|c} & z & x_1 & x_2 & s_1 & s_2 & s_3 & \text{RHS} \\ \hline z & 1 & -4 & 0 & 0 & 3 & 0 & 480 \\ s_1 & 0 & \boxed{2.5} & 0 & 1 & -0.5 & 0 & 40 \\ x_2 & 0 & 0.5 & 1 & 0 & 0.5 & 0 & 80 \\ s_3 & 0 & 1 & 0 & 0 & 0 & 1 & 35 \end{array} \qquad \begin{array}{c} \underline{\text{MRT } (x_1)} \\ \\ 16 \\ 160 \\ 35 \end{array}$$

We now pivot on the element we have boxed to obtain the new tableau[1]:

$$(5.33) \quad \begin{array}{c|c|ccccc|c} & z & x_1 & x_2 & s_1 & s_2 & s_3 & \text{RHS} \\ \hline z & 1 & 0 & 0 & 1.6 & 2.2 & 0 & 544 \\ x_1 & 0 & 1 & 0 & 0.4 & -0.2 & 0 & 16 \\ x_2 & 0 & 0 & 1 & -0.2 & 0.6 & 0 & 72 \\ s_3 & 0 & 0 & 0 & -0.4 & 0.2 & 1 & 19 \end{array}$$

All the reduced costs of the non-basic variables (s_1 and s_2) are positive and so this is the optimal solution to the linear programming problem. We can also see that this solution agrees with our previous computations on the Toy Maker Problem.

---

[1]Thanks to Ethan Wright for catching a typo here.

5. Identifying Unboundedness

We have already identified a theorem for detecting unboundedness. Recall Theorem 5.7: *In a maximization problem, if $\bar{a}_{j_i} < 0$ for all $i = 1, \ldots, m$, and $z_j - c_j < 0$, then the linear programming problem is unbounded.*

This condition occurs when a variable x_j should enter the basis because $\partial z / \partial x_j > 0$ and there is no blocking basis variable. That is, we can arbitrarily increase the value of x_j without causing any variable to become negative. We give an example:

EXAMPLE 5.11. Consider the Linear programming problem from Example 2.9:

$$\begin{cases} \max \ z(x_1, x_2) = 2x_1 - x_2 \\ \text{s.t. } x_1 - x_2 \leq 1 \\ \qquad 2x_1 + x_2 \geq 6 \\ \qquad x_1, x_2 \geq 0 \end{cases}$$

We can convert this problem into standard form by adding a slack variable s_1 and a surplus variable s_2:

$$\begin{cases} \max \ z(x_1, x_2) = 2x_1 - x_2 \\ \text{s.t. } x_1 - x_2 + s_1 = 1 \\ \qquad 2x_1 + x_2 - s_2 = 6 \\ \qquad x_1, x_2, s_1, s_2 \geq 0 \end{cases}$$

This yields the matrices:

$$\mathbf{c} = \begin{bmatrix} 2 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & -1 & 1 & 0 \\ 2 & 1 & 0 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 6 \end{bmatrix}$$

We have both slack and surplus variables, so the case when $x_1 = x_2 = 0$ is not a valid initial solution. We can chose a valid solution based on our knowledge of the problem. Assume that $s_1 = s_2 = 0$ and so we have:

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

In this case we have:

$$\mathbf{x_B} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \mathbf{x_N} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad \mathbf{c_B} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \quad \mathbf{c_N} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This yields:

$$\mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 7/3 \\ 4/3 \end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} 1/3 & -1/3 \\ -2/3 & -1/3 \end{bmatrix}$$

We also have the cost information:

$$\mathbf{c_B}\mathbf{B}^{-1}\mathbf{b} = \frac{10}{3} \quad \mathbf{c_B}\mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} \frac{4}{3} & -\frac{1}{3} \end{bmatrix} \quad \mathbf{c_B}\mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N} = \begin{bmatrix} \frac{4}{3} & -\frac{1}{3} \end{bmatrix}$$

Based on this information, we can construct the tableau for this problem as:

$$(5.34) \qquad \begin{array}{c} z \\ x_1 \\ x_2 \end{array} \left[\begin{array}{c|ccccc} z & x_1 & x_2 & s_1 & s_2 & \text{RHS} \\ \hline 1 & 0 & 0 & \frac{4}{3} & \frac{-1}{3} & \frac{10}{3} \\ \hline 0 & 1 & 0 & \frac{1}{3} & \frac{-1}{3} & \frac{7}{3} \\ 0 & 0 & 1 & \frac{-2}{3} & \frac{-1}{3} & \frac{4}{3} \end{array} \right]$$

We see that s_2 should enter the basis because $\mathbf{c_B B}^{-1}\mathbf{A}_{.4} - \mathbf{c}_4 < 0$. But the column corresponding to s_2 in the tabluau is all negative. Therefore there is no minimum ratio test. We can let s_2 become as large as we like and we will keep increasing the objective function without violating feasibility.

What we have shown is that the ray with vertex

$$\mathbf{x}_0 = \begin{bmatrix} 7/3 \\ 4/3 \\ 0 \\ 0 \end{bmatrix}$$

and direction:

$$\mathbf{d} = \begin{bmatrix} 1/3 \\ 1/3 \\ 0 \\ 1 \end{bmatrix}$$

is entirely contained inside the polyhedral set defined by $\mathbf{Ax} = \mathbf{b}$. This can be see from the fact that:

$$\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{Nx_N}$$

When applied in this case, we have:

$$\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{A}_{.4}s_2$$

We know that

$$-\mathbf{B}^{-1}\mathbf{A}_{.4} = \begin{bmatrix} 1/3 \\ 1/3 \end{bmatrix}$$

We will be increasing s_2 (which acts like λ in the definition of ray) and leaving s_1 equal to 0. It's now easy to see that the ray we described is contained entirely in the feasible region. This is illustrated in the original constraints in Figure 5.2.

Based on our previous example, we have the following theorem that extends Theorem 5.7:

THEOREM 5.12. *In a maximization problem, if* $\overline{a}_{j_i} \leq 0$ *for all* $i = 1, \ldots, m$, *and* $z_j - c_j < 0$, *then the linear programming problem is unbounded furthermore, let* $\overline{\mathbf{a}}_j$ *be the* j^{th} *column of* $\mathbf{B}^{-1}\mathbf{A}_{.j}$ *and let* $\mathbf{e}_k$ *be a standard basis column vector in* $\mathbb{R}^{m\times(n-m)}$ *where* k *corresponds to the position of* j *in the matrix* $\mathbf{N}$. *Then the direction:*

$$(5.35) \qquad \mathbf{d} = \begin{bmatrix} -\overline{\mathbf{a}}_j \\ \mathbf{e}_k \end{bmatrix}$$

is an extreme direction of the feasible region $X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b}, \, \mathbf{x} \geq \mathbf{0}\}$.
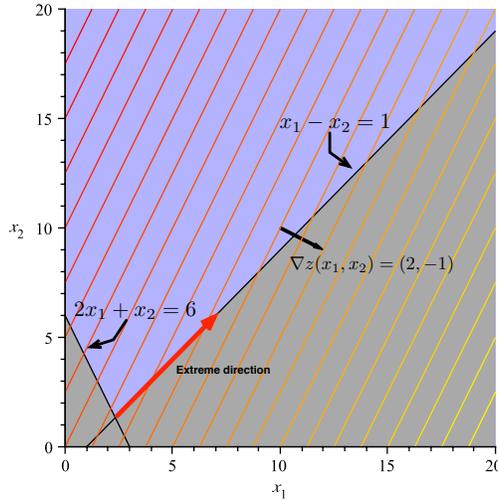
Figure 5.2. Unbounded Linear Program: The existence of a negative column $\bar{\mathbf{a}}_j$ in the simplex tableau for entering variable x_j indicates an unbounded problem and feasible region. The recession direction is shown in the figure.

PROOF. The fact that $\mathbf{d}$ is a direction is easily verified by the fact there is an extreme point $\mathbf{x} = [\mathbf{x_B}\ \mathbf{x_N}]^T$ and for all $\lambda \geq 0$ we have:

$$(5.36) \quad \mathbf{x} + \lambda\mathbf{d} \in X$$

Thus it follows from the proof of Theorem 4.24 that $\mathbf{Ad} \leq \mathbf{0}$. The fact that $\mathbf{d} \geq \mathbf{0}$ and $\mathbf{d} \neq \mathbf{0}$ follows from our assumptions. Now, we know that we can write $\mathbf{A} = [\mathbf{B}|\mathbf{N}]$. Further, we know that $\bar{\mathbf{a}}_j = \mathbf{B}^{-1}\mathbf{A}_{\cdot j}$. Let us consider $\mathbf{Ad}$:

$$(5.37) \quad \mathbf{Ad} = [\mathbf{B}|\mathbf{N}] \begin{bmatrix} -\bar{\mathbf{a}}_j \\ \mathbf{e}_k \end{bmatrix} = -\mathbf{BB}^{-1}\mathbf{A}_{\cdot j} + \mathbf{Ne}_k$$

Remember, $\mathbf{e}_k$ is the standard basis vector that has have 1 precisely in the position corresponding to column $\mathbf{A}_{\cdot j}$ in matrix $\mathbf{N}$, so $\mathbf{A}_{\cdot j} = \mathbf{Ne}_j$. Thus we have:

$$(5.38) \quad -\mathbf{BB}^{-1}\mathbf{A}_{\cdot j} + \mathbf{Ne}_k = -\mathbf{A}_{\cdot j} + \mathbf{A}_{\cdot j} = \mathbf{0}$$

Thus, $\mathbf{Ad} = \mathbf{0}$. We can scale $\mathbf{d}$ so that $\mathbf{e}^T\mathbf{d} = 1$. We know that $n - m - 1$ elements of $\mathbf{d}$ are zero (because of $\mathbf{e}_k$) and we know that $\mathbf{Ad} = \mathbf{0}$. Thus $\mathbf{d}$ can be made to represent the intersection of n-hyperplanes in $\mathbb{R}^n$. Thus, $\mathbf{d}$ is an extreme point of the polyhedron $D = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{Ad} \leq \mathbf{0}, \mathbf{d} \geq \mathbf{0}, \mathbf{e}^T\mathbf{d} = 1\}$. It follows from Theorem 4.39, we know that $\mathbf{d}$ is an extreme direction of X. $\square$

EXERCISE 51. Consider the problem

$$\begin{cases} \min\ z(x_1, x_2) = 2x_1 - x_2 \\ s.t.\ x_1 - x_2 + s_1 = 1 \\ \qquad 2x_1 + x_2 - s_2 = 6 \\ \qquad x_1, x_2, s_1, s_2 \geq 0 \end{cases}$$

Using the rule you developed in Exercise 49, show that the minimization problem has an unbounded feasible solution. Find an extreme direction for this set. [Hint: The minimum ratio test is the same for a minimization problem. Execute the simplex algorithm as we did in Example 5.11 and use Theorem 5.12 to find the extreme direction of the feasible region.]

6. Identifying Alternative Optimal Solutions

We saw in Theorem 5.6 that is $z_j - c_j > 0$ for all $j \in \mathcal{J}$ (the indices of the non-basic variables), then the basic feasible solution generated by the current basis was optimal. Suppose that $z_j - c_j \geq 0$. Then we have a slightly different result:

THEOREM 5.13. *In Problem P for a given set of non-basic variables $\mathcal{J}$, if $z_j - c_j \geq 0$ for all $j \in \mathcal{J}$, then the current basic feasible solution is optimal. Further, if $z_j - c_j = 0$ for at least one $j \in \mathcal{J}$, then there are alternative optimal solutions. Furthermore, let $\bar{\mathbf{a}}_j$ be the jth column of $\mathbf{B}^{-1}\mathbf{A}_{\cdot j}$. Then the solutions to P are:*

$$(5.39) \quad \begin{cases} \mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} - \bar{\mathbf{a}}_j x_j \\ x_j \in \left[0, \min\left\{\dfrac{\bar{b}_i}{\bar{a}_{j_i}} : i = 1, \ldots, m, \ \bar{a}_{j_i} > 0\right\}\right] \\ x_r = 0, \forall r \in \mathcal{J}, r \neq j \end{cases}$$

PROOF. It follows from the proof of Theorem 5.6 that the solution must be optimal as $\partial z/\partial x_j \leq 0$ for all $j \in \mathcal{J}$ and therefore increasing and x_j will *not* improve the value of the objective function. If there is some $j \in \mathcal{J}$ so that $z_j - c_j = 0$, then $\partial z/\partial x_j = 0$ and we may increase the value of x_j up to some point specified by the minimum ratio test, while keeping other non-basic variables at zero. In this case, we will neither increase nor decrease the objective function value. Since that objective function value is optimal, it follows that the set of all such values (described in Equation 5.39) are alternative optimal solutions. $\square$

EXAMPLE 5.14. Let us consider the toy maker problem again from Example 2.3 and 5.9 with our adjusted objective

$$(5.40) \quad z(x_1, x_2) = 18x_1 + 6x_2$$

Now consider the penultimate basis from Example 5.9 in which we had as basis variables x_1, s_2 and x_2.

$$\mathbf{x_B} = \begin{bmatrix} x_1 \\ x_2 \\ s_2 \end{bmatrix} \quad \mathbf{x_N} = \begin{bmatrix} s_1 \\ s_3 \end{bmatrix} \quad \mathbf{c_B} = \begin{bmatrix} 18 \\ 6 \\ 0 \end{bmatrix} \quad \mathbf{c_N} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The matrices become:

$$\mathbf{B} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

The derived matrices are then:

$$\mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 35 \\ 15 \\ 95 \end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} 0 & 1 \\ 1 & -3 \\ -2 & 5 \end{bmatrix}$$

The cost information becomes:
$$\mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} = 720 \quad \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} 6 & 0 \end{bmatrix} \quad \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N} = \begin{bmatrix} 6 & 0 \end{bmatrix}$$

This yields the tableau:

(5.41)

	z	x_1	x_2	s_1	s_2	s_3	RHS
z	1	0	0	6	0	**0**	720
s_1	0	1	0	0	0	1	35
s_2	0	0	1	1	0	-3	15
s_3	0	0	0	-2	1	5	95

Unlike example 5.9, the reduced cost for s_3 is 0. This means that if we allow s_3 to enter the basis, the objective function value will not change. Performing the minimum ratio test however, we see that s_2 will still leave the basis:

(5.42)

	z	x_1	x_2	s_1	s_2	s_3	RHS	MRT (s_3)
z	1	0	0	6	0	**0**	720	
x_1	0	1	0	0	0	1	35	35
x_2	0	0	1	1	0	-3	15	$-$
s_2	0	0	0	-2	1	$\boxed{5}$	95	19

Therefore any solution of the form:

$$s_3 \in [0, 19]$$

(5.43)
$$\begin{bmatrix} x_1 \\ x_2 \\ s_2 \end{bmatrix} = \begin{bmatrix} 35 \\ 15 \\ 95 \end{bmatrix} - \begin{bmatrix} 1 \\ -3 \\ 5 \end{bmatrix} s_3$$

is an optimal solution to the linear programming problem. This precisely describes the edge shown in Figure 5.3.
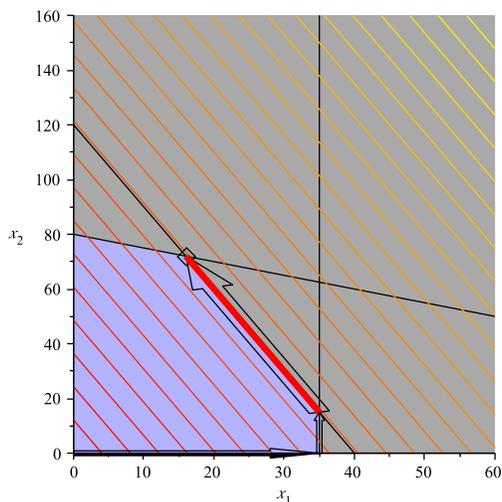
Figure 5.3. Infinite alternative optimal solutions: In the simplex algorithm, when $z_j - c_j \geq 0$ in a maximization problem with at least one j for which $z_j - c_j = 0$, indicates an infinite set of alternative optimal solutions.

EXERCISE 52. Consider the diet problem we covered in Example 3.50. I wish to design a diet consisting of Raman noodles and ice cream. I'm interested in spending as little money as possible but I want to ensure that I eat at least 1200 calories per day and that I get at least 20 grams of protein per day. Assume that each serving of Raman costs $1 and contains 100 calories and 2 grams of protein. Assume that each serving of ice cream costs $1.50 and contains 200 calories and 3 grams of protein.

(1) Develop a linear programming problem that will help me minimize the cost of my food intake.
(2) Remembering to transform the linear programming problem you found above into standard form, use the simplex algorithm to show that this problem has an infinite set of alternative optimal solutions.
(3) At an optimal extreme point, find an expression for the set of infinite alternative optimal exteme points like the one shown in Equation 5.43.
(4) Plot the feasible region and the level curves of the objective function. Highlight the face of the polyhedral set on which the alternative optimal solutions can be found.

7. Degeneracy and Convergence

In this section we give an example of degeneracy and its impact on the simplex algorithm.

EXAMPLE 5.15. Consider the modified form of the toy maker problem originally stated in Example 4.37:

$$(5.44) \quad \begin{cases} \max \ 7x_1 + 6x_2 \\ \text{s.t. } 3x_1 + x_2 \leq 120 \\ \quad x_1 + 2x_2 \leq 160 \\ \quad x_1 \leq 35 \\ \quad \dfrac{7}{4}x_1 + x_2 \leq 100 \\ \quad x_1, x_2 \geq 0 \end{cases}$$

The polyhedral set and level curves of the objective function are shown Figure 5.4. We can convert the problem to standard form by introducing slack variables:

$$(5.45) \quad \begin{cases} \max \ 7x_1 + 6x_2 \\ \text{s.t. } 3x_1 + x_2 + s_1 = 120 \\ \quad x_1 + 2x_2 + s_2 = 160 \\ \quad x_1 + s_3 = 35 \\ \quad \dfrac{7}{4}x_1 + x_2 + s_4 = 100 \\ \quad x_1, x_2, s_1, s_2, s_3, s_4 \geq 0 \end{cases}$$
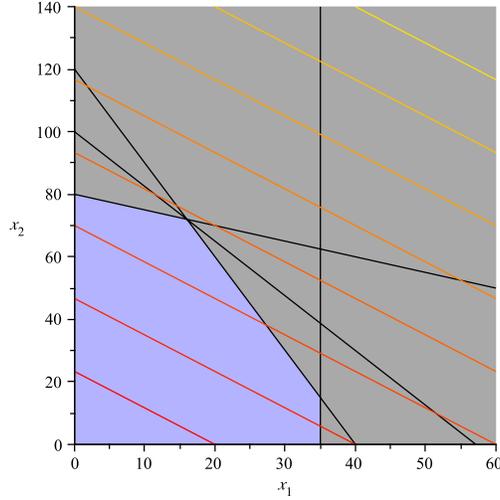
Figure 5.4. An optimization problem with a degenerate extreme point: The optimal solution to this problem is still $(16, 72)$, but this extreme point is degenerate, which will impact the behavior of the simplex algorithm.

Suppose we start at the extreme point where $x_1 = 35$ and $x_2 = 15$ and $s_2 = 95$ and $s_4 = 23.75$. In this case, the matrices are:

$$\mathbf{B} = \begin{bmatrix} 3 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 7/4 & 1 & 0 & 1 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 35 \\ 15 \\ 95 \\ \frac{95}{4} \end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} 0 & 1 \\ 1 & -3 \\ -2 & 5 \\ -1 & \frac{5}{4} \end{bmatrix}$$

$$\mathbf{c_B}\mathbf{B}^{-1}\mathbf{b} = 335 \quad \mathbf{c_B}\mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N} = \begin{bmatrix} 6 & -11 \end{bmatrix}$$

The tableau representation is:

(5.46)

	z	x_1	x_2	s_1	s_2	s_3	s_4	RHS	MRT (s_3)
z	1	0	0	6	0	-11	0	335	
x_1	0	1	0	0	0	1	0	35	35
x_2	0	0	1	1	0	-3	0	15	—
s_2	0	0	0	-2	1	5	0	95	19
s_4	0	0	0	-1	0	$\boxed{5/4}$	1	$\frac{95}{4}$	19

From this, we see that the variable s_3 should enter (because its reduce cost is negative). In this case, there is a tie for the leaving variables: we see that $95/5 = 19 = (95/4)/(5/4)$, therefore, either s_2 or s_4 could be chosen as the leaving variable. This is because we will move to a degenerate extreme point when s_3 enters the basis.

Suppose we choose s_4 as the leaving variable. Then our tableau will become:

(5.47)

	z	x_1	x_2	s_1	s_2	s_3	s_4	RHS	MRT (s_1)
z	1	0	0	$-14/5$	0	0	$44/5$	544	
x_1	0	1	0	$4/5$	0	0	$-4/5$	16	20
x_2	0	0	1	$-7/5$	0	0	$12/5$	72	$-$
s_2	0	0	0	$\boxed{2}$	1	0	-4	0	0
s_3	0	0	0	$-4/5$	0	1	$4/5$	19	$-$

We now observe two things:

(1) One of the basic variables (s_2) is zero, even though it is basic. This is *the indicator of degeneracy at an extreme point*.
(2) The reduced cost of s_1 is negative, indicating that s_1 should enter the basis.

If we choose s_1 as an entering variable, then using the minimum ratio test, we will choose s_2 as the leaving variable (by the minimum ratio test)[2]. Then the tableau becomes:

(5.48)

	z	x_1	x_2	s_1	s_2	s_3	s_4	RHS
z	1	0	0	0	$7/5$	0	$16/5$	544
x_1	0	1	0	0	$-2/5$	0	$4/5$	16
x_2	0	0	1	0	$7/10$	0	$-2/5$	72
s_1	0	0	0	1	$1/2$	0	-2	0
s_3	0	0	0	0	$2/5$	1	$-4/5$	19

Notice the objective function value $\mathbf{c_B B}^{-1}\mathbf{b}$ has *not* changed, because we really have not moved to a new extreme point. We have simply changed from one representation of the degenerate extreme point to another. This was to be expected, the fact that the minimum ratio was zero showed that we could not increase s_1 and maintain feasibility. As such $s_1 = 0$ in the new basic feasible solution. The reduced cost vector $\mathbf{c_B B}^{-1}\mathbf{N} - \mathbf{c_N}$ has changed and we could now terminate the simplex method.

THEOREM 5.16. *Consider Problem P (our linear programming problem). Let $\mathbf{B} \in \mathbb{R}^{m \times m}$ be a basis matrix corresponding to some set of basic variables $\mathbf{x_B}$. Let $\overline{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$. If $\overline{b}_j = 0$ for some $j = 1, \ldots, m$, then $\mathbf{x}_B = \overline{\mathbf{b}}$ and $\mathbf{x_N} = \mathbf{0}$ is a degenerate extreme point of the feasible region of Problem P.*

PROOF. At any basic feasible solutions we have chosen m variables as basic. This basic feasible solution satisfies $\mathbf{B x_B} = \mathbf{b}$ and thus provides m binding constraints. The remaining variables are chosen as non-basic and set to zero, thus $\mathbf{x_N} = \mathbf{0}$, which provides $n - m$ binding constraints on the non-negativity constraints (i.e., $\mathbf{x} \geq \mathbf{0}$). If there is a basic variable that is zero, then an extra non-negativity constraint is binding at that extreme point. Thus $n + 1$ constraints are binding and, by definition, the extreme point must be degenerate. $\square$

7.1. The Simplex Algorithm and Convergence. Using the work we've done in this chapter, we can now state the following implementation of the Simplex algorithm in matrix form.

---

[2]The minimum ratio test still applies when $\overline{b}_j = 0$. In this case, we will remain at the same extreme point.

> **Simplex Algorithm in Algebraic Form**
>
> (1) Given Problem P in **standard form** with cost vector $\mathbf{c}$, coefficient matrix $\mathbf{A}$ and right hand side $\mathbf{b}$, identify an initial basic feasible solution $\mathbf{x_B}$ and $\mathbf{x_N}$ by any means. Let $\mathcal{J}$ be the set of indices of non-basic variables. If no basic feasible solution can be found, **STOP**, the problem has no solution.
> (2) Compute the row vector $\mathbf{c_B}^T \mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N}^T$. This vector contains $z_j - c_j$ for $j \in \mathcal{J}$.
> (3) If $z_j - c_j \geq 0$ for all $j \in \mathcal{J}$, **STOP**, the current basic feasible solution is optimal. Otherwise, Goto 4.
> (4) Choose a non-basic variable x_j with $z_j - c_j < 0$. Select $\bar{\mathbf{a}}_j$ from $\mathbf{B}^{-1}\mathbf{N}$. If $\bar{\mathbf{a}}_j \leq 0$, then the problem is unbounded, **STOP**. Otherwise Goto 5.
> (5) Let $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$. Find the index i solving:
> $$\min\left\{ \bar{\mathbf{b}}_i/\bar{\mathbf{a}}_{j_i} : i = 1, \ldots, m \text{ and } \bar{\mathbf{a}}_{j_i} \geq 0 \right\}$$
> (6) Set $\mathbf{x_B}_i = 0$ and $x_j = \bar{\mathbf{b}}_i/\bar{\mathbf{a}}_{j_i}$.
> (7) Update $\mathcal{J}$ and GOTO Step 2

Algorithm 6. The Matrix form of the Simplex Algorithm

EXERCISE 53. State the simplex algorithm in Tableau Form. [Hint: Most of the simplex algorithm is the same, simply add in the row-operations executed to compute the new reduced costs and $\mathbf{B}^{-1}\mathbf{N}$.]

THEOREM 5.17. *If the feasible region of Problem P has no degenerate extreme points, then the simplex algorithm will terminate in a finite number of steps with an optimal solution to the linear programming problem.*

SKETCH OF PROOF. In the absence of degeneracy, the value of the objective function improves (increases in the case of a maximization problem) each time we exchange a basic variable and non-basic variable. This is ensured by the fact that the entering variable always has a negative reduced cost. There are a finite number of extreme points for each polyhedral set, as shown in Lemma 4.41. Thus, the process of moving from extreme point to extreme point of X, the polyhedral set in Problem P must terminate with the largest possible objective function value. $\square$

REMARK 5.18. The correct proof that the simplex algorithm converges to a point of optimality is actually proved by showing that the algorithm terminates with something called a Karush-Kuhn-Tucker (KKT) point. Unfortunately, we will not study the Karush-Kuhn-Tucker conditions until later. There are a few proofs that do not rely on showing that the point of optimality is a KKT point (see [**Dan60**] for example), but most rely on some intimate knowledge or assumptions on polyhedral sets and are not satisfying. Thus, for students who are not as concerned with the intricate details of the proof, the previous proof sketch is more than sufficient to convince you that the simplex algorithm is correct. For those students who prefer the rigorous detail, please see Chapter 8.

CHAPTER 6

Simplex Initialization

In the previous chapter, we introduced the Simplex Algorithm and showed how to manipulate the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{N}$ matrices as we execute it. In this chapter, we will discuss the issue of finding an initial basic feasible solution to start execution of the Simplex Algorithm.

1. Artificial Variables

So far we have investigated linear programming problems that had form:

$$\max \ \mathbf{c}^T\mathbf{x}$$
$$s.t. \ \mathbf{Ax} \leq \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

In this case, we use slack variables to convert the problem to:

$$\max \ \mathbf{c}^T\mathbf{x}$$
$$s.t. \ \mathbf{Ax} + \mathbf{I}_m\mathbf{x_s} = \mathbf{b}$$
$$\mathbf{x}, \mathbf{x_s} \geq \mathbf{0}$$

where $\mathbf{x_s}$ are slack variables, one for each constraint. If $\mathbf{b} \geq \mathbf{0}$, then our initial basic feasible solution can be $\mathbf{x} = \mathbf{0}$ and $\mathbf{x_s} = \mathbf{b}$ (that is, our initial basis matrix is $\mathbf{B} = \mathbf{I}_m$). We have also explored small problems where a graphical technique could be used to identify an initial extreme point of a polyhedral set and thus an initial basic feasible solution for the problem.

Suppose now we wish to investigate problems in which we do not have a problem structure that lends itself to easily identifying an initial basic feasible solution. The simplex algorithm requires an initial BFS to begin execution and so we must develop a method for finding such a BFS.

For the remainder of this chapter we will assume, unless told otherwise, that we are interested in solving a linear programming problem provided in Standard Form. That is:

$$(6.1) \quad P \begin{cases} \max \ \mathbf{c}^T\mathbf{x} \\ s.t. \ \mathbf{Ax} = \mathbf{b} \\ \quad \mathbf{x} \geq \mathbf{0} \end{cases}$$

and that $\mathbf{b} \geq \mathbf{0}$. Clearly our work in Chapter 3 shows that any linear programming problem can be put in this form.

Suppose to each constraint $\mathbf{A}_{i\cdot}\mathbf{x} = \mathbf{b}_i$ we associate an *artificial variable* x_{a_i}. We can replace constraint i with:

$$(6.2) \quad \mathbf{A}_{i\cdot}\mathbf{x} + x_{a_i} = \mathbf{b}_i$$

Since $\mathbf{b}_i \geq 0$, we will require $x_{a_i} \geq 0$. If $x_{a_i} = 0$, then this is simply the original constraint. Thus if we can find values for the ordinary decision variables $\mathbf{x}$ so that $x_{a_i} = 0$, then constraint i is satisfied. If we can identify values for $\mathbf{x}$ so that all the artificial variables are zero and m variables of $\mathbf{x}$ are non-zero, then the modified constraints described by Equation 6.2 are satisfied and we have identified an initial basic feasible solution.

Obviously, we would like to penalize non-zero artificial variables. This can be done by writing a new linear programming problem:

$$(6.3) \qquad P_1 \begin{cases} \min \; \mathbf{e}^T \mathbf{x_a} \\ s.t. \; \mathbf{Ax} + \mathbf{I}_m \mathbf{x_a} = \mathbf{b} \\ \qquad \mathbf{x}, \mathbf{x_a} \geq \mathbf{0} \end{cases}$$

REMARK 6.1. We can see that the artificial variables are similar to slack variables, but they should have zero value because they have no true meaning in the original problem P. They are introduced *artificially* to help identify an initial basic feasible solution to Problem P.

LEMMA 6.2. *The optimal objective function value in Problem P_1 is bounded below by 0. Furthermore, if the optimal solution to problem P_1 has $\mathbf{x_a} = \mathbf{0}$, then the values of $\mathbf{x}$ form a feasible solution to Problem P.*

PROOF. Clearly, setting $\mathbf{x_a} = \mathbf{0}$ will produce an objective function value of zero. Since $\mathbf{e} > \mathbf{0}$, we cannot obtain a smaller objective function value. If at optimality we have $\mathbf{x_a} = \mathbf{0}$, then we know that m of the variables in $\mathbf{x}$ are in the basis and the remaining variables (in both $\mathbf{x}$ and $\mathbf{x_a}$) are not in the basis and hence at zero. Thus we have found a basic feasible solution to Problem P. $\qquad \square$

EXAMPLE 6.3. Consider the following problem:

$$(6.4) \qquad \begin{aligned} \min \; & x_1 + 2x_2 \\ s.t. \; & x_1 + 2x_2 \geq 12 \\ & 2x_1 + 3x_2 \geq 20 \\ & x_1, x_2 \geq 0 \end{aligned}$$

We can convert the problem to standard form by adding two surplus variables:

$$(6.5) \qquad \begin{aligned} \min \; & x_1 + 2x_2 \\ s.t. \; & x_1 + 2x_2 - s_1 = 12 \\ & 2x_1 + 3x_2 - s_2 = 20 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{aligned}$$

It's not clear what a good basic feasible solution would be for this. Clearly, we cannot set $x_1 = x_2 = 0$ because we would have $s_1 = -12$ and $s_2 = -20$, which is not feasible. We can introduce two artificial variables (x_{a_1} and x_{a_2}) and create a new problem P_1.

$$(6.6) \qquad \begin{aligned} \min \; & x_{a_1} + x_{a_2} \\ s.t. \; & x_1 + 2x_2 - s_1 + x_{a_1} = 12 \\ & 2x_1 + 3x_2 - s_2 + x_{a_2} = 20 \\ & x_1, x_2, s_1, s_2, x_{a_1}, x_{a_2} \geq 0 \end{aligned}$$

A basic feasible solution for our *artificial problem* would let $x_{a_1} = 12$ and $x_{a_2} = 20$. The pertinent matrices in this case are:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & -1 & 0 & 1 & 0 \\ 2 & 3 & 0 & -1 & 0 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 12 \\ 20 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 2 & 3 & 0 & -1 \end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 12 \\ 20 \end{bmatrix}$$

$$\mathbf{c_B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{c_N} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{c_B}^T B^{-1}\mathbf{b} = 32 \quad \mathbf{c_B}^T \mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N}^T = \begin{bmatrix} 3 & 5 & -1 & -1 \end{bmatrix}$$

We can construct an initial tableau for this problem as:

(6.7)

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS
z	1	3	5	−1	−1	0	0	32
x_{a_1}	0	1	2	−1	0	1	0	12
x_{a_2}	0	2	3	0	−1	0	1	20

This is a minimization problem, so if $z_j - c_j > 0$, then entering x_j will improve (decrease) the objective value because $\partial z/\partial x_j < 0$. In this case, we could enter either x_1 or x_2 to improve the objective value. Let's assume we enter variable x_1. Performing the minimum ratio test we see:

(6.8)

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS	MRT(x_1)
z	1	3	5	−1	−1	0	0	32	
x_{a_1}	0	1	2	−1	0	1	0	12	12
x_{a_2}	0	2	3	0	−1	0	1	20	20/2 = 10

Thus x_{a_2} leaves the basis and x_1 enters. The new tableau becomes:

(6.9)

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS
z	1	0	1/2	−1	1/2	0	−3/2	2
x_{a_1}	0	0	1/2	−1	1/2	1	−1/2	2
x_1	0	1	3/2	0	−1/2	0	1/2	10

In this case, we see that x_2 should enter the basis. Performing the minimum ratio test, we obtain:

(6.10)

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS	MRT(x_2)
z	1	0	1/2	−1	1/2	0	−3/2	2	
x_{a_1}	0	0	1/2	−1	1/2	1	−1/2	2	4
x_1	0	1	3/2	0	−1/2	0	1/2	10	20/3

Thus we see that x_{a_2} leaves the basis and we obtain:

(6.11)

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS
z	1	0	0	0	0	−1	−1	0
x_2	0	0	1	−2	1	2	−1	4
x_1	0	1	0	3	−2	−3	2	4

At this point, we have eliminated both artificial variables from the basis and we have identified and initial basic feasible solution to the original problem: $x_1 = 4$, $x_2 = 4$, $s_1 = 0$ and $s_2 = 0$. The process of moving to a feasible solution in the original problem is shown in Figure 6.1.
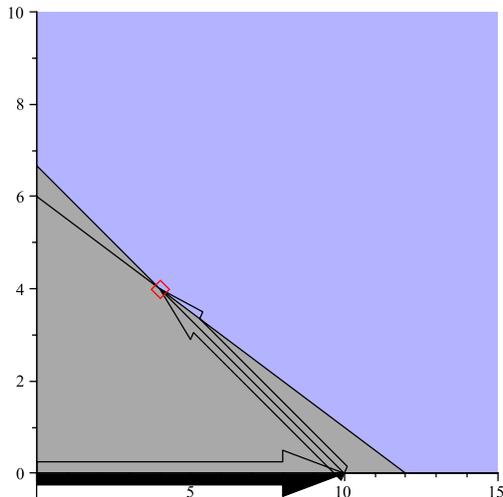
Figure 6.1. Finding an initial feasible point: Artificial variables are introduced into the problem. These variables allow us to move through non-feasible space. Once we reach a feasible extreme point, the process of optimizing Problem P_1 stops.

We could now continue on to solve the initial problem we were given. At this point, our basic feasible solution makes x_2 and x_1 basic variables and s_1 and s_2 non-basic variables. Our problem data are:

$$\mathbf{x_B} = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} \quad \mathbf{x_N} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

Note that we keep the basic variables in the order in which we find them at the end of the solution to our first problem.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 2 & 3 & 0 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 12 \\ 20 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

$$\mathbf{c_B} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \mathbf{c_N} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{c_B}^T B^{-1}\mathbf{b} = 12 \quad \mathbf{c_B}^T \mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N}^T = \begin{bmatrix} -1 & 0 \end{bmatrix}$$

Notice that we don't have to do a lot of work to get this information out of the last tableau in Expression 6.11. The matrix $\mathbf{B}^{-1}$ is actually positioned in the columns below the artificial variables. This is because we started with an identity matrix in this position. As always, the remainder of the matrix holds $\mathbf{B}^{-1}\mathbf{N}$. Thus, we can read this final tableau as:

$$(6.12) \quad \begin{array}{c} z \\ x_2 \\ x_1 \end{array} \left[\begin{array}{c|ccc|c} z & \mathbf{x_B} & \mathbf{s} & \mathbf{x}_a & \text{RHS} \\ \hline 1 & \mathbf{0} & \mathbf{0} & -\mathbf{e} & 0 \\ \hline 0 & \mathbf{I}_2 & \mathbf{B}^{-1}\mathbf{N} & \mathbf{B}^{-1} & \mathbf{B}^{-1}\mathbf{b} \end{array} \right]$$

In our case from Expression 6.11 we have:

$$(6.13) \quad \begin{array}{c} z \\ x_2 \\ x_1 \end{array} \left[\begin{array}{c|cc|cc|cc|c} z & x_1 & x_2 & s_1 & s_2 & x_{a_1} & x_{a_2} & \text{RHS} \\ \hline 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ \hline 0 & 0 & 1 & -2 & 1 & 2 & -1 & 4 \\ 0 & 1 & 0 & 3 & -2 & -3 & 2 & 4 \\ \hline - & \multicolumn{2}{c}{\mathbf{I}_2} & \multicolumn{2}{c}{\mathbf{B}^{-1}\mathbf{N}} & \multicolumn{2}{c}{\mathbf{B}^{-1}} & \mathbf{B}^{-1}\mathbf{b} \end{array} \right]$$

We can use this information (and the reduced costs and objective function we computed) to start our tableau to solve the problem with which we began. Our next initial tableau will be:

$$(6.14) \quad \begin{array}{c} z \\ x_2 \\ x_1 \end{array} \left[\begin{array}{c|cc|cc|c} z & x_1 & x_2 & s_1 & s_2 & \text{RHS} \\ \hline 1 & 0 & 0 & -1 & 0 & 12 \\ 0 & 0 & 1 & -2 & 1 & 4 \\ 0 & 1 & 0 & 3 & -2 & 4 \end{array} \right]$$

Notice all we've done is removed the artificial variables from the problem and substituted the newly computed reduced costs for s_1 and s_2 (-1 and 0) into Row 0 of the tableau. We've also put the correct objective function value (12) into Row 0 of the right hand side. We're now ready to solve the original problem. However, since this is a *minimization problem* we can see we're already at a point of optimality. Notice that all reduced costs are either negative or zero, suggesting that entering any non-basic variable will at best keep the objective function value the same and at worst make the objective function worse. Thus we conclude that an optimal solution for our original problem is $x_1^* = x_2^* = 4$ and $s_1^* = s_2^* = 0$.

THEOREM 6.4. *Let* $\mathbf{x}^*, \mathbf{x_a}^*$ *be an optimal feasible solution to problem* P_1*. Problem* P *is feasible if and only if* $\mathbf{x_a}^* = \mathbf{0}$*.*

PROOF. We have already proved in Lemma 6.2 that if $\mathbf{x_a}^* = \mathbf{0}$, then $\mathbf{x}^*$ is a feasible solution to P and thus P is feasible.

Conversely, suppose that P is feasible. Then P has at least one basic feasible solution because the feasible region of P is a polyhedral set and we are assured by Lemma 4.41 that this set has at least one extreme point. Now we can simply let $\mathbf{x_a}^* = \mathbf{0}$ and $\mathbf{x}$ be this basic feasible solution to problem P. Then this is clearly an optimal solution to problem P_1 because it forces the objective value to its lower bound (zero). □

2. The Two-Phase Simplex Algorithm

The two phase simplex algorithm applies the results from the previous section to develop an end-to-end algorithm for solving an arbitrary linear programming problem.

When we solve the Phase I problem, if $\mathbf{x_a^*} \neq \mathbf{0}$ at optimality, then there is no solution. If $\mathbf{x_a^*} = \mathbf{0}$, then there are two possibilities:

Two-Phase Simplex Algorithm

   (1) Given a problem of the form of the general maximization (or minimization) problem from Equation 2.1, convert it to standard form:

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

     with $\mathbf{b \geq 0}$.

   (2) Introduce auxiliary variables $\mathbf{x_a}$ and solve the **Phase I** problem:

$$P_1 \begin{cases} \min \ \mathbf{e}^T\mathbf{x_a} \\ s.t. \ \mathbf{Ax + I_m x_a = b} \\ \quad\ \ \mathbf{x, x_a \geq 0} \end{cases}$$

   (3) If $\mathbf{x_a^* = 0}$, then an initial feasible solution has been identified. This solution can be converted into a basic feasible solution as we discuss below. Otherwise, there is no solution to Problem P.

   (4) Use the Basic Feasible solution identified in Step 3 to start the Simplex Algorithm (compute the reduced costs given the $\mathbf{c}$ vector).

   (5) Solve the **Phase II** problem:

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

Algorithm 7. Two-Phase Simplex Algorithm

   (1) The basis consists only of variables in the vector $\mathbf{x}$; i.e., no auxiliary variable is in the basis.

   (2) There is some auxiliary variable $x_{a_i} = 0$ and this variable is in the basis; i.e., the solution is degenerate and the degeneracy is expressed in an auxiliary variable.

    2.1. Case I: $\mathbf{x_a = 0}$ and is out of the basis. If $\mathbf{x_a = 0}$ and there are not elements of the vector $\mathbf{x_a}$ in the basis, then we have identified a basic feasible solution $\mathbf{x} = [\mathbf{x_B} \ \ \mathbf{x_N}]^T$. Simply allow the non-zero basic elements (in $\mathbf{x}$) to be $\mathbf{x_B}$ and the remainder of the elements (not in $\mathbf{x_a}$) are in $\mathbf{x_N}$. We can then begin Phase II using this basic feasible solution.

    2.2. Case II: $\mathbf{x_a = 0}$ and is not out of the basis. If $\mathbf{x_a = 0}$ and there is at least one artificial variable still in the basis, then we have identified a degenerate solution to the Phase I problem. Theoretically we could proceed directly to Phase II, assigning 0 coefficients to the artificial variables as long as we ensure that *no artificial variable ever becomes positive again.* [**BJS04**] notes that this can be accomplished by selective pivoting, however it is often more efficient and simpler to remove the artificial variables completely from the basis before proceeding to Phase II.

To remove the artificial variables from the basis, let us assume that we can arrange Rows $1 - m$ of the Phase I simplex tableau as follows:

(6.15)

	$\mathbf{x_B}$	$\mathbf{x_{B_a}}$	$\mathbf{x_N}$	$\mathbf{x_{N_a}}$	RHS
$\mathbf{x_B}$	$\mathbf{I}_k$	$\mathbf{0}$	$\mathbf{R_1}$	$\mathbf{R_3}$	$\bar{\mathbf{b}}$
$\mathbf{x_{B_a}}$	$\mathbf{0}$	$\mathbf{I}_{m-k}$	$\mathbf{R_2}$	$\mathbf{R_4}$	$\mathbf{0}$

Column swapping ensures we can do this, if we so desire. Our objective is to replace elements in $\mathbf{x_{B_a}}$ (the basic artificial variables) with elements from $\mathbf{x_N}$ non-basic, non-artificial variables. Thus, we will attempt to pivot on elements in the matrix $\mathbf{R_2}$. Clearly since the Phase I coefficients of the variables in $\mathbf{x_N}$ are zero, pivoting in these elements will not negatively impact the Phase I objective value. Thus, if the element in position $(1,1)$ is non-zero, then we can enter the variable x_{N_1} into the basis and remove the variable $x_{B_{a_1}}$. This will produce a new tableau with structure similar to the one given in Equation 6.15 except there will be $k + 1$ non-artificial basic variables and $m - k - 1$ artificial basic variables. Clearly if the element in position $(1,1)$ in matrix $\mathbf{R_2}$ is zero, then we must move to a different element for pivoting.

In executing the procedure discussed above, one of two things will occur:

(1) The matrix $\mathbf{R_2}$ will be transformed into $\mathbf{I}_{m-k}$ or
(2) A point will be reached where there are no longer any variables in $\mathbf{x_N}$ that can be entered into the basis because all the elements of $\mathbf{R_2}$ are zero.

In the first case, we have removed all the artificial variables from the basis in Phase I and we can proceed to Phase II with the current basic feasible solution. In the second case, we will have shown that:

(6.16) $\quad \mathbf{A} \sim \begin{bmatrix} \mathbf{I}_k & \mathbf{R_1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$

This shows that the $m - k$ rows of $\mathbf{A}$ are *not* linearly independent of the first k rows and thus the matrix $\mathbf{A}$ did not have full row rank. When this occurs, we can discard the last $m - k$ rows of $\mathbf{A}$ and simply proceed with the solution given in $\mathbf{x_B} = \bar{\mathbf{b}}$, $\mathbf{x_N} = \mathbf{0}$. This is a basic feasible solution to the new matrix $\mathbf{A}$ in which we have removed the redundant rows.

EXAMPLE 6.5. Once execution of the Phase I simplex algorithm is complete, the reduced costs of the current basic feasible solution must be computed. These can be computed during Phase I by adding an additional "z" row to the tableau. In this case, the initial tableau has the form:

(6.17)

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS
z_{II}	1	-1	-2	0	0	0	0	0
z	1	3	5	-1	-1	0	0	32
x_{a_1}	0	1	2	-1	0	1	0	12
x_{a_2}	0	2	3	0	-1	0	1	20

The first row (z_{II}) is computed for the objective function:

(6.18) $\quad x_1 + 2x_2 + 0s_1 + 0s_2 + 0x_{a_1} + 0x_{a_2},$

which is precisely the Phase II problem, except we never allow the artificial variables x_{a_1} and x_{a_2} to carry into the Phase II problem. If we carry out the same steps we did in Example

6.3 then we obtain the sequence of tableaux:

TABLEAU I

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS	MRT(x_1)
z_{II}	1	-1	-2	0	0	0	0	0	
z	1	3	5	-1	-1	0	0	32	
x_{a_1}	0	1	2	-1	0	1	0	12	12
x_{a_2}	0	$\boxed{2}$	3	0	-1	0	1	20	$20/2 = 10$

TABLEAU II

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS	MRT(x_1)
z_{II}	1	0	$-1/2$	0	$-1/2$	0	$1/2$	10	
z	1	0	$1/2$	-1	$1/2$	0	$-3/2$	2	
x_{a_1}	0	0	$\boxed{1/2}$	-1	$1/2$	1	$-1/2$	2	4
x_1	0	1	$3/2$	0	$-1/2$	0	$1/2$	10	$20/3$

TABLEAU III

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS
z_{II}	1	0	0	-1	0	1	0	12
z	1	0	0	0	0	-1	-1	0
x_2	0	0	1	-2	1	2	-1	4
x_1	0	1	0	3	-2	-3	2	4

We again arrive at the end of Phase I, but we are now prepared to immediately execute Phase II with the tableau:

	z	x_1	x_2	s_1	s_2	RHS
z_{II}	1	0	0	-1	0	12
x_2	0	0	1	-2	1	4
x_1	0	1	0	3	-2	4

In this case, we see that we are already at an optimal solution for a minimization problem because the reduced costs are all less than or equal to zero. We also note that since the reduced cost of the non-basic variable s_2 is zero, there are alternative optimal solutions.

3. The Big-M Method

The Big-M method is similar to the two-phase simplex algorithm, except that it essentially attempts to execute Phase I and Phase II in a single execution of the simplex algorithm.

In the Big-M method we modify problem P with artificial variables as we did in the two-phase simplex method but we also modify the objective function:

$$(6.19) \quad P_M \begin{cases} \max \ \mathbf{c}^T\mathbf{x} - M\mathbf{e}^T\mathbf{x_a} \\ s.t. \ \mathbf{Ax} + \mathbf{I}_m\mathbf{x_a} = \mathbf{b} \\ \mathbf{x}, \mathbf{x_a} \geq \mathbf{0} \end{cases}$$

Here, M is a large positive constant, much larger than the largest coefficient in the vector $\mathbf{c}$. The value of M is usually chosen to be at least 100 times larger than the largest coefficient in the original objective function.

REMARK 6.6. In the case of a minimization problem, the objective function in the Big-M method becomes:

$$(6.20) \quad \min \ \mathbf{c}^T\mathbf{x} + M\mathbf{e}^T\mathbf{x_a}$$

EXERCISE 54. In Exercise 4 we showed that every maximization problem can be written as a minimization problem (and vice-versa). Show that Equation 6.20 follows by changing Problem P_M into a minimization problem.

LEMMA 6.7. *Suppose that problem P_M is unbounded. If problem P is feasible, then it is unbounded.*

PROOF. If P_M is unbounded, then there is some direction direction:

$$\mathbf{d}_M = \begin{bmatrix} \mathbf{d} \\ \mathbf{d_a} \end{bmatrix}$$

to the feasible region of Problem P_M. Furthermore, $\mathbf{d} \geq \mathbf{0}$ and $\mathbf{d_a} \geq \mathbf{0}$ and as a whole $\mathbf{d}_M \neq \mathbf{0}$. For this problem to be unbounded, it suffices that:

$$(6.21) \quad \mathbf{c}^T\mathbf{d} - M\mathbf{e}^T\mathbf{d}_a > 0$$

by Corollary 5.2.

Since we are free to choose M as large as we like, it follows that for a large value of M, the left-hand-side of Inequality 6.21 must be negative *unless $\mathbf{d_a} = \mathbf{0}$*.

The fact that $\mathbf{d}_M$ is a direction implies that $\mathbf{Ad} + \mathbf{I}_m\mathbf{d_a} = \mathbf{0}$ and therefore $\mathbf{Ad} = \mathbf{0}$. We know further that $\mathbf{d} \geq \mathbf{0}$ and $\mathbf{d} \neq \mathbf{0}$. Thus it follows that we have identified a direction $\mathbf{d}$ of the feasible region for Problem P. Furthermore, we know that following this direction must result in an unbounded objective function for P since the coefficients of the artificial variables are all negative. □

REMARK 6.8. Lemma 6.7 tells us that if the Problem P_M is unbounded, then we know that there is no *useful* solution to Problem P. If Problem P has non-empty feasible region, then it [Problem P] is unbounded and thus there is no useful solution. On the other hand, if Problem P has no feasible solution, there is still no useful solution to Problem P. *In either case*, we may need to re-model the problem to obtain a useful result.

THEOREM 6.9. *If Problem P is feasible and has a finite solution. Then there is an $M > 0$ so that the optimal solution to P_M has all artificial variables non-basic and thus the solution to Problem P can be extracted from the solution to Problem P_M.*

PROOF. By contrapositive applied to Lemma 6.7 we know that Problem P_M is bounded. By the Cartheodory Characterization theorem, we may enumerate the extreme points of the feasible region of Problem P_M: call these $\mathbf{y}_1, \ldots, \mathbf{y}_k$ where:

$$\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x_a} \end{bmatrix}$$

Let $z_{M_1}, \ldots z_{M_k}$ be the objective function value of Problem P_M at each of these extreme points. Since P is feasible, at least one of these extreme points has $\mathbf{x_a} = \mathbf{0}$. Let us sub-divide the extreme points in $\mathcal{Y}_a = \{\mathbf{y}_1, \ldots, \mathbf{y}_l\}$ and $\mathcal{Y} = \{\mathbf{y}_{l+1}, \ldots, \mathbf{y}_k\}$ where the points in $\mathcal{Y}_a$ are the extreme points such that there is at least one non-zero artificial variable and the points in $\mathcal{Y}$ are the extreme points where all artificial variables are zero. At any extreme

point in $\mathcal{Y}$ we know that *at most m* elements of the vector $\mathbf{x}$ are non-zero and therefore, every extreme point in $\mathcal{Y}$ corresponds to an extreme point of the original problem P. Since Problem P has a finite solution, it follows that the optimal solution to problem P occurs at some point in $\mathcal{Y}$, by Theorem 5.1. Furthermore the value of the objective function for Problem P is precisely the same as the value of the objective function of Problem P_M for each point in $\mathcal{Y}$ because $\mathbf{x_a} = \mathbf{0}$. Define:

$$(6.22) \quad z_P^{\min} = \min_{\mathbf{y} \in \mathcal{Y}} \{\mathbf{c}^T \mathbf{x} : \mathbf{y} = [\mathbf{x} \ \mathbf{x_a}]^T\}$$

At each extreme point in $\mathcal{Y}_a$, the value of the objective function of Problem P_M is a function of the value of M, which we are free to choose. Therefore, choose M so that:

$$(6.23) \quad \max_{\mathbf{y} \in \mathcal{Y}_a} \{\mathbf{c}^T \mathbf{x} - M\mathbf{e}^T \mathbf{x_a} : \mathbf{y} = [\mathbf{x} \ \mathbf{x_a}]^T\} < z_P^{\min}$$

Such a value exists for M since there are only a finite number of extreme points in $\mathcal{Y}$. Our choice of M ensures that the optimal solution to P_M occurs at an extreme point where $\mathbf{x_a} = \mathbf{0}$ and the $\mathbf{x}$ component of $\mathbf{y}$ is the solution to Problem P. $\square$

REMARK 6.10. Another way to look at the proof of this theorem is to think of defining M in such a way so that at any extreme point where $\mathbf{x_a} \neq \mathbf{0}$, the objective function can always be made larger by moving to any extreme point that is feasible to Problem P. Thus the simplex algorithm will move among the extreme points seeking to leave those that are not feasible to Problem P because they are less desirable.

THEOREM 6.11. *Suppose Problem P is infeasible. Then there is no value of M that will drive the all the artificial variables from the basis of Problem P_M.*

PROOF. If such an M existed, then $\mathbf{x_a} = \mathbf{0}$ and the resulting values of $\mathbf{x}$ represents a feasible solution to Problem P, which contradicts our assumption that Problem P was infeasible. $\square$

REMARK 6.12. The Big-M method is not particularly effective for solving real-world problems. The introduction of a set of variables with large coefficients (M) can lead to round-off errors in the execution of the simplex algorithm. (Remember, computers can only manipulate numbers in binary, which means that all floating point numbers are restricted in their precision to the machine precision of the underlying system OS. This is generally given in terms of the largest amount of memory that can be addressed in bits. This has led, in recent times, to operating system manufacturers selling their OS's as "32 bit" or "64 bit." When solving real-world problems, these issue can become a real factor with which to contend.

Another issue is we have no way of telling how large M should be without knowing that Problem P is feasible, which is precisely what we want the Big-M method to tell us! The general rule of thumb provided earlier will suffice.

EXAMPLE 6.13. Suppose we solve the problem from Example 6.3 using the Big-M method. Our problem is:

$$
(6.24) \quad
\begin{aligned}
\min \ & x_1 + 2x_2 \\
s.t. \ & x_1 + 2x_2 \geq 12 \\
& 2x_1 + 3x_2 \geq 20 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

Again, this problem has standard form:

$$
(6.25) \quad
\begin{aligned}
\min \ & x_1 + 2x_2 \\
s.t. \ & x_1 + 2x_2 - s_1 = 12 \\
& 2x_1 + 3x_2 - s_2 = 20 \\
& x_1, x_2, s_1, s_2 \geq 0
\end{aligned}
$$

To execute the Big-M method, we'll choose $M = 300$ which is larger than 100 times the largest coefficient of the objective function of the original problem. Our new problem becomes:

$$
(6.26) \quad
\begin{aligned}
\min \ & x_1 + 2x_2 + 300x_{a_1} + 300x_{a_2} \\
s.t. \ & x_1 + 2x_2 - s_1 + x_{a_1} = 12 \\
& 2x_1 + 3x_2 - s_2 + x_{a_2} = 20 \\
& x_1, x_2, s_1, s_2, x_{a_1}, x_{a_2} \geq 0
\end{aligned}
$$

Since this is a *minimization* problem, we *add* $M\mathbf{e}^T\mathbf{x_a}$ to the objective function. Letting x_{a_1} and x_{a_2} be our initial basis, we have the series of tableaux:

TABLEAU I

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS	MRT(x_1)
z	1	899	1498	-300	-300	0	0	9600	
x_{a_1}	0	1	2	-1	0	1	0	12	12
x_{a_2}	0	$\boxed{2}$	3	0	-1	0	1	20	$20/2 = 10$

TABLEAU II

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS	MRT(x_1)
z	1	0	$299/2$	-300	$299/2$	0	$-899/2$	610	
x_{a_1}	0	0	$\boxed{1/2}$	-1	$1/2$	1	$-1/2$	2	4
x_1	0	1	$3/2$	0	$-1/2$	0	$1/2$	10	$20/3$

TABLEAU III

	z	x_1	x_2	s_1	s_2	x_{a_1}	x_{a_2}	RHS
z	1	0	0	-1	0	-299	-300	12
x_2	0	0	1	-2	1	2	-1	4
x_1	0	1	0	3	-2	-3	2	4

It is worth noting that this is essentially the same series of tableau we had when executing the Two-Phase method, but we have to deal with the large M coefficients in our arithmetic.

4. The Single Artificial Variable Technique

Consider the system of equations $\mathbf{Ax} = \mathbf{b}$ that composes a portion of the feasible region of Problem P. Suppose we chose some sub-matrix of $\mathbf{A}$ to be our basis matrix $\mathbf{B}$ irrespective of whether the solution $\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$. If $\mathbf{A}$ has full row rank, then clearly such a matrix exists. The resulting basic solution with basis $\mathbf{B}$ is called a *crash basis*.

If $\overline{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$, then we have (by luck) identified an initial basic feasible solution and we can proceed directly to execute the simplex algorithm as we did in Chapter 5. Suppose that $\overline{\mathbf{b}} \not\geq \mathbf{0}$. Then we can form the new system:

$$(6.27) \quad \mathbf{I}_m \mathbf{x_B} + \mathbf{B}^{-1}\mathbf{N} + \mathbf{y}_a x_a = \overline{\mathbf{b}}$$

where x_a is a single artificial variable and $\mathbf{y}_a$ is a (row) vector of coefficients for x_a so that:

$$(6.28) \quad \mathbf{y}_{a_i} = \begin{cases} -1 & \text{if } \overline{\mathbf{b}}_i < 0 \\ 0 & \text{else} \end{cases}$$

LEMMA 6.14. *Suppose we enter x_a into the basis by pivoting on the row of the simplex tableau with most negative right hand side. That is, x_a is exchanged with variable $\mathbf{x_B}_j$ having most negative value. Then the resulting solution is a basic feasible solution to the constraints:*

$$(6.29) \quad \begin{aligned} \mathbf{I}_m \mathbf{x_B} + \mathbf{B}^{-1}\mathbf{N} + \mathbf{y}_a x_a &= \overline{\mathbf{b}} \\ \mathbf{x}, x_a &\geq \mathbf{0} \end{aligned}$$

EXERCISE 55. Prove Lemma 6.14.

The resulting basic feasible solution can either be used as a starting solution for the two-phase simplex algorithm with the single artificial variable *or* the Big-M method. For the two-phase method, we would solve the Phase I problem:

$$(6.30) \quad \begin{aligned} \min \;\; & x_a \\ s.t. \;\; & \mathbf{Ax} + \mathbf{B}_0 \mathbf{y}_a x_a = \mathbf{b} \\ & \mathbf{x}, x_a \geq \mathbf{0} \end{aligned}$$

where $\mathbf{B}_0$ is the initial crash basis we used to identify the coefficients of single artificial variable. Equation 6.30 is generated by multiplying by $\mathbf{B}_0$ on both sides of the inequalities.

EXAMPLE 6.15. Suppose we were interested in the constraint set:

$$(6.31) \quad \begin{aligned} x_1 + 2x_2 - s_1 &= 12 \\ 2x_1 + 3x_2 - s_2 &= 20 \\ x_1, x_2, s_1, s_2 &\geq 0 \end{aligned}$$

We can choose the crash basis:

$$(6.32) \quad \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

corresponding to the variables s_1 and s_2. Then we obtain the system:

$$(6.33) \quad \begin{aligned} -x_1 - 2x_2 + s_1 &= -12 \\ -2x_1 - 3x_2 + s_2 &= -20 \end{aligned}$$

That is, $s_1 = -12$ and $s_2 = -20$ is our basic solution, which is not feasible. We append the artificial variable with coefficient vector $\mathbf{y}_a = [-1 \quad -1]^T$ (since both elements of the right-hand-side are negative) to obtain:

(6.34)
$$- x_1 - 2x_2 + s_1 - x_a = -12$$
$$- 2x_1 - 3x_2 + s_2 - x_a = -20$$

If we build a tableau for Phase I with this current BFS, we obtain:

	z	x_1	x_2	s_1	s_2	x_a	RHS
z	1	0	0	0	0	-1	0
s_1	0	-1	-2	1	0	-1	-12
s_2	0	-2	-3	0	1	-1	-20

We enter the variable x_a and pivot out variable s_2 which has the **most negative** right hand side to obtain the initial feasible tableau:

	z	x_1	x_2	s_1	s_2	x_a	RHS
z	1	2	3	0	1	0	20
s_1	0	1	1	1	-1	0	8
x_a	0	2	3	0	-1	1	20

We can now complete the Phase I process and execute the simplex algorithm until we drive x_a from the basis and reduce the right-hand-side to 0. At this point we will have identified an initial basic feasible solution to the initial problem and we can execute Phase II.

REMARK 6.16. Empirical evidence suggests that the single artificial variable technique is not as efficient as the two-phase or Big-M method. Thus, it is presented as an historical component of the development of efficient implementations of the Simplex algorithm and not as a realistic technique for implementation in production systems.

5. Problems that Can't be Initialized by Hand

In these notes, we have so far considered very small problems that could easily be solved graphically. Determining an initial basic feasible solution requires little more than trial and error. These problems hardly require the use of Phase I methods.

To provide an example of a class of problems that can easily generate large numbers of variables, we will consider a multi-period inventory control problem. These problems can easily generate large numbers of variables and constraints, even in small problems.

EXAMPLE 6.17. McLearey's Shamrock Emporium produces and sells shamrocks for three days each year: the day before St. Patrick's Day, St. Patrick's Day and the day after St. Patrick's day. This year, McLearey had 10 shamrocks left over from last year's sale. This year, he expects to sell 100 shamrocks the day before St. Patrick's Day, 200 shamrocks the day of St. Patrick's day and 50 shamrocks the day after St. Patrick's day.

It costs McLearey $2 to produce each Shamrock and $0.01 to store a Shamrock over night. Additionally, McLearey can put shamrocks into long term storage for $0.05 per shamrock.

McLearey can produce at most 150 shamrocks per day. Shamrocks must be produced within two days of being sold (or put into long term storage) otherwise, they wilt. Assuming that McLearey must meet his daily demand and will not start producing Shamrocks early,

he wants to know how many shamrocks he should make and store on each day to minimize his cost.

To determine an answer to this problem, note that we have time as a parameter: time runs over three days. Let x_t be the number of shamrocks McLearey makes on day t ($t = 1, 2, 3$) and let y_t be the number of shamrocks McLearey stores on day t. There is also a parameter $y_0 = 10$, the number of shamrocks left over from last year.

McLearey's total cost (in cents) can the be written as:

$$(6.35) \quad z = 200x_1 + 200x_2 + 200x_3 + y_1 + y_2 + 5y_3$$

Additionally, there are some constraints linking production, storage and demand. These constraints are depicted graphically in Figure 6.2. Multiperiod inventory models operate on a principle of conservation of flow. Manufactured goods and previous period inventories flow into the box representing each period. Demand and next period inventories flow out of the box representing each period. This inflow and outflow must be equal to account for all shamrocks produced. This is depicted below:
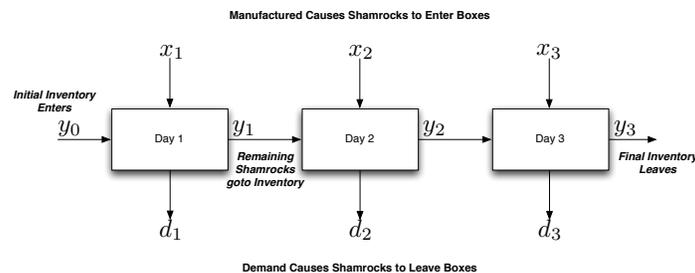
Figure 6.2. Multiperiod inventory models operate on a principle of conservation of flow. Manufactured goods and previous period inventories flow into the box representing each period. Demand and next period inventories flow out of the box representing each period. This inflow and outflow must be equal to account for all production.

This means that:

$$(6.36) \quad y_{t-1} + x_t = y_t + d_t \quad \forall t$$

This equation says that at period t the amount of inventory carried over from period $t - 1$ plus amount of shamrocks produced in period t must be equal to the total demand in period t plus any left over shamrocks at the end of period t. Clearly we also know that $x_t \geq 0$ for all t, since you cannot make a negative number of shamrocks. However, by also requiring $y_t \geq 0$ for all t, then we assert that our inventory can never be negative. A *negative inventory* is a *backorder*. Thus by saying that $y_t \geq 0$ we are also satisfying the requirement that McLearey satisfy his demand in each period. Note, when $t = 1$, then $y_{t-1} = y_0$, which is the parameter we defined above.

The complete problem describing McLearey's situation is:

$$(6.37) \quad \begin{cases} \min & 200x_1 + 200x_2 + 200x_3 + y_1 + y_2 + 5y_3 \\ s.t. & y_{t-1} + x_t = y_t + d_t \quad \forall t \in \{1, 2, 3\} \\ & x_t \leq 150 \quad \forall t \in \{1, 2, 3\} \\ & x_t, y_t \geq 0 \quad \forall t \in \{1, 2, 3\} \end{cases}$$

Constraints of the form $x_t \leq 150$ for all t come from the fact that McLearey can produce at most 150 shamrocks per day.

This simple problem now has 6 variables and 6 constraints plus 6 non-negativity constraints and it is non-trivial to determine a good initial basic feasible solution, especially since the problem contains both equality and inequality constraints.

A problem like this can be solved in Matlab (see Chapter 3.11), or on a commercial or open source solver like the GNU Linear Programming Kit (GLPK, http://www.gnu.org/software/glpk/). In Figure 6.3 we show an example *model file* that describes the problem. In Figure 6.4, we show the data section of the GLPK model file describing McLearey's problem. Finally, figure 6.5 shows a portion of the output generated by the GLPK solver `glpsol` using this model. Note that there is no inventory in Year 3 (because it is too expensive) even though it might be beneficial to McLearey to hold inventory for next year. This is because the problem has no information about any other time periods and so, in a sense, the *end of the world* occurs immediately after period 3. This type of *end of the world* phenomenon is common in multi-period problems.

```
#
# This finds the optimal solution for McLearey
#

/* sets */
set DAY;
set DAY2;

/* parameters */
param makeCost {t in DAY};
param holdCost {t in DAY};
param demand {t in DAY};
param start;
param S;

/* decision variables: */
var x {t in DAY} >= 0;
var y {t in DAY} >= 0;

/* objective function */
minimize z: sum{t in DAY} (makeCost[t]*x[t]+holdCost[t]*y[t]);

/* Flow Constraints */
s.t. FLOWA : x[1] - y[1] = demand[1] - start;
s.t. FLOWB{t in DAY2} : x[t] + y[t-1] - y[t] = demand[t];

/* Manufacturing constraints */
s.t. MAKE{t in DAY} : x[t] <= S;


end;
```

Figure 6.3. Input model to GLPK describing McLearey's Problem

```
/* data section */
data;
set DAY := 1 2 3;
set DAY2 := 2 3;

param makeCost:=
1 200
2 200
3 200;

param holdCost:=
1 1
2 1
3 5;

param demand:=
1 100
2 200
3 50;

param start:=10;
param S:=150;
end;
```

Figure 6.4. Input data to GLPK describing McLearey's Problem

```
Problem:    Shamrock
Rows:       7
Columns:    6
Non-zeros:  17
Status:     OPTIMAL
Objective:  z = 68050 (MINimum)


    No. Column name  St   Activity     Lower bound   Upper bound   Marginal
------ ------------ --  ------------- ------------- ------------- -------------
     1 x[1]          B           140             0
     2 x[2]          B           150             0
     3 x[3]          B            50             0
     4 y[1]          B            50             0
     5 y[2]          NL            0             0                           2
     6 y[3]          NL            0             0                         205
```

Figure 6.5. Output from `glpsol` on the McLearey Problem.

Degeneracy and Convergence

In this section, we will consider the problem of degeneracy and prove (at last) that there is an implementation of the Simplex Algorithm that is guaranteed to converge to an optimal solution, assuming one exists.

1. Degeneracy Revisited

We've already discussed degeneracy. Recall the following theorem from Chapter 5 that defines degeneracy in terms of the simplex tableau:

Theorem 5.16. *Consider Problem P (our linear programming problem). Let* $\mathbf{B} \in \mathbb{R}^{m \times m}$ *be a basis matrix corresponding to some set of basic variables* $\mathbf{x_B}$. *Let* $\overline{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$. *If* $\overline{\mathbf{b}}_j = \mathbf{0}$ *for some* $j = 1, \ldots, m$, *then* $\mathbf{x}_B = \overline{\mathbf{b}}$ *and* $\mathbf{x_N} = \mathbf{0}$ *is a degenerate extreme point of the feasible region of Problem P.*

We have seen in Example 5.15 that degeneracy can cause us to take extra steps on our way from an initial basic feasible solution to an optimal solution. When the simplex algorithm takes extra steps while remaining at the same degenerate extreme point, this is called *stalling*. The problem can become much worse; for certain entering variable rules, the simplex algorithm can become locked in a cycle of pivots each one moving from one characterization of a degenerate extreme point to the next. The following example from Beale and illustrated in Chapter 4 of [**BJS04**] demonstrates the point.

EXAMPLE 7.1. Consider the following linear programming problem:

$$\min \quad -\frac{3}{4}x_4 + 20x_5 - \frac{1}{2}x_6 + 6x_7$$

(7.1)

$$s.t \ \ x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = 0$$

$$x_2 + \frac{1}{2}x_4 - 12x_5 - \frac{1}{2}x_6 + 3x_7 = 0$$

$$x_3 + x_6 = 1$$

$$x_i \geq 0 \ \ i = 1, \ldots, 7$$

It is conducive to analyze the $\mathbf{A}$ matrix of the constraints of this problem. We have:

(7.2)
$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1/4 & -8 & -1 & 9 \\ 0 & 1 & 0 & 1/2 & -12 & -1/2 & 3 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The fact that the $\mathbf{A}$ matrix contains an identity matrix embedded within it suggests that an initial basic feasible solution with basic variables x_1, x_2 and x_3 would be a good choice.

This leads to a vector of reduced costs given by:

$$(7.3) \qquad \mathbf{c_B}^T \mathbf{B}^{-1} \mathbf{N} - \mathbf{c_N}^T = \begin{bmatrix} 3/4 & -20 & 1/2 & -6 \end{bmatrix}$$

These yield an initial tableau with structure:

	z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	1	0	0	0	3/4	−20	1/2	−6	0
x_1	0	1	0	0	1/4	−8	−1	9	0
x_2	0	0	1	0	1/2	−12	−1/2	3	0
x_3	0	0	0	1	0	0	1	0	1

    If we apply an entering variable rule where we always chose the non-basic variable to enter with the *most positive* reduced cost (since this is a minimization problem), and we choose the leaving variable to be the first row that is in a tie, then we will obtain the following sequence of tableaux:

Tableau I:

	z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	1	0	0	0	3/4	−20	1/2	−6	0
x_1	0	1	0	0	1/4	−8	−1	9	0
x_2	0	0	1	0	1/2	−12	−1/2	3	0
x_3	0	0	0	1	0	0	1	0	1

Tableau II:

	z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	1	−3	0	0	0	4	7/2	−33	0
x_4	0	4	0	0	1	−32	−4	36	0
x_2	0	−2	1	0	0	4	3/2	−15	0
x_3	0	0	0	1	0	0	1	0	1

Tableau III:

	z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	1	−1	−1	0	0	0	2	−18	0
x_4	0	−12	8	0	1	0	8	−84	0
x_5	0	−1/2	1/4	0	0	1	3/8	−15/4	0
x_3	0	0	0	1	0	0	1	0	1

Tableau IV:

	z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	1	2	−3	0	−1/4	0	0	3	0
x_6	0	−3/2	1	0	1/8	0	1	−21/2	0
x_5	0	1/16	−1/8	0	−3/64	1	0	3/16	0
x_3	0	3/2	−1	1	−1/8	0	0	21/2	1

Tableau V:

$$\begin{bmatrix}
 & z & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & RHS \\
\hline
z & 1 & 1 & -1 & 0 & 1/2 & -16 & 0 & 0 & 0 \\
\hline
x_6 & 0 & \boxed{2} & -6 & 0 & -5/2 & 56 & 1 & 0 & 0 \\
x_7 & 0 & 1/3 & -2/3 & 0 & -1/4 & 16/3 & 0 & 1 & 0 \\
x_3 & 0 & -2 & 6 & 1 & 5/2 & -56 & 0 & 0 & 1
\end{bmatrix}$$

Tableau VI:

$$\begin{bmatrix}
 & z & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & RHS \\
\hline
z & 1 & 0 & 2 & 0 & 7/4 & -44 & -1/2 & 0 & 0 \\
\hline
x_1 & 0 & 1 & -3 & 0 & -5/4 & 28 & 1/2 & 0 & 0 \\
x_7 & 0 & 0 & \boxed{1/3} & 0 & 1/6 & -4 & -1/6 & 1 & 0 \\
x_3 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1
\end{bmatrix}$$

Tableau VII:

$$\begin{bmatrix}
 & z & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & RHS \\
\hline
z & 1 & 0 & 0 & 0 & 3/4 & -20 & 1/2 & -6 & 0 \\
\hline
x_1 & 0 & 1 & 0 & 0 & 1/4 & -8 & -1 & 9 & 0 \\
x_2 & 0 & 0 & 1 & 0 & 1/2 & -12 & -1/2 & 3 & 0 \\
x_3 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1
\end{bmatrix}$$

We see that the last tableau (VII) is the same as the first tableau and thus we have constructed an instance where (using the given entering and leaving variable rules), the Simplex Algorithm will cycle forever at this degenerate extreme point.

2. The Lexicographic Minimum Ratio Leaving Variable Rule

Given the example of the previous section, we require a method for breaking ties in the case of degeneracy is required that prevents cycling from occurring. There is a large literature on cycling prevention rules, however the most well known is the *lexicographic rule for selecting the entering variable.*

DEFINITION 7.2 (Lexicographic Order). Let $\mathbf{x} = [x_1, \dots, x_n]^T$ and $\mathbf{y} = [y_1, \dots, y_n]^T$ be vectors in $\mathbb{R}^n$. We say that $\mathbf{x}$ is *lexicographically greater* than $\mathbf{y}$ if: there exists $m < n$ so that $x_i = y_i$ for $i = 1, \dots, m$, and $x_{m+1} > y_{m+1}$.

Clearly, if there is no such $m < n$, then $x_i = y_i$ for $i = 1, \dots, n$ and thus $\mathbf{x} = \mathbf{y}$. We write $\mathbf{x} \succ \mathbf{y}$ to indicate that $\mathbf{x}$ is lexicographically greater than $\mathbf{y}$. Naturally, we can write $\mathbf{x} \succeq \mathbf{y}$ to indicate that $\mathbf{x}$ is lexicographically greater than or equal to $\mathbf{y}$.

Lexicographic ordering is simply the standard order operation $>$ applied to the individual elements of a vector in $\mathbb{R}^n$ with a precedence on the index of the vector.

DEFINITION 7.3. A vector $\mathbf{x} \in \mathbb{R}^n$ is *lexicographically positive* if $\mathbf{x} \succ \mathbf{0}$ where $\mathbf{0}$ is the zero vector in $\mathbb{R}^n$.

LEMMA 7.4. *Let $\mathbf{x}$ and $\mathbf{y}$ be two lexicographically positive vectors in $\mathbb{R}^n$. Then $\mathbf{x} + \mathbf{y}$ is lexicographically positive. Let $c > 0$ be a constant in $\mathbb{R}$, then $c\mathbf{x}$ is a lexicographically positive vector.*

EXERCISE 56. Prove Lemma 7.4.

2.1. Lexicographic Minimum Ratio Test. Suppose we are considering a linear programming problem and we have chosen an entering variable x_j according to a fixed entering variable rule. Assume further, we are given some current basis matrix $\mathbf{B}$ and as usual, the right-hand-side vector of the constraints is denoted $\mathbf{b}$, while the coefficient matrix is denoted $\mathbf{A}$. Then the minimum ratio test asserts that we will chose as the leaving variable the basis variable with the minimum ratio in the minimum ratio test. Consider the following set:

$$(7.4) \quad I_0 = \left\{ r : \frac{\bar{b}_r}{\bar{a}_{j_r}} = \min\left[\frac{\bar{b}_i}{\bar{a}_{j_i}} : i = 1, \ldots, m \text{ and } a_{j_i} > 0 \right] \right\}$$

In the absence of degeneracy, I_0 contains a *single* element: the row index that has the smallest ratio of $\bar{b}_i$ to $\bar{a}_{j_i}$, where naturally: $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$ and $\bar{\mathbf{a}}_j = \mathbf{B}^{-1}\mathbf{A}_{\cdot j}$. In this case, x_j is swapped into the basis in exchange for x_{B_r} (the rth basic variable).

When we have a degenerate basic feasible solution, then I_0 is not a singleton set and contains all the rows that have *tied* in the minimum ratio test. In this case, we can form a new set:

$$(7.5) \quad I_1 = \left\{ r : \frac{\bar{a}_{1_r}}{\bar{a}_{j_r}} = \min\left[\frac{\bar{a}_{1_i}}{\bar{a}_{j_i}} : i \in I_0 \right] \right\}$$

Here, we are taking the elements in column 1 of $\mathbf{B}^{-1}\mathbf{A}_{\cdot 1}$ to obtain $\bar{\mathbf{a}}_1$. The elements of this (column) vector are then being divided by the elements of the (column) vector $\bar{\mathbf{a}}_j$ on a index-by-index basis. If this set is a singleton, then basic variable x_{B_r} leaves the basis. If this set is not a singleton, we may form a new set I_2 with column $\bar{\mathbf{a}}_2$. In general, we will have the set:

$$(7.6) \quad I_k = \left\{ r : \frac{\bar{a}_{k_r}}{\bar{a}_{j_r}} = \min\left[\frac{\bar{a}_{k_i}}{\bar{a}_{j_i}} : i \in I_{k-1} \right] \right\}$$

LEMMA 7.5. *For any degenerate basis matrix* $\mathbf{B}$ *for any linear programming problem, we will ultimately find a k so that I_k is a singleton.*

EXERCISE 57. Prove Lemma 7.5. [Hint: Assume that the tableau is arranged so that the identity columns are columns 1 through m. (That is $\bar{\mathbf{a}}_j = \mathbf{e}_j$ for $i = 1, \ldots, m$.) Show that this configuration will easily lead to a singleton I_k for $k < m$.]

In executing the lexicographic minimum ratio test, we can see that we are essentially comparing the tied rows in a lexicographic manner. If a set of rows ties in the minimum ratio test, then we execute a minimum ratio test on the first column of the tied rows. If there is a tie, then we move on executing a minimum ratio test on the second column of the rows that tied in both previous tests. This continues until the tie is broken and a single row emerges as the leaving row.

EXAMPLE 7.6. Let us consider the example from Beale again using the lexicographic minimum ratio test. Consider the tableau shown below.

Tableau I:

	z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	1	0	0	0	$3/4$	-20	$1/2$	-6	0
x_1	0	1	0	0	$1/4$	-8	-1	9	0
x_2	0	0	1	0	$\boxed{1/2}$	-12	$-1/2$	3	0
x_3	0	0	0	1	0	0	1	0	1

112

Again, we chose to enter variable x_4 as it has the most positive reduced cost. Variables x_1 and x_2 tie in the minimum ratio test. So we consider a new minimum ratio test on the first column of the tableau:

$$(7.7) \quad \min\left\{\frac{1}{1/4}, \frac{0}{1/2}\right\}$$

From this test, we see that x_2 is the leaving variable and we pivot on element $1/2$ as indicated in the tableau. Note, we *only* need to execute the minimum ratio test on variables x_1 and x_2 since those were the tied variables in the standard minimum ratio test. That is, $I_0 = \{1, 2\}$ and we construct I_1 from these indexes alone. In this case $I_1 = \{2\}$. Pivoting yields the new tableau:

Tableau II:

	z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	1	0	$-3/2$	0	0	-2	$5/4$	$-21/2$	0
x_1	0	1	$-1/2$	0	0	-2	$-3/4$	$15/2$	0
x_4	0	0	2	0	1	-24	-1	6	0
x_3	0	0	0	1	0	0	$\boxed{1}$	0	1

There is no question this time of the entering or leaving variable, clearly x_6 must enter and x_3 must leave and we obtain[1]:

Tableau III:

	z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	1	0	$-3/2$	$-5/4$	0	-2	0	$-21/2$	$-5/4$
x_1	0	1	$-1/2$	$3/4$	0	-2	0	$15/2$	$3/4$
x_4	0	0	2	1	1	-24	0	6	1
x_6	0	0	0	1	0	0	1	0	1

Since this is a minimization problem and the reduced costs of the non-basic variables are now all negative, we have arrived at an optimal solution. The lexicographic minimum ratio test successfully prevented cycling.

2.2. Convergence of the Simplex Algorithm Under Lexicographic Minimum Ratio Test.

LEMMA 7.7. *Consider the problem:*

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

Suppose the following hold:

(1) $\mathbf{I}_m$ *is embedded in the matrix* $\mathbf{A}$ *and is used as the starting basis,*
(2) *a consistent entering variable rule is applied (e.g., largest reduced cost first), and*
(3) *the lexicographic minimum ratio test is applied as the leaving variable rule.*

Then each row of the sequence of augmented matrices $[\overline{\mathbf{b}}|\mathbf{B}^{-1}]$ *is lexicographically positive. Here* $\mathbf{B}$ *is the basis matrix and* $\overline{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$.

---

[1]Thanks to Ethan Wright for finding a small typo in this example, that is now fixed.

113

PROOF. The initial basis $\mathbf{I}_m$ yields an augmented matrix $[\mathbf{b}|\mathbf{I}_m]$. This matrix clearly has every row lexicographically positive, since $\mathbf{b} \geq 0$. Assume that the rows of $[\overline{\mathbf{b}}|\mathbf{B}^{-1}] \succ \mathbf{0}$ for the first n iterations of the simplex algorithm with fixed entering variable rule and lexicographic minimum ratio test. We will show that it is also true for step $n + 1$.

Suppose (after iteration n) we have the following tableau:

	z	x_1	$\ldots$	x_j	$\ldots$	x_m	x_{m+1}	$\ldots$	x_k	$\ldots$	x_n	RHS
z	1	$z_1 - c_1$	$\ldots$	$z_j - c_j$	$\ldots$	$z_m - c_m$	$z_{m+1} - c_{m+1}$	$\ldots$	$z_k - c_k$	$\ldots$	$z_n - c_n$	$\overline{z}$
x_{B_1}	0	$\overline{a}_{11}$	$\ldots$	$\overline{a}_{1j}$	$\ldots$	$\overline{a}_{1m}$	$\overline{a}_{1m+1}$	$\ldots$	$\overline{a}_{1k}$	$\ldots$	$\overline{a}_{1n}$	$\overline{b}_1$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
x_{B_i}	0	$\overline{a}_{i1}$	$\ldots$	$\overline{a}_{ij}$	$\ldots$	$\overline{a}_{im}$	$\overline{a}_{im+1}$	$\ldots$	$\overline{a}_{ik}$	$\ldots$	$\overline{a}_{in}$	$\overline{b}_i$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
x_{B_r}	0	$\overline{a}_{r1}$	$\ldots$	$\overline{a}_{rj}$	$\ldots$	$\overline{a}_{rm}$	$\overline{a}_{rm+1}$	$\ldots$	$\boxed{\overline{a}_{rk}}$	$\ldots$	$\overline{a}_{rn}$	$\overline{b}_r$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
x_{B_m}	0	$\overline{a}_{m1}$	$\ldots$	$\overline{a}_{mj}$	$\ldots$	$\overline{a}_{mm}$	$\overline{a}_{mm+1}$	$\ldots$	$\overline{a}_{mk}$	$\ldots$	$\overline{a}_{mn}$	$\overline{b}_m$

Table 1. Tableau used for Proof of Lemma 7.7

Assume using the entering variable rule of our choice that x_k will enter. Let us consider what happens as we choose a leaving variable and execute a pivot. Suppose that after executing the lexicographic minimum ratio test, we pivot on element $\overline{a}_{rk}$. Consider the pivoting operation on row i: there are two cases:

Case I $i \notin I_0$: If $i \notin I_0$, then we replace b_i with

$$\overline{b}'_i = \overline{b}_i - \frac{\overline{a}_{ik}}{\overline{a}_{rk}} \overline{b}_r$$

If $\overline{a}_{ij} < 0$, then clearly $b'_i > 0$. Otherwise, since $i \notin I_0$, then:

$$\frac{\overline{b}_r}{\overline{a}_{rk}} < \frac{\overline{b}_i}{\overline{a}_{ik}} \implies \overline{b}_r \frac{\overline{a}_{ik}}{\overline{a}_{rk}} < \overline{b}_i \implies 0 < \overline{b}_i - \frac{\overline{a}_{ik}}{\overline{a}_{rk}} \overline{b}_r$$

Thus we know that $\overline{b}'_i > 0$. It follows then that row i of the augmented matrix $[\overline{\mathbf{b}}|\mathbf{B}^{-1}]$ is lexicographically positive.

Case II $i \in I_0$: Then $\overline{b}'_i = \overline{b}_i - (\overline{a}_{ik}/\overline{a}_{rk})\overline{b}_r = 0$ since

$$\frac{\overline{b}_r}{\overline{a}_{rk}} = \frac{\overline{b}_i}{\overline{a}_{ik}}$$

There are now two possibilities, either $i \in I_1$ or $i \notin I_1$. In the first, case we can argue that

$$\overline{a}'_{i1} = \overline{a}_{i1} - \frac{\overline{a}_{ik}}{\overline{a}_{rk}} \overline{a}_{r1} > 0$$

for the same reason that $\overline{b}'_i > 0$ in the case when $i \in I_0$, namely that the lexicographic minimum ratio test ensures that:

$$\frac{\overline{a}_{r1}}{\overline{a}_{rk}} < \frac{\overline{a}_{i1}}{\overline{a}_{ik}}$$

114

if $i \notin I_1$. This confirms (since $\bar{b}'_i = 0$) row i of the augmented matrix $[\bar{\mathbf{b}}|\mathbf{B}^{-1}]$ is lexicographically positive.

In the second case that $i \in I_1$, then we may proceed to determine whether $i \in I_2$. This process continues until we identify the j for which I_j is the singleton index r. Such a j must exist by Lemma 7.5. In each case, we may reason that row i of the augmented matrix $[\bar{\mathbf{b}}|\mathbf{B}^{-1}]$ is lexicographically positive.

The preceding argument shows that at step $n+1$ of the simplex algorithm we will arrive an augmented matrix $[\bar{\mathbf{b}}|\mathbf{B}^{-1}]$ for which every row is lexicographically positive. This completes the proof. □

REMARK 7.8. The assumption that we force $\mathbf{I}_m$ into the basis can be justified in one of two ways:

(1) We may assume that we first execute a Phase I simplex algorithm with artificial variables. Then the forgoing argument applies.
(2) Assume we are provided with a crash basis $\mathbf{B}$ and we form the equivalent problem:

$$P' \begin{cases} \max \ \mathbf{0}^T\mathbf{x_B} + (\mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N})\mathbf{x_N} \\ s.t. \ \mathbf{I}_m\mathbf{x_B} + \mathbf{B}^{-1}\mathbf{N}\mathbf{x_N} = \mathbf{B}^{-1}\mathbf{b} \\ \quad \mathbf{x_B}, \mathbf{x_N} \geq \mathbf{0} \end{cases}$$

where $\mathbf{B}^{-1}\mathbf{b} \geq 0$. This problem is clearly equivalent because its initial simplex tableau will be identical to a simplex tableau generated by Problem P with basis matrix $\mathbf{B}$. If no such crash basis exists, then the problem is infeasible.

LEMMA 7.9. *Under the assumptions of Lemma 7.7, let $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$ be row vectors in $\mathbb{R}^{n+1}$ corresponding to Row 0 from the simplex tableau at iterations i and $i+1$ respectively. Assume, however, that we exchange the z column (column 1) and the RHS column (column $n+2$). Then $\mathbf{z}_{i+1} - \mathbf{z}_i$ is lexicographically positive.*

PROOF. Consider the tableau in Table 1. If we are solving a maximization problem, then clearly for x_k to be an entering variable (as we assumed in the proof of Lemma 7.7) we must have $z_k - c_k < 0$. Then the new Row Zero is obtained by adding:

$$\mathbf{y} = \frac{-(z_k - c_k)}{\bar{a}_{rk}} \begin{bmatrix} 0 & \bar{a}_{r1} & \dots & \bar{a}_{rj} & \dots & \bar{a}_{rm} & \bar{a}_{rm+1} & \dots & \bar{a}_{rk} & \dots & \bar{a}_{rn} & \bar{b}_r \end{bmatrix}$$

to the current row zero consisting of $\begin{bmatrix} 1 & z_1 - c_1 & \dots & z_n - c_n & \bar{z} \end{bmatrix}$. That is: $\mathbf{z}_{i+1} = \mathbf{z}_i + \mathbf{y}$, or $\mathbf{y} = \mathbf{z}_{i+1} - \mathbf{z}_i$.

The fact that $z_k - c_k < 0$ and $\bar{a}_{rk} > 0$ (in order to pivot at that element) implies that $-(z_k - c_k)/\bar{a}_{rk} > 0$. Further, Lemma 7.7 asserts that the vector $\begin{bmatrix} 0 & \bar{a}_{r1} & \dots & \bar{a}_{rn} & \bar{b}_r \end{bmatrix}$ is lexicographically positive (if we perform the exchange of column 1 and column $n + 2$ as we assumed we would). Thus, $\mathbf{y}$ is lexicographically positive by Lemma 7.4. This completes the proof. □

THEOREM 7.10. *Under the assumptions of Lemma 7.7, the simplex algorithm converges in a finite number of steps.*

PROOF. Assume by contradiction that we begin to cycle. Then there is a sequence of row 0 vectors $\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_l$ so that $\mathbf{z}_l = \mathbf{z}_0$. Consider $\mathbf{y}_i = \mathbf{z}_i - \mathbf{z}_{i-1}$. By Lemma 7.9, $\mathbf{y}_i \succ \mathbf{0}$

for $i = 1, \ldots, n$. Then we have:

$$(7.8) \quad \mathbf{y}_1 + \mathbf{y}_2 + \cdots + \mathbf{y}_l = (\mathbf{z}_1 - \mathbf{z}_0) + (\mathbf{z}_2 - \mathbf{z}_1) + \cdots + (\mathbf{z}_l - \mathbf{z}_{l-1}) =$$
$$(\mathbf{z}_1 - \mathbf{z}_0) + (\mathbf{z}_2 - \mathbf{z}_1) + \cdots + (\mathbf{z}_0 - \mathbf{z}_{l-1}) = \mathbf{z}_0 - \mathbf{z}_0 = \mathbf{0}$$

But by Lemma 7.4, the sum of lexicographically positive vectors is lexicographically positive. Thus we have established a contradiction. This cycle cannot exist and the simplex algorithm must converge by the same argument we used in the proof of Theorem 5.17. This completes the proof. □

REMARK 7.11. Again, the proof of correctness, i.e., that the simplex algorithm with the lexicographic minimum ratio test finds a point of optimality, is left until the next chapter when we'll argue that the simplex algorithm finds a so-called KKT point.

3. Bland's Rule, Entering Variable Rules and Other Considerations

There are many other anti-cycling rules that each have their own unique proofs of convergence. Bland's rule is a simple one: All the variables are ordered (say by giving them an index) and strictly held in that order. If many variables may enter, then the variable with lowest index is chosen to enter. If a tie occurs in the minimum ratio test, then variable with smallest index leaves the basis. It is possible to show that this rule will prevent cycling. However, it can also lead to excessively long simplex algorithm execution.

In general, there are many rules for choosing the entering variable. Two common ones are:

(1) Largest absolute reduced cost: In this case, the variable with most negative (in maximization) or most positive (in minimization) reduced cost is chosen to enter.
(2) Largest impact on the objective: In this case, the variable whose entry will cause the greatest increase (or decrease) to the objective function is chosen. This of course requires pre-computation of the value of the objective function for each possible choice of entering variable and can be time consuming.

Leaving variable rules (like Bland's rule or the lexicographic minimum ratio test) can also be expensive to implement. Practically, many systems ignore these rules and use floating point error to break ties. This does not ensure that cycling does not occur, but often is useful in a practical sense. However, care must be taken. In certain simplex instances, floating point error cannot be counted on to ensure tie breaking and consequently cycling prevention rules *must* be implemented. This is particularly true in network flow problems that are coded as linear programs. It is also important to note that *none* of these rules prevent stalling. Stalling prevention is a complicated thing and there are still open questions on whether certain algorithms admit or prevent stalling. See Chapter 4 of [**BJS04**] for a treatment of this subject.

The Revised Simplex Method and Optimality Conditions

1. The Revised Simplex Method

Consider an arbitrary linear programming problem, which we will assume is written in standard form:

$$(8.1) \quad P \begin{cases} \max & \mathbf{c}^T \mathbf{x} \\ s.t. & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{cases}$$

The tableau method is a substantially data intensive process as we carry the entire simplex tableau with us as we execute the simplex algorithm. However, consider the data we need at each iteration of the algorithm:

(1) Reduced costs: $\mathbf{c}_\mathbf{B}^T \mathbf{B}^{-1} \mathbf{A}_{\cdot j} - c_j$ for each variable x_j where $j \in \mathcal{J}$ and $\mathcal{J}$ is the set of indices of non-basic variables.
(2) Right-hand-side values: $\overline{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b}$ for use in the minimum ratio test.
(3) $\overline{\mathbf{a}}_j = \mathbf{B}^{-1}\mathbf{A}_{\cdot j}$ for use in the minimum ratio test.
(4) $\overline{z} = \mathbf{c}_\mathbf{B}^T \mathbf{B}^{-1}\mathbf{b}$, the current objective function value.

The one value that is clearly critical to the computation is $\mathbf{B}^{-1}$ as it appears in each and every computation. It would be far more effective to keep only the values: $\mathbf{B}^{-1}$, $\mathbf{c}_\mathbf{B}^T \mathbf{B}^{-1}$, $\overline{\mathbf{b}}$ and $\overline{z}$ and compute the reduced cost values and vectors $\overline{\mathbf{a}}_j$ as we need them.

Let $\mathbf{w} = \mathbf{c}_\mathbf{B}^T \mathbf{B}^{-1}$, then the pertinent information may be stored in a new *revised simplex tableau* with form:

$$(8.2) \quad \mathbf{x_B} \begin{bmatrix} \mathbf{w} & \overline{z} \\ \hline \mathbf{B}^{-1} & \overline{\mathbf{b}} \end{bmatrix}$$

The revised simplex algorithm is detailed in Algorithm 8. In essence, the revised simplex algorithm allows us to avoid computing $\overline{\mathbf{a}}_j$ until we absolutely need to do so. In fact, if we do not apply Dantzig's entering variable rule and simply select the first acceptable entering variable, then we may be able to avoid computing a substantial number of columns in the tableau.

EXAMPLE 8.1. Consider a software company who is developing a new program. The company has identified two types of bugs that remain in this software: *non-critical* and *critical*. The company's actuarial firm predicts that the risk associated with these bugs are uniform random variables with mean $100 per non-critical bug and mean $1000 per critical bug. The software currently has 50 non-critical bugs and 5 critical bugs.

Assume that it requires 3 hours to fix a non-critical bug and 12 hours to fix a critical bug. For each day (8 hour period) beyond two business weeks (80 hours) that the company fails to ship its product, the actuarial firm estimates it will loose $500 per day.

Algorithm 8. Revised Simplex Algorithm

We can find the optimal number of bugs of each type the software company should fix assuming it wishes to minimize its exposure to risk using a linear programming formulation.

Let x_1 be the number of non-critical bugs corrected and x_2 be the number of critical software bugs corrected. Define:

(8.3) $\quad y_1 = 50 - x_1$

(8.4) $\quad y_2 = 5 - x_2$

Here y_1 is the number of non-critical bugs that are not fixed while y_2 is the number of critical bugs that are not fixed.

The time (in hours) it takes to fix these bugs is:

(8.5) $\quad 3x_1 + 12x_2$

Let:

$$(8.6) \qquad y_3 = \frac{1}{8}(80 - 3x_1 - 12x_2)$$

Then y_3 is a variable that is unrestricted in sign and determines the amount of time (in days) either over or under the two-week period that is required to ship the software. As an unrestricted variable, we can break it into two components:

$$(8.7) \qquad y_3 = z_1 - z_2$$

We will assume that $z_1, z_2 \geq 0$. If $y_3 > 0$, then $z_1 > 0$ and $z_2 = 0$. In this case, the software is completed ahead of the two-week deadline. If $y_3 < 0$, then $z_1 = 0$ and $z_2 > 0$. In this case the software is finished after the two-week deadline. Finally, if $y_3 = 0$, then $z_1 = z_2 = 0$ and the software is finished precisely on time.

We can form the objective function as:

$$(8.8) \qquad z = 100y_1 + 1000y_2 + 500z_2$$

The linear programming problem is then:

$$(8.9) \qquad \begin{aligned} \min \quad z &= y_1 + 10y_2 + 5z_2 \\ \text{s.t.} \quad x_1 + y_1 &= 50 \\ x_2 + y_2 &= 5 \\ \frac{3}{8}x_1 + \frac{3}{2}x_2 + z_1 - z_2 &= 10 \\ x_1, x_2, y_1, y_2, z_1, z_2 &\geq 0 \end{aligned}$$

Notice we have modified the objective function by dividing by 100. This will make the arithmetic of the simplex algorithm easier. The matrix of coefficients for this problem is:

$$(8.10) \qquad \begin{bmatrix} x_1 & x_2 & y_1 & y_2 & z_1 & z_2 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ \frac{3}{8} & \frac{3}{2} & 0 & 0 & 1 & -1 \end{bmatrix}$$

Notice there is an identity matrix embedded inside the matrix of coefficients. Thus a good initial basic feasible solution is $\{y_1, y_2, z_1\}$. The initial basis matrix is $\mathbf{I}_3$ and naturally, $\mathbf{B}^{-1} = \mathbf{I}_3$ as a result. We can see that $\mathbf{c_B} = [1\ 10\ 0]^T$. It follows that $\mathbf{c_B}^T\mathbf{B}^{-1} = \mathbf{w} = [1\ 10\ 0]$.

Our initial revised simplex tableau is thus:

$$(8.11) \qquad \begin{array}{c} z \\ y_1 \\ y_2 \\ z_1 \end{array} \begin{bmatrix} 1 & 10 & 0 & 100 \\ 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 10 \end{bmatrix}$$

There are three variables that might enter at this point, x_1, x_2 and z_1. We can compute the reduced costs for each of these variables using the columns of the $\mathbf{A}$ matrix, the coefficients of these variables in the objective function and the current $\mathbf{w}$ vector (in row 0 of the revised

simplex tableau). We obtain:

$$z_1 - c_1 = \mathbf{w}\mathbf{A}_{\cdot 1} - c_1 = \begin{bmatrix} 1 & 10 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 3/8 \end{bmatrix} - 0 = 1$$

$$z_2 - c_2 = \mathbf{w}\mathbf{A}_{\cdot 2} - c_2 = \begin{bmatrix} 1 & 10 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 3/2 \end{bmatrix} - 0 = 10$$

$$z_6 - c_6 = \mathbf{w}\mathbf{A}_{\cdot 6} - c_6 = \begin{bmatrix} 1 & 10 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} - 5 = -5$$

By Dantzig's rule, we enter variable x_2. We append $\mathbf{B}^{-1}\mathbf{A}_{\cdot 2}$ and the reduced cost to the revised simplex tableau to obtain:

(8.12)

							MRT
z	1	10	0	100	10		
y_1	1	0	0	50	0		$-$
y_2	0	1	0	5	$\boxed{1}$		5
z_1	0	0	1	10	3/2		20/3

After pivoting on the indicated element, we obtain the new tableau:

(8.13)

z	1	0	0	50
y_1	1	0	0	50
x_2	0	1	0	5
z_1	0	$-3/2$	1	5/2

We can compute reduced costs for the non-basic variables (except for y_2, which we know will **not** re-enter the basis on this iteration) to obtain:

$$z_1 - c_1 = \mathbf{w}\mathbf{A}_{\cdot 1} - c_1 = 1$$
$$z_6 - c_6 = \mathbf{w}\mathbf{A}_{\cdot 6} - c_6 = -5$$

In this case, x_1 will enter the basis and we augment our revised simplex tableau to obtain:

(8.14)

						MRT
z	1	0	0	50	1	
y_1	1	0	0	50	1	50
x_2	0	1	0	5	0	$-$
z_1	0	$-3/2$	1	5/2	3/8	$\boxed{20/3}$

Note that:

$$\mathbf{B}^{-1}\mathbf{A}_{\cdot 1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3/2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 3/8 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 3/8 \end{bmatrix}$$

120

This is the $\bar{\mathbf{a}}_1$ column that is appended to the right hand side of the tableau along with $z_1 - c_1 = 1$. After pivoting, the tableau becomes:

$$
(8.15) \quad
\begin{array}{c}
z \\
y_1 \\
x_2 \\
x_1
\end{array}
\left[
\begin{array}{ccc|c}
1 & 4 & -8/3 & 130/3 \\
\hline
1 & 4 & -8/3 & 130/3 \\
0 & 1 & 0 & 5 \\
0 & -4 & 8/3 & 20/3
\end{array}
\right]
$$

We can now check our reduced costs. Clearly, z_1 will not re-enter the basis. Therefore, we need only examine the reduced costs for the variables y_2 and z_2.

$$z_4 - c_4 = \mathbf{w}\mathbf{A}_{\cdot 4} - c_4 = -6$$

$$z_6 - c_6 = \mathbf{w}\mathbf{A}_{\cdot 6} - c_6 = -7/3$$

Since all reduced costs are now negative, no further minimization is possible and we conclude we have arrived at an optimal solution.

Two things are interesting to note: first, the solution for the number of non-critical software bugs to fix is non-integer. Thus, in reality the company must fix either 6 or 7 of the non-critical software bugs. The second thing to note is that this economic model helps to explain why some companies are content to release software that contains known bugs. In making a choice between releasing a flawless product or making a quicker (larger) profit, a selfish, profit maximizer will always choose to fix only those bugs it must fix and release sooner rather than later.

EXERCISE 58. Solve the following problem using the revised simplex algorithm.

$$
\begin{aligned}
\max \quad & x_1 + x_2 \\
\text{s.t.} \quad & 2x_1 + x_2 \leq 4 \\
& x_1 + 2x_2 \leq 6 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

2. Farkas' Lemma and Theorems of the Alternative

LEMMA 8.2 (Farkas' Lemma). *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{c} \in \mathbb{R}^n$ be a row vector. Suppose $\mathbf{x} \in \mathbb{R}^n$ is a column vector and $\mathbf{w} \in \mathbb{R}^m$ is a row vector. Then exactly one of the following systems of inequalities has a solution:*

(1) $\mathbf{A}\mathbf{x} \geq \mathbf{0}$ *and* $\mathbf{c}\mathbf{x} < 0$ *or*
(2) $\mathbf{w}\mathbf{A} = \mathbf{c}$ *and* $\mathbf{w} \geq \mathbf{0}$

REMARK 8.3. Before proceeding to the proof, it is helpful to restate the lemma in the following way:

(1) *If there is a vector $\mathbf{x} \in \mathbb{R}^n$ so that $\mathbf{A}\mathbf{x} \geq \mathbf{0}$ and $\mathbf{c}\mathbf{x} < 0$, then there **is no** vector $\mathbf{w} \in \mathbb{R}^m$ so that $\mathbf{w}\mathbf{A} = \mathbf{c}$ and $\mathbf{w} \geq \mathbf{0}$.*

(2) *Conversely, if there is a vector $\mathbf{w} \in \mathbb{R}^m$ so that $\mathbf{w}\mathbf{A} = \mathbf{c}$ and $\mathbf{w} \geq \mathbf{0}$, then there **is no** vector $\mathbf{x} \in \mathbb{R}^n$ so that $\mathbf{A}\mathbf{x} \geq \mathbf{0}$ and $\mathbf{c}\mathbf{x} < 0$.*

PROOF. We can prove Farkas' Lemma using the fact that a bounded linear programming problem has an extreme point solution. Suppose that System 1 has a solution $\mathbf{x}$. If System 2 also has a solution $\mathbf{w}$, then

$$(8.16) \quad \mathbf{wA} = \mathbf{c} \implies \mathbf{wAx} = \mathbf{cx}.$$

The fact that System 1 has a solution ensures that $\mathbf{cx} < 0$ and therefore $\mathbf{wAx} < 0$. However, it also ensures that $\mathbf{Ax} \geq \mathbf{0}$. The fact that System 2 has a solution implies that $\mathbf{w} \geq \mathbf{0}$. Therefore we must conclude that:

$$(8.17) \quad \mathbf{w} \geq \mathbf{0} \text{ and } \mathbf{Ax} \geq \mathbf{0} \implies \mathbf{wAx} \geq \mathbf{0}.$$

This contradiction implies that if System 1 has a solution, then System 2 cannot have a solution.

Now, suppose that System 1 has no solution. We will construct a solution for System 2. If System 1 has no solution, then there is no vector $\mathbf{x}$ so that $\mathbf{cx} < 0$ and $\mathbf{Ax} \geq \mathbf{0}$. Consider the linear programming problem:

$$(8.18) \quad P_F \begin{cases} \min \ \mathbf{cx} \\ s.t. \ \mathbf{Ax} \geq \mathbf{0} \end{cases}$$

Clearly $\mathbf{x} = \mathbf{0}$ is a feasible solution to this linear programming problem and furthermore is optimal. To see this, note that the fact that there is no $\mathbf{x}$ so that $\mathbf{cx} < 0$ and $\mathbf{Ax} \geq \mathbf{0}$, it follows that $\mathbf{cx} \geq 0$; i.e., 0 is a lower bound for the linear programming problem P_F. At $\mathbf{x} = \mathbf{0}$, the objective achieves its lower bound and therefore this must be an optimal solution. Therefore P_F is bounded and feasible.

We can covert P_F to standard form through the following steps:

(1) Introduce two new vectors $\mathbf{y}$ and $\mathbf{z}$ with $\mathbf{y}, \mathbf{z} \geq \mathbf{0}$ and write $\mathbf{x} = \mathbf{y} - \mathbf{z}$ (since $\mathbf{x}$ is unrestricted).
(2) Append a vector of surplus variables $\mathbf{s}$ to the constraints.

This yields the new problem:

$$(8.19) \quad P_F' \begin{cases} \min \ \mathbf{cy} - \mathbf{cz} \\ s.t. \ \mathbf{Ay} - \mathbf{Az} - \mathbf{I}_m \mathbf{s} = \mathbf{0} \\ \quad \ \mathbf{y}, \mathbf{z}, \mathbf{s} \geq \mathbf{0} \end{cases}$$

Applying Theorems 5.1 and 5.5, we see we can obtain an optimal basic feasible solution for Problem P_F' in which the reduced costs for the variables are all negative (that is, $z_j - c_j \leq 0$ for $j = 1, \ldots, 2n + m$). Here we have n variables in vector $\mathbf{y}$, n variables in vector $\mathbf{z}$ and m variables in vector $\mathbf{s}$. Let $\mathbf{B} \in \mathbb{R}^{m \times m}$ be the basis matrix at this optimal feasible solution with basic cost vector $\mathbf{c_B}$. Let $\mathbf{w} = \mathbf{c_B} \mathbf{B}^{-1}$ (as it was defined for the revised simplex algorithm).

Consider the columns of the simplex tableau corresponding to a variable x_k (in our original $\mathbf{x}$ vector). The variable $x_k = y_k - z_k$. Thus, these two columns are additive inverses. That is, the column for y_k will be $\mathbf{B}^{-1} \mathbf{A}_{\cdot k}$, while the column for z_k will be $\mathbf{B}^{-1}(-\mathbf{A}_{\cdot k}) = -\mathbf{B}^{-1} \mathbf{A}_{\cdot k}$. Furthermore, the objective function coefficient will be precisely opposite as well.

Thus the fact that $z_j - c_j \leq 0$ for all variables implies that:

$$\mathbf{w}\mathbf{A}_{\cdot k} - c_k \leq 0 \textbf{ and}$$
$$-\mathbf{w}\mathbf{A}_{\cdot k} + c_k \leq 0 \textbf{ and}$$

That is, we obtain

(8.20) $\mathbf{w}\mathbf{A} = \mathbf{c}$

since this holds for all columns of $\mathbf{A}$.

    Consider the surplus variable s_k. Surplus variables have zero as their coefficient in the objective function. Further, their simplex tableau column is simply $\mathbf{B}^{-1}(-\mathbf{e}_k) = -\mathbf{B}^{-1}\mathbf{e}_k$. The fact that the reduced cost of this variable is non-positive implies that:

(8.21) $\mathbf{w}(-\mathbf{e}_k) - 0 = -\mathbf{w}\mathbf{e}_k \leq 0$

Since this holds for all surplus variable columns, we see that $-\mathbf{w} \leq \mathbf{0}$ which implies $\mathbf{w} \geq \mathbf{0}$. Thus, the optimal basic feasible solution to Problem P'_F must yield a vector $\mathbf{w}$ that solves System 2.

    Lastly, the fact that if System 2 does not have a solution, then System 1 does follows from contrapositive on the previous fact we just proved. $\square$

    EXERCISE 59. Suppose we have two statements A and B so that:

    $A \equiv$ System 1 has a solution.

    $B \equiv$ System 2 has a solution.

Our proof showed explicitly that NOT $A \implies B$. Recall that contrapositive is the logical rule that asserts that:

(8.22) $X \implies Y \equiv \text{NOT } Y \implies \text{NOT } X$

Use contrapositive to prove explicitly that if System 2 has no solution, then System 1 must have a solution. [Hint: NOT NOT $X \equiv X$.]

    2.1. Geometry of Farkas' Lemma. Farkas' Lemma has a pleasant geometric interpretation[1]. Consider System 2: namely:

    $\mathbf{w}\mathbf{A} = \mathbf{c}$ and $\mathbf{w} \geq \mathbf{0}$

Geometrically, this states that $\mathbf{c}$ is inside the positive cone generated by the *rows of* $\mathbf{A}$. That is, let $\mathbf{w} = (w_1, \ldots, w_m)$. Then we have:

(8.23) $w_1 \mathbf{A}_{1\cdot} + \cdots + w_m \mathbf{A}_{m\cdot}$

and $w_i \geq 0$ for $i = 1, \ldots, m$. Thus $\mathbf{c}$ is a positive combination of the rows of $\mathbf{A}$. This is illustrated in Figure 8.1. On the other hand, suppose System 1 has a solution. Then let $\mathbf{y} = -\mathbf{x}$. System 1 states that $\mathbf{A}\mathbf{y} \leq \mathbf{0}$ and $\mathbf{c}\mathbf{y} > 0$. That means that each row of $\mathbf{A}$ (as a vector) must be at a right angle or obtuse to $\mathbf{y}$. (Since $\mathbf{A}_i.\mathbf{x} \geq \mathbf{0}$.) Further, we know that the vector $\mathbf{y}$ must be acute with respect to the vector $\mathbf{c}$. This means that System 1 has a solution only if the vector $\mathbf{c}$ is not in the positive cone of the rows of $\mathbf{A}$ or equivalently the intersection of the open half-space $\{\mathbf{y} : \mathbf{c}\mathbf{y} > 0\}$ and the set of vectors $\{\mathbf{y} : \mathbf{A}_i.\mathbf{y} \leq \mathbf{0}, i = 1, \ldots m\}$ is

---

[1]Thanks to Akinwale Akinbiyi for pointing out a typo in this discussion.
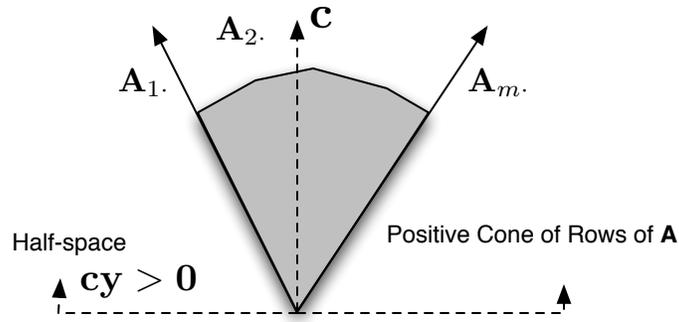
Figure 8.1. System 2 has a solution if (and only if) the vector **c** is contained inside the positive cone constructed from the rows of **A**.

non-empty. This set is the cone of vectors perpendicular to the rows of **A**. This is illustrated in Figure 8.2
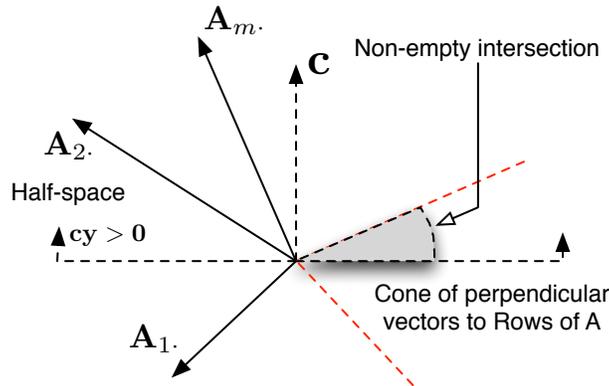
Figure 8.2. System 1 has a solution if (and only if) the vector **c** is not contained inside the positive cone constructed from the rows of **A**.

EXAMPLE 8.4. Consider the matrix:
$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
and the vector $\mathbf{c} = \begin{bmatrix} 1 & 2 \end{bmatrix}$. Then clearly, we can see that the vector $\mathbf{w} = \begin{bmatrix} 1 & 2 \end{bmatrix}$ will satisfy System 2 of Farkas' Lemma, since $\mathbf{w} \geq \mathbf{0}$ and $\mathbf{wA} = \mathbf{c}$.

Contrast this with $\mathbf{c}' = \begin{bmatrix} 1 & -1 \end{bmatrix}$. In this case, we can choose $\mathbf{x} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. Then $\mathbf{Ax} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T \geq \mathbf{0}$ and $\mathbf{c}'\mathbf{x} = -1$. Thus **x** satisfies System 1 of Farkas' Lemma.

These two facts are illustrated in Figure 8.3. Here, we see that **c** is inside the positive cone formed by the rows of **A**, while $\mathbf{c}'$ is not.

EXERCISE 60. Consider the following matrix:
$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$
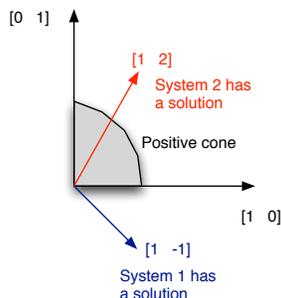
Figure 8.3. An example of Farkas' Lemma: The vector $\mathbf{c}$ is inside the positive cone formed by the rows of $\mathbf{A}$, but $\mathbf{c}'$ is not.

and the vector $\mathbf{c} = \begin{bmatrix} 1 & 2 \end{bmatrix}$. For this matrix and this vector, does System 1 have a solution or does System 2 have a solution? [Hint: Draw a picture illustrating the positive cone formed by the rows of $\mathbf{A}$. Draw in $\mathbf{c}$. Is $\mathbf{c}$ in the cone or not?]

2.2. Theorems of the Alternative. Farkas' lemma can be manipulated in many ways to produce several equivalent statements. The collection of all such theorems are called *Theorems of the Alternative* and are used *extensively* in optimization theory in proving optimality conditions. We state two that will be useful to us.

COROLLARY 8.5. *Let $\mathbf{A} \in \mathbb{R}^{k \times n}$ and $\mathbf{E} \in \mathbb{R}^{l \times n}$. Let $\mathbf{c} \in \mathbb{R}^n$ be a row vector. Suppose $\mathbf{d} \in \mathbb{R}^n$ is a column vector and $\mathbf{w} \in \mathbb{R}^k$ is a row vector and $\mathbf{v} \in \mathbf{R}^l$ is a row vector. Let:*

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} \\ \mathbf{E} \end{bmatrix}$$

and

$$\mathbf{u} = \begin{bmatrix} \mathbf{w} & \mathbf{v} \end{bmatrix}$$

Then exactly one of the following systems has a solution:

(1) $\mathbf{Md} \leq \mathbf{0}$ *and* $\mathbf{cd} > 0$ *or*

(2) $\mathbf{uM} = \mathbf{c}$ *and* $\mathbf{u} \geq \mathbf{0}$

PROOF. Let $\mathbf{x} = -\mathbf{d}$. Then $\mathbf{Md} \leq \mathbf{0}$ implies $\mathbf{Mx} \geq \mathbf{0}$ and $\mathbf{cd} > 0$ implies $\mathbf{cx} < 0$. This converts System 1 to the System 1 of Farkas' Lemma. System 2 is already in the form found in Farkas' Lemma. This completes the proof. □

EXERCISE 61. Prove the following corollary to Farkas' Lemma:

COROLLARY 8.6. *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{c} \in \mathbb{R}^n$ be a row vector. Suppose $\mathbf{d} \in \mathbb{R}^n$ is a column vector and $\mathbf{w} \in \mathbb{R}^m$ is a row vector and $\mathbf{v} \in \mathbf{R}^n$ is a row vector. Then exactly one of the following systems of inequalities has a solution:*

(1) $\mathbf{Ad} \leq \mathbf{0}$, $\mathbf{d} \geq \mathbf{0}$ *and* $\mathbf{cd} > 0$ *or*

(2) $\mathbf{wA} - \mathbf{v} = \mathbf{c}$ *and* $\mathbf{w}, \mathbf{v} \geq \mathbf{0}$

[Hint: Write System 2 from this corollary as $\mathbf{wA} - \mathbf{I}_n\mathbf{v} = \mathbf{c}$ and then re-write the system with an augmented vector $\begin{bmatrix} \mathbf{w} & \mathbf{v} \end{bmatrix}$ with an appropriate augmented matrix. Let $\mathbf{M}$ be the augmented matrix you identified. Now write System 1 from Farkas' Lemma using $\mathbf{M}$ and $\mathbf{x}$. Let $\mathbf{d} = -\mathbf{x}$ and expand System 1 until you obtain System 1 for this problem.]

3. The Karush-Kuhn-Tucker Conditions

THEOREM 8.7. *Consider the linear programming problem:*

$$(8.24) \quad P \begin{cases} \max \ \mathbf{cx} \\ s.t. \ \mathbf{Ax} \leq \mathbf{b} \\ \quad \mathbf{x} \geq \mathbf{0} \end{cases}$$

with $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ *and (row vector)* $\mathbf{c} \in \mathbb{R}^n$. *Then* $\mathbf{x}^* \in \mathbb{R}^n$ *is an optimal solution*[2] *to P if and only if there exists (row) vectors* $\mathbf{w}^* \in \mathbb{R}^m$ *and* $\mathbf{v}^* \in \mathbb{R}^n$ *so that:*

$$(8.25) \qquad \textit{Primal Feasibility} \begin{cases} \mathbf{Ax}^* \leq \mathbf{b} \\ \quad \mathbf{x}^* \geq \mathbf{0} \end{cases}$$

$$(8.26) \qquad \textit{Dual Feasibility} \begin{cases} \mathbf{w}^*\mathbf{A} - \mathbf{v}^* = \mathbf{c} \\ \qquad \mathbf{w}^* \geq \mathbf{0} \\ \qquad \mathbf{v}^* \geq \mathbf{0} \end{cases}$$

$$(8.27) \quad \textit{Complementary Slackness} \begin{cases} \mathbf{w}^* \left(\mathbf{Ax}^* - \mathbf{b} \right) = 0 \\ \qquad\qquad \mathbf{v}^*\mathbf{x}^* = 0 \end{cases}$$

REMARK 8.8. The vectors $\mathbf{w}^*$ and $\mathbf{v}^*$ are sometimes called *dual variables* for reasons that will be clear in the next chapter. They are also sometimes called *Lagrange Multipliers*. You may have encountered Lagrange Multipliers in your Math 230 or Math 231 class. These are the same kind of variables except applied to linear optimization problems. There is one element in the dual variable vector $\mathbf{w}^*$ for each constraint of the form $\mathbf{Ax} \leq \mathbf{b}$ and one element in the dual variable vector $\mathbf{v}^*$ for each constraint of the form $\mathbf{x} \geq \mathbf{0}$.

PROOF. Suppose that $\mathbf{x}^*$ is an optimal solution to Problem P. Consider only the *binding constraints* at $\mathbf{x}^*$. For simplicity, write the constraints $\mathbf{x} \geq \mathbf{0}$ as $-\mathbf{x} \leq \mathbf{0}$. Then we can form a new system of equations of the form:

$$(8.28) \quad \begin{bmatrix} \mathbf{A}_E \\ \mathbf{E} \end{bmatrix} \mathbf{x} = \mathbf{b}_E$$

where $\mathbf{E}$ is a matrix of negative identity matrix rows corresponding to the variables x_k that are equal to zero. That is, if $x_k = 0$, then $-x_k = 0$ and the negative identity matrix row $-\mathbf{e}_k^T$ will appear in $\mathbf{E}$ where $-\mathbf{e}_k^T \in \mathbb{R}^{1 \times n}$. Here $\mathbf{b_E}$ are the right hand sides of the binding constraints. Let:

$$\mathbf{M} = \begin{bmatrix} \mathbf{A}_E \\ \mathbf{E} \end{bmatrix}$$

The fact that $\mathbf{x}^*$ is optimal implies that there is no improving direction $\mathbf{d}$ at the point $\mathbf{x}^*$. That is, there is no $\mathbf{d}$ so that $\mathbf{Md} \leq \mathbf{0}$ and $\mathbf{c}^T\mathbf{d} > 0$. Otherwise, by moving in this direction we could find a new point $\hat{\mathbf{x}} = \mathbf{x}^* + \lambda\mathbf{d}$ (with λ sufficiently small) so that:

$$\mathbf{M}\hat{\mathbf{x}} = \mathbf{Mx}^* + \lambda\mathbf{Md} \leq \mathbf{b}_E$$

---

[2]Thanks to Rich Benjamin for pointing out the fact I was missing "...is an optimal solution..."

The (negative) identity rows in $\mathbf{E}$ ensure that $\hat{\mathbf{x}} \geq \mathbf{0}$. It follows that:

$$\mathbf{c}\hat{\mathbf{x}} = \mathbf{c}^T \left(\mathbf{x}^* + \lambda\mathbf{d}\right) = \mathbf{c}\mathbf{x}^* + \lambda\mathbf{c}\mathbf{d} > \mathbf{c}\mathbf{x}^*$$

That is, this point is both feasible and has a larger objective function value than $\mathbf{x}^*$.

We can now apply Corollary 8.5 to show that there are vectors $\mathbf{w}$ and $\mathbf{v}$ so that:

$$(8.29) \quad \begin{bmatrix} \mathbf{w} & \mathbf{v} \end{bmatrix} \begin{bmatrix} \mathbf{A}_E \\ \mathbf{E} \end{bmatrix} = \mathbf{c}$$

$$(8.30) \qquad\qquad \mathbf{w} \geq \mathbf{0}$$

$$(8.31) \qquad\qquad \mathbf{v} \geq \mathbf{0}$$

Let I be the indices of the rows of $\mathbf{A}$ making up $\mathbf{A}_E$ and let J be the indices of the variables that are zero (i.e., binding in $\mathbf{x} \geq \mathbf{0}$). Then we can re-write Equation 8.29 as:

$$(8.32) \quad \sum_{i \in I} w_i \mathbf{A}_{i\cdot} - \sum_{j \in J} v_j \mathbf{e}_j = \mathbf{c}$$

The vector $\mathbf{w}$ has dimension equal to the number of binding constraints of the form $\mathbf{A}_{i\cdot}\mathbf{x} = \mathbf{b}$ while the vector $\mathbf{v}$ has dimension equal to the number of binding constraints of the form $\mathbf{x} \geq \mathbf{0}$. We can extend $\mathbf{w}$ to $\mathbf{w}^*$ by adding 0 elements for the constraints where $\mathbf{A}_{i\cdot}\mathbf{x} < \mathbf{b}_i$. Similarly we can extend $\mathbf{v}$ to $\mathbf{v}^*$ by adding 0 elements for the constraints where $x_j > 0$. The result is that:

$$(8.33) \quad \mathbf{w}^* \left(\mathbf{A}\mathbf{x}^* - \mathbf{b}\right) = 0$$

$$(8.34) \qquad\qquad \mathbf{v}^* \mathbf{x}^* = 0$$

In doing this we maintain $\mathbf{w}^*, \mathbf{v}^* \geq \mathbf{0}$ and simultaneously guarantee that $\mathbf{w}^*\mathbf{A} - \mathbf{v}^* = \mathbf{c}$.

To prove the converse we assume that the KKT conditions hold and we are given vectors $\mathbf{x}^*$, $\mathbf{w}^*$ and $\mathbf{v}^*$. We will show that $\mathbf{x}^*$ solves the linear programming problem. By dual feasibility, we know that Equation 8.32 holds with $\mathbf{w} \geq \mathbf{0}$ and $\mathbf{v} \geq \mathbf{0}$ defined as before and the given point $\mathbf{x}^*$. Let $\mathbf{x}$ be an alternative point. We can multiply both side of Equation 8.32 by $(\mathbf{x}^* - \mathbf{x})$. This leads to:

$$(8.35) \quad \left\{ \sum_{i \in I} w_i \mathbf{A}_{i\cdot}\mathbf{x}^* - \sum_{j \in J} v_j \mathbf{e}_j \mathbf{x}^* \right\} - \left\{ \sum_{i \in I} w_i \mathbf{A}_{i\cdot}\mathbf{x} - \sum_{j \in J} v_j \mathbf{e}_j \mathbf{x}^* \right\} = \mathbf{c}\mathbf{x}^* - \mathbf{c}\mathbf{x}$$

We know that $\mathbf{A}_{i\cdot}\mathbf{x}^* = \mathbf{b}_i$ for $i \in I$ and that $x_j^* = 0$ for $j \in J$. We can use this to simplify Equation 8.35:

$$(8.36) \quad \sum_{i \in I} w_i \left(\mathbf{b}_i - \mathbf{A}_{i\cdot}\mathbf{x}\right) + \sum_{j \in J} v_j \mathbf{e}_j \mathbf{x} = \mathbf{c}\mathbf{x}^* - \mathbf{c}\mathbf{x}$$

The left hand side must be non-negative, since $\mathbf{w} \geq \mathbf{0}$ and $\mathbf{v} \geq \mathbf{0}$, $\mathbf{b}_i - \mathbf{A}_{i\cdot}\mathbf{x} \geq 0$ for all i, and $\mathbf{x} \geq \mathbf{0}$ and thus it follows that $\mathbf{x}^*$ must be an optimal point since $\mathbf{c}\mathbf{x}^* - \mathbf{c}\mathbf{x} \geq 0$. This completes the proof. $\qquad\square$

REMARK 8.9. The expressions:

$$(8.37) \qquad \text{Primal Feasibility} \begin{cases} \mathbf{Ax^*} \leq \mathbf{b} \\ \mathbf{x^*} \geq \mathbf{0} \end{cases}$$

$$(8.38) \qquad \text{Dual Feasibility} \begin{cases} \mathbf{w^*A} - \mathbf{v^*} = \mathbf{c} \\ \mathbf{w^*} \geq \mathbf{0} \\ \mathbf{v^*} \geq \mathbf{0} \end{cases}$$

$$(8.39) \quad \text{Complementary Slackness} \begin{cases} \mathbf{w^*}\left(\mathbf{Ax^*} - \mathbf{b}\right) = 0 \\ \mathbf{v^*x^*} = 0 \end{cases}$$

are called the *Karush-Kuhn-Tucker (KKT) conditions* of optimality. Note there is one element in $\mathbf{w}$ for each row of the constraints $\mathbf{Ax} \leq \mathbf{b}$ and one element in the vector $\mathbf{v}$ for each constraint of the form $\mathbf{x} \geq \mathbf{0}$. The vectors $\mathbf{w}$ and $\mathbf{v}$ are sometimes called *dual variables* and sometimes called Lagrange Multipliers.

We can think of dual feasibility as expressing the following interesting fact: at optimality, the gradient of the objective function $\mathbf{c}$ can be expressed as a positive combination of the gradients of the binding constraints written as less-than-or-equal-to inequalities. That is, the gradient of the constraint $\mathbf{A}_i.\mathbf{x} \leq \mathbf{b}_i$ is $\mathbf{A}_i.$ and the gradient of the constraint $-x_j \leq 0$ is $-\mathbf{e}_j$. More specifically, the vector $\mathbf{c}$ is in the *cone* generated by the binding constraints at optimality.

EXAMPLE 8.10. Consider the Toy Maker Problem (Equation 2.8) with Dual Variables (Lagrange Multipliers) listed next to their corresponding constraints:

$$\begin{cases} \max \ z(x_1, x_2) = 7x_1 + 6x_2 & \textbf{Dual Variable} \\ \text{s.t. } 3x_1 + x_2 \leq 120 & (w_1) \\ \quad x_1 + 2x_2 \leq 160 & (w_1) \\ \quad x_1 \leq 35 & (w_3) \\ \quad x_1 \geq 0 & (v_1) \\ \quad x_2 \geq 0 & (v_2) \end{cases}$$

In this problem we have:

$$\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 1 & 0 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 120 \\ 160 \\ 35 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 7 & 6 \end{bmatrix}$$

Then the KKT conditions can be written as:

$$\text{Primal Feasibility} \begin{cases} \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 120 \\ 160 \\ 35 \end{bmatrix} \\ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{cases}$$

$$\text{Dual Feasibility} \begin{cases} \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 7 & 6 \end{bmatrix} \\ \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \geq \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} v_1 & v_2 \end{bmatrix} \geq \begin{bmatrix} 0 & 0 \end{bmatrix} \end{cases}$$

$$\text{Complementary Slackness} \begin{cases} \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \left(\begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 120 \\ 160 \\ 35 \end{bmatrix} \right) \\ \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix} = 0 \end{cases}$$

Recall that at optimality, we had $x_1 = 16$ and $x_2 = 72$. The binding constraints in this case where

$$3x_1 + x_2 \leq 120$$
$$x_1 + 2x_2 \leq 160$$

To see this note that if $3(16) + 72 = 120$ and $16 + 2(72) = 160$. Then we should be able to express $\mathbf{c} = \begin{bmatrix} 7 & 6 \end{bmatrix}$ (the vector of coefficients of the objective function) as a positive combination of the gradients of the binding constraints:

$$\nabla(7x_1 + 6x_2) = \begin{bmatrix} 7 & 6 \end{bmatrix}$$
$$\nabla(3x_1 + x_2) = \begin{bmatrix} 3 & 1 \end{bmatrix}$$
$$\nabla(x_1 + 2x_2) = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

That is, we wish to solve the linear equation:

$$(8.40) \quad \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 6 \end{bmatrix}$$

Note, this is how Equation 8.32 looks when we apply it to this problem. The result is a system of equations:

$$3w_1 + w_2 = 7$$
$$w_1 + 2w_2 = 6$$

A solution to this system is $w_1 = \frac{8}{5}$ and $w_2 = \frac{11}{5}$. This fact is illustrated in Figure 8.4.

Figure 8.4 shows the gradient cone formed by the binding constraints at the optimal point for the toy maker problem. Since $x_1, x_2 > 0$, we must have $v_1 = v_2 = 0$. Moreover, since $x_1 < 35$, we know that $x_1 \leq 35$ is not a binding constraint and thus its dual variable w_3 is also zero. This leads to the conclusion:

$$\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} 16 \\ 72 \end{bmatrix} \quad \begin{bmatrix} w_1^* & w_2^* & w_3^* \end{bmatrix} = \begin{bmatrix} 8/5 & 11/5 & 0 \end{bmatrix} \quad \begin{bmatrix} v_1^* & v_2^* \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

and the KKT conditions are satisfied.
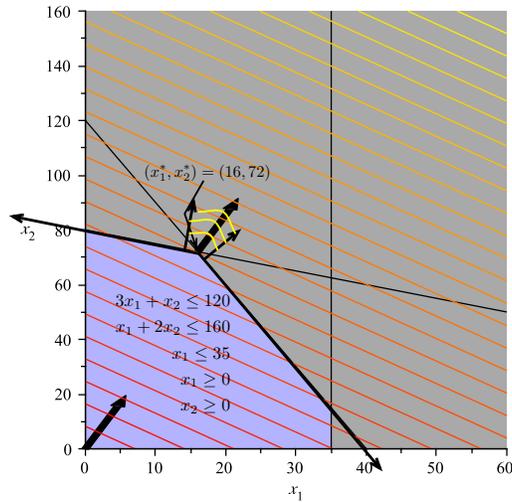
Figure 8.4. The Gradient Cone: At optimality, the cost vector **c** is obtuse with respect to the directions formed by the binding constraints. It is also contained inside the cone of the gradients of the binding constraints, which we will discuss at length later.

EXERCISE 62. Consider the problem:

$$\max \ x_1 + x_2$$
$$s.t. \ 2x_1 + x_2 \le 4$$
$$x_1 + 2x_2 \le 6$$
$$x_1, x_2 \ge 0$$

Write the KKT conditions for an optimal point for this problem. (You will have a vector $\mathbf{w} = [w_1 \ \ w_2]$ and a vector $\mathbf{v} = [v_1 \ \ v_2]$).

Draw the feasible region of the problem. At the optimal point you identified in Exercise 58, identify the binding constraints and draw their gradients. Show that the objective function is in the positive cone of the gradients of the binding constraints at this point. (Specifically find **w** and **v**.)

The Karush-Kuhn-Tucker Conditions for an Equality Problem. The KKT conditions can be modified to deal with problems in which we have equality constraints (i.e., $\mathbf{Ax} = \mathbf{b}$).

COROLLARY 8.11. *Consider the linear programming problem:*

$$(8.41) \quad P \begin{cases} \max \ \mathbf{cx} \\ s.t. \ \mathbf{Ax} = \mathbf{b} \\ \quad \mathbf{x} \ge \mathbf{0} \end{cases}$$

with $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ and (row vector) $\mathbf{c} \in \mathbb{R}^n$. Then $\mathbf{x}^* \in \mathbb{R}^n$ if and only if there exists (row) vectors $\mathbf{w}^* \in \mathbb{R}^m$ and $\mathbf{v}^* \in \mathbb{R}^n$ so that:

(8.42) $\qquad$ *Primal Feasibility* $\begin{cases} \mathbf{A}\mathbf{x}^* = \mathbf{b} \\ \quad \mathbf{x}^* \geq \mathbf{0} \end{cases}$

(8.43) $\qquad$ *Dual Feasibility* $\begin{cases} \mathbf{w}^*\mathbf{A} - \mathbf{v}^* = \mathbf{c} \\ \mathbf{w}^* \text{ unrestricted} \\ \qquad \quad \mathbf{v}^* \geq \mathbf{0} \end{cases}$

(8.44) $\quad$ *Complementary Slackness* $\{\mathbf{v}^*\mathbf{x}^* = 0$

PROOF. Replace the constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$ with the equivalent constraints:

(8.45) $\qquad \mathbf{A}\mathbf{x} \leq \mathbf{b}$

(8.46) $\quad -\mathbf{A}\mathbf{x} \leq -\mathbf{b}$

Let $\mathbf{w}_1$ be the vector corresponding to the rows in $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{w}_2$ be the vector corresponding to the rows in $-\mathbf{A}\mathbf{x} \leq -\mathbf{b}$. Then the KKT conditions are:

(8.47) $\qquad$ Primal Feasibility $\begin{cases} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ \quad -\mathbf{A}\mathbf{x} \leq -\mathbf{b} \\ \mathbf{x}^* \geq \mathbf{0} \end{cases}$

(8.48) $\qquad$ Dual Feasibility $\begin{cases} \mathbf{w}_1^*\mathbf{A} - \mathbf{w}_2\mathbf{A} - \mathbf{v}^* = \mathbf{c} \\ \qquad\qquad\qquad \mathbf{w}_1^* \geq \mathbf{0} \\ \qquad\qquad\qquad \mathbf{w}_2^* \geq \mathbf{0} \\ \qquad\qquad\qquad \mathbf{v}^* \geq \mathbf{0} \end{cases}$

(8.49) $\quad$ Complementary Slackness $\begin{cases} \mathbf{w}_1^*\left(\mathbf{A}\mathbf{x}^* - \mathbf{b}\right) = 0 \\ \mathbf{w}_2^*\left(\mathbf{b} - \mathbf{A}\mathbf{x}^*\right) = 0 \\ \qquad\qquad \mathbf{v}^*\mathbf{x}^* = 0 \end{cases}$

The fact that $\mathbf{A}\mathbf{x} = \mathbf{b}$ at optimality ensures we can re-write primal-feasibility as:

$\qquad$ Primal Feasibility $\begin{cases} \mathbf{A}\mathbf{x} = \mathbf{b} \\ \quad \mathbf{x}^* \geq \mathbf{0} \end{cases}$

Furthermore, since $\mathbf{A}\mathbf{x} = \mathbf{b}$ we know that complementary slackness of

$\qquad \mathbf{w}_1^*\left(\mathbf{A}\mathbf{x}^* - \mathbf{b}\right) = 0$
$\qquad \mathbf{w}_2^*\left(\mathbf{b} - \mathbf{A}\mathbf{x}^*\right) = 0$

is always satisfied and thus, we only require $\mathbf{v}^*\mathbf{x}^* = 0$ as our complementary slackness condition.

Finally, let $\mathbf{w} = \mathbf{w}_1 - \mathbf{w}_2$. Then dual feasibility becomes:

(8.50) $\quad \mathbf{w}\mathbf{A} - \mathbf{v} = \mathbf{w}_1^*\mathbf{A} - \mathbf{w}_2\mathbf{A} - \mathbf{v}^* = \mathbf{c}$

Since $\mathbf{w}_1, \mathbf{w}_2 \geq \mathbf{0}$, we know that $\mathbf{w}$ is unrestricted in sign. Thus we have:

$$\text{(8.51)} \quad \text{Dual Feasibility} \begin{cases} \mathbf{wA} - \mathbf{v}^* = \mathbf{c} \\ \mathbf{w}^* \text{ unrestricted} \\ \mathbf{v}^* \geq \mathbf{0} \end{cases}$$

This completes the proof. $\square$

EXERCISE 63. Use the trick of converting a minimization problem to a maximization problem to identify the KKT conditions for the following problem:

$$\text{(8.52)} \quad \begin{array}{l} \min \ \mathbf{cx} \\ s.t. \ \mathbf{Ax} \geq \mathbf{b} \\ \quad \mathbf{x} \geq \mathbf{0} \end{array}$$

[Hint: Remember, $\mathbf{Ax} \geq \mathbf{b}$ is the same as writing $-\mathbf{Ax} \leq -\mathbf{b}$. Now use the KKT conditions for the maximization problem to find the KKT conditions for this problem.]

4. Relating the KKT Conditions to the Tableau

Consider a linear programming problem in Standard Form:

$$\text{(8.53)} \quad P \begin{cases} \max \ \mathbf{cx} \\ s.t. \ \mathbf{Ax} = \mathbf{b} \\ \quad \mathbf{x} \geq \mathbf{0} \end{cases}$$

with $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ and (row vector) $\mathbf{c} \in \mathbb{R}^n$.

The KKT conditions for a problem of this type assert that

$$\mathbf{wA} - \mathbf{v} = \mathbf{c}$$
$$\mathbf{vx} = 0$$

at an optimal point $\mathbf{x}$ for some vector $\mathbf{w}$ unrestricted in sign and $\mathbf{v} \geq \mathbf{0}$. (Note, for the sake of notational ease, we have dropped the * notation.)

Suppose at optimality we have a basis matrix $\mathbf{B}$ corresponding to a set of basic variables $\mathbf{x_B}$ and we simultaneously have non-basic variables $\mathbf{x_N}$. We may likewise divide $\mathbf{v}$ into $\mathbf{v_B}$ and $\mathbf{v_N}$.

Then we have:

$$\text{(8.54)} \quad \mathbf{wA} - \mathbf{v} = \mathbf{c} \implies \mathbf{w} \begin{bmatrix} \mathbf{B} & \mathbf{N} \end{bmatrix} - \begin{bmatrix} \mathbf{v_B} & \mathbf{v_N} \end{bmatrix} = \begin{bmatrix} \mathbf{c_B} & \mathbf{c_N} \end{bmatrix}$$

$$\text{(8.55)} \quad \mathbf{vx} = 0 \implies \begin{bmatrix} \mathbf{v_B} & \mathbf{v_N} \end{bmatrix} \begin{bmatrix} \mathbf{x_B} \\ \mathbf{x_N} \end{bmatrix} = 0$$

We can rewrite Expression 8.54 as:

$$\text{(8.56)} \quad \begin{bmatrix} \mathbf{wB} - \mathbf{v_B} & \mathbf{wN} - \mathbf{v_N} \end{bmatrix} = \begin{bmatrix} \mathbf{c_B} & \mathbf{c_N} \end{bmatrix}$$

This simplifies to:

$$\mathbf{wB} - \mathbf{v_B} = \mathbf{c_B}$$
$$\mathbf{wN} - \mathbf{v_N} = \mathbf{c_N}$$

Let $\mathbf{w} = \mathbf{c_B}\mathbf{B}^{-1}$. Then we see that:

$$(8.57) \quad \mathbf{wB} - \mathbf{v_B} = \mathbf{c_B} \implies \mathbf{c_B}\mathbf{B}^{-1}\mathbf{B} - \mathbf{v_B} = \mathbf{c_B} \implies \mathbf{c_B} - \mathbf{v_B} = \mathbf{c_B} \implies \mathbf{v_B} = \mathbf{0}$$

Since we know that $\mathbf{x_B} \geq \mathbf{0}$, we know that $\mathbf{v_B}$ should be equal to zero to ensure complementary slackness. Thus, this is consistent with the KKT conditions.

We further see that:

$$(8.58) \quad \mathbf{wN} - \mathbf{v_N} = \mathbf{c_N} \implies \mathbf{c_B}\mathbf{B}^{-1}\mathbf{N} - \mathbf{v_N} = \mathbf{c_N} \implies \mathbf{v_N} = \mathbf{c_B}\mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N}$$

Thus, the $\mathbf{v_N}$ are just the reduced costs of the non-basic variables. ($\mathbf{v_B}$ are the reduced costs of the basic variables.) Furthermore, dual feasibility requires that $\mathbf{v} \geq \mathbf{0}$. Thus we see that at optimality we require:

$$(8.59) \quad \mathbf{c_B}\mathbf{B}^{-1}\mathbf{N} - \mathbf{c_N} \geq \mathbf{0}$$

This is *precisely* the condition for optimality in the simplex tableau.

We now can see the following facts are true about the Simplex Method:

(1) At each iteration of the Simplex Method, primal feasibility is satisfied. This is ensured by the minimum ratio test and the fact that we start at a feasible point.
(2) At each iteration of the Simplex Method, complementary slackness is satisfied. After all, the vector $\mathbf{v}$ is just the reduced cost vector (Row 0) of the Simplex tableau. If a variable is basic x_j (and hence non-zero), then the its reduced cost $v_j = 0$. Otherwise, v_j may be non-zero.
(3) At each iteration of the Simplex Algorithm, we may violate dual feasibility because we may not have $\mathbf{v} \geq \mathbf{0}$. It is *only* at optimality that we achieve dual feasibility and satisfy the KKT conditions.

We can now prove the following theorem:

THEOREM 8.12. *Assuming an appropriate cycling prevention rule is used, the simplex algorithm converges in a finite number of iterations to an optimal solution to the linear programming problem.*

PROOF. Convergence is guaranteed by the proof of Theorem 7.10 in which we show that when the lexicographic minimum ratio test is used, then the simplex algorithm will always converge. Our work above shows that at optimality, the KKT conditions are satisfied because the termination criteria for the simplex algorithm are precisely the same as the criteria in the Karush-Kuhn-Tucker conditions. This completes the proof. □

EXAMPLE 8.13. Consider the following linear programming problem:

$$\begin{cases} \max \ z(x_1, x_2) = 3x_1 + 5x_2 \\ \ s.t. \ x_1 + 2x_2 \le 60 \quad (w_1) \\ \qquad x_1 + x_2 \le 40 \quad (w_2) \\ \qquad x_1 \ge 0 \quad (v_1) \\ \qquad x_2 \ge 0 \quad (v_2) \end{cases}$$

Note we have assigned dual variables corresponding to each constraint on the right-hand-side of the constraints. That is, dual variable w_1 corresponds to the constraint $x_1 + 2x_2 \le 60$. We can write this problem in standard form as:

$$\begin{cases} \max \ z(x_1, x_2) = 3x_1 + 5x_2 \\ \ s.t. \ x_1 + 2x_2 + s_1 = 60 \quad (w_1) \\ \qquad x_1 + x_2 + s_2 = 40 \quad (w_2) \\ \qquad x_1 \ge 0 \quad (v_1) \\ \qquad x_2 \ge 0 \quad (v_2) \\ \qquad s_1 \ge 0 \quad (v_3) \\ \qquad s_2 \ge 0 \quad (v_4) \end{cases}$$

Note we have added two new dual variables v_3 and v_4 for the non-negativity constraints on slack variables s_1 and s_2. Our dual variable vectors are: $\mathbf{w} = \begin{bmatrix} w_1 & w_2 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}$. We can construct an initial simplex tableau as:

$$\begin{array}{c} z \\ s_1 \\ s_2 \end{array} \left[\begin{array}{c|ccccc} z & x_1 & x_2 & s_1 & s_2 & RHS \\ \hline 1 & -3 & -5 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 60 \\ 0 & 1 & 1 & 0 & 1 & 40 \end{array} \right]$$

In this initial configuration, we note that $v_1 = -3$, $v_2 = -5$, $v_3 = 0$ and $v_4 = 0$. This is because s_1 and s_2 are basic variables. We also notice that complementary slackness is satisfied. That is at the current values of x_1, x_2, s_1 and s_2 we have:

$$\begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \end{bmatrix} = 0$$

Applying the Simplex Algorithm yields the final tableau:

$$\begin{array}{c} z \\ x_2 \\ x_1 \end{array} \left[\begin{array}{c|ccccc} z & x_1 & x_2 & s_1 & s_2 & RHS \\ \hline 1 & 0 & 0 & 2 & 1 & 160 \\ 0 & 0 & 1 & 1 & -1 & 20 \\ 0 & 1 & 0 & -1 & 2 & 20 \end{array} \right]$$

The optimal value for $\mathbf{v}$ is $\begin{bmatrix} 0 & 0 & 2 & 1 \end{bmatrix}$. Note $\mathbf{v} \ge \mathbf{0}$ as required. Further, complementary slackness is still maintained. Notice further that the current value of $\mathbf{B}^{-1}$ can be found in

the portion of the matrix where the identity matrix stood in the initial tableau. Thus we can compute $\mathbf{w}$ as:

$$\mathbf{w} = \mathbf{c_B}\mathbf{B}^{-1}$$

Since $\mathbf{c_B} = \begin{bmatrix} 5 & 3 \end{bmatrix}$ (since x_2 and x_1 the basic variables at optimality) we see that:

$$\mathbf{w} = \begin{bmatrix} 5 & 3 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 \end{bmatrix}$$

That is, $w_1 = 2$ and $w_2 = 1$.

Note that $\mathbf{w} \geq \mathbf{0}$. This is because $\mathbf{w}$ is also a dual variable vector for our original problem (not in standard form). The KKT conditions for a maximization problem in canonical form require $\mathbf{w} \geq \mathbf{0}$ (see Theorem 8.7). Thus, it makes sense that we have $\mathbf{w} \geq \mathbf{0}$. Note this does not always have to be the case if we do not begin with a problem in canonical form.

Last, we can see that the constraints:

$$x_1 + 2x_2 \leq 60$$
$$x_1 + x_2 \leq 40$$

are both binding at optimality (since s_1 and s_2 are both zero). This means we should be able to express $\mathbf{c} = \begin{bmatrix} 3 & 5 \end{bmatrix}^T$ as a positive combination of the gradients of the left-hand-sides of these constraints using $\mathbf{w}$. To see this, note that w_1 corresponds to $x_1 + 2x_2 \leq 60$ and w_2 to $x_1 + x_2 \leq 40$. We have:

$$\nabla(x_1 + 2x_2) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\nabla(x_1 + x_2) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Then:

$$w_1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + w_2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = (2) \begin{bmatrix} 1 \\ 2 \end{bmatrix} + (1) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

Thus, the objective function gradient is in the dual cone of the binding constraint. That is, it is a positive combination of the gradients of the left-hand-sides of the binding constraints at optimality. This is illustrated in Figure 8.5.

We can also verify that the KKT conditions hold for the problem in standard form. Naturally, complementary slackness and primal feasibility hold. To see that dual feasibility holds note that $\mathbf{v} = \begin{bmatrix} 0 & 0 & 2 & 1 \end{bmatrix} \geq \mathbf{0}$. Further:

$$\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 5 & 0 & 0 \end{bmatrix}$$

Here $\begin{bmatrix} 3 & 5 & 0 & 0 \end{bmatrix}$ is the objective function coefficient vector for the problem in Standard Form.

EXERCISE 64. Use a full simplex tableau to find the values of the Lagrange multipliers (dual variables) at optimality for the problem from Exercise 62. Confirm that complementary slackness holds at optimality. Lastly show that dual feasibility holds by showing that the
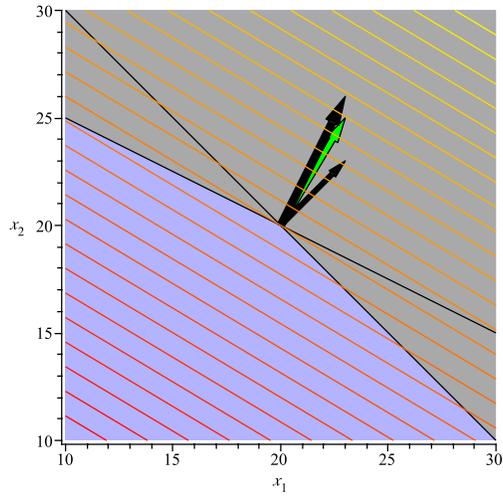
Figure 8.5. This figure illustrates the optimal point of the problem given in Example 8.13. Note that at optimality, the objective function gradient is in the dual cone of the binding constraint. That is, it is a positive combination of the gradients of the left-hand-sides of the binding constraints at optimality. The gradient of the objective function is shown in green.

gradient of the objective function ($\mathbf{c}$) is a positive combination of the gradients of the binding constraints at optimality. [Hint: Use the vector $\mathbf{w}$ you should have identified.]

Duality

In the last chapter, we explored the Karush-Kuhn-Tucker (KKT) conditions and identified constraints called the *dual feasibility* constraints. In this section, we show that to each linear programming problem (the primal problem) we may associate another linear programming problem (the dual linear programming problem). These two problems are closely related to each other and an analysis of the dual problem can provide deep insight into the primal problem.

1. The Dual Problem

Consider the linear programming problem

$$(9.1) \quad P \begin{cases} \max \ \mathbf{c}^T \mathbf{x} \\ s.t. \ \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ \quad \mathbf{x} \geq 0 \end{cases}$$

Then the dual problem for Problem P is:

$$(9.2) \quad D \begin{cases} \min \ \mathbf{wb} \\ s.t. \ \mathbf{wA} \geq \mathbf{c} \\ \quad \mathbf{w} \geq \mathbf{0} \end{cases}$$

REMARK 9.1. Let $\mathbf{v}$ be a vector of *surplus* variables. Then we can transform Problem D into standard form as:

$$(9.3) \quad D_S \begin{cases} \min \ \mathbf{wb} \\ s.t. \ \mathbf{wA} - \mathbf{v} = \mathbf{c} \\ \quad \mathbf{w} \geq \mathbf{0} \\ \quad \mathbf{v} \geq \mathbf{0} \end{cases}$$

Thus we already see an intimate relationship between duality and the KKT conditions. The feasible region of the dual problem (in standard form) is precisely the the dual feasibility constraints of the KKT conditions for the primal problem.

In this formulation, we see that we have assigned a dual variable w_i $(i = 1, \ldots, m)$ to each constraint in the system of equations $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ of the primal problem. Likewise dual variables $\mathbf{v}$ can be thought of as corresponding to the constraints in $\mathbf{x} \geq \mathbf{0}$.

LEMMA 9.2. *The dual of the dual problem is the primal problem.*

PROOF. Rewrite Problem D as:

$$(9.4) \quad \begin{cases} \max & -\mathbf{b}^T\mathbf{w}^T \\ s.t. & -\mathbf{A}^T\mathbf{w}^T \leq -\mathbf{c}^T \\ & \mathbf{w}^T \geq \mathbf{0} \end{cases}$$

Let $\boldsymbol{\beta} = -\mathbf{b}^T$, $\mathbf{G} = -\mathbf{A}^T$, $\mathbf{u} = \mathbf{w}^T$ and $\boldsymbol{\kappa} = -\mathbf{c}^T$. Then this new problem becomes:

$$(9.5) \quad \begin{cases} \max & \boldsymbol{\beta}\mathbf{u} \\ s.t. & \mathbf{G}\mathbf{u} \leq \boldsymbol{\kappa} \\ & \mathbf{u} \geq \mathbf{0} \end{cases}$$

Let $\mathbf{x}^T$ be the vector of dual variables (transposed) for this problem. We can formulate the dual problem as:

$$(9.6) \quad \begin{cases} \min & \mathbf{x}^T\boldsymbol{\kappa} \\ s.t. & \mathbf{x}^T\mathbf{G} \geq \boldsymbol{\beta} \\ & \mathbf{x}^T \geq \mathbf{0} \end{cases}$$

Expanding, this becomes:

$$(9.7) \quad \begin{cases} \min & -\mathbf{x}^T\mathbf{c}^T \\ s.t. & -\mathbf{x}^T\mathbf{A}^T \geq -\mathbf{b}^T \\ & \mathbf{x}^T \geq \mathbf{0} \end{cases}$$

This can be simplified to:

$$(9.8) \quad P\begin{cases} \max & \mathbf{c}^T\mathbf{x} \\ s.t. & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{cases}$$

as required. This completes the proof. $\qquad\square$

Lemma 9.2 shows that the notion of dual and primal can be exchanged and that it is simply a matter of perspective which problem is the dual problem and which is the primal problem. Likewise, by transforming problems into canonical form, we can develop dual problems for any linear programming problem.

The process of developing these formulations can be exceptionally tedious, as it requires enumeration of all the possible combinations of various linear and variable constraints. The following table summarizes the process of converting an arbitrary primal problem into its dual. This table can be found in Chapter 6 of [**BJS04**].

	MINIMIZATION PROBLEM	MAXIMIZATION PROBLEM	
VARIABLES	≥ 0 ≤ 0 UNRESTRICTED	$\leq$ $\geq$ $=$	CONSTRAINTS
CONSTRAINTS	$\geq$ $\leq$ $=$	≥ 0 ≤ 0 UNRESTRICTED	VARIABLES

Table 1. Table of Dual Conversions: To create a dual problem, assign a dual variable to each constraint of the form $\mathbf{Ax} \circ \mathbf{b}$, where $\circ$ represents a binary relation. Then use the table to determine the appropriate sign of the inequality in the dual problem as well as the nature of the dual variables.

EXAMPLE 9.3. Consider the problem of finding the dual problem for the Toy Maker Problem (Example 2.3) in standard form. The primal problem is:

$$\max \ 7x_1 + 6x_2$$
$$s.t. \ 3x_1 + x_2 + s_1 = 120 \qquad (w_1)$$
$$x_1 + 2x_2 + s_2 = 160 \qquad (w_2)$$
$$x_1 + s_3 = 35 \qquad (w_3)$$
$$x_1, x_2, s_1, s_2, s_3 \geq 0$$

Here we have placed dual variable names (w_1, w_2 and w_3) next to the constraints to which they correspond.

The primal problem variables in this case are all positive, so using Table 1 we know that the constraints of the dual problem will be greater-than-or-equal-to constraints. Likewise, we know that the dual variables will be unrestricted in sign since the primal problem constraints are all equality constraints.

The coefficient matrix is:

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Clearly we have:

$$\mathbf{c} = \begin{bmatrix} 7 & 6 & 0 & 0 & 0 \end{bmatrix}$$
$$\mathbf{b} = \begin{bmatrix} 120 \\ 160 \\ 35 \end{bmatrix}$$

Since $\mathbf{w} = [w_1 \ w_2 \ w_3]$, we know that $\mathbf{wA}$ will be:

$$\mathbf{wA} = \begin{bmatrix} 3w_1 + w_2 + w_3 & w_1 + 2w_2 & w_1 & w_2 & w_3 \end{bmatrix}$$

This vector will be related to **c** in the constraints of the dual problem. **Remember, in this case, all constraints are greater-than-or-equal-to**. Thus we see that the constraints of the dual problem are:

$$3w_1 + w_2 + w_3 \geq 7$$
$$w_1 + 2w_2 \geq 6$$
$$w_1 \geq 0$$
$$w_2 \geq 0$$
$$w_3 \geq 0$$

We also have the redundant set of constraints that tell us **w** is unrestricted because the primal problem had equality constraints. This *will always* happen in cases when you've introduced slack variables into a problem to put it in standard form. This should be clear from the definition of the dual problem for a maximization problem in canonical form.

Thus the whole dual problem becomes:

(9.9)
$$\min\ 120w_1 + 160w_2 + 35w_3$$
$$s.t.\ 3w_1 + w_2 + w_3 \geq 7$$
$$w_1 + 2w_2 \geq 6$$
$$w_1 \geq 0$$
$$w_2 \geq 0$$
$$w_3 \geq 0$$
$$\mathbf{w}\ \text{unrestricted}$$

Again, note that in reality, the constraints we derived from the $\mathbf{wA} \geq \mathbf{c}$ part of the dual problem make the constraints "**w** unrestricted" redundant, for in fact $\mathbf{w} \geq \mathbf{0}$ just as we would expect it to be if we'd found the dual of the Toy Maker problem given in canonical form.

EXERCISE 65. Identify the dual problem for:

$$\max\ x_1 + x_2$$
$$s.t.\ 2x_1 + x_2 \geq 4$$
$$x_1 + 2x_2 \leq 6$$
$$x_1, x_2 \geq 0$$

EXERCISE 66. Use the table or the definition of duality to determine the dual for the problem:

(9.10)
$$\begin{cases} \min\ \mathbf{cx} \\ s.t.\ \mathbf{Ax} \leq \mathbf{b} \\ \quad \mathbf{x} \geq 0 \end{cases}$$

Compare it to the KKT conditions you derived in Exercise 63.

2. Weak Duality

There is a deep relationship between the objective function value, feasibility and bound-edness of the primal problem and the dual problem. We will explore these relationships in the following lemmas.

LEMMA 9.4 (Weak Duality). *For the primal problem P and dual problem D let $\mathbf{x}$ and $\mathbf{w}$ be feasible solutions to Problem P and Problem D respectively. Then:*

$$(9.11) \quad \mathbf{wb} \geq \mathbf{cx}$$

PROOF. Primal feasibility ensures that:

$$\mathbf{Ax} \leq \mathbf{b}$$

Therefore, we have:

$$(9.12) \quad \mathbf{wAx} \leq \mathbf{wb}$$

Dual feasibility ensure that:

$$\mathbf{wA} \geq \mathbf{c}$$

Therefore we have:

$$(9.13) \quad \mathbf{wAx} \geq \mathbf{cx}$$

Combining Equations 9.12 and 9.13 yields Equation 9.11:

$$\mathbf{wb} \geq \mathbf{cx}$$

This completes the proof. $\qquad\square$

REMARK 9.5. Lemma 9.4 ensures that the optimal solution $\mathbf{w}^*$ for Problem D must provide an *upper bound* to Problem P, since for any feasible $\mathbf{x}$, we know that:

$$(9.14) \quad \mathbf{w}^*\mathbf{b} \geq \mathbf{cx}$$

Likewise, any optimal solution to Problem P provides a lower bound on solution D.

COROLLARY 9.6. *If Problem P is unbounded, then Problem D is infeasible. Likewise, if Problem D is unbounded, then Problem P is infeasible.*

PROOF. For any $\mathbf{x}$, feasible to Problem P we know that $\mathbf{wb} \geq \mathbf{cx}$ for any feasible $\mathbf{w}$. The fact that Problem P is unbounded implies that for any $V \in \mathbb{R}$ we can find an $\mathbf{x}$ feasible to Problem P that $\mathbf{cx} > V$. If $\mathbf{w}$ were feasible to Problem D, then we would have $\mathbf{wb} > V$ for any arbitrarily chosen V. There can be no finite vector $\mathbf{w}$ with this property and we conclude that Problem D must be infeasible.

The alternative case that when Problem D is unbounded, then Problem P is infeasible follows by reversing the roles of the problem. This completes the proof. $\qquad\square$

3. Strong Duality

LEMMA 9.7. *Problem D has an optimal solution $\mathbf{w}^* \in \mathbb{R}^m$ if and only if there exists vector $\mathbf{x}^* \in \mathbb{R}^n$ and $\mathbf{s}^* \in \mathbb{R}^m$ such that:*

$$(9.15) \qquad \text{Primal Feasibility} \begin{cases} \mathbf{w}^*\mathbf{A} \geq \mathbf{c} \\ \quad \mathbf{w}^* \geq 0 \end{cases}$$

$$(9.16) \qquad \text{Dual Feasibility} \begin{cases} \mathbf{A}\mathbf{x}^* + \mathbf{s}^* = \mathbf{b} \\ \qquad \mathbf{x}^* \geq 0 \\ \qquad \mathbf{s}^* \geq 0 \end{cases}$$

$$(9.17) \qquad \text{Complementary Slackness} \begin{cases} (\mathbf{w}^*\mathbf{A} - \mathbf{c})\,\mathbf{x}^* = 0 \\ \qquad\qquad \mathbf{w}^*\mathbf{s}^* = 0 \end{cases}$$

Furthermore, these KKT conditions are equivalent to the KKT conditions for the primal problem.

PROOF. Following the proof of Lemma 9.2, let $\boldsymbol{\beta} = -\mathbf{b}^T$, $\mathbf{G} = -\mathbf{A}^T$, $\mathbf{u} = \mathbf{w}^T$ and $\boldsymbol{\kappa} = -\mathbf{c}^T$. Then the dual problem can be rewritten as:

$$\begin{cases} \max\ \boldsymbol{\beta}\mathbf{u} \\ \quad s.t.\ \mathbf{G}\mathbf{u} \leq \boldsymbol{\kappa} \\ \qquad \mathbf{u} \geq \mathbf{0} \end{cases}$$

Let $\mathbf{x}^T \in \mathbb{R}^{1 \times n}$ and $s^T \in \mathbb{R}^{1 \times m}$ be the dual variables for this problem. Then applying Theorem 8.7, we obtain KKT conditions for this problem:

$$(9.18) \qquad \text{Primal Feasibility} \begin{cases} \mathbf{G}\mathbf{u}^* \leq \boldsymbol{\kappa} \\ \quad \mathbf{u}^* \geq \mathbf{0} \end{cases}$$

$$(9.19) \qquad \text{Dual Feasibility} \begin{cases} \mathbf{x}^{*T}\mathbf{G} - \mathbf{s}^{*T} = \boldsymbol{\beta} \\ \qquad\quad \mathbf{x}^{*T} \geq \mathbf{0} \\ \qquad\quad \mathbf{s}^{*T} \geq \mathbf{0} \end{cases}$$

$$(9.20) \qquad \text{Complementary Slackness} \begin{cases} \mathbf{x}^{*T}\left(\mathbf{G}\mathbf{u}^* - \boldsymbol{\kappa}\right) = 0 \\ \qquad\qquad \mathbf{s}^{*T}\mathbf{u}^* = 0 \end{cases}$$

We can rewrite:

$$\mathbf{G}\mathbf{u}^* \leq \boldsymbol{\kappa} \equiv -\mathbf{A}^T\mathbf{w}^T \leq -\mathbf{c}^T \equiv \mathbf{w}\mathbf{A} \geq \mathbf{c}$$
$$\mathbf{x}^{*T}\mathbf{G} - \mathbf{s}^{*T} = \boldsymbol{\beta} \equiv \mathbf{x}^{*T}(-\mathbf{A}^T) - \mathbf{s}^{*T} = -\mathbf{b}^T \equiv \mathbf{A}\mathbf{x}^* + \mathbf{s}^* = \mathbf{b}$$
$$\mathbf{x}^{*T}\left(\mathbf{G}\mathbf{u}^* - \boldsymbol{\kappa}\right) = 0 \equiv \mathbf{x}^{*T}\left((-\mathbf{A}^T)\mathbf{w}^{*T} - (-\mathbf{c}^T)\right) = 0 \equiv (\mathbf{w}^*\mathbf{A} - \mathbf{c})\mathbf{x}^* = 0$$
$$\mathbf{s}^{*T}\mathbf{u}^* = 0 \equiv \mathbf{s}^{*T}\mathbf{w}^{*T} = 0 \equiv \mathbf{w}^*\mathbf{s}^* = 0$$

Thus, we have shown that the KKT conditions for the dual problem are:

$$\text{Primal Feasibility} \begin{cases} \mathbf{w}^*\mathbf{A} \geq \mathbf{c} \\ \mathbf{w}^* \geq 0 \end{cases}$$

$$\text{Dual Feasibility} \begin{cases} \mathbf{A}\mathbf{x}^* + \mathbf{s}^* = \mathbf{b} \\ \mathbf{x}^* \geq 0 \\ \mathbf{s}^* \geq 0 \end{cases}$$

$$\text{Complementary Slackness} \begin{cases} (\mathbf{w}^*\mathbf{A} - \mathbf{c})\,\mathbf{x}^* = 0 \\ \mathbf{w}^*\mathbf{s}^* = 0 \end{cases}$$

To prove the equivalence to the KKT conditions for the primal problem, define:

(9.21) $\quad \mathbf{s}^* = \mathbf{b} - \mathbf{A}\mathbf{x}^*$

(9.22) $\quad \mathbf{v}^* = \mathbf{w}^*\mathbf{A} - \mathbf{c}$

That is, $\mathbf{s}^*$ is a vector slack variables for the primal problem P at optimality and $\mathbf{v}^*$ is a vector of surplus variables for the dual problem D at optimality. Recall the KKT conditions for the primal problem are:

$$\text{Primal Feasibility} \begin{cases} \mathbf{A}\mathbf{x}^* \leq \mathbf{b} \\ \mathbf{x}^* \geq \mathbf{0} \end{cases}$$

$$\text{Dual Feasibility} \begin{cases} \mathbf{w}^*\mathbf{A} - \mathbf{v}^* = \mathbf{c} \\ \mathbf{w}^* \geq \mathbf{0} \\ \mathbf{v}^* \geq \mathbf{0} \end{cases}$$

$$\text{Complementary Slackness} \begin{cases} \mathbf{w}^*\,(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = 0 \\ \mathbf{v}^*\mathbf{x}^* = 0 \end{cases}$$

We can rewrite these as:

(9.23) $\qquad \text{Primal Feasibility} \begin{cases} \mathbf{A}\mathbf{x}^* + \mathbf{s}^* = \mathbf{b} \\ \mathbf{x}^* \geq \mathbf{0} \end{cases}$

(9.24) $\qquad \text{Dual Feasibility} \begin{cases} \mathbf{w}^*\mathbf{A} - \mathbf{v}^* = \mathbf{c} \\ \mathbf{w}^* \geq \mathbf{0} \\ \mathbf{v}^* \geq \mathbf{0} \end{cases}$

(9.25) $\quad \text{Complementary Slackness} \begin{cases} \mathbf{w}^*\,(\mathbf{s}^*) = 0 \\ \mathbf{v}^*\mathbf{x}^* = 0 \end{cases}$

Where the complementary slackness condition $\mathbf{w}^*(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = 0$ is re-written as:

$$\mathbf{w}^*(\mathbf{A}\mathbf{x}^* - \mathbf{b}) = 0 \equiv \mathbf{w}^*(-(\mathbf{b} - \mathbf{A}\mathbf{x}^*)) = 0 \equiv \mathbf{w}^*(-\mathbf{s}^*) = 0 \equiv \mathbf{w}^*(\mathbf{s}^*)$$

Likewise beginning with the KKT conditions for the dual problem (Expression 9.15 - 9.17) we can write:

$$\text{Primal Feasibility} \begin{cases} \mathbf{w}^*\mathbf{A} - \mathbf{v}^* \geq \mathbf{c} \\ \mathbf{w}^* \geq 0 \end{cases}$$

$$\text{Dual Feasibility} \begin{cases} \mathbf{A}\mathbf{x}^* + \mathbf{s}^* = \mathbf{b} \\ \mathbf{x}^* \geq 0 \\ \mathbf{s}^* \geq 0 \end{cases}$$

$$\text{Complementary Slackness} \begin{cases} (\mathbf{v}^*)\mathbf{x}^* = 0 \\ \mathbf{w}^*\mathbf{s}^* = 0 \end{cases}$$

Here we substitute $\mathbf{v}^*$ for $\mathbf{w}^*\mathbf{A} - \mathbf{c}$ in the complementary slackness terms. Thus we can see that the KKT conditions for the primal and dual problems are equivalent. This completes the proof. $\square$

REMARK 9.8. Notice that the KKT conditions for the primal and dual problems are equivalent, but the dual feasibility conditions for the primal problem are identical to the primal feasibility conditions for the dual problem and vice-versa. Thus, two linear programming problems are dual to each other if they share KKT conditions with the primal and dual feasibility conditions swapped.

EXERCISE 67. Compute the dual problem for a canonical form minimization problem:

$$P \begin{cases} \min \ \mathbf{cx} \\ s.t. \ \mathbf{Ax} \geq \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

Find the KKT conditions for the dual problem you just identified. Use the result from Exercise 63 to show that KKT conditions for Problem P are identical to the KKT conditions for the dual problem you just found.

LEMMA 9.9 (Strong Duality). *There is a bounded optimal solution $\mathbf{x}^*$ for Problem P if and only if there is a bounded optimal solution $\mathbf{w}^*$ for Problem D. Furthermore, $\mathbf{cx}^* = \mathbf{w}^*\mathbf{b}$.*

PROOF. Suppose that there is a solution $\mathbf{x}^*$ for Problem P. Let $\mathbf{s}^* = \mathbf{b} - \mathbf{A}\mathbf{x}^*$. Clearly $\mathbf{s}^* \geq \mathbf{0}$.

By Theorem 8.7 there exists dual variables $\mathbf{w}^*$ and $\mathbf{v}^*$ satisfying dual feasibility and complementary slackness. Dual feasibility in the KKT conditions implies that:

$$(9.26) \quad \mathbf{v}^* = \mathbf{w}^*\mathbf{A} - \mathbf{c}$$

We also know that $\mathbf{w}^*, \mathbf{v}^* \geq \mathbf{0}$. Complementary Slackness (from Theorem 8.7) states that $\mathbf{v}^*\mathbf{x}^* = 0$. But $\mathbf{v}^*$ is defined above and we see that:

$$(9.27) \quad \mathbf{v}^*\mathbf{x}^* = 0 \implies (\mathbf{w}^*\mathbf{A} - \mathbf{c})\mathbf{x} = 0$$

Likewise, since we have $\mathbf{s}^* = \mathbf{b} - \mathbf{A}\mathbf{x}^*$. Complementary slackness assures us that $\mathbf{w}^*(\mathbf{b} - \mathbf{A}\mathbf{x}^*) = 0$. Thus we see that:

$$(9.28) \quad \mathbf{w}^*(\mathbf{b} - \mathbf{A}\mathbf{x}^*) = 0 \implies \mathbf{w}^*\mathbf{s}^* = 0$$

Thus it follows from Lemma 9.7 that $\mathbf{w}^*$ is an optimal solution to Problem D since it satisfies the KKT conditions. The fact that Problem P has an optimal solution when Problem D has an optimal solution can be proved in a similar manner starting with the KKT conditions given in Lemma 9.7 and applying the same reasoning as above.

Finally, at optimality, we know from Lemma 9.7 that:

$$(9.29) \quad (\mathbf{w}^*\mathbf{A} - \mathbf{c})\,\mathbf{x}^* = 0 \implies \mathbf{w}^*\mathbf{A}\mathbf{x}^* = \mathbf{c}\mathbf{x}^*$$

We also know from Theorem 8.7 that:

$$(9.30) \quad \mathbf{w}^*\,(\mathbf{b} - \mathbf{A}\mathbf{x}^*) = 0 \implies \mathbf{w}^*\mathbf{b} = \mathbf{w}^*\mathbf{A}\mathbf{x}^*$$

Therefore we have: $\mathbf{w}^*\mathbf{b} = \mathbf{c}\mathbf{x}^*$. This completes the proof. $\qquad\square$

COROLLARY 9.10. *If Problem P is infeasible, then Problem D is either unbounded or infeasible. If Problem D is infeasible, then either Problem P is unbounded or infeasible.*

PROOF. This result follows by contrapositive from Lemma 9.9. To see this, suppose that Problem P is infeasible. Then Problem P has no bounded optimal solution. Therefore, Problem D has no bounded optimal solution (by Lemma 9.9). If Problem D has no bounded optimal solution, then either Problem D is unbounded or it is infeasible. A symmetric argument on Problem D completes the proof of the Lemma. $\qquad\square$

EXERCISE 68. Consider the problem

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

(1) Show that this problem is infeasible.
(2) Compute its dual.
(3) Show that the dual is infeasible, thus illustrating Corollary 9.10.

The following theorem summarizes all of the results we have obtained in the last two sections.

THEOREM 9.11 (Strong Duality Theorem). *Consider Problem P and Problem D. Then exactly one of the following statements is true:*

(1) *Both Problem P and Problem D possess optimal solutions $\mathbf{x}^*$ and $\mathbf{w}^*$ respectively and $\mathbf{c}\mathbf{x}^* = \mathbf{w}^*\mathbf{b}$.*
(2) *Problem P is unbounded and Problem D is infeasible.*
(3) *Problem D is unbounded and Problem P is infeasible.*
(4) *Both problems are infeasible.*

4. Geometry of the Dual Problem

The geometry of the dual problem is, in essence, exactly the same the geometry of the primal problem insofar as they are both linear programming problems and thus their feasible

regions are polyhedral sets[1]. Certain examples have a very nice geometric visualization. Consider the problem:

$$(9.31) \quad \begin{cases} \max \; 6x_1 + 6x_2 \\ \text{s.t.} \; 3x_1 + 2x_2 \leq 6 \\ \qquad 2x_1 + 3x_2 \leq 6 \\ \qquad x_1, x_2 \geq 0 \end{cases}$$

In this problem we have:

$$\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 6 & 6 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}$$

Notice that $\mathbf{A}$ is a symmetric matrix and $\mathbf{c}^T = \mathbf{b}$ and the dual problem is:

$$(9.32) \quad \begin{cases} \min \; 6w_1 + 6w_2 \\ \text{s.t.} \; 3w_1 + 2w_2 \geq 6 \\ \qquad 2w_1 + 3w_2 \leq 6 \\ \qquad w_1, w_2 \geq 0 \end{cases}$$

This results in a geometry in which the dual feasible region is a reflection of the primal feasible region (ignoring non-negativity constraints). This is illustrated in Figure 9.1.
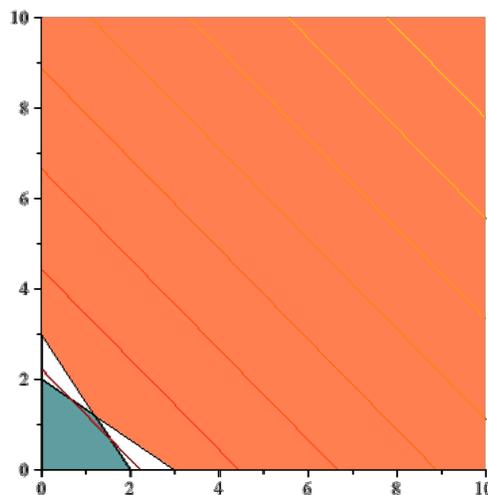
Figure 9.1. The dual feasible region in this problem is a mirror image (almost) of the primal feasible region. This occurs when the right-hand-side vector $\mathbf{b}$ is equal to the objective function coefficient column vector $\mathbf{c}^T$ and the matrix $\mathbf{A}$ is symmetric.

---

[1]Thanks for Michael Cline for suggesting this section.

We can also illustrate the process of solving this problem using the (revised) simplex algorithm. In doing so, we first convert Problem 9.33 to standard form:

$$(9.33) \quad \begin{cases} \max \ 6x_1 + 6x_2 \\ \text{s.t. } 3x_1 + 2x_2 + s_1 \quad = 6 \\ \qquad 2x_1 + 3x_2 \quad + s_2 = 6 \\ \qquad x_1, x_2 \geq 0 \end{cases}$$

This yields an initial tableau:

$$(9.34) \quad \begin{matrix} z \\ s_1 \\ s_2 \end{matrix} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 6 \\ 0 & 1 & 6 \end{bmatrix}$$

This is because our initial $\mathbf{c_B} = \begin{bmatrix} 0 & 0 \end{bmatrix}$ and thus $\mathbf{w} = \begin{bmatrix} 0 & 0 \end{bmatrix}$. This means at the start of the simplex algorithm, $w_1 = 0$ and $w_2 = 0$ and $x_1 = 0$ and $x_2 = 0$, so we begin at the origin, which is **in the primal feasible region** and **not in the dual feasible region**. If we iterate and choose x_1 as an entering variable, our updated tableau will be:

$$(9.35) \quad \begin{matrix} z \\ x_1 \\ s_2 \end{matrix} \begin{bmatrix} 2 & 0 & 12 \\ 1/3 & 0 & 2 \\ -2/3 & 1 & 2 \end{bmatrix}$$

Notice at this point, $x_1 = 2$, $x_2 = 0$ and $w_1 = 2$ and $w_2 = 0$. This point is again feasible for the primal problem but still infeasible for the dual. This step is illustrated in Figure 9.2. Entering x_2 yields the final tableau:

$$(9.36) \quad \begin{matrix} z \\ x_1 \\ x_2 \end{matrix} \begin{bmatrix} 6/5 & 6/5 & 72/5 \\ 3/5 & -2/5 & 6/5 \\ -2/5 & 3/5 & 6/5 \end{bmatrix}$$

At this final point, $x_1 = x_2 = w_1 = w_2 = \frac{6}{5}$, which is feasible to both problems. This step is also illustrated in Figure 9.2. We should note that this problem is atypical in that the primal and dual feasible regions share one common point. In more standard problems, the two feasible regions cannot be drawn in this convenient way, *but* the simplex process is the same. The simplex algorithm begins at a point in the primal feasible region with a corresponding dual vector that is *not* in the feasible region of the dual problem. As the simplex algorithm progresses, this dual vector approaches and finally enters the dual feasible region.

EXERCISE 69. Draw the dual feasible region for the following problem.

$$\max \ 3x_1 + 5x_2$$
$$\text{s.t. } x_1 + 2x_2 \leq 60$$
$$\qquad x_1 + x_2 \leq 40$$
$$\qquad x_1, x_2 \geq 0$$

Solve the problem using the revised simplex algorithm and trace the path of dual variables (the $\mathbf{w}$ vector) in your plot of the dual feasible region. Also trace the path of the primal vector $\mathbf{x}$ through the primal feasible region. [Hint: Be sure to draw the area around the dual
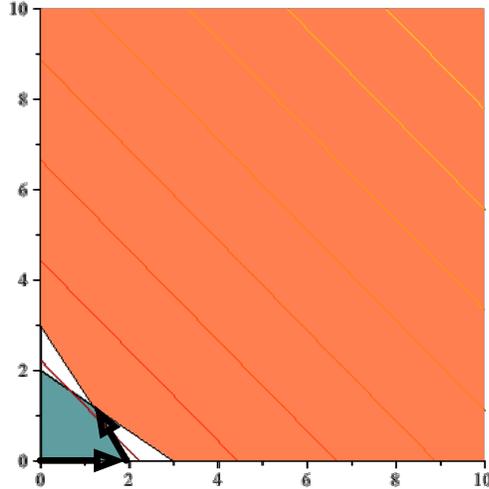
Figure 9.2. The simplex algorithm begins at a feasible point in the feasible region of the primal problem. In this case, this is also the same starting point in the dual problem, which is infeasible. The simplex algorithm moves through the feasible region of the primal problem towards a point in the dual feasible region. At the conclusion of the algorithm, the algorithm reaches the unique point that is both primal and dual feasible.

feasible region. Your dual vector $\mathbf{w}$ will not enter the feasible region until the last simplex pivot.]

5. Economic Interpretation of the Dual Problem

Consider again, the value of the objective function in terms of the values of the non-basic variables (Equation 5.11):

$$(9.37) \quad z = \mathbf{cx} = \mathbf{c_B B}^{-1}\mathbf{b} + \left(\mathbf{c_N} - \mathbf{c_B B}^{-1}\mathbf{N}\right)\mathbf{x}_N$$

Suppose we are at a non-degenerate optimal point. We've already observed that:

$$(9.38) \quad \frac{\partial z}{\partial x_j} = -(z_j - c_j) = c_j - \mathbf{c_B B}^{-1}\mathbf{A}_{.j}$$

We can rewrite all these equations in terms of our newly defined term:

$$(9.39) \quad \mathbf{w} = \mathbf{c_B B}^{-1}$$

to obtain:

$$(9.40) \quad z = \mathbf{wb} + (\mathbf{c_N} - \mathbf{wN})\mathbf{x}_N$$

Remember, $\mathbf{w}$ is the vector of dual variables corresponding to the constraints in our original problem P.

Suppose we fix the values of $\mathbf{x_N}$. Then we can see that the vector $\mathbf{w}$ has individual elements with the property that:

$$(9.41) \quad \frac{\partial z}{\partial b_i} = w_i$$

148

That is, the ith element of **w** represents the amount that the objective function value would change at optimality assuming we could modify the right-hand-side of the constraints. **Note, this result holds only in the absence of degeneracy, for reasons we will see in an example.**

Thus, we can think of w_i as the *shadow price* for resource i (the right-hand-side of the ith constraint). A shadow price is the *fair price* one would pay for an extra unit of resource i.

EXAMPLE 9.12. Consider a leather company that requires 1 square yard of leather to make a regular belt and a 1 square yard of leather to make a deluxe belt. If the leather company can use up to 40 square yards per week to construct belts, then one constraint it may have is:

$$x_1 + x_2 \leq 40$$

In the absence of degeneracy, the dual variable (say w_1) will tell the fair price we would pay for 1 extra yard of leather. Naturally, if this were not a binding constraint, then $w_1 = 0$ indicating that extra leather is worth *nothing* to us since we already have a surplus of leather.

To understand the economics of the situation, suppose that we a manufacturer is to produce products $1, \ldots, n$ and we produce $x_1, \ldots, x_n$ of each. If we can sell each product for a profit of $c_1, \ldots, c_n$, then we wish to find values for $x_1, \ldots, x_n$ to solve:

$$(9.42) \quad \max \ \sum_{j=1}^{n} c_j x_j$$

Simultaneously, suppose that m resources (leather, wood, time etc.) are used to make these n products and that a_{ij} units of resource i are used to manufacture product j. Then clearly our constraints will be:

$$(9.43) \quad a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i$$

where b_i is the amount of resource i available to the company. Suppose now that the company decides to sell off some of its resources (instead of manufacturing products). Suppose we sell each resource for a price w_i $(i = 1, \ldots, m)$ we'd like to know what a fair price for these resources would be. Each unit of product j not manufactured would result in a *loss of profit* of c_j. At the same time, we would obtain a gain (from selling the excess resources) of:

$$(9.44) \quad \sum_{i=1}^{m} a_{ij} w_i$$

Because we would save a_{ij} units of unit i from *not* manufacturing 1 unit of x_j $(i = 1, \ldots, m)$. Selling this resource would require us to make more money in the sale of the resource then we could in manufacturing the product, or:

$$(9.45) \quad \sum_{i=1}^{m} a_{ij} w_i \geq c_j$$

If a selfish profit maximizer wishes to buy these items, then we will seek a price per resource that minimizes the total he could pay for all the items, that is:

$$(9.46) \quad \min \sum_{i=1}^{m} w_i b_i$$

The Strong Duality Theorem asserts that the optimal solution to this problem will produce *fair shadow prices* that force the total amount an individual could purchase the resources of the company for to be equal to the amount the company could make in manufacturing products itself.

EXAMPLE 9.13. Assume that a leather company manufactures two types of belts: regular and deluxe. Each belt requires 1 square yard of leather. A regular belt requires 1 hour of skilled labor to produce, while a deluxe belt requires 2 hours of labor. The leather company receives 40 square yards of leather each week and a total of 60 hours of skilled labor is available. Each regular belt nets \$3 in profit, while each deluxe belt nets \$5 in profit. The company wishes to maximize profit. We can compute the fair price the company could sell its time or labor (or the amount the company should be willing to pay to obtain more leather or more hours).

The problem for the leather manufacturer is to solve the linear programming problem:

$$\max \ 3x_1 + 5x_2$$
$$\text{s.t.} \ x_1 + 2x_2 \leq 60$$
$$x_1 + x_2 \leq 40$$
$$x_1, x_2 \geq 0$$

The dual problem for this problem is given as:

$$\max \ 60w_1 + 40w_2$$
$$\text{s.t.} \ w_1 + w_2 \geq 3$$
$$2w_1 + w_2 \geq 5$$
$$w_1, w_2 \geq 0$$

If we solve the primal problem, we obtain the final revised simplex tableau as:

$$\begin{array}{c} z \\ x_1 \\ x_2 \end{array} \left[\begin{array}{cc|c} 2 & 1 & 160 \\ \hline -1 & 2 & 20 \\ 1 & -1 & 20 \end{array} \right]$$

Note that both x_1 and x_2 are in the basis. In this case, we have $w_1 = 2$ and $w_2 = 1$ from Row 0 of the tableau.

We can likewise solve the dual-problem by converting it to standard form and then using the simplex algorithm, we would have:

$$\max \ 60w_1 + 40w_2$$
$$\text{s.t.} \ w_1 + w_2 - v_1 = 3$$
$$2w_1 + w_2 - v_2 = 5$$
$$w_1, w_2, v_1, v_2 \geq 0$$

In this case, it is more difficult to solve the dual problem because there is no conveniently obvious initial basic feasible solution (that is, the identity matrix is not embedded inside the coefficient matrix).

The final full simplex tableau for the dual problem would look like:

$$
\begin{array}{c|cccccc}
 & z & w_1 & w_2 & v_1 & v_2 & RHS \\
\hline
z & 1 & 0 & 0 & -20 & -20 & 160 \\
\hline
w_1 & 0 & 1 & 0 & 1 & -1 & 2 \\
w_2 & 0 & 0 & 1 & -2 & 1 & 1
\end{array}
$$

We notice two things: The reduced costs of v_1 and v_2 are precisely the negatives of the values of x_1 and x_2. This was to be expected, these variables are duals of each other. However, in a minimization problem, the reduced costs have opposite sign. The second thing to notice is that $w_1 = 2$ and $w_2 = 1$. These are the same values we determined in the primal simplex tableau.

Lastly, let's see what happens if we increase the amount of leather available by 1 square yard. If w_2 (the dual variable that corresponds to the leather constraint) is truly a shadow price, then we should predict our profit will increase by 1 unit. Our new problem will become:

$$
\begin{aligned}
\max\ & 3x_1 + 5x_2 \\
\text{s.t.}\ & x_1 + 2x_2 \leq 60 \\
& x_1 + x_2 \leq 41 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

At optimality, our new revised simplex tableau will be:

$$
\begin{array}{c|ccc}
z & 2 & 1 & 161 \\
x_1 & -1 & 2 & 22 \\
x_2 & 1 & -1 & 19
\end{array}
$$

Thus, if our Leather Manufacturer could obtain leather for a price *under* \$1 per yard. He would be a fool not buy it. Because he could make an immediate profit. This is what economists call *thinking at the margin*.

Shadow Prices Under Degeneracy. We have asserted that the value of the dual variable does not necessarily provide a shadow price at a degenerate constraint. To see this, consider the example of the degenerate toy maker problem.

EXAMPLE 9.14.

$$
\begin{aligned}
\max\ & 7x_1 + 6x_2 \\
\text{s.t.}\ & 3x_1 + x_2 + s_1 = 120 \\
& x_1 + 2x_2 + s_2 = 160 \\
& x_1 + s_3 = 35 \\
& \frac{7}{4}x_1 + x_2 + s_4 = 100 \\
& x_1, x_2, s_1, s_2, s_3, s_4 \geq 0
\end{aligned}
$$

Recall the optimal full tableau for this problem was:

$$
\begin{array}{c|c|cccccc|c}
 & z & x_1 & x_2 & s_1 & s_2 & s_3 & s_4 & \text{RHS} \\
\hline
z & 1 & 0 & 0 & 0 & 7/5 & 0 & 16/5 & 544 \\
\hline
x_1 & 0 & 1 & 0 & 0 & -2/5 & 0 & 4/5 & 16 \\
x_2 & 0 & 0 & 1 & 0 & 7/10 & 0 & -2/5 & 72 \\
s_2 & 0 & 0 & 0 & 1 & 1/2 & 0 & -2 & 0 \\
s_3 & 0 & 0 & 0 & 0 & 2/5 & 1 & -4/5 & 19 \\
\end{array}
$$

We can compute the dual variables $\mathbf{w}$ for this by using $\mathbf{c_B}$ and $\mathbf{B}^{-1}$ at optimality. You'll notice that $\mathbf{B}^{-1}$ can always be found in the columns of the slack variables for this problem because we would have begun the simplex algorithm with an identity matrix in that position. We also know that $\mathbf{c_B} = [7\ 6\ 0\ 0]$ at optimality. Therefore, we can compute $\mathbf{c_B}\mathbf{B}^{-1}$ as:

$$
\mathbf{c_B}\mathbf{B}^{-1} = \begin{bmatrix} 7 & 6 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & -2/5 & 0 & 4/5 \\ 1 & 0 & 7/10 & 0 & -2/5 \\ 0 & 1 & 1/2 & 0 & -2 \\ 0 & 0 & 2/5 & 1 & -4/5 \end{bmatrix} = \begin{bmatrix} 0 & 7/5 & 0 & 16/5 \end{bmatrix}
$$

In this case, it would seem that modifying the right-hand-side of constraint 1 would have no affect. This is true, if we were to *increase* the value by a increment of 1. Suppose however we decreased the value of the right-hand-side by 1. Since we claim that:

$$
(9.47) \qquad \frac{\partial z}{\partial b_1} = w_1
$$

there should be no change to the optimal objective function value. However, our new optimal point would occur at $x_1 = 15.6$ and $x_2 = 72.2$ with an objective function value of 542.4, clearly the value of the dual variable for constraint 1 is not a true representation the shadow price of resource 1. This is illustrated in Figure 9.3 where we can see that modifying the right-hand-side of Constraint 1 is transforming the feasible region in a way that substantially changes the optimal solution. This is simply not detected because degeneracy in the primal problem leads to alternative optimal solutions in the dual problem.

It should be noted that a true margin price can be computed, however this is outside the scope of the notes. The reader is referred to [**BJS04**] (Chapter 6) for details.

6. The Dual Simplex Method

Occasionally when given a primal problem, it is easier to solve the dual problem and extract the information about the primal problem from the dual.

EXAMPLE 9.15. Consider the problem:

$$
\begin{aligned}
\min\ & x_1 + 2x_2 \\
s.t.\ & x_1 + 2x_2 \geq 12 \\
& 2x_1 + 3x_2 \geq 20 \\
& x_1, x_2 \geq 0
\end{aligned}
$$
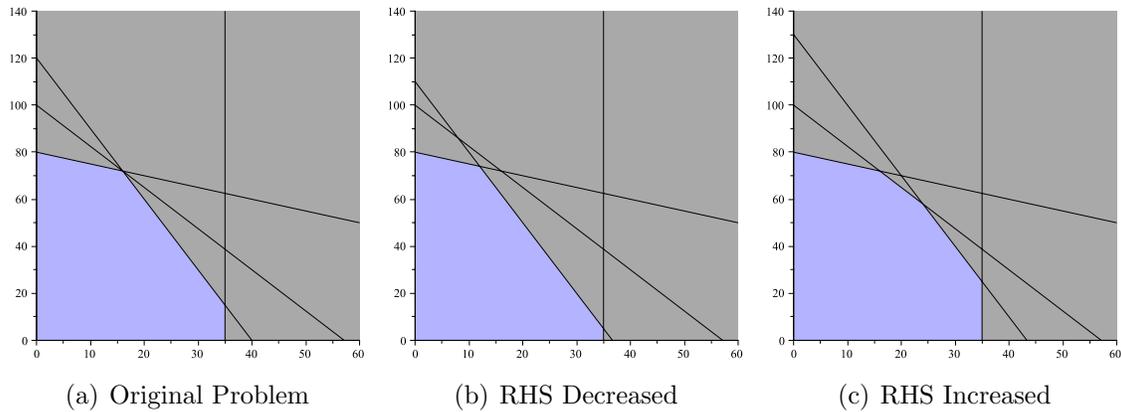
Figure 9.3. Degeneracy in the primal problem causes alternative optimal solutions in the dual problem and destroys the direct relationship between the resource margin price that the dual variables represent in a non-degenerate problem.

To transform this problem to standard form, we would have to introduce surplus variables to obtain:

$$\min \; x_1 + 2x_2$$
$$\text{s.t.} \; x_1 + 2x_2 - s_1 = 12$$
$$2x_1 + 3x_2 - s_2 = 20$$
$$x_1, x_2, s_1, s_2 \geq 0$$

In this case there is no immediately obvious initial basic feasible solution and we would have to solve a Phase I problem. Consider the dual of the original maximization problem:

$$\max \; 12w_1 + 20w_2$$
$$\text{s.t.} \; w_1 + 2w_2 \leq 1$$
$$2w_1 + 3w_2 \leq 1$$
$$w_1, w_2 \geq 0$$

This is a maximization problem whose standard form is given by:

$$\max \; 12w_1 + 20w_2$$
$$\text{s.t.} \; w_1 + 2w_2 + v_1 = 1$$
$$2w_1 + 3w_2 + v_2 = 1$$
$$w_1, w_2, v_1, v_2 \geq 0$$

In this case, a reasonable initial basic feasible solution for the dual problem is to set $v_1 = v_2 = 1$ and $w_1 = w_2 = 0$ (i.e., w_1 and w_2 are non-basic variables) and proceed with the simplex algorithm from this point.

In cases like the one illustrated in Example 9.15, we can solve the dual problem directly in the simplex tableau of the primal problem instead of forming the dual problem and solving it as a primal problem in its own tableau. The resulting algorithm is called the *dual simplex algorithm*.

For the sake of space, we will provide the dual simplex algorithm for a maximization problem:

$$P \begin{cases} \max \ \mathbf{cx} \\ s.t. \ \mathbf{Ax} = \mathbf{b} \\ \quad \mathbf{x} \geq \mathbf{0} \end{cases}$$

We will then shown how to adjust the dual simplex algorithm for minimization problems.

Dual Simplex Algorithm in Algebraic Form

(1) Choose an initial basic solution $\mathbf{x}_B$ and corresponding basis matrix $\mathbf{B}$ so that $\mathbf{wA}_{\cdot j} - c_j \geq 0$ for all $j \in \mathcal{J}$, where $\mathcal{J}$ is the set of non-basic variables and $\mathbf{w} = \mathbf{c_B B}^{-1}$.

(2) Construct a simplex tableau using this initial solution.

(3) If $\bar{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$, then an optimal solution has been achieved; STOP. Otherwise, the dual problem is feasible (since $z_j - c_j \geq 0$). GOTO STEP 4.

(4) Choose a *leaving variable* (row) $x_{\mathbf{B}_i} = \bar{\mathbf{b}}_i$ so that $\bar{\mathbf{b}}_i < 0$.

(5) Choose the index of the entering variable (column) x_j ($j \in \mathcal{J}$) using the following minimum ratio test:

$$\frac{z_j - c_j}{\bar{a}_{j_i}} = \min \left\{ \frac{z_k - c_k}{|\bar{a}_{k_i}|} : k \in \mathcal{J}, \bar{a}_{k_i} < 0 \right\}$$

(6) If no entering variable can be selected ($\bar{a}_{k_i} \geq 0$ for all $k \in \mathcal{K}$) then the dual problem is unbounded and the primal problem is infeasible. STOP.

(7) Using a standard simplex pivot, pivot on element $\bar{a}_{j_i}$, thus causing $\mathbf{x}_{\mathbf{B}_i}$ to become 0 (and thus feasible) and causing x_j to enter the basis. GOTO STEP 3.

Algorithm 9. The Matrix form of the Dual Simplex Algorithm

The pivoting step works because we choose the entering variable specifically so that the reduced costs will remain positive. Just as we chose the leaving variable in the standard simplex algorithm using a minimum ratio test to ensure that $\mathbf{B}^{-1}\mathbf{b}$ remains positive, here we use it to ensure that $z_j - c_j$ remains non-negative for all $j \in \mathcal{J}$ and thus we assure dual feasibility is maintained.

The convergence of the dual simplex algorithm is outside of the scope of this course. However, it suffices to understand that we are essentially solving the dual problem in the primal simplex tableau using the simplex algorithm applied to the dual problem. Therefore under appropriate cycle prevention rules, the dual simplex does in fact converge to the optimal (primal) solution.

THEOREM 9.16. *In the absence of degeneracy, or when using an appropriate cycling prevention rule, the dual simplex algorithm converges and is correct.*

EXAMPLE 9.17. Consider the following linear programming problem:

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

Then the standard form problem is given as:

$$\max \quad -x_1 - x_2$$
$$\text{s.t.} \quad 2x_1 + x_2 - s_1 = 4$$
$$x_1 + 2x_2 - s_2 = 2$$
$$x_1, x_2 \geq 0$$

The coefficient matrix for this problem is:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & -1 & 0 \\ 1 & 2 & 0 & -1 \end{bmatrix}$$

In standard form, there is no clearly good choice for a starting basic feasible solution. However, since this is a maximization problem and we know that $x_1, x_2 \geq 0$, we know that the objective function $-x_1 - x_2$ must be bounded above by 0. A basic solution that yields this objective function value occurs when s_1 and s_2 are both non-basic and x_1 and x_2 are both non-basic.

If we let

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

Then we obtain the infeasible solution:

$$\overline{\mathbf{b}} = \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} -4 \\ -2 \end{bmatrix}$$

Likewise we have:

$$\mathbf{w} = \mathbf{c_B}\mathbf{B}^{-1} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

since both s_1 and s_2 do not appear in the objective function. We can compute the reduced costs in this case to obtain:

$$z_1 - c_1 = \mathbf{w}\mathbf{A}_{.1} - c_1 = 1 \geq 0$$
$$z_2 - c_2 = \mathbf{w}\mathbf{A}_{.2} - c_2 = 1 \geq 0$$
$$z_3 - c_3 = \mathbf{w}\mathbf{A}_{.3} - c_3 = 0 \geq 0$$
$$z_4 - c_4 = \mathbf{w}\mathbf{A}_{.4} - c_4 = 0 \geq 0$$

Thus, the fact that $\mathbf{w} \geq \mathbf{0}$ and the fact that $z_j - c_j \geq 0$ for all j, shows us that we have a *dual feasible solution* and based on our use of a basic solution, we know that complementary slackness is ensured.

We can now set up our initial simplex tableau for the dual simplex algorithm. This is given by:

	z	x_1	x_2	s_1	s_2	RHS
z	1	1	1	0	0	0
s_1	0	-2	-1	1	0	-4
s_2	0	-1	$\boxed{-2}$	0	1	-2

We can choose either s_1 or s_2 as a leaving variable. For the sake of argument, suppose we choose s_2, the variable that's most negative as the leaving variable. Then our entering variable is chosen by comparing:

$$\frac{z_1 - c_1}{|\bar{a}_{21}|} = \frac{1}{|-1|}$$

$$\frac{z_2 - c_2}{|\bar{a}_{22}|} = \frac{1}{|-2|}$$

Clearly, $1/2 < 1$ and therefore, x_2 is our entering variable.

	z	x_1	x_2	s_1	s_2	RHS
z	1	1/2	0	0	1/2	-1
s_1	0	-3/2	0	1	$-1/2$	-3
x_2	0	1/2	1	0	$-1/2$	1

At this point, we see we have maintained dual feasibility, but we still do not have primal feasibility. We can therefore choose a new leaving variable (s_1) corresponding to the negative element in the RHS. The minimum ratio test shows that this time x_1 will enter and the final simplex tableau will be:

	z	x_1	x_2	s_1	s_2	RHS
z	1	0	0	1/3	1/3	-2
x_1	0	1	0	$-2/3$	1/3	2
x_2	0	0	1	1/3	$-2/3$	0

It's clear this is the optimal solution to the problem since we've achieved primal and dual feasibility and complementary slackness. It's also worth noting that this optimal solution is degenerate, since there is a zero in the right hand side.

EXERCISE 70. Prove that the minimum ratio test given in the dual simplex algorithm will maintain dual feasibility from one iteration of the simplex tableau to the next. [Hint: Prove that the reduced costs remain greater than or equal to zero, just as we proved that $\bar{b}$ remains positive for the standard simplex algorithm.]

Bibliography

[BJS04] Mokhtar S. Bazaraa, John J. Jarvis, and Hanif D. Sherali, *Linear programming and network flows*, Wiley-Interscience, 2004.

[BTN02] A. Ben-Tal and A. Nemirovski, *Robust optimization – methodology and applications*, Math. Program. Ser. B **92** (2002), 453–480.

[Cul72] C. G. Cullen, *Matrices and Linear Transformations*, 2 ed., Dover Press, 1972.

[Dan60] G. B. Danzig, *Inductive proof of the simplex method*, Tech. report, RAND Corporation, 1960.

[KW94] P. Kall and S. W. Wallace, *Stochastic programming*, John Wiley and Sons, 1994.

[Lan87] S. Lang, *Linear Algebra*, Springer-Verlag, 1987.

[MT03] J. E. Marsden and A. Tromba, *Vector calculus*, 5 ed., W. H. Freeman, 2003.

[PS98] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: Algorithms and complexity*, Dover Press, 1998.

[Ste07] J. Stewart, *Calculus*, 6 ed., Brooks Cole, 2007.

[WN99] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization*, Wiley-Interscience, 1999.