Math 340: Linear Programming

Omar Antolín Camarena

September 10, 2018

Contents

1 What is Linear Programming?

1.1 About the name *Linear Programming*

Linear Programming might best be called *Linear Optimization*: it means finding maxima and minima of linear functions of several variables subject to constraints that are linear equations or linear inequalities. The word "programming" has the old-fashioned meaning of "planning" and was chosen in the forties, before the advent of computers.

1.2 Example: Pig Farming

Say you are a pig farmer and want to find the cheapest way to make sure your pigs get all the nutrition they need. Let's say there are three types of food you are considering and that their nutritional contents, prices and the pigs' daily requirements are as follows:

	Corn	Silage[1]	Alfalfa	Daily requirement
Carbs	0.9	0.2	0.4	2
Protein	3	8	6	18
Vitamins	1	2	4	15
Cost	7	6	5	

Notice that we didn't specify the units, and it won't really matter what they are as long as they are consistent. Let's say the amounts of nutrients shown are in units (of carbs, protein or vitamins) per kilogram of food and that the prices are in cents per kilogram.

To express this problem mathematically choose variables for the amounts of each the foods we should buy: let c be the number of kilograms of *corn* per day you'll buy and similarly let s be the amount of *silage* and a the amount of *alfalfa* (both in kg/day).

The cost of buying those amounts will be $7c + 6s + 5a$ cents per day. This is the amount we wish to minimize. The nutritional requirements give inequalities that the variables must satisfy. For example, getting two units of carbs per day means that $0.9c + 0.2s + 0.4a \geq 2$.

Putting all of this together we get a **linear program** (often abbreviated **LP**):

$$\begin{aligned} &\text{Minimize } 7c + 6s + 5a \\ &\text{subject to} \\ &\qquad 0.9c + 0.2s + 0.4a \geq 2 \\ &\qquad 3c + 8s + 6a \geq 18 \\ &\qquad c + 2s + 4a \geq 15 \\ &\qquad c, s, a \geq 0 \end{aligned}$$

We added the obvious constraint that all three variable must be non-negative because while it is commonsense for us, if we didn't include them "the math wouldn't know about it".

The bulk of this course will be learning how to solve this type of problem using the **Simplex Method**.

1.2.1 Terminology

As we've mentioned this kind of problem is called a **linear program**. The function you are trying to maximize or minimize is called the **objective**

[1]Silage is grass or other green fodder that is stored without drying first so that it ferments, and is then fed to cattle, pigs or sheep. It's also the name of a Christian rock band.

function. Each of the inequalities or equations the variables must satisfy is called a **constraint**. Constraint that simply specify that a variable is non-negative, such as $c \geq 0$, are called **positivity constraints**. We'll almost always assume that each variable has a positivity constraint and the Simplex Method relies on this (we'll also explain what to do when you don't have positivity constraints).

1.2.2 Solving linear programs on a computer

There is software that can solve linear programs. Later on in this course we'll often use LINDO to solve LPs. In LINDO you type in a pretty recognizable version of the LP; the pig farming problem would look like this:

```
minimize 7c + 6s + 5a
subject to
  0.9 c + 0.2 s + 0.4 a >  2
  3   c +   8 s +   6 a > 18
      c +   2 s +   4 a > 15
end
```

Notice that:

- You only need to type > and LINDO assumes you mean $\geq$.
- You don't put in the positivity constraints, LINDO assumes them by default.
- You need the word `end` at, well, the end.

1.2.3 More sophisticated models

This example is a simplified versions of a real world problem, you can easily imagine adding more types of food, or breaking down the nutritional requirements further. If you want to see more sophisticated models for this problem, you could take a look at the following articles[2]:

- A mixed integer linear programming model for optimal delivery of fattened pigs to the abattoir by Lluís M. Plà-Aragonés, Sara V. Rodríguez-Sánchez and Victoria Rebillas-Loredo.

[2]We won't cover those articles in the course, I've only included them if you are curious to see what more realistic models look like.

This uses integer programming, which we will discuss much later in this course.

- Feeding Strategies for Maximising Gross Margin in Pig Production by David L.J. Alexander, Patrick C.H. Morel and Graham R. Wood.

  This uses non-linear functions, so isn't really linear programming at all, although it *uses* linear programming too.

1.3 Example: Gas Blending

Say you need to blend 4000 barrels of aviation gas from three available components: Toluene, Alkylate and Pentane. The prices and certain constraints which must be satisfied by the blend are summarized in the following table[3]:

Constraint	Toluene	Alkylate	Pentane	Specification
% aromatics	100	0	0	5 (min)
Reid Vapor Pressure	2.0	4.8	19.7	5.5 (min), 7.0 (max)
Performance Number	100	125	125	115 (max)
Cost ($) per Barrel	45	30	30	

The first row of the table says that we need at least 5% of "aromatics" in the blend, and also that only toluene is an "aromatic". The objective is to find the amount of each resource to use in order to minimize the cost of producing the aviation gas. What happens if resource specifications or costs or available amounts change?

We can make several choices of how to define our decision variables; there should definitely be one variable for each of the amounts of toluene (t), alkylate (a) and pentane (p), but we could choose to represent:

- the *number of barrels*, so we'd have $t + a + p = 4000$,
- the *percentage* of the total blend, so that $t + a + p = 100$, or,
- the *fraction* of the blend, so that $t + a + p = 1$.

Each of these would be a valid choice and would give LPs that are only slightly different. If we let t be the *fraction* of toluene in the blend (and similarly for alkylate and pentane), we'd get the following LP:

[3] I learned about this problem from Richard Anstee who in turn got it from Bill Pulleyblank around 1980 and I am uncertain of the original source.

Minimize $45t + 30a + 30p$

subject to

$$\begin{aligned} 100t &\geq 5 \\ 2t + 4.8a + 19.7p &\geq 5.5 \\ 2t + 4.8a + 19.7p &\leq 7 \\ 100t + 125a + 125p &\geq 115 \end{aligned}$$

2 Standard form for linear programs

2.1 What is standard form?

The Simplex Method, which is the procedure we will use for solving linear programs, is easiest to explain for linear programs that are in a fixed format we will call the standard form. A linear program in **standard form** looks like:

Maximize $c_1x_1 + c_2x_2 + \cdots c_nx_n$

subject to

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \\ x_1, x_2, \ldots, x_n &\geq 0 \end{aligned}$$

We can rewrite this is matrix form, by setting:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top$,
- $\mathbf{c} = (c_1, c_2, \ldots, c_n)^\top$,
- $\mathbf{b} = (b_1, b_2, \ldots, b_m)^\top$, and
- $A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$.

With those definitions we can write the LP as:

Maximize $\mathbf{c} \cdot \mathbf{x}$

$$\text{subject to } \begin{cases} A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases}$$

2.1.1 Characteristics of standard form LPs

- They are about maximizing, not minimizing.
- They have a positivity constraint for each variable.
- The other constraints are all of the form "linear combination of variables $\leq$ constant".

2.2 Converting a LP to standard form

You can always convert a LP to an equivalent one that is in standard form. There are several "errors" which you need to know how to fix.

2.2.1 "Error": the objective function is to be minimized

This is easy: minimizing $c_1x_1 + c_2x_2 + \cdots + c_nx_n$ is the "same" as maximizing $-c_1x_1 - c_2x_2 - \cdots - c_nx_n$.

In what sense is it the same? Well, the maximum value of the new objective function won't be the same as the minimum of the old objective function, but it is predictable: it's just minus the minimum of the old function. Also, the values of the variables that lead to the optimum stay the same.

2.2.2 "Error": a constraint is lower bound

Also easy: you can replace $a_{i1}x_1 + \cdots + a_{in}x_n \geq b_i$ with $-a_{i1}x_1 - \cdots - a_{in}x_n \leq -b_i$. This doesn't change which values of the x_j satisfy the constraints.

2.2.3 "Error": there is an equality constraint

An equality $u = v$ is equivalent to the system of inequalities $u \leq v$ and $u \geq v$. We can use the previous trick to turn those inequalities into something

acceptable for a standard form LP. All together, an equality $a_{i1}x_1 + \cdots + a_{in}x_n = b_i$ gets replaced by the pair of inequalities

$$\begin{aligned} a_{i1}x_1 + \cdots + a_{in}x_n &\leq b_i \\ -a_{i1}x_1 - \cdots - a_{in}x_n &\leq -b_i \end{aligned}$$

2.2.4 "Error": a variable lacks a positivity constraint

The most subtle "error" a linear program can have that keeps from being in standard form is to have a variable that lacks a positivity constraint, such a variable is called **free**. We'll explain two ways to fix that and why you probably only want to ever use the second way.

1. (Bad) strategy: divide and conquer

   If a variable x is not constrained to be non-negative, in the optimal solution to the LP we don't know if it should take a positive or negative value. So why not just try both ways? Here's an example:

   Maximize $-2x + 3y - 5z$
   subject to
   $$\begin{aligned} 7x - 5y + 6z &\leq 10 \\ -2x + 8y - 4z &\leq 3 \\ 9x - 2y - 5z &\leq 4 \\ y, z &\geq 0 \end{aligned}$$

   This problem is almost in standard form, the only issue is that x is missing a positivity constraint. The maximum of the objective can be found as the maximum of two subproblems: one where we add the constraint $x \geq 0$ and one where we add instead $x \leq 0$:

   (A) Maximize $-2x + 3y - 5z$
   subject to
   $$\begin{aligned} 7x - 5y + 6z &\leq 10 \\ -2x + 8y - 4z &\leq 3 \\ 9x - 2y - 5z &\leq 4 \\ x, y, z &\geq 0 \end{aligned}$$

   (B) Maximize $-2x + 3y - 5z$

subject to

$$\begin{aligned} 7x - 5y + 6z &\leq 10 \\ -2x + 8y - 4z &\leq 3 \\ 9x - 2y - 5z &\leq 4 \\ y, z &\geq 0 \\ x &\leq 0 \end{aligned}$$

Problem (B) can be recast in standard form by a change of variables: flipping the sign of x, say, letting $x' = -x$ it becomes:

(B′) Maximize $2x' + 3y - 5z$
subject to

$$\begin{aligned} -7x' - 5y + 6z &\leq 10 \\ 2x' + 8y - 4z &\leq 3 \\ -9x' - 2y - 5z &\leq 4 \\ x', y, z &\geq 0 \end{aligned}$$

Later on we will learn how to solve this LPs, but for know I'll just list the solutions:

- For (A) the maximum is 1.125 (achieved, for example, for $x = 0, y = 0.375, z = 0$).
- For (B′) the maximum is 3 (achieved, for example, for $x = 1.5, y = 0, z = 0$).

So the maximum for the original problem that had no positivity constraint for x is 3, and as this came from (B′), the maximum for the original problem will have a negative value of x.

The main reason to avoid this strategy is that it creates a lot of extra work. Even in this example it turned solving one LP into solving two. But it gets worse if there are more variables lacking a positivity constraint: if we had k variables lacking a positivity constraint this strategy would have us solve one LP for every combination of signs of those k variables, that is, 2^k different LPs!

2. Good Strategy: make a difference!

   There is a way to turn an LP with free variables into just one equivalent LP via a change of variables. If x is free, we can set $x = x' - x''$

where $x', x'' \geq 0$ —a difference of non-negative numbers can have any sign at all!

Our example from above becomes the following LP in standard form:

$$\begin{aligned}
&\text{Maximize } -2x' + 2x'' + 3y - 5z \\
&\text{subject to} \\
&\qquad 7x' - 7x'' - 5y + 6z \leq 10 \\
&\qquad -2x' + 2x'' + 8y - 4z \leq 3 \\
&\qquad 9x' - 9x'' - 2y - 5z \leq 4 \\
&\qquad x', x'', y, z \geq 0
\end{aligned}$$

In what sense is this equivalent to the old LP? Well, given any values x', x'', y, z satisfying the constraints of the new LP, setting $x = x' - x''$ and keeping the value of y and z yields valid values for the original LP, with the same value for the objective functions. Conversely, if x, y, z are values satisfying the constraints of the original we can keep y and z and define non-negative x', x'' as follows:

- If $x \geq 0$, we can take $x' = x, x'' = 0$.
- If $x < 0$, we can take $x' = 0, x'' = -x$.

This gets us values satisfying the constraints of the new LP, with the same value for the objective function. In particular, notice that the maximum of the objective functions will be the same for the original and the new LP.

Notice that there are infinitely many choices we could have made above when picking values for x' and x''. For example, if $x = -5$ we said take $x' = 0, x'' = 5$, but $x' = 10, x'' = 15$ would also work (since they are non-negative and $x' - x'' = x$).

3 Thinking of linear programs geometrically

3.1 Representing a two-variable LP graphically

Consider the following linear program:

$$\text{Maximize } x + y$$

subject to

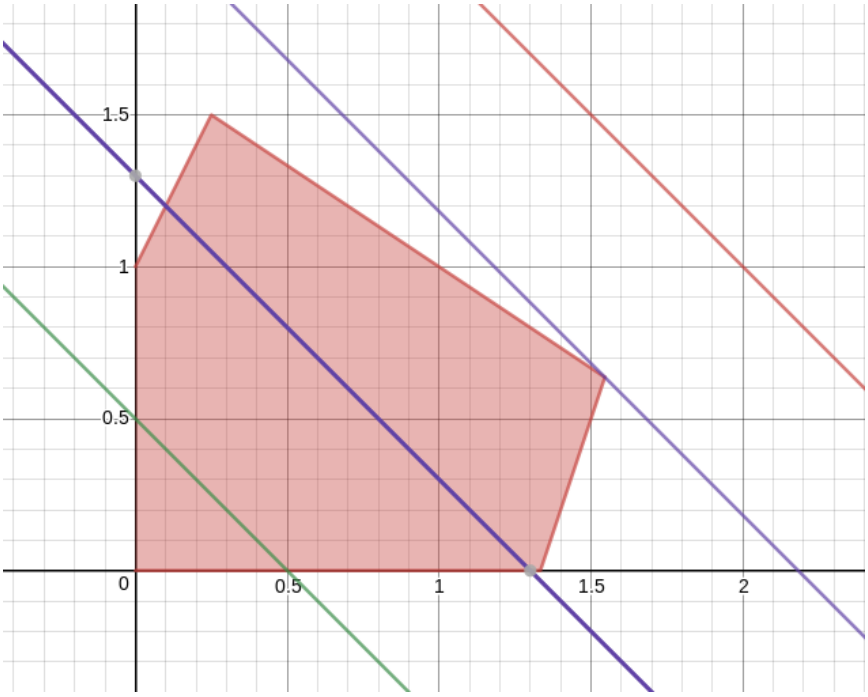
$$\begin{aligned} 2x + 3y &\leq 5 \\ -2x + y &\leq 1 \\ 3x - y &\leq 4 \\ x, y &\geq 0 \end{aligned}$$

We can represent this graphically in the xy-plane. For each constraint, the set of points (x, y) that satisfies it is a half-plane; the set of points satisfying all of the constraints is then the intersection of all these half planes. In this example it turns out to be a pentagon, shaded red in the image below.

The set of points that satisfy all the constraints is called the **feasible region**. It is always **convex**, meaning that whenever two points are in the feasible region, so is the entire line segment connecting them. It doesn't have to be of finite size as it is for this linear program, it can "go off to infinity". It can also be empty! (It is a good exercise to draw half-planes that would give a feasible region of infinite extent, and to draw half-plane that don't have any point common to all, leading to an empty feasible region.)

How can we represent the objective function? The *graph* of the function would require three dimensions to plot (i.e., the graph is the *plane* $z = x + y$ in three-dimensional space). To get something we can draw in the same figure as the feasible region we will instead draw level curves $x + y = k$ that you get by setting the objective function equal to a constant k. (Since the objective function is linear these level "curves" are actually straight lines.)

Here is a picture of the feasible region together with various level lines for the objective function:



As you change the constant k, the level line slides around, always keeping the same slope. If for some value of k that line goes through the feasible region, that means that it is possible to obtain the value k for some choice of x and y satisfying the constraints. We want the largest such k, so imagine sliding the line around until it is about to fall off the feasible region: the corresponding k is the maximum[4] value of the objective function in the feasible region and the answer to the optimization question posed by the linear problem.

3.1.1 Explore the example interactively with Desmos

If you're reading this on a computer[5] you're in luck: using the lovely online graphing tool Desmos you can explore the above example interactively.

- Here's a picture of just the constraints. Each one is drawn as a translucent half-plane, where the half-planes overlap their colors combine. You can turn each constraint on and off by clicking on the colored circle next to the inequality in the left-hand column.

[4] Or minimum, depending on which side it's falling off! In this example clearly the minimum is 0, attained at the origin.

[5] If you are reading this on paper, I apologize.

- Here's a picture of the feasible region[6] with a movable level line for the objective function. Play with the slider to approximately maximize the objective function!

3.2 Why don't we just always do this?

For the example above this graphical way of thinking works wonderfully: we see right away that the optimal solution is at the point where the two constraints $2x + 3y \leq 5$ and $3x - y \leq 4$ both turn into equalities. Solving the system of equations we get $x = 17/11, y = 7/11$, so the maximum of the objective function is $24/11$.

But this isn't a practical method in general. This LP had only two variables so our picture was two-dimensional. Natural LPs tend to have many, many more variables and geometric intuition tends to give out after three dimensions!

4 Introduction to the simplex method

We'll start by explaining the "easy case" of the Simplex Method: when you start with a linear program in standard form where all the right-hand sides of the constraints are non-negative.

Roughly speaking, you turn the LP into a dictionary[7], and then repeatedly pivot to get new dictionaries until at some point the numbers in the dictionary indicate you are done. We'll illustrate the procedure with the following example:

Maximize $x_1 + 2x_2 - x_3$

subject to

$$\begin{aligned} 2x_1 + x_2 + x_3 &\leq 14 \\ 4x_1 + 2x_2 + 3x_3 &\leq 28 \\ 2x_1 + 5x_2 + 5x_3 &\leq 30 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

[6] Notice the way all the constraints are combined into one to get the feasible region. This is just because there doesn't seem to currently be a way of asking Desmos to intersect various regions.

[7] There is an alternative way of presenting the Simplex Method using *tableaux* instead of dictionaries. This really just a different way of writing the same calculation. If you've studied linear programming before you are likely to have seen tableaux. It's a good exercise to solve the same LP both ways, so you can see how this dictionary method corresponds to the tableau. If you don't know about tableaux, don't worry about them.

4.1 Slack variables

The first step will be to introduce **slack variables**, one for each of the constraints (except the positivity constraints). These are simply the difference between the right-hand side and the left-hand side of a constraint. For example, for the first constraint we define a slack variable $x_4 = 14 - 2x_1 - x_2 - x_3$. In terms of this new variable, the first constraint is equivalent simply to $x_4 \geq 0$, the positivity constraint for x_4.

Introducing these slack variables leads to a linear program equivalent to the original one[8] but for which all constraints are either *equations* or *positivity constraints*:

Maximize $x_1 + 2x_2 - x_3$

subject to

$$\begin{array}{rrrrrrr} 2x_1 & +x_2 & +x_3 & +x_4 & & & = 14 \\ 4x_1 & +2x_2 & +3x_3 & & +x_5 & & = 28 \\ 2x_1 & +5x_2 & +5x_3 & & & +x_6 & = 30 \\ x_1, & x_2, & x_3, & x_4, & x_5, & x_6 & \geq 0 \end{array}$$

4.2 Dictionaries

Solving for the slack variables we get the initial **dictionary** for our problem:

$$\begin{array}{rrrrr} x_4 = & 14 & -2x_1 & -x_2 & -x_3 \\ x_5 = & 28 & -4x_1 & -2x_2 & -3x_3 \\ x_6 = & 30 & -2x_1 & -5x_2 & -5x_3 \\ z = & 0 & +x_1 & +2x_2 & -x_3 \end{array}$$

The variables on the left, x_4, x_5, x_6, are the **basic variables** for this dictionary; the set of these variables is called the **basis**. (In the initial dictionary the basic variables are the slack variables, that changes after pivoting.) The rest of the variables are called **non-basic**. Notice that at the bottom we've added a variable z for the objective function.

Each dictionary is a system of equations that is equivalent to the equality constraints from the LP obtained from the original LP by adding slack variables. In terms of dictionaries, the LP we want to solve is equivalent to "maximize z subject to the equations in the dictionary and to positivity constraints for all x_i".

[8] Notice that this equivalent LP is *not* in standard form.

4.2.1 The solution associated to a dictionary

To any dictionary there is an **associated solution** of the system of equations: just set all the non-basic variables to zero and compute the values of the basic variables from the dictionary. For the above dictionary the associated solution is

$$x_1 = x_2 = x_3 = 0, x_4 = 14, x_5 = 28, x_6 = 30.$$

4.2.2 Feasibility

The solution associated to a dictionary certainly satisfies all of the constraints that are equations, but may fail to satisfy the positivity constraints. When it does also satisfy the positivity constraints it is said to be a **feasible solution** and the dictionary is called **feasible** too.

Here the initial dictionary is feasible because the LP we started with happened to have all the right-hand sides of constraints non-negative. This needn't always happen and later on we will cover what to do when some constraints have negative right-hand sides; but for now, let's focus on the case when the initial dictionary is feasible, like in this example.

4.3 Pivoting

The main idea of the Simplex Method is to go from dictionary to dictionary by exchanging a basic variable for a non-basic one, in such a way that:

- The objective function increases at each step[9].
- The dictionary is feasible at every step.

Let's explain how to pick the variables you swap.

4.3.1 Picking a non-basic variable to enter the basis

In the dictionary we have, the z**-row** is $z = 0 + x_1 + 2x_2 - x_3$. This tells us that for the solution associated to the dictionary, the objective function has the value 0; and it also tells us, for each non-basic variable, whether increasing that variable will increase or decrease z. Since x_1 and x_2 have positive coefficients, increasing them would increase z. On the other hand, increasing x_3 will *decrease* z.

[9]Or, at least, doesn't decrease. Sometimes you are forced to make **degenerate pivots** which don't change the associated solution or the value of the objective function.

So, to increase z we will choose either x_1 or x_2 to enter the basis.

Either one would be a valid choice and make some progress toward solving the LP, but we'll use a definite rule to pick one:

Standard rule[10] **for picking the entering variable**: among the non-basic variables that have a *positive* coefficient in the objective function, choose the one with the *largest* coefficient. If more than one variable is tied for largest coefficient, pick the one with smallest index (e.g., if both x_4 and x_2 had a coefficient of 3, and 3 was the largest, you'd pick x_2).

This is not done for mathematical reasons but for logistical reasons! By having a standard rule for the course, I can make sure that when you solve an LP you don't get crazy numbers, and the course grader has an easier time, too.

4.3.2 Picking a basic variable to exit the basis

We've settled on putting x_2 in the basis and now we need to decide which basic variable to swap it with. The solution associated to our current dictionary has $x_2 = 0$. The idea now is to increase the value of x_2 as much as possible while keeping the resulting dictionary feasible. Let's focus on how the values of the basic variables depend on x_2 (keeping $x_1 = x_3 = 0$):

$$\begin{aligned} x_4 &= 14 - x_2 \\ x_5 &= 28 - 2x_2 \\ x_6 &= 30 - 5x_2 \\ z &= 0 + 2x_2 \end{aligned}$$

We see that:

- To keep $x_4 \geq 0$, we can only increase x_2 up to 14.
- To keep $x_5 \geq 0$, we can only increase x_2 up to 14.
- To keep $x_6 \geq 0$, we can only increase x_2 up to 6.

So, to keep all variables non-negative, we can only go up to 6. Since x_6 had the strictest requirement on the entering variable x_2, we pick x_6 as the exiting variable.

In case two variable are tied for strictest bound on the entering variable we need a tie breaking rule:

[10] In some course materials you might see this standard rule called "Anstee's rule" instead, named tongue in cheek after Richard Anstee who has taught this course many times. Don't expect non-UBC people to call it Anstee's rule.

Standard rule for picking the exiting variable: In case of ties, pick the variable with the smallest index.

1. Performing the pivot

   So we've decided that x_2 will enter the basis and x_6 will exit. We get the next dictionary by:

   - Solving for x_2 in the equation for x_6.
   - Plugging that value for x_2 into the equations for the other basic variables and the equation for the objective function.

   This gives us:

$$\begin{aligned} x_2 &= 6 - \tfrac{2}{5}x_1 - x_3 - \tfrac{1}{5}x_6 \\ x_4 &= 8 - \tfrac{8}{5}x_1 + \tfrac{1}{5}x_6 \\ x_5 &= 16 - \tfrac{16}{5}x_1 - x_3 + \tfrac{2}{5}x_6 \\ z &= 12 + \tfrac{1}{5}x_1 - 3x_3 - \tfrac{2}{5}x_6 \end{aligned}$$

   This is a new dictionary that still represents the same LP. The associated solution has $x_1 = x_3 = x_6 = 0, x_2 = 6, x_4 = 8, x_5 = 16$ and $z = 12$. This is a feasible solution so we learn that getting $z = 12$ is possible, and thus the maximum z, which is what we seek, is 12 or larger.

2. One way to tell you made a mistake

   The solution turned out feasible again by virtue of the way we picked the exiting variable. If we had, for example, decided to make x_2 enter but x_4 exit we would have gotten the following dictionary:

$$\begin{aligned} x_2 &= 14 - 2x_1 - x_3 - x_4 \\ x_5 &= 0 - x_3 + 2x_4 \\ x_6 &= -40 + 8x_1 + 5x_4 \\ z &= 28 - 3x_1 - 3x_3 - 2x_4 \end{aligned}$$

   The associated solution has $x_6 = -40$ violating the positivity constraint. If that ever happens, you'll know you made a mistake somewhere!

4.3.3 When to stop pivoting

At this point you know how to pivot to increase the objective function while keeping the dictionary feasible. For example, in the last feasible dictionary we can pivot by having x_1 enter. Either x_4 or x_5 would have to exit and our tie-breaking rule has us choose x_4. After pivoting we get:

$$\begin{aligned} x_1 &= 5 & & -\tfrac{5}{8}x_4 & +\tfrac{1}{8}x_6 \\ x_2 &= 4 & -x_3 & +\tfrac{1}{4}x_4 & -\tfrac{1}{4}x_6 \\ x_5 &= 0 & -x_3 & +2x_4 & \\ z &= 13 & -3x_3 & -\tfrac{1}{8}x_4 & -\tfrac{3}{8}x_6 \end{aligned}$$

This new dictionary has all of the coefficients in the z-row *negative*, so we can't pick a new entering variable. This means we are done with this problem and $z = 13$ is the maximum value of the objective function. The solution associated to this dictionary, $x_3 = x_4 = x_6 = 0, x_1 = 5, x_2 = 4, x_5 = 0$ is an **optimal solution**.

Why exactly is $z = 13$ the maximum? Clearly the Simplex Method stops here, since there is no way to pick variables for the next pivot according to the rules, but can't there be a sneaky way to increase z, say, by doing a pivot that's against the rules because it *decreases* z, followed by a pivot that makes z even bigger than 13?

There's no need to worry, that can't happen and the dictionary tells us why: we have $z = 13 - 3x_3 - \frac{1}{8}x_4 - \frac{3}{8}x_6$ and each variable has a positivity constraint, so that equation really tells us that $z = 13 -$ (something non-negative) ≤ 13.

Additionally we learn that $z = 13$ *only* when $x_3 = x_4 = x_6 = 0$. And having those variable be zero tells us that $x_1 = 5, x_2 = 4, x_5 = 0$, so that the optimal solution we found is the *only* optimal solution.

5 Multiple optimal solutions

When you reach a feasible dictionary where all the coefficients in the row for the objective function are negative, why does that mean you have an optimal solution? For example, if you come to the following dictionary[11] you've found an optimal solution:

[11] This dictionary is very similar, but not exactly the same as the final dictionary in the example we did earlier.

$$\begin{aligned} x_1 &= 5 & & -\tfrac{5}{8}x_4 & +\tfrac{1}{8}x_6 \\ x_2 &= 4 & -x_3 & +\tfrac{1}{4}x_4 & -\tfrac{1}{4}x_6 \\ x_5 &= 0 & -x_3 & +2x_4 & \\ z &= 13 & -3x_3 & & -\tfrac{3}{8}x_6 \end{aligned}$$

The argument was as follows: since we know that $x_3, x_4, x_6 \geq 0$, we have $z = 13 - 3x_3 - \frac{3}{8}x_6 \leq 13$. This tells us that 13 is the maximum value of z, so the associated solution to this dictionary, namely $x_3 = x_4 = x_6 = 0$, $x_1 = 5, x_2 = 4, x_5 = 0$, is an optimal solution.

Now, in an earlier example we reached a final dictionary that was almost like this except the z-row was $z = 13 - 3x_3 - \frac{1}{8}x_4 - \frac{3}{8}x_6$. In that case to get $z = 13$ we needed all three non-basic variables to be 0 so that the solution associated to the dictionary was the *only* optimal solution. In our current example, to get $z = 13$ only $x_3 = x_6 = 0$ is required, so it is possible that there are other optimal solutions in which $x_4 > 0$. Let's try to find them:

Setting $x_3 = x_6 = 0, x_4 = t$ we get $x_1 = 5 - \frac{5}{8}t, x_2 = 4 + \frac{1}{4}t, x_5 = 2t$. Now this solution will be optimal (because it has $z = 13$) as long as it is feasible, that is, as long as all x_i are non-negative. This boils down to $t \geq 0$ and $5 - \frac{5}{8}t$, because the other inequalities (namely $2t \geq 0$, $4 + \frac{1}{4}t \geq 0$) have a positive coefficient of t and thus follow from $t \geq 0$.

Putting it all together we see that for this second dictionary we have that $x_3 = x_6 = 0, x_4 = t, x_1 = 5 - \frac{5}{8}t, x_2 = 4 + \frac{1}{4}t, x_5 = 2t$ is an optimal solution for $0 \leq t \leq 8$, and that this formula gives *all* the optimal solutions.

5.1 Warning: zeros in the z-row don't automatically imply multiple optimal solutions

What if we had reached the following dictionary instead? What are the optimal solutions?

$$\begin{aligned} x_1 &= 5 & & -\tfrac{5}{8}x_4 & +\tfrac{1}{8}x_6 \\ x_2 &= 4 & -x_3 & +\tfrac{1}{4}x_4 & -\tfrac{1}{4}x_6 \\ x_5 &= 0 & -x_3 & +2x_4 & \\ z &= 13 & & -\tfrac{1}{8}x_4 & -\tfrac{3}{8}x_6 \end{aligned}$$

Well, since the non-basic variable x_3 is missing from the z-row we might think there are again multiple optimal solutions, but see what happens when we carry out the reasoning we used above:

We have $z \leq 13$ and to get $z = 13$ we need $x_4 = x_6 = 0$, but x_3 needn't be zero. Setting $x_3 = t$ we see that $x_1 = 5, x_2 = 4 - t, x_5 = -t$. What values can t take? The positivity constraint on x_3 tells us that $t \geq 0$; the positivity constraint on x_5 tells us that $-t \geq 0$, or, $t \leq 0$. So we can only have $t = 0$, which means that if we had found this final dictionary the only optimal solution to the LP would be the associated solution to the dictionary, namely $x_3 = x_4 = x_6 = 0, x_1 = 5, x_2 = 4, x_5 = 0$. This happened even though the non-basic variable x_3 had a coefficient of zero in the z-row, because the basic variable x_5 had value 0 and a negative coefficient for x_3.

5.2 Warning: when more than non-basic variable has a zero coefficient things can get complicated

Don't get the impression that it is always easy to find formulas for all of the optimal solution! The examples above gave unique solutions or a line segment of solutions because only one of the non-basic variables was free to be non-zero. If more non-basics can be positive the inequalities can become tricky. This makes sense if you think geometrically: for example if you have three variables in the original problem, the feasible solutions form a convex polytope in three dimensional space and the optimal solutions are the intersection of that polytope with a plane; this intersection could be either (1) a single vertex, (2) an edge of the polytope, (3) and entire face, which could be any convex polygon!

For example, imagine we had reached the final dictionary:

$$\begin{aligned} x_1 &= 5 & & -\tfrac{5}{8}x_4 & +\tfrac{1}{8}x_6 \\ x_2 &= 4 & -x_3 & +\tfrac{1}{4}x_4 & -\tfrac{1}{4}x_6 \\ x_5 &= 0 & -x_3 & +2x_4 & \\ z &= 13 & -3x_3 & & \end{aligned}$$

Then the optimal solutions would have $x_3 = 0$, but x_4 and x_6 could take any values as long all the variables are non-negative, that is, they could take any values solving the following system of inequalities:

$$\begin{aligned} 5 \quad & -\tfrac{5}{8}x_4 & +\tfrac{1}{8}x_6 & \geq 0 \\ 4 \quad & +\tfrac{1}{4}x_4 & -\tfrac{1}{4}x_6 & \geq 0 \\ & 2x_4 & & \geq 0 \\ x_4 \quad & \geq 0, & x_6 & \geq 0 \end{aligned}$$

While it may not be easy in such case (specially if there are more variables!) to find *all* optimal solutions, it's usually not hard to find *at least two*

optimal solutions. You can always take all the non-basic variables to be 0 (this is the solution associated to the final dictionary), and then by trial and error guess a second solution. In this example, $x_4 = x_6 = 1$ works.

6 Unbounded linear programs

Let's think about how the Simplex Method stops. In other words: when can we no longer pivot? We've already seen one way this can happen: if all coefficients in the z-row are non-positive, then we have found an optimal solution. In terms of pivoting, that happens when we can't choose an entering variable.

But can pivoting instead fail at the stage where you choose an exiting variable? That is, could it be that you do have an entering variable but no exiting variable? Well, the exiting variable is chosen to be the basic variable imposing the most stringent constraint on the entering variable. What if none of the basic variable imposes any constraint on how much you can increase the entering variable? For that to happen we'd need all the equations for the basic variables to contain the entering variable with a non-negative coefficient. This can certainly happen, for example, the dictionary might be:

$$\begin{aligned} x_1 &= 5 + 5x_4 + x_6 \\ x_2 &= 4 - x_3 + x_4 - x_6 \\ x_5 &= 2 - x_3 + 2x_4 \\ z &= 13 - 3x_3 + 2x_4 - 3x_6 \end{aligned}$$

Here x_4 is the entering variable and we can increase x_4 as much as we want without x_1, x_2 or x_5 ever becoming negative. And as x_4 keeps increasing, z also increases without bound. In this situation z does not have a maximum and we say the linear program is **unbounded**. We can use the entering variable to find formulas for a family of feasible solutions for which the objective function tends to infinity. Set $x_4 = t$ and all other non-basic variables to zero: $x_3 = x_6 = 0$. Then the basic variables become $x_1 = 5 + 5t$, $x_2 = 4 + t$, $x_5 = 2 + 2t$ and the objective function is $z = 13 + 2t$. These solutions are feasible as long as $t \geq 0$ and we have $\lim_{t\to\infty} z = \infty$.

Whenever a linear problem is unbounded the Simplex Method will eventually tell us (by reaching a dictionary that has an entering variable but no exiting variable) and we can produce an unbounded one-parameter family of feasible solutions as above.

6.1 A shortcut that is sometimes available

Consider this slight variation on the previous example, imagine that while solving a linear program you came across this dictionary:

$$\begin{aligned} x_1 &= 5 + 5x_4 + x_6 \\ x_2 &= 4 - x_3 + x_4 - x_6 \\ x_5 &= 2 - x_3 + 2x_4 \\ z &= 13 + 3x_3 + 2x_4 - 3x_6 \end{aligned}$$

The standard pivoting rule would lead us to choose x_3 as the entering variable and then x_5 as the exiting variable. But here x_4 has positive coefficients in every row of the dictionary; this tells us already that the linear program is unbounded and that

$$x_1 = 5 + 5t, x_2 = 4 + t, x_3 = 0, x_4 = t, x_5 = 2 + 2t, x_6 = 0$$

for $t \geq 0$ is a family of feasible solutions with $\lim_{t\to\infty} z = \infty$. So, if you notice that some variable (here x_4) shows the problem is unbounded, you can stop pivoting and go straight to the unbounded family of feasible solutions.

What happens if you don't spot x_4 and instead do the standard pivot? Nothing terrible: you just do a little extra work: since the problem *is* unbounded at some later stage you will be *forced* to discover that the problem is unbounded.

7 The 2-phase Simplex Method and infeasible linear programs

So far we've only discussed how to solve linear programs (that are in standard form and) for which the right-hand sides of the constraints are all non-negative. Why did we have that restriction? Take this LP, for instance:

Maximize $x_1 - x_2 + x_3$

subject to

$$\begin{aligned} 2x_1 - x_2 + 2x_3 &\leq 4 \\ 2x_1 - 3x_2 + x_3 &\leq -5 \\ -x_1 + x_2 - 2x_3 &\leq -1 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

This problem *is* in standard form, but since there are negative numbers in the right hand sides of the constraints, introducing slack variables produces an initial dictionary that is *not* feasible:

$$\begin{aligned} x_4 &= 4 - 2x_1 + x_2 - 2x_3 \\ x_5 &= -5 - 2x_1 + 3x_2 - x_3 \\ x_6 &= -1 + x_1 - x_2 + 2x_3 \\ z &= 0 + x_1 - x_2 + 3x_3 \end{aligned}$$

The solution associated to that dictionary has, for example, $x_5 = -5$, which is negative. That means that the second constraint is violated and indeed, the associated solution has $x_1 = x_2 = x_3 = 0$ so that $2x_1 - 3x_2 + x_3 = 0 \not\leq -5$.

Remember that the simplex method proceeds by pivoting from feasible dictionary to feasible dictionary until you reach a dictionary from which you cannot pivot indicating you either have an optimal solution or the LP is unbounded (in which case you can find an 1-parameter family of solutions with objective function tending to infinity). So we can't use this infeasible dictionary.

It's possible that choosing a different set of variables (instead of the slack variables) to be the basic variables and solving for them would produce a feasible dictionary from which we can start the simplex method. But how can we find which set of variables to use?

7.1 Phase 1: Finding a feasible starting dictionary

If we had a feasible dictionary to begin applying the simplex method, the associated solution would be a feasible solution of the LP, that is, it would have values for the decision variables that satisfy all of the constraints in original LP. Such values don't always exist! We'll now explain how to simultaneously find out whether or not there are feasible solutions and, if there are, how to pick a feasible starting dictionary.

We'll do it by considering an **auxiliary linear program** that is constructed from the original LP by a adding a variable we will call x_0 as follows:

Minimize x_0 (or, in standard form, maximize $-x_0$)

subject to

$$\begin{aligned} 2x_1 - x_2 + 2x_3 - x_0 &\leq 4 \\ 2x_1 - 3x_2 + x_3 - x_0 &\leq -5 \\ -x_1 + x_2 - 2x_3 - x_0 &\leq -1 \\ x_0, x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Notice that this auxiliary LP only uses the constraints from the original LP, not the objective function. This is reasonable because in this first phase we are only concerned with finding a feasible starting point for the second phase (that we've already seen how to do); feasibility is all about the constraints and not the objective function.

This auxiliary LP obviously has feasible solutions: just take x_0 very large and all constraints are satisfied. (In this example, we can take $x_1 = x_2 = x_3 = 0$ and $x_0 = 5$.) And this auxiliary LP has a feasible solution *with* $x_0 = 0$ if and only if the original LP has a feasible solution! This is an important point, make sure you understand why that's true.

If we add slack variables to the auxiliary LP we still get an infeasible starting dictionary (where I've called the objective function w so we can still use z for the objective function in the original LP):

$$\begin{aligned} x_4 &= 4 - 2x_1 + x_2 - 2x_3 + x_0 \\ x_5 &= -5 - 2x_1 + 3x_2 - x_3 + x_0 \\ x_6 &= -1 + x_1 - x_2 + 2x_3 + x_0 \\ w &= - x_0 \end{aligned}$$

But this kind of infeasible dictionary coming from the auxiliary LP becomes feasible after just one pivot! The variables are chosen by a different rule than the standard pivots we've seen, so we'll call this a **special pivot to feasibility**. The entering variable can only be x_0, and for the exiting variable we'll take the one with the *most negative value*, here x_5. After pivoting we get:

$$\begin{aligned} x_0 &= 5 + 2x_1 - 3x_2 + x_3 + x_5 \\ x_4 &= 9 - 2x_2 - x_3 + x_5 \\ x_6 &= 4 + 3x_1 - 4x_2 + 3x_3 + x_5 \\ w &= -5 - 2x_1 + 3x_2 - x_3 - x_5 \end{aligned}$$

This dictionary is feasible and tells us that we can achieve $w = -x_0 = -5$. Remember we're hoping to achieve $x_0 = 0$. We can use the simplex method with standard pivots starting from this feasible dictionary to maximize w:

x_2 enters, x_6 exits:

$$\begin{aligned} x_2 &= 1 + \tfrac{3}{4}x_1 + \tfrac{3}{4}x_3 + \tfrac{1}{4}x_5 - \tfrac{1}{4}x_6 \\ x_0 &= 2 - \tfrac{1}{4}x_1 - \tfrac{5}{4}x_3 + \tfrac{1}{4}x_5 + \tfrac{3}{4}x_6 \\ x_4 &= 7 - \tfrac{3}{2}x_1 - \tfrac{5}{2}x_3 + \tfrac{1}{2}x_5 + \tfrac{1}{2}x_6 \\ w &= -2 + \tfrac{1}{4}x_1 + \tfrac{5}{4}x_3 - \tfrac{1}{4}x_5 - \tfrac{3}{4}x_6 \end{aligned}$$

Now x_3 enters, x_0 exits:

$$\begin{array}{rrrrrr} x_2 = & \frac{11}{5} & +\frac{3}{5}x_1 & +\frac{2}{5}x_5 & +\frac{1}{5}x_6 & -\frac{3}{5}x_0 \\ x_3 = & \frac{8}{5} & -\frac{1}{5}x_1 & +\frac{1}{5}x_5 & +\frac{3}{5}x_6 & -\frac{4}{5}x_0 \\ x_4 = & 3 & -x_1 & & -x_6 & +2x_0 \\ w = & & & & & -x_0 \end{array}$$

At this point, the auxiliary LP is solved: the maximum value of $w = -x_0$ is 0, which means the auxiliary LP *does* have feasible solutions with $x_0 = 0$, namely the solution associated to this dictionary: $x_0 = x_1 = x_5 = x_6 = 0, x_2 = \frac{11}{5}, x_3 = \frac{8}{5}, x_4 = 3$. This (ignoring $x_0 = 0$) is also a feasible solution for the constraints of the original LP. Moreover, setting $x_0 = 0$ we get a feasible dictionary for the original LP that we can use for the Simplex Method:

$$\begin{array}{rrrrr} x_2 = & \frac{11}{5} & +\frac{3}{5}x_1 & +\frac{2}{5}x_5 & +\frac{1}{5}x_6 \\ x_3 = & \frac{8}{5} & -\frac{1}{5}x_1 & +\frac{1}{5}x_5 & +\frac{3}{5}x_6 \\ x_4 = & 3 & -x_1 & & -x_6 \end{array}$$

This dictionary is just missing a z-row for the objective function. Now $z = x_1 - x_2 + x_3$, but we can't use this directly since the non-basic variable for the dictionary are x_1, x_5, x_6. So we simply plug in the values for x_2 and x_3 given by the dictionary:

$$\begin{aligned} z &= x_1 - x_2 + x_3 \\ &= x_1 - (\frac{11}{5} + \frac{3}{5}x_1 + \frac{2}{5}x_5 + \frac{1}{5}x_6) + (\frac{8}{5} - \frac{1}{5}x_1 + \frac{1}{5}x_5 + \frac{3}{5}x_6) \\ &= -\frac{3}{5} + \frac{1}{5}x_1 - \frac{1}{5}x_5 + \frac{2}{5}x_6. \end{aligned}$$

7.2 Phase 2

Adding that formula for z to the above dictionary gives us a complete *feasible* dictionary for the original linear problem, namely:

$$\begin{array}{rrrrr} x_2 = & \frac{11}{5} & +\frac{3}{5}x_1 & +\frac{2}{5}x_5 & +\frac{1}{5}x_6 \\ x_3 = & \frac{8}{5} & -\frac{1}{5}x_1 & +\frac{1}{5}x_5 & +\frac{3}{5}x_6 \\ x_4 = & 3 & -x_1 & & -x_6 \\ z = & -\frac{3}{5} & +\frac{1}{5}x_1 & -\frac{1}{5}x_5 & +\frac{2}{5}x_6 \end{array}$$

From there we can do standard pivots until we solve the LP. In this particular example one pivot suffices: x_6 enters, x_4 exits and we get the final dictionary (which is optimal):

$$\begin{aligned} x_2 &= \tfrac{14}{5} + \tfrac{2}{5}x_1 - \tfrac{1}{5}x_4 + \tfrac{2}{5}x_5 \\ x_3 &= \tfrac{17}{5} - \tfrac{4}{5}x_1 - \tfrac{3}{5}x_4 + \tfrac{1}{5}x_5 \\ x_6 &= 3 - x_1 - x_4 \\ z &= \tfrac{3}{5} - \tfrac{1}{5}x_1 - \tfrac{2}{5}x_4 - \tfrac{1}{5}x_5 \end{aligned}$$

So the Simplex Method as we studied it initially is really only "Phase 2" of the full 2-phase Simplex Method! It's just that we initially discussed only the case where the starting dictionary was feasible, so we could skip Phase 1.

7.3 Example of an infeasible LP

As we mention above, not all LPs are feasible: sometimes the constraints are just impossible to satisfy simultaneously. In that case, the Simplex Method discovers this in Phase 1. For example, consider the following LP:

Maximize $x_1 - x_2 + x_3$

subject to

$$\begin{aligned} 2x_1 - x_2 - 2x_3 &\leq 4 \\ 2x_1 - 3x_2 - x_3 &\leq -5 \\ -x_1 + x_2 + x_3 &\leq -1 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

We form the auxiliary LP for Phase 1:

Maximize $-x_0$

subject to

$$\begin{aligned} 2x_1 - x_2 - 2x_3 - x_0 &\leq 4 \\ 2x_1 - 3x_2 - x_3 - x_0 &\leq -1 \\ -x_1 + x_2 + x_3 - x_0 &\leq -2 \\ x_0, x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Adding slack variables we get the initial dictionary:

$$\begin{aligned} x_4 &= 4 - 2x_1 + x_2 + 2x_3 + x_0 \\ x_5 &= -5 - 2x_1 + 3x_2 + x_3 + x_0 \\ x_6 &= -1 + x_1 - x_2 - x_3 + x_0 \\ w &= - x_0 \end{aligned}$$

For the special pivot to feasibility, x_0 enters and x_5 exits (since -5 is the most negative value of any slack variable).

$$\begin{array}{rrrrrr} x_0 = & 5 & +2x_1 & -3x_2 & -x_3 & +x_5 \\ x_4 = & 9 & & -2x_2 & +x_3 & +x_5 \\ x_6 = & 4 & +3x_1 & -4x_2 & -2x_3 & +x_5 \\ w = & -5 & -2x_1 & +3x_2 & +x_3 & -x_5 \end{array}$$

Now standard pivots: x_2 enters, x_6 exits.

$$\begin{array}{rrrrrr} x_0 = & 2 & -\frac{1}{4}x_1 & +\frac{1}{2}x_3 & +\frac{1}{4}x_5 & +\frac{3}{4}x_6 \\ x_2 = & 1 & +\frac{3}{4}x_1 & -\frac{1}{2}x_3 & +\frac{1}{4}x_5 & -\frac{1}{4}x_6 \\ x_4 = & 7 & -\frac{3}{2}x_1 & +2x_3 & +\frac{1}{2}x_5 & +\frac{1}{2}x_6 \\ w = & -2 & +\frac{1}{4}x_1 & -\frac{1}{2}x_3 & -\frac{1}{4}x_5 & -\frac{3}{4}x_6 \end{array}$$

Now x_1 enters, x_4 exits.

$$\begin{array}{rrrrrr} x_0 = & \frac{5}{6} & +\frac{1}{6}x_3 & +\frac{1}{6}x_4 & +\frac{1}{6}x_5 & +\frac{2}{3}x_6 \\ x_1 = & \frac{14}{3} & +\frac{4}{3}x_3 & -\frac{2}{3}x_4 & +\frac{1}{3}x_5 & +\frac{1}{3}x_6 \\ x_2 = & \frac{9}{2} & +\frac{1}{2}x_3 & -\frac{1}{2}x_4 & +\frac{1}{2}x_5 & \\ w = & -\frac{5}{6} & -\frac{1}{6}x_3 & -\frac{1}{6}x_4 & -\frac{1}{6}x_5 & -\frac{2}{3}x_6 \end{array}$$

This dictionary is optimal, so we discover that for the auxiliary LP the maximum value of w is $-\frac{5}{6}$, or, in other words, the minimum value of x_0 is $\frac{5}{6}$. This tells us that the original LP is infeasible. (So we're done for the original LP: there is no Phase 2, and we just answer: "the LP is infeasible".)

Let's go over *why* this means that the original is infeasible. Indeed, we just learned that in the auxiliary LP there are feasible solutions with $x_0 = \frac{5}{6}$ but no feasible solutions with any smaller x_0. That is, the inequalities:

$$\begin{aligned} 2x_1 - x_2 - 2x_3 - \frac{5}{6} &\leq 4 \\ 2x_1 - 3x_2 - x_3 - \frac{5}{6} &\leq -1 \\ -x_1 + x_2 + x_3 - \frac{5}{6} &\leq -2 \\ x_1, x_2, x_3 &\geq 0, \end{aligned}$$

have a simultaneous solution (namely, $x_1 = \frac{14}{3}$, $x_2 = \frac{9}{2}$, $x_3 = 0$, as we see from the final dictionary), but, if we replace $\frac{5}{6}$ with any small number they have no solution. The inequalities of the original LP correspond to setting $x_0 = 0$ and so they have no solution —which is exactly what it means to say the LP is infeasible.

7.3.1 Sneak Preview: "Magic Coefficients"

The w-row of the final dictionary contains information you can use to find a quick proof of infeasibility for the original LP:

$$w = -\frac{5}{6} - \frac{1}{6}x_3 - \frac{1}{6}x_4 - \frac{1}{6}x_5 - \frac{2}{3}x_6.$$

The coefficients of the *slack* variables x_4, x_5 and x_6 are $-\frac{1}{6}$, $-\frac{1}{6}$ and $-\frac{2}{3}$, respectively. The negatives of these numbers can be used as coefficients for a linear combination of the constraints of the original LP. Multiply both sides of each inequality by the corresponding coefficient and add them up:

$$\begin{aligned}
\frac{1}{6}(2x_1 - x_2 - 2x_3) &\leq \frac{1}{6}(4) \\
\frac{1}{6}(2x_1 - 3x_2 - x_3) &\leq \frac{1}{6}(-1) \\
\frac{2}{3}(-x_1 + x_2 + x_3) &\leq \frac{2}{3}(-2) \\
\hline
\frac{1}{6}x_3 &\leq -\frac{5}{6}
\end{aligned}$$

The answer, $\frac{1}{6}x_3 \leq -\frac{5}{6}$, must true for any feasible solution, but this is impossible because x_3 has a positivity constraint! this contradiction proves there can be no feasible solution.

We won't be able to prove this recipe works (i.e., that the coefficients of the slacks to combine the inequalities always gives an obvious contradiction for infeasible problem) until we discuss Duality Theory later on.

8 Degenerate pivots and cycling

8.1 Degenerate pivots

In each pivot of the Simplex Method we are attempting to do two things simultaneously: (1) maintain feasibility of the dictionary (this restricts the choice of *exiting* variable), (2) increase the value of the objective function (which restricts the choice of *entering* variable). It's in fact not always possible to achieve the second goal: sometimes we are forced to perform **degenerate pivots** that do not increase the objective function, but merely keep it the same. (Following our pivoting rules does at least guarantee that the objective function never decreases.)

When does a degenerate pivot occur? Well, when you pick the entering variable you pick a non-basic variable with positive coefficient in the objective function, so that any actual increase in the entering variable will also increase the objective function. This means that a degenerate pivot can only occur if the entering variable stays at zero without actually increasing. How much the entering variable is alowed to increase depends on the formulas for the basic variables in the dictionary: it is stuck at zero only if there is a basic variable whose value in the associated solution is zero and that includes the entering variable with a negative coefficient.

A **degenerate dictionary** is one in which at least one of the basic variables has the value zero; the associated solution is called a **degenerate solution**. The above discussion says that a degenerate pivot can only happen starting from a degenerate dictionary, but not every pivot from a degenerate dictionary is degenerate: for a degenerate pivot to occur you don't just need a basic variable to be zero, you need one *that has the entering variable with negative coefficient* to have the value zero.

8.1.1 An example of degeneracy

Consider the following LP in standard form:

Maximize $2x_1 + 2x_2 + x_3$

subject to

$$\begin{array}{rrrl} 2x_1 & & +x_3 & \leq 4 \\ x_1 & +x_2 & & \leq 1 \\ x_1 & & +x_3 & \leq 1 \\ x1, & x2, & x3 & \geq 0 \end{array}$$

Add slack variables to get the initial dictionary (no need for Phase 1 here):

$$\begin{array}{rrrrr} x_4 = & 4 & -2x_1 & & -x_3 \\ x_5 = & 1 & -x_1 & -x_2 & \\ x_6 = & 1 & -x_1 & & -x_3 \\ z = & & +2x_1 & +2x_2 & +x_3 \end{array}$$

Now x_1 enters, x_5 exits.

$$\begin{array}{rrrrr} x_1 = & 1 & -x_2 & & -x_5 \\ x_4 = & 2 & +2x_2 & -x_3 & +2x_5 \\ x_6 = & & +x_2 & -x_3 & +x_5 \\ z = & 2 & & +x_3 & -2x_5 \end{array}$$

Here x_3 enters. Of the basic variables, two limit the increase of the entering variable: x_4 tells us that x_3 can only increase up to 2, but x_6 tells us that x_3 can only "increase" up to zero! So we have a degenerate pivot in which x_6 is the exiting variable. The next dictionary is:

$$\begin{aligned} x_1 &= 1 - x_2 - x_5 \\ x_3 &= \quad + x_2 + x_5 - x_6 \\ x_4 &= 2 + x_2 + x_5 + x_6 \\ z &= 2 + x_2 - x_5 - x_6 \end{aligned}$$

Notice that this dictionary has the *same* associated solution as the previous one, namely, $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = 2$, $x_5 = 0$, $x_6 = 0$, $z = 2$. The difference is not in the values of the variables, but rather in which variables are basic.

Let's continue: x_2 enters, x_1 exits (a non-degenerate pivot).

$$\begin{aligned} x_2 &= 1 - x_1 - x_5 \\ x_3 &= 1 - x_1 \quad - x_6 \\ x_4 &= 3 - x_1 \quad + x_6 \\ z &= 3 - x_1 - 2x_5 - x_6 \end{aligned}$$

And we've reached an optimal solution!

8.1.2 Degenerate dictionaries don't always lead to degenerate pivots

Notice that if instead of the dictionary where we made a degenerate pivot we had instead come across a dictionary like the following, the next pivot would not have been degenerate, even though the dictionary is degenerate:

$$\begin{aligned} x_1 &= 1 - x_2 \quad - x_5 \\ x_4 &= 2 + 2x_2 - x_3 + 2x_5 \\ x_6 &= \quad + x_2 + x_3 + x_5 \\ z &= 2 \quad + x_3 - 2x_5 \end{aligned}$$

Here, the dictionary is degenerate because the basic variable x_6 is zero in the associated solution, but the pivot we'd make from here has x_3 entering and x_4 exiting. This pivot does increase x_3 (to 2) and is thus non-degenerate.

8.2 Cycling

Chvátal has a wonderful example showing that the standard rule can get trapped in a cycle of degenerate pivots!

Maximize $10x_1 - 57x_2 - 9x_3 - 24x_4$

subject to

$$\begin{aligned} \frac{1}{2}x_1 - \frac{11}{2}x_2 - \frac{5}{2}x_3 + 9x_4 &\leq 0 \\ \frac{1}{2}x_1 - \frac{3}{2}x_2 - \frac{1}{2}x_3 + x_4 &\leq 0 \\ x_1 &\leq 1 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

Putting in the slack variables:

$$\begin{array}{rrrrrr} x_5 = & & -\frac{1}{2}x_1 & +\frac{11}{2}x_2 & +\frac{5}{2}x_3 & -9x_4 \\ x_6 = & & -\frac{1}{2}x_1 & +\frac{3}{2}x_2 & +\frac{1}{2}x_3 & -x_4 \\ x_7 = & 1 & -x_1 & & & \\ z = & & +10x_1 & -57x_2 & -9x_3 & -24x_4 \end{array}$$

Here, x_1 enters and x_5 exits.

$$\begin{array}{rrrrrr} x_1 = & & +11x_2 & +5x_3 & -18x_4 & -2x_5 \\ x_6 = & & -4x_2 & -2x_3 & +8x_4 & +x_5 \\ x_7 = & 1 & -11x_2 & -5x_3 & +18x_4 & +2x_5 \\ z = & & +53x_2 & +41x_3 & -204x_4 & -20x_5 \end{array}$$

Now x_2 enters, x_6 exits.

$$\begin{array}{rrrrrr} x_1 = & & -\frac{1}{2}x_3 & +4x_4 & +\frac{3}{4}x_5 & -\frac{11}{4}x_6 \\ x_2 = & & -\frac{1}{2}x_3 & +2x_4 & +\frac{1}{4}x_5 & -\frac{1}{4}x_6 \\ x_7 = & 1 & +\frac{1}{2}x_3 & -4x_4 & -\frac{3}{4}x_5 & +\frac{11}{4}x_6 \\ z = & & +\frac{29}{2}x_3 & -98x_4 & -\frac{27}{4}x_5 & -\frac{53}{4}x_6 \end{array}$$

Here x_3 enters, x_1 exits.

$$\begin{array}{rrrrrr} x_2 = & & +x_1 & -2x_4 & -\frac{1}{2}x_5 & +\frac{5}{2}x_6 \\ x_3 = & & -2x_1 & +8x_4 & +\frac{3}{2}x_5 & -\frac{11}{2}x_6 \\ x_7 = & 1 & -x_1 & & & \\ z = & & -29x_1 & +18x_4 & +15x_5 & -93x_6 \end{array}$$

Next x_4 enters, x_2 exits.

$$\begin{array}{rrrrrr} x_3 = & & +2x_1 & -4x_2 & -\frac{1}{2}x_5 & +\frac{9}{2}x_6 \\ x_4 = & & +\frac{1}{2}x_1 & -\frac{1}{2}x_2 & -\frac{1}{4}x_5 & +\frac{5}{4}x_6 \\ x_7 = & 1 & -x_1 & & & \\ z = & & -20x_1 & -9x_2 & +\frac{21}{2}x_5 & -\frac{141}{2}x_6 \end{array}$$

Then x_5 enters, x_3 exits.

$$\begin{array}{rrrrrr} x_4 = & & -\frac{1}{2}x_1 & +\frac{3}{2}x_2 & +\frac{1}{2}x_3 & -x_6 \\ x_5 = & & +4x_1 & -8x_2 & -2x_3 & +9x_6 \\ x_7 = & 1 & -x_1 & & & \\ z = & & +22x_1 & -93x_2 & -21x_3 & +24x_6 \end{array}$$

Now x_6 enters, x_4 exits.

$$\begin{array}{rrrrrr} x_5 = & & -\frac{1}{2}x_1 & +\frac{11}{2}x_2 & +\frac{5}{2}x_3 & -9x_4 \\ x_6 = & & -\frac{1}{2}x_1 & +\frac{3}{2}x_2 & +\frac{1}{2}x_3 & -x_4 \\ x_7 = & 1 & -x_1 & & & \\ z = & & +10x_1 & -57x_2 & -9x_3 & -24x_4 \end{array}$$

And we've returned to the very first dictionary!

The moral of the story is that using the standard pivoting rule can lead to cycling. This is not really a problem for two different reasons:

1. There are other pivoting rules that do guarantee termination, such as Bland's Rule.
2. Cycling with the standard rule is pretty unlikely, in particular, you shouldn't fear degenerate pivots as they normally do *not* lead to cycling. Cycling is so rare, many computer implementations don't bother checking for it![citation needed]

So we'll stick with the standard rule for quizzes and such.

In this particular example, we could have chosen to pivot differently at the last step: instead of following the standard rule and choosing x_6 to enter, we could choose x_1 to enter. Then x_4 would still exit and instead of coming back to the first dictionary we would have:

$$\begin{array}{rrrrrr} x_1 = & & +3x_2 & +x_3 & -2x_4 & -2x_6 \\ x_5 = & & +4x_2 & +2x_3 & -8x_4 & +x_6 \\ x_7 = & 1 & -3x_2 & -x_3 & +2x_4 & +2x_6 \\ z = & & -27x_2 & +x_3 & -44x_4 & -20x_6 \end{array}$$

Now the only choice is for x_3 to enter and x_7 to exit, which is finally a non-degenerate pivot! And, luckily, we get an optimal dictionary:

$$\begin{array}{rrrrrr} x_1 = & 1 & & & & -x_7 \\ x_3 = & 1 & -3x_2 & +2x_4 & +2x_6 & -x_7 \\ x_5 = & 2 & -2x_2 & -4x_4 & +5x_6 & -2x_7 \\ z = & 1 & -30x_2 & -42x_4 & -18x_6 & -x_7 \end{array}$$

9 Matrix formulas for dictionaries

9.1 Matrix form of a linear program

Just like linear algebra benefits from formulating linear systems of equations in matrix notation, so too does linear programming. Recall our standard form linear program:

Maximize $c_1x_1 + c_2x_2 + \cdots c_nx_n$

subject to

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \\ x_1, x_2, \ldots, x_n &\geq 0 \end{aligned}$$

We can rewrite this is matrix form, by setting:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top$,
- $\mathbf{c} = (c_1, c_2, \ldots, c_n)^\top$,
- $\mathbf{b} = (b_1, b_2, \ldots, b_m)^\top$, and
- $A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$.

With those definitions the LP becomes:

Maximize $\mathbf{c}^\top\mathbf{x}$

subject to $\begin{cases} A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases}$

9.2 Adding the slack variables

The first thing we did to solve this LP in the Simplex Method was add slack variables $x_{n+1}, \ldots, x_{n+m}$, one per constraint (other than the positivity constraints). In matrix terms that corresponds to considering the following linear program:

$$\text{Maximize } \mathbf{c}^\top \mathbf{x}_{\mathsf{dec}}$$
$$\text{subject to } \begin{cases} A\mathbf{x}_{\mathsf{dec}} + \mathbf{x}_{\mathsf{slack}} = \mathbf{b} \\ \mathbf{x}_{\mathsf{dec}}, \mathbf{x}_{\mathsf{slack}} \geq 0, \end{cases}$$

where $\mathbf{x}_{\mathsf{dec}} = (x_1, x_2, \ldots, x_n)^\top$ is the vector of original or decision variables and $\mathbf{x}_{\mathsf{slack}} = (x_{n+1}, x_{n+2}, \ldots, x_{n+m})^\top$ is the vector of slack variables.

We can rewrite this last LP in terms of the augmented $m \times (n+m)$ matrix $A^{\mathsf{aug}} = \begin{pmatrix} A & I \end{pmatrix}$, where I denotes the $m \times m$ identity matrix. Using $\mathbf{x}$ to now denote the vector of *all* variables, both original and slack, and setting $\mathbf{c}^{\mathsf{aug}} = (c_1, \ldots, c_n, 0, \ldots, 0)^\top$ (with a zero for every slack variable) the LP is:

$$\text{Maximize } (\mathbf{c}^{\mathsf{aug}})^\top \mathbf{x}$$
$$\text{subject to } \begin{cases} A^{\mathsf{aug}}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases}$$

9.3 Dictionaries

A warm up, let's consider the initial dictionary. Roughly speaking, it looks like:

> "slack variables = right-hand sides of constraints - left-hand sides"
>
> "z = objective function"

This is exactly what we get by solving for $\mathbf{x}_{\mathsf{slack}}$ in the next to last LP we wrote above:

$$\mathbf{x}_{\mathsf{slack}} = \mathbf{b} - A\mathbf{x}_{\mathsf{dec}}$$
$$z = \mathbf{c}^\top \mathbf{x}_{\mathsf{dec}}$$

Later on in the Simplex Method, after performing some pivots, the non-basic variables are no longer $\mathbf{x}_{\mathsf{dec}}$ and the basic variables are no longer $\mathbf{x}_{\mathsf{slack}}$.

Let's say that we are interested in some dictionary where the basic variables for a vector $\mathbf{x}_B$ and the non-basic variables form a vector $\mathbf{x}_N$. Let A_B and A_N denote the submatrices of A^{aug} formed by the columns corresponding to basic and non-basic variables, respectively. The first order of business is to see that the equation $A^{\mathsf{aug}}\mathbf{x} = \mathbf{b}$ can be written as $A_B\mathbf{x}_B + A_N\mathbf{x}_N = b$.

One way to think about this is to use block matrix multiplication. Think of ordering the variables in $\mathbf{x}$ so that all the basics come first and the vector

is partitioned as $\mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix}$; rearranging the columns of A^{aug} in the corresponding order, we get $A^{\text{aug}} = \begin{pmatrix} A_B & A_N \end{pmatrix}$. Then, by block multiplication, $A^{\text{aug}}\mathbf{x} = \begin{pmatrix} A_B & A_N \end{pmatrix} \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix} = A_B\mathbf{x}_B + A_N\mathbf{x}_N$.

I definitely encourage everyone to learn all about block matrix multiplication, but there is a less fancy way to deal with this particular situation: if you have a matrix M with columns $M_1, M_2, \ldots, M_p$ and a vector $\mathbf{v} = (v_1, \ldots, v_p) =$, then the product $M\mathbf{v}$ is the linear combination of the columns of M with coefficients the entries of the vector $\mathbf{v}$, in symbols, $M\mathbf{v} = v_1M_1 + \cdots + v_pM_p$ —which you can easily check by computing both sides. So, if the columns of A^{aug} are $A_1, \ldots, A_{m+n}$, then $A^{\text{aug}}\mathbf{x} = A_B\mathbf{x}_B + A_N\mathbf{x}_N$ simply because both sides work out to be $x_1A_1 + \cdots + x_{m+n}A_{m+n}$.

OK, now that we've justified the equation, solving for $\mathbf{x}_B$, we get $\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N$. This is the top part of the dictionary that gives formulas for the basic variables. To get the z-row, we can split the entries of $\mathbf{c}^{\text{aug}}$ into the coefficients of basic and non-basic variables, say, $\mathbf{c}_B$ and $\mathbf{c}_N$. Then the objective function is

$$\begin{aligned} z &= (\mathbf{c}^{\text{aug}})^\top x \\ &= \mathbf{c}_B^\top \mathbf{x}_B + \mathbf{c}_N^\top \mathbf{x}_N \\ &= \mathbf{c}_B^\top (A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N) + \mathbf{c}_N^\top \mathbf{x}_N \\ &= \mathbf{c}_B^\top A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N \end{aligned}$$

Putting it all together, we see that the dictionary whose basic variables are $\mathbf{x}_B$ is given by

$$\begin{aligned} \mathbf{x}_B &= A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N \\ z &= \mathbf{c}_B^\top A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N \end{aligned}$$

10 Bland's Rule guarantees termination

Robert G. Bland]] of Cornell University discovered a pivoting rule that is very similar to what we've been calling the standard rule, but that guarantees that the Simplex Method terminates!

Bland's Rule: For the entering variable choose, among the variables with a positive coefficient in the objective function, the one with the smallest index. For the exiting variable choose, the variable imposing the strictest

limit on the increase of the entering variable, breaking ties by choosing the variable with smallest index.

Notice that the rule for the exiting variable is the *same* as the standard rule for exiting variables.

Theorem 1. *The Simplex Method using Bland's rule is guaranteed to terminate.*

We'll present the proof from Chvátal's textbook:

Proof. First we'll explain why the only way for the Simplex Method to fail to finish when following a given pivoting rule is for it to **cycle**, which means to get stuck in a loop of degenerate pivots that bring you back exactly to a previous dictionary (and once a dictionary repeats, the rule your following will make you choose the same pivots over and over again).

Imagine that the Simplex Method fails to finish, pivoting for all eternity. Since there are only a finite number of possible sets of basic variables, at some point you must reach a dictionary with exactly the same set of basic variables as a previous dictionary. But then not just the basic variables but the whole dictionary must repeat: the matrix formulas for the dictionary we found prove that once you know which variables are basic, the entire dictionary is determined.

So now all we need to show is that following Bland's rule we never repeat a dictionary (or equivalently, never repeat a basis).

Assume for the sake of obtaining a contradiction that D_0, D_1, $\ldots$, D_{k-1}, $D_k = D_0$ is a cycle of dictionaries each obtained from the previous by Bland's rule. Of course, all of the pivots must be degenerate since otherwise the value of the objective function would be larger in the last dictionary than in the first.

For the purposes of this proof, call a variable *fickle* if it enters or leaves the basis at some point on the cycle; that is, if it is basic in some dictionary and non-basic in some other dictionary of the cycle.

Let x_t be the fickle variable with largest index. For some dictionary D_p in the cycle, x_t must exit with, say, x_s entering. At some later point x_t must enter the basis again, let's say that it is chosen to enter in the pivot from dictionary D_q to D_{q+1}.

Say the dictionary D_p looks like:

$$x_i = b_i - \sum_{j \notin B_p} a_{ij} x_j$$
$$z = v + \sum_{j \notin B_p} c_j x_j$$

(Here B_p is the set of indices of basic variables for dictionary D_p, so the sums $\sum_{j \notin B_p}$ range over non-basic variables for D_p, as you'd expect.)

Since all the pivots are degenerate in D_q the objective function still has value v, so its last row must look like $z = v + \sum_{j=1}^{m+n} c_j^* x_j$ (where, for convenience, we've listed all the variables, but any basic variable for D_q has coefficient 0).

All the dictionaries are systems of equations with the same solutions, so any solution (even if not feasible) of D_p we manage to write down must also satisfy D_q. Take, for the non-basic variable in D_p the values $x_s = y$, $x_j = 0$ for $j \neq s, j \notin B_p$, which force $x_i = b_i - a_{is}y$ for the basic variables. Since this choice must also solve D_q, we have, equating the objective functions, that

$$v + c_s y = v + c_s^* y + \sum_{i \in B_p} c_i^* (b_i - a_{is} y).$$

This must be true for any value of y, so the two sides must be the same function of y. The graph of each side as a function of y is a straight line and since they are equal they must have the same slope, so we learn that

$$c_s = c_s^* - \sum_{i \in B_p} c_i^* a_{is}. \tag{1}$$

Now we will gather some inequalities about these various quantities with the eventual aim of showing that Bland's Rule would not actually have chosen x_t to exit the basis of D_p. That contradiction will show that Bland's Rule cannot lead to cycling.

- $c_s > 0$. This is because x_s is chosen as the entering variable for D_p, so it must appear in the objective function with positive coefficient.
- $c_s^* \leq 0$ and $c_t^* > 0$. This is because in D_q the entering variable is x_t and according to Bland's rule, x_t must the variable with smallest index that has a positive coefficient in the z-row of D_q. Since $s < t$ (because we assumed x_t had the largest index among fickle variables), we have $c_s^* \leq 0$.

Then, by equation (1), we must have that $\sum_{i \in B_p} c_i^* a_{is} < 0$, so at least one of the terms $c_i^* a_{is}$ must be negative. What can we say about the variable x_i?

- It's basic in D_p because the sum only ranges over $i \in B_p$.
- It's non-basic in D_q because $c_i^* \neq 0$ (if c_i^* *were* 0, then $c_i^* a_{is}$ would be zero rather than negative).

- Therefore x_i is fickle and $i \leq t$ (because t was chosen to be the largest index of a fickle variable).
- But we can't have $i = t$! This is because i was chosen so that $c_i^* a_{is} < 0$ but $c_t^* a_{ts} > 0$: we already saw that $c_t^* > 0$; and in dictionary D_p the variable x_s was chosen to enter and x_t was chosen to exit, so $a_{ts} > 0$ because x_t had to impose a restriction on the increase of x_s.

Now we can argue that Bland's Rule actually would have chosen x_i over x_t to exit the basis of dictionary D_p. We know that $i < t$, so if x_i is viable candidate for the exiting variable, then it would be chosen. We need to show:

1. That $a_{is} > 0$, so that x_i does impose a limit on how far x_s could increase.

   To see this, it is enough to show $c_i^* \leq 0$, because i was chosen to satisfy $c_i^* a_{is} < 0$. Recall that x_t was selected by Bland's Rule to enter the basis when the dictionary was D_q, so we must have $c_i^* \leq 0$ because we know $i < t$.

2. That the restriction x_i imposes, $x_s \leq b_i / a_{is}$, is at least as strong as the restriction imposed by x_t, which is $x_s \leq b_t / a_{ts}$.

   In fact both $b_i = b_t = 0$! Every pivot in a cycle must be degenerate so that all variables conserve their values throughout the process and every *fickle* variable must therefore be zero in the solution associated to any dictionary in the cycle.

□

11 Recap of the Simplex Method

- Convert your LP to standard form if necessary.
- If the right-hand sides of the constraints are all non-negative, add slack variables, write the initial dictionary (by solving for the slack variables) and skip to Phase 2, otherwise perform Phase 1.
- *Phase 1*.
  - Form the auxiliary LP by adding a variable x_0 that you subtract from all the left-hand sides of constraints and by replacing the objective function with $w = -x_0$.

- Add slack variables and write down the initial dictionary, which will be infeasible.
- Perform the special pivot to feasibility and then successive standard pivots until you reach an optimal solution of the auxiliary LP (this is guaranteed to happen!).
- If the optimal value is negative, then original LP is **infeasible**. If the optimal value is zero, then the final dictionary for the auxiliary LP furnishes a starting dictionary for Phase 2: set $x_0 = 0$ in the top part of the dictionary and replace the w-row by the result of plugging in this top part into the formula for the original objective function z.

- *Phase 2*. Repeatedly perform standard pivots until either you reach an **optimal solution** (when all coefficients or variables in the z-row are ≤ 0) or you see the LP is **unbounded** (when there is a choice of entering variable upon which no basic variable imposes a limit).

12 Duality theorems

12.1 Bounding the optimal value

As motivation for the dual, let's discuss how we can bound the **optimal value** of a linear program. Let's stick to the case of a linear program in standard form. We'll illustrate with the following example:

Maximize $x_1 - 2x_2$

subject to

$$\begin{aligned} x_1 + 3x_2 &\leq 4 \\ x_1 - 4x_2 &\leq 2 \\ x_1, x_2 &\geq 0 \end{aligned}$$

12.1.1 Lower bounds

For a maximization problem, finding a lower bound is simply a matter of finding a feasible solution: the value the objective function takes on *some* feasible solution is certainly less than or equal to the *maximum* value it can take on feasible solutions! Here, for instance, we can easily see that $x_1 = 2, x_2 = 0$ is feasible, which shows the optimal value is at least $x_1 - 2x_2 = 2$.

12.1.2 Upper bounds

Finding an upper bound for a maximization problem is necessarily more subtle: to show that, for example, 2.5 is an upper bound[12] for the optimal value, we would need some argument that explains why *none* of the feasible solutions could possibly give an objective function value of more than 2.5.

Here's one example of an upper bound we can prove for the optimal value in the example LP:

> Suppose x_1, x_2 are a feasible solution of the LP. Then, $x_1, x_2 \geq 0$ and $x_1 + 3x_2 \leq 4$, from which we can conclude that $x_1 \leq 4$ —and $3x_2 \leq 4$, or, $x_2 \leq \frac{4}{3}$. Because $x_1 \leq 4$ and $x_2 \geq 0$, we conclude $z = x_1 - 2x_2 \leq 4$.

How can we search for better (i.e., smaller) upper bounds? Here we only used the first inequality in the LP (plus the positivity constraints), we can try combining both. For example, multiplying the first constraint by $\frac{1}{3}$ and the second by $\frac{2}{3}$ and adding we get $\frac{1}{3}(x_1 + 3x_2) + \frac{2}{3}(x_1 - 4x_2) \leq \frac{4}{3} + \frac{4}{3}$, that is, $x_1 - \frac{5}{3}x_2 \leq \frac{8}{3}$. Since $x_2 \geq 0$, we have $z = x_1 - 2x_2 \leq x_1 - \frac{5}{3}x_2 \leq \frac{8}{3}$.

Can we find even better upper bound by using different coefficients? Let's be systematic about it. Say we multiply the first constraint by y_1 and the second by y_2. To keep the inequalities going in the same direction we will require $y_1, y_2 \geq 0$. Adding those multiples of the constraints we get:

$$y_1(x_1 + 3x_2) + y_2(x_1 - 4x_2) \leq 4y_1 + 2y_2.$$

When will the left hand side be $\geq z$? Let's regroup to see the coefficients of x_1 and x_2 explicitly, we can rewrite the inequality as:

$$(y_1 + y_2)x_1 + (3y_1 - 4y_2)x_2 \leq 4y_1 + 2y_2.$$

Since $z = x_1 - 2x_2$ (and $x_1, x_2 \geq 0$!), we will have $z \leq (y_1 + y_2)x_1 + (3y_1 - 4y_2)x_2$ as long as:

$$\begin{aligned} y_1 + y_2 &\geq 1 \\ 3y_1 - 4y_2 &\geq -2 \end{aligned}$$

All together we have shown that whenever $y_1, y_2 \geq 0$ satisfy the above inequalities, then the number $4y_1 + 2y_2$ is an upper bound for the optimal value of the LP we started with.

[12] We'll see that it is not! But this is irrelevant to the point being made.

We might now ask what is the best (i.e., smallest) upper bound obtainable by this method? Well, the answer is given by the optimal value of the following LP called the **dual** of the original LP:

Minimize $4y_1 + 2y_2$

subject to

$$\begin{aligned} y_1 + y_2 &\geq 1 \\ 3y_1 - 4y_2 &\geq -2 \\ y_1, y_2 &\geq 0 \end{aligned}$$

12.2 The dual of a linear program in standard form

Repeating the steps that led to the dual LP in the above example but now for a general LP in standard form, we see that the dual of:

Maximize $c_1x_1 + c_2x_2 + \cdots c_nx_n$

subject to

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \\ x_1, x_2, \ldots, x_n &\geq 0 \end{aligned}$$

is given by:

Minimize $b_1y_1 + b_2y_2 + \cdots b_my_m$

subject to

$$\begin{aligned} a_{11}y_1 + a_{21}y_2 + \cdots + a_{m1}y_m &\geq c_1 \\ a_{12}y_1 + a_{22}y_2 + \cdots + a_{m2}y_m &\geq c_2 \\ &\vdots \\ a_{1n}y_1 + a_{2n}y_2 + \cdots + a_{mn}y_m &\geq c_m \\ y_1, y_2, \ldots, y_n &\geq 0 \end{aligned}$$

Or, even better, in matrix form the dual of:

Maximize $\mathbf{c} \cdot \mathbf{x}$

subject to $\begin{cases} A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases}$

is the problem:

$$\text{Minimize } \mathbf{b} \cdot \mathbf{y}$$
$$\text{subject to } \begin{cases} A^\top \mathbf{y} \geq \mathbf{c} \\ \mathbf{y} \geq 0 \end{cases}$$

This formula for dual of a LP *only* applies to problems in standard form! To find the dual of a LP that is *not* in standard form you need to convert to standard form first. A good exercise is to show that the dual of the dual of a problem is equivalent to the problem you started with!

12.3 Weak duality theorem

From the way we constructed the dual it is clear that the value of the dual objective function on any feasible solution of the dual is an upper bound for the objective function of the original or **primal** problem. This fact is important enough to be recorded as a theorem:

Theorem 2. ***Weak duality****. Consider the following pair of dual linear programs:*

Primal LP	*Dual LP*
max $\mathbf{c} \cdot \mathbf{x}$	*min* $\mathbf{b} \cdot \mathbf{y}$
subject to	*subject to*
$A\mathbf{x} \leq \mathbf{b}$	$A^\top \mathbf{y} \geq \mathbf{c}$
$\mathbf{x} \geq 0$	$\mathbf{y} \geq 0$

If $\mathbf{x}$ *is any feasible solution of the primal and* $\mathbf{y}$ *is any feasible solution of the dual, then* $\mathbf{c} \cdot \mathbf{x} \leq \mathbf{b} \cdot \mathbf{y}$.

Moreover, if equality holds, that is if $\mathbf{c} \cdot \mathbf{x} = \mathbf{b} \cdot \mathbf{y}$, *then* $\mathbf{x}$ *is be an optimal solution of the primal LP and* $\mathbf{y}$ *is an optimal solution of the dual LP.*

Proof. The first part, the part before "moreover", we already proved at the same time as we introduced the dual!

Here's another proof though, to get some practice with matrix manipulations:

Assume $\mathbf{x}$ is feasible for the primal and $\mathbf{y}$ is feasible for the dual. Then

$$\mathbf{c}^\top \mathbf{x} \leq (A^\top \mathbf{y})^\top \mathbf{x} = \mathbf{y}^\top A\mathbf{x} \leq \mathbf{y}^\top \mathbf{b}.$$

Where:

- the first inequality is because $A^\top \mathbf{y} \geq \mathbf{c}$ and $\mathbf{x} \geq 0$,

- the equality is by standard properties of the transpose,
- and the second inequality is because $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{y} \geq 0$.

Now for the "moreover" part.

Assume, as before, that $\mathbf{x}$ is feasible for the primal and $\mathbf{y}$ is feasible for the dual, and that $\mathbf{c} \cdot \mathbf{x} = \mathbf{b} \cdot \mathbf{y}$. Let's show that $\mathbf{x}$ is optimal for the primal LP; a very similar argument would show that $\mathbf{y}$ is optimal for the dual LP.

Let $\mathbf{x}'$ be any other feasible solution for the primal. To show that $\mathbf{x}$ is optimal, we need to show that $\mathbf{c} \cdot \mathbf{x}' \leq \mathbf{c} \cdot \mathbf{x}$. But by the first part of the theorem applied to $\mathbf{x}'$ and $\mathbf{y}$, we know that $\mathbf{c} \cdot \mathbf{x}' \leq \mathbf{b} \cdot \mathbf{y}$; and we are assuming that $\mathbf{b} \cdot \mathbf{y} = \mathbf{c} \cdot \mathbf{x}$, which finishes the proof. □

12.4 Possibilities for a primal-dual pair of LPs

Recall that the fundamental theorem of linear programming says that any LP falls into exactly one of three cases: it's infeasible, unbounded or has an optimal solution. For a LP and it's dual which combinations are possible? The duality theorems rule some combinations out. Already the weak duality theorem tells us that if both primal and dual are feasible, then neither can be unbounded ("each bounds the other"). A stronger version of the duality theorem, which we will discuss next rules out the possibility of one problem of the pair being infeasible while the other has an optimal solution. It turns out all other combinations are possible (this needs to be shown by giving examples of each case, maybe this would be a good problem for a later assignment!). Here's a table summarizing:

Primal/Dual	infeasible	unbounded	optimal
infeasible	yes	yes	no (SD)
unbounded	yes	no (WD)	no (WD)
optimal	no (SD)	no (WD)	yes

12.5 Strong duality theorem

Theorem 3. ***Strong duality****. Assume a primal-dual pair of linear problems satisfies* any *of the following conditions:*

1. *the primal has a optimal solution,*
2. *the dual has an optimal solution, or,*
3. *both primal and dual are feasible,*

Then both primal and dual must have optimal solutions (say $\mathbf{x}^*$ and $\mathbf{y}^*$) and the optimal values of the objective functions are equal ($\mathbf{c} \cdot \mathbf{x}^* = \mathbf{b} \cdot \mathbf{y}^*$, using the above notation for the primal and dual).

Proof. We'll first argue it is enough to show the following claim:

> If the primal has an optimal solution $\mathbf{x}^*$, then the dual has a feasible solution $\mathbf{y}^*$ such that $\mathbf{c} \cdot \mathbf{x}^* = \mathbf{b} \cdot \mathbf{y}^*$.

Notice that by the second part of the weak duality theorem, the conclusion of the claim implies that $\mathbf{y}^*$ is optimal for the dual. So the claim is enough to show strong duality in case 1. Case 2 is just case 1 applied to the dual problem, and case 3, by weak duality implies that both primal and dual have optimal solutions, so we can establish case 3 using either case 1 or 2.

Now we show the claim. Imagine we solve the primal problem by the Simplex Method and we get as final dictionary the one corresponding to a basis $\mathbf{x}_B$. Then the final dictionary is

$$\begin{aligned} \mathbf{x}_B &= A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N \\ z &= \mathbf{c}_B^\top A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N \end{aligned}$$

Since this is the final dictionary and we have an optimal solution, all the coefficients in the z-row must be non-positive, so we know that

$$\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N \leq 0.$$

That vector of coefficients has one entry for every non-basic variable, and the entry corresponding to a particular non-basic x_j is $c_j - \mathbf{c}_B^\top A_B^{-1}A_j$ (where A_j is the j-th column of A^{aug}, the column corresponding to x_j). So we've learned that $c_j - \mathbf{c}_B^\top A_B^{-1}A_j \leq 0$ for every j such that x_j is non-basic in the final dictionary.

We want to know that this inequality is also true for j's such that x_j is a basic variable. Those inequalities for x_j basic put together would say that

$$\mathbf{c}_B^\top - \mathbf{c}_B^\top A_B^{-1}A_B \leq 0,$$

which is certainly true because the left-hand side is zero!

So now we know that $c_j - \mathbf{c}_B^\top A_B^{-1}A_j \leq 0$ for every j. Now separate the variables into original variables and slack variables to get:

$$\mathbf{c}^\top - \mathbf{c}_B^\top A_B^{-1} A \leq 0 \tag{2}$$
$$\mathbf{0}^\top - \mathbf{c}_B^\top A_B^{-1} I \leq 0 \tag{3}$$

(Recall that the columns of A^{aug} corresponding to the original variables form the matrix A and the columns for slack variables form an identity matrix; also, the entries of $\mathbf{c}^{\mathsf{aug}}$ for the original variables form the vector $\mathbf{c}$ and the entries for the slacks are all 0.)

The feasible solution of the dual that we seek is given by ${\mathbf{y}^*}^\top = \mathbf{c}_B^\top A_B^{-1}$ (so $\mathbf{y}^* = (A_B^{-1})^\top \mathbf{c}_B$). We just need to check it really is feasible for the dual and that $\mathbf{c} \cdot \mathbf{x}^* = \mathbf{b} \cdot \mathbf{y}^*$:

Main constraints: $A^\top \mathbf{y}^* \geq \mathbf{c}$ is equivalent to (2) (just transpose both sides of the inequality).

Positivity constraints: $\mathbf{y}^* \geq 0$ is equivalent to (3).

Equality of objective functions: we have $\mathbf{b} \cdot \mathbf{y}^* = {\mathbf{y}^*}^\top \mathbf{b} = (\mathbf{c}_B^\top A_B^{-1})\mathbf{b}$, which, as you can see from the final dictionary for the primal, is the optimal value of the primal objective function.

□

12.6 Magic coefficients

Notice from the *proof* (not just the statement!) of the strong duality theorem that an optimal solution of the dual can be read off from the final dictionary for the primal! The proof shows you can take ${\mathbf{y}^*}^\top = \mathbf{c}_B^\top A_B^{-1}$ as an optimal dual solution, and looking at how we obtained (3), we see we can describe the vector $\mathbf{c}_B^\top A_B^{-1}$ more intuitively as *minus the coefficients of the (primal) slack variables in the z-row of the final dictionary*. We'll call this phenomenon, **magic coefficients**.

Take, for example, the following LP:

Maximize $x_1 + 2x_2 - x_3$

subject to

$$\begin{aligned} 2x_1 + x_2 + x_3 &\leq 14 \\ 4x_1 + 2x_2 + 3x_3 &\leq 28 \\ 2x_1 + 5x_2 + 5x_3 &\leq 30 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

After running the Simplex Method on it we obtained as final dictionary:

$$\begin{array}{rrrrr} x_1 = & 5 & & -\frac{5}{8}x_4 & +\frac{1}{8}x_6 \\ x_2 = & 4 & -x_3 & +\frac{1}{4}x_4 & -\frac{1}{4}x_6 \\ x_5 = & 0 & -x_3 & +2x_4 & \\ z = & 13 & -3x_3 & -\frac{1}{8}x_4 & -\frac{3}{8}x_6 \end{array}$$

The slack variables are x_4, x_5, x_6, which appear with coefficients $-\frac{1}{8}, 0, -\frac{3}{8}$ in the z-row (the coefficient of x_5 is 0 because it is basic in the final dictionary). So by "magic coefficients", $y_1 = \frac{1}{8}, y_2 = 0, y_3 = \frac{3}{8}$ is an[13] optimal solution of the dual.

13 Complementary Slackness

13.1 The theorem

Theorem 4. *Consider the usual primal-dual pair of problems where A is an $m \times n$ matrix:*

Primal LP	*Dual LP*
max $\mathbf{c} \cdot \mathbf{x}$	*min* $\mathbf{b} \cdot \mathbf{y}$
subject to	*subject to*
$A\mathbf{x} \leq \mathbf{b}$	$A^\top \mathbf{y} \geq \mathbf{c}$
$\mathbf{x} \geq 0$	$\mathbf{y} \geq 0$

Let $\mathbf{x}$ *be feasible for the primal LP and* $\mathbf{y}$ *be feasible for the dual. Then we have that* $\mathbf{x}$ *and* $\mathbf{y}$ *are optimal solution of their respective LPs if and only if* ***complementary slackness*** *holds:*

1. *For $1 \leq i \leq m$, either $y_i = 0$ or $b_i - \sum_{j=1}^n a_{ij}x_j = 0$ (or both).*
   Notice that $b_i - \sum_{j=1}^n a_{ij}x_j$ is the slack in the i-th constraint in the primal.
2. *For $1 \leq j \leq n$, either $x_j = 0$ or $\sum_{i=1}^m a_{ij}y_i - c_j = 0$ (or both).*
   Notice that $\sum_{i=1}^m a_{ij}y_i - c_j$ is the slack in the j-th constraint in the dual.

Proof. Each statement in this list is equivalent to the next:

- $\mathbf{x}$ and $\mathbf{y}$ are optimal for their respective LPs

[13] It's not the only one! Can you find all of them without using the Simplex Method on the dual?

$\iff$ [14] *(by weak duality)*

- $\mathbf{c}^\top \mathbf{x} = \mathbf{y}^\top \mathbf{b}$

  $\iff$ *(recall the proof of Weak Duality:* $\mathbf{c}^\top \mathbf{x} \leq \mathbf{y}^\top A\mathbf{x} \leq \mathbf{y}^\top \mathbf{b}$ *)*

- $\mathbf{c}^\top \mathbf{x} = \mathbf{y}^\top A\mathbf{x}$ *and* $\mathbf{y}^\top A\mathbf{x} = \mathbf{y}^\top \mathbf{b}$.

  $\iff$ *(algebra)*

- $\left(A^\top \mathbf{y} - \mathbf{c}\right)^\top \mathbf{x} = 0$ *and* $\mathbf{y}^\top (\mathbf{b} - A\mathbf{x}) = 0$.

  $\iff$ *(see below)*

- The complementary slackness conditions hold.

The last equivalence is for the following reason: all of $A^\top \mathbf{y} - \mathbf{c}$, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{b} - A\mathbf{x}$ are vectors with non-negative entries, so their dot product is zero if and only if for each j one of the two vectors has a zero in the j-th entry. □

13.2 Examples

One thing we can use complementary slackness for is to verify claims about optimal solutions.

Example 1. Say someone tells us that $x_1^* = \frac{9}{7}, x_2^* = 0, x_3^* = \frac{1}{7}$ is an optimal solution for the following LP:

Maximize $x_1 - 2x_2 + 3x_3$

subject to

$$\begin{aligned} x_1 + x_2 - 2x_3 &\leq 1 \\ 2x_1 - x_2 - 3x_3 &\leq 4 \\ x_1 + x_2 + 5x_3 &\leq 2 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Let's try to verify that claim.

At least those values satisfy the constraints! Now let's see what complementary slackness would tells us about an optimal solution y_1^*, y_2^*, y_3^* of the dual.

[14] Read this as "if and only if".

Because x_1^* and x_3^* are non-zero, the first and third constraints of the dual have no slack:

$$y_1^* + 2y_2^* + y_3^* = 1$$
$$-2y_1^* - 3y_2^* + 5y_3^* = 3$$

That's only two equations for three unknowns! But checking the primal, we see that the alleged optimal solution shows some slack in the second constraint (that is $2x_1^* - x_2^* - 3x_3^* = \frac{15}{7} < 4$), and thus, by complementary slackness again, $y_2^* = 0$. Plugging that in we get the following system:

$$y_1^* + y_3^* = 1$$
$$-2y_1^* + 5y_3^* = 3$$

The only solution is $y_1^* = \frac{2}{7}, y_3^* = \frac{5}{7}$. So the dual should have optimal solution $y_1^* = \frac{2}{7}, y_2^* = 0, y_3^* = \frac{5}{7}$. Does it really? We can use complementary slackness again to check! We need to check it is feasible for the dual and that complementary slackness holds, and most of that has already been done in the process of coming up with these values of the y_i^*.

We still need to check one inequality for feasibility: $y_1^* - y_2^* + y_3^* \geq -2$, which clearly holds.

And we still need to check, since $y_1^*, y_3^* > 0$, that in the primal the first and third constraints have no slack. Plugging in we see that indeed they don't, and so we conclude that the x_i^* and y_i^* are indeed optimal for the primal and dual respectively.

Example 2. Now imagine someone told us instead that $x_1^* = \frac{9}{7}, x_2^* = 0, x_3^* = \frac{1}{7}$ was an optimal solution for the following different LP:

Maximize $x_1 + 2x_2 + 3x_3$

subject to

$$x_1 + x_2 - 2x_3 \leq 1$$
$$2x_1 - x_2 - 3x_3 \leq 4$$
$$x_1 + x_2 + 5x_3 \leq 2$$
$$x_1, x_2, x_3 \geq 0$$

(The only difference is the sign of the coefficient of x_2 in the objective function!)

We can argue similarly to the first example and conclude that $y_1^* = \frac{2}{7}, y_2^* = 0, y_3^* = \frac{5}{7}$ is the only possibility for an optimal solution of the dual. But now this dual solution is not feasible! This shows that the proposed optimal solution of the primal is not actually optimal.

14 Theorem of the Alternative

We can apply duality theory of linear programs to prove statements that don't mention linear programming at all! Here's an example. Notice that the statement of the theorem doesn't mention maximizing or minimizing at all, just linear inequalities.

Theorem 5. *(**of the Alternative**) Let A be an $m \times n$ matrix and $\mathbf{b}$ be a vector with m entries. **Either** there exists a vector $\mathbf{x}$ such that $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$, **or** there exists a vector $\mathbf{y}$ such that $A^\top \mathbf{y} \geq 0$, $\mathbf{y} \geq 0$ and $\mathbf{b} \cdot \mathbf{y} < 0$, **but not both**.*

Proof. Since the alternatives both are reminiscent of LPs, let's try to build up a primal-dual pair based on the alternatives we're aiming for. First of all, $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ definitely look like primal constraints; $A^\top \mathbf{y} \geq 0$ and $\mathbf{y} \geq 0$ look like dual constraints. Now, what to do with $\mathbf{b} \cdot \mathbf{y} < 0$? Despite being an inequality, it doesn't look too much like a constraint: it says something is *strictly* less than zero and our constraints usually include the possibility of equality; also, the dot product looks more like an objective function for the dual. So let's try $\mathbf{b} \cdot \mathbf{y}$ as the dual objective function and deal with the "< 0" part later, in the course of the proof.

The only thing left to decide is what the objective function of the primal should be, but having the dual complete, we can just remember that the primal is the dual of the dual, and since the main constraint of the dual is $A^\top \mathbf{y} \geq 0$, the primal objective function must be $0 \cdot \mathbf{x}$. The finished pair looks like this:

Primal LP	Dual LP
$\max 0 \cdot \mathbf{x}$	$\min \mathbf{b} \cdot \mathbf{y}$
subject to	subject to
$A\mathbf{x} \leq \mathbf{b}$	$A^\top \mathbf{y} \geq 0$
$\mathbf{x} \geq 0$	$\mathbf{y} \geq 0$

In terms of these LPs, the theorem we are trying to prove can be expressed as follows:

> **Either** the primal is feasible, **or** the dual has a feasible solution with negative objective function, **but not both**.

Let's see what's special about these LPs:

- Since the objective function of the primal is the constant zero, the primal can't possibly be unbounded.

- The dual is automatically feasible! Indeed, taking $\mathbf{y} = 0$ we can satisfy both constraints in the dual.

So the primal can only be infeasible or have an optimal solution, while the dual can only be unbounded or have an optimal solution. Which combinations are possible? By strong duality, if either one has an optimal solution, then both do. So the only cases are:

Both primal and dual have optimal solutions: In particular the primal is feasible, which is the first of the alternatives we were aiming for.

The primal is infeasible and the dual is unbounded: In this case the dual has a family of feasible solutions with objective function tending[15] to $-\infty$. In particular, the dual must have feasible solutions with $\mathbf{b} \cdot \mathbf{y} < 0$. So in this case, the second alternative holds.

We still need to show that both alternatives can't happen at the same time, but that's just from weak duality: $\mathbf{x}$ would be feasible for the primal, $\mathbf{y}$ would be feasible for the dual, but then weak duality would tell us that $0 \cdot \mathbf{x} \leq \mathbf{b} \cdot \mathbf{y}$, which is incompatible with $\mathbf{b} \cdot \mathbf{y} < 0$. □

14.1 Variants of the theorem of the alternative

The theorem of the alternative presented above "corresponds" to the standard form of a linear program. There are many variations that deal with LPs that are in other forms. The basic strategy for all variants is to setup a primal dual-pair whose constraints and objective functions match the alternatives the variant is about and then reason using the basic theorems we have at our disposal: the fundamental theorem of linear programming and the duality theorems.

Notice that you can even use your knowledge of duals to come up with the full statement given one of the alternatives. For example, say we wanted a variant that goes like:

> **Either** there exists a vector $\mathbf{x}$ such that $A\mathbf{x} \leq \mathbf{b}$, **or** there exists a vector $\mathbf{y}$ such that ???, **but not both**.

We can write down the primal with those constraints for $\mathbf{x}$, and take it's dual (remember free variables produce equations in the dual):

[15] Remember that the dual is a *minimization* question, so for it being unbounded means the objective function can be as *negative as you want*.

Primal LP	Dual LP
max $? \cdot \mathbf{x}$	min $\mathbf{b} \cdot \mathbf{y}$
subject to	subject to
$A\mathbf{x} \leq \mathbf{b}$	$A^\top \mathbf{y} = ?$
	$\mathbf{y} \geq 0$

The question mark still needs to be decided. If we take $? = 0$, we can directly reuse the logic of the proof above (check this!). We get the following variant of the theorem of the alternative:

> **Either** there exists a vector $\mathbf{x}$ such that $A\mathbf{x} \leq \mathbf{b}$, **or** there exists a vector $\mathbf{y}$ such that $A^\top \mathbf{y} = 0$ and $\mathbf{y} \geq 0$, **but not both**.

14.2 Streamlined logic

When I wrote the proof of theorem of the alternative above I tried to motivate each step, but strictly speaking that's not necessary for a proof to be *correct* (though motivation sure helps to make the proof easier to understand, and above all, to feel *non-magical*). Here's a streamlined proof that tries not to do more than necessary. When you later prove variants of the theorem of the alternative, you might want to *think* as in the first proof, but only *write* in the style of this next proof. (Although, there absolutely nothing wrong with writing your motivation anyway, to help the reader!)

Proof. Consider the following primal dual pair:

Primal LP	Dual LP
max $0 \cdot \mathbf{x}$	min $\mathbf{b} \cdot \mathbf{y}$
subject to	subject to
$A\mathbf{x} \leq \mathbf{b}$	$A^\top \mathbf{y} \geq 0$
$\mathbf{x} \geq 0$	$\mathbf{y} \geq 0$

Notice the dual is automatically feasible, because $\mathbf{y} = 0$ is a feasible solution. By the fundamental theorem of linear programming that leaves two cases:

The dual is unbounded: Then the dual has a family of feasible solutions with objective function tending to $-\infty$. In particular, the dual must have feasible solutions with $\mathbf{b} \cdot \mathbf{y} < 0$. So in this case, the second alternative holds.

The dual has an optimal solution: Then so does the primal, by strong duality, in particular the primal is feasible which is the first alternative.

We've shown that in any case one of the alternatives holds. We still need to show that not both of them can hold simultaneously. If they did, then $\mathbf{x}$ and $\mathbf{y}$ would be feasible for the primal and dual respectively, and weak duality would tells us that $0 \cdot \mathbf{x} \leq \mathbf{b} \cdot \mathbf{y}$, contradicting that $\mathbf{b} \cdot \mathbf{y} < 0$. □

15 Marginal values

This section is dedicated to all the economics majors taking the course (but everyone else needs to learn this stuff too!)

15.1 The archetypal factory problem

Say you run a factory and are trying to decide how many units of each kind of product to manufacture; let x_j be the number of units of product #j you plan to make. Each of the n products requires some amount of each of m different resource; let a_{ij} be the amount of resource #i needed to manufacture one unit of product #i. Say also that b_i is the amount of resource #i you have available and that c_j is the predicted profit you make from each unit of product #j. Then to maximize profit you would want to solve the standard form linear program:

Maximize $c_1x_1 + c_2x_2 + \cdots c_nx_n$

subject to

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \\ x_1, x_2, \ldots, x_n &\geq 0 \end{aligned}$$

Here the constraints[16] express that the manufacture of all the products together should not require more than the available amount of each resource.

[16] The ones that are not just positivity constraints, I mean.

(Notice that not all standard form linear programs can be interpreted as a factory-type situation: in these situations all $a_{ij} \geq 0$ and all $b_i \geq 0$. It's still a good guide for intuition in general.)

In this model, what do the dual variables $y_1, \ldots, y_m$ mean? Let's start by figuring out their units. The dual constraints look like $\sum_{i=1}^{m} a_{ij} y_i \geq c_j$. So the units of y_i must be something that whose product with the units of a_{ij} matches the units of c_j. The units of a_{ij} are (whatever unit resource #i is measured in) per (unit of product #j); and the units of c_j are, say, dollars per (unit of product #j). This means that y_i is measured in units of dollars per (whatever unit resource #i is measured in).

That should make it plausible that, in an optimal dual solution, y_i is the **marginal value** of resource #i, that is, y_i is the amount of extra profit you'd expect to make for each additional unit of resource #i available. This is therefore also the price that you, as factory planner, should think is fair for one unit of resource #i. So, if someone offers to sell you resource #i at less than y_i dollars per unit, that's a good deal and you should buy some if you can.

This advice is only "local", that is, for small enough changes in the amount of resource #i available, y_i correctly predicts the change in profit, but for larger changes the situation can change. If you like multi-variable calculus, you can think of the marginal values this way: the optimal value of the primal LP is a function of all the a_{ij}, b_i and the c_j; let's focus on the b_i and regard the a_{ij} and c_j as constants and let $z(b_1, \ldots, b_m)$ be the maximum profit as a function of available resources. Then we are saying that $y_i = \frac{\partial z}{\partial b_i}$.

We will now prove a theorem which says that, with some caveats, this interpretation is correct.

15.2 The marginal values theorem

The following theorem won't mention anything about resources or factories, but it might be more intuitive if you keep the above interpretation in mind. The theorem has two parts: the first shows that the marginal values bound the optimal value of the altered primal, the second gives a condition under which that bound really is the new optimal value.

Theorem 6. *Consider the following standard form LP, its dual and an altered primal with slightly modified constraints:*

Primal LP	*Dual LP*	*Altered Primal LP*
max $\mathbf{c} \cdot \mathbf{x}$	*min* $\mathbf{b} \cdot \mathbf{y}$	*max* $\mathbf{c} \cdot \mathbf{x}$
subject to	*subject to*	*subject to*
$A\mathbf{x} \leq \mathbf{b}$	$A^\top \mathbf{y} \geq \mathbf{c}$	$A\mathbf{x} \leq \mathbf{b} + \Delta\mathbf{b}$
$\mathbf{x} \geq 0$	$\mathbf{y} \geq 0$	$\mathbf{x} \geq 0$

1. *Assume the primal (and therefore also its dual) has an optimal solution, and let* $\mathbf{y}^*$ *be* any *optimal solution of the dual; so that the optimal value of the primal is* $z^* = \mathbf{b} \cdot \mathbf{y}^*$. *If* $\mathbf{x}$ *is any feasible solution of the* altered *primal, then* $\mathbf{c} \cdot \mathbf{x} \leq z^* + \mathbf{y}^* \cdot \Delta\mathbf{b}$. *In particular,* if the altered primal is feasible *(it may not be, depending on* $\Delta\mathbf{b}$*), then its optimal value* z' *satisfies* $z' \leq z^* + \mathbf{y}^* \cdot \Delta\mathbf{b}$.

2. *Consider the basis* $\mathbf{x}_B$ *of the final dictionary that the Simplex Method produces for the primal LP and use* A_B, $\mathbf{c}_B$, *etc. as usual. Take* $\mathbf{y}^* = \left(\mathbf{c}_B^\top A_B^{-1}\right)^\top$ *as the optimal dual solution. If* $A_B^{-1}(\mathbf{b} + \Delta\mathbf{b}) \geq 0$, *then the altered primal is guaranteed to have an optimal solution and the optimal value is* $z' = z^* + \mathbf{y}^* \cdot \Delta\mathbf{b}$.

Proof. 1. Notice that the dual of the altered primal is

$$\text{Minimize } (\mathbf{b} + \Delta\mathbf{b}) \cdot \mathbf{y}$$
$$\text{subject to } \begin{cases} A^\top \mathbf{y} \geq \mathbf{c} \\ \mathbf{y} \geq 0 \end{cases}$$

In particular, notice that it has the same constraints as the dual of the original primal. This tells us that $\mathbf{y}^*$, the optimal dual solution, is still feasible for the altered dual. And so, weak duality for the altered LPs directly says that given any feasible solution of the *altered* primal, say $\mathbf{x}$, we have $\mathbf{c} \cdot \mathbf{x} \leq (\mathbf{b} + \Delta\mathbf{b}) \cdot \mathbf{y} = z^* + \Delta\mathbf{b} \cdot \mathbf{y}$.

2. Now, consider the final dictionary in the Simplex Method for the original primal LP:

$$\begin{aligned} \mathbf{x}_B &= A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N \\ z &= \mathbf{c}_B^\top A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N \end{aligned}$$

What if we try to use the same basis for the altered LP? The dictionary would look like this:

$$\mathbf{x}_B = A_B^{-1}(\mathbf{b} + \Delta\mathbf{b}) - A_B^{-1}A_N\mathbf{x}_N$$
$$z = \mathbf{c}_B^\top A_B^{-1}(\mathbf{b} + \Delta\mathbf{b}) + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N$$

Is this an optimal dictionary for the altered LP? Well, the z-row has the same coefficients as before so they are all non-positive as required for optimality. But this new dictionary is not necessarily feasible! Feasibility requires $A_B^{-1}(\mathbf{b} + \Delta\mathbf{b}) \geq 0$. So, under that assumption, we *do* get an optimal dictionary for the altered primal, and the optimal value of the altered LP is $\mathbf{c}_B^\top A_B^{-1}(\mathbf{b} + \Delta\mathbf{b}) = \mathbf{y}^* \cdot (\mathbf{b} + \Delta\mathbf{b}) = z^* + \mathbf{y}^* \cdot \Delta\mathbf{b}$, as desired.

□

15.2.1 The case of non-degenerate optimal solutions

How often should you expect the second half of the theorem to be applicable? To answer that notice that typically optimal solutions are non-degenerate: typically all the basic variables in the final dictionary have strictly positive values, that is, $A_B^{-1}\mathbf{b} > 0$. In that case, any vector $\Delta\mathbf{b}$ whose entries are close enough to zero will satisfy $A_B^{-1}(\mathbf{b} + \Delta\mathbf{b}) \geq 0$.

You may have noticed that the first part of the theorem is for *any* optimal dual solution but the second part is only for $\mathbf{y}^* = \left(\mathbf{c}_B^\top A_B^{-1}\right)^\top$. In the non-degenerate case where $A_B^{-1}\mathbf{b} > 0$, the dual optimal solution is unique, so you needn't worry then about the distinction! Indeed, all the basic variables $\mathbf{x}_B$ are positive, so complementary slackness gives us m different equations for $y_1, \ldots, y_m$; namely, we get that $A_B^\top \mathbf{y}^* = \mathbf{c}_B$. The only solution of that system is $\mathbf{y}^* = \left(A_B^\top\right)^{-1}\mathbf{c}_B = \left(\mathbf{c}_B^\top A_B^{-1}\right)^\top$.

16 The Revised Simplex Method

The way we've performing the Simplex Method so far is by writing a full dictionary at each step, but this is potentially wasteful: the matrix formulas for the dictionary tells us that knowing the basic variables is enough to reconstruct the whole dictionary and we don't even need *all* of the dictionary to figure out what pivot to perform, and thus to figure out what the next basis will be. For example, we never need to know the current value of the objective function. From the z-row we only need the coefficients of

non-basic variables, to pick the entering variable, and then to pick the exiting variables we only need two columns of the dictionary: we need the values of the basic variables in the solution associated to the dictionary and we need the coefficients of the exiting variable in the formulas for the basic variables.

For example in the following dictionary we don't need any of the question marks to figure out that x_2 should enter and x_6 should exit:

$$\begin{aligned} x_3 &= 3 + ?x_1 - 4x_2 + ?x_5 \\ x_4 &= 1 + ?x_1 + 2x_2 + ?x_5 \\ x_6 &= 2 + ?x_1 - 3x_2 + ?x_5 \\ z &= ? - 2x_1 + 4x_2 + 2x_5 \end{aligned}$$

The idea of the Revised Simplex Method is to avoid having to compute the full dictionary after every pivot. Instead, we'll only keep track of some of the information, including the current basis, and use the matrix formulas to compute the portions of the dictionary we need.

We'll describe two versions of the Revised Simplex Method: one where we only keep track of the current basis variables, the current solution and compute everything else (to avoid computing A_B^{-1}, we'll solve systems of equations instead), and one where we additionally keep track of A_B^{-1}.

16.1 Revised Simplex Method, version 1

In this version, we keep track only of the basis $\mathbf{x}_B$ and the current solution[17], $\mathbf{x}_B^* = A_B^{-1}\mathbf{b}$. To deal with any occurrences of A_B^{-1} in the formulas, we'll solve systems of equations. When we say we "keep track" of some data, it means that when we start a pivoting step we will have that data available for the current basis, and also that we must compute that data for the next basis. Throughout we will need to refer to the matrix formulas for the dictionary:

$$\begin{aligned} \mathbf{x}_B &= A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N \\ z &= \mathbf{c}_B^\top A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N \end{aligned}$$

We're **given** $\mathbf{x}_B$ and the numbers $\mathbf{x}_B^*$. Now we must determine the entering variable. For that we need to calculate the coefficients of the non-basic

[17] The $\mathbf{x}_B^*$ are the values of the *basic* variables in the solution associated to the dictionary for the basis B. This determines the entire associated solution, because the non-basic variables have the value 0.

variables in the z-row, i.e., $\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1} A_N$. All of the pieces, $\mathbf{c}_B$, $\mathbf{c}_N$, A_B and A_N, can be found from the original LP (or more, precisely, from the augmented $\mathbf{c}^{\mathsf{aug}}$ and A^{aug} you get after adding the slack variables). But we'd like to avoid calculating A_B^{-1}, so we'll get $\mathbf{y}^\top := \mathbf{c}_B^\top A_B^{-1}$ by solving the system of equations $A_B^\top \mathbf{y} = \mathbf{c}_B$.

Once we have $\mathbf{y}$ we can compute the vector of coefficients of $\mathbf{x}_N$ in the z-row as $\mathbf{c}_N^\top - \mathbf{y}^\top A_N$. We can then following the standard pivoting rules to choose the entering variable: we look for the largest positive coefficient, and in case of ties pick the one corresponding to the non-basic variable x_j with smallest j.

To choose the entering variable, we need $\mathbf{x}_B^*$ and the vector of coefficients of x_j in the top portion of the dictionary (the $\mathbf{x}_B$ part). From the matrix formulation of the dictionary, this vector is $-A_B^{-1} A_j$ (recall that A_j is the column of A^{aug} corresponding to the variable x_j). Again, we avoid inverting A_B by solving a system of equations, namely, we get $\mathbf{d} := A_B^{-1} A_j$ by solving $A_B \mathbf{d} = A_j$.

Once we have $\mathbf{d}$, we can choose the exiting variable according to the standard pivoting rules: x_j starts at 0 and is allowed to increase as long as all the $\mathbf{x}_B$ stay ≥ 0. So, the value of x_j in the solution associated to the *next* basis (in which x_j will be basic!), is given by $\max\{t \geq 0 : \mathbf{x}_B^* - t\mathbf{d} \geq 0\}$. Of course, there may be no maximum, that is, it might happen that $\mathbf{x}_B^* - t\mathbf{d} \geq 0$ for all $t \geq 0$; if so, that means the LP is unbounded and that $\mathbf{x}_N = 0$, $\mathbf{x}_B = \mathbf{x}_B^* - t\mathbf{d}$ is an unbounded family of feasible solutions.

If there is a maximum value of t, one or more of the entries of $\mathbf{x}_B^* - t\mathbf{d}$ becomes 0 (when t has the maximum value). Each entry of that vector corresponds to a basic variable; and we choose the exiting variable x_i to be the one whose entry becomes 0 —or, if there are several, the one with smallest i.

It only remains to make sure we have the next-basis versions of all the data we gave ourselves. The new $\mathbf{x}_B$ is, of course, just obtained from the previous one by removing the exiting x_i and putting in the entering x_j. The new associated solution requires is also pretty easy: as we said above, the value x_j^* of x_j in the next dictionary is that maximum t, and the new values of $\mathbf{x}_B$ are given by $\mathbf{x}_B^* - t\mathbf{d}$ (again, for that maximum t). Note that the vector $\mathbf{x}_B^* - t\mathbf{d}$ includes a zero in the entry corresponding to the exiting variable x_i, all the other entries correspond to variables that are still basic in the next dictionary.

16.1.1 Recipe

Summarizing the above discussion, these are the steps you take given the list of basic variables $\mathbf{x}_B$ and their values in the associated solution, $\mathbf{x}_B^*$:

1. Find $\mathbf{y}$ by solving $A_B^\top \mathbf{y} = \mathbf{c}_B$.
2. Compute $\mathbf{c}_N^\top - \mathbf{y}^\top A_N$, which is the vector of coefficients in the z-row. Use it to choose the entering variable x_j by looking for the largest positive entry. If all entries of the coefficient vector are non-positive, then stop: you have reached optimality!
3. Find $\mathbf{d}$ by solving $A_B \mathbf{d} = A_j$.
4. Find the maximum value of t such that $\mathbf{x}_B^* - t\mathbf{d} \geq 0$. From now on, let t be that maximum.
5. The exiting variable x_i will be the one (with smallest i, in case of ties) for which the corresponding entry of $\mathbf{x}_B^* - t\mathbf{d}$ is zero.
6. Find the values of the basic variables in the next dictionary:
   - For x_j the value will be that maximum t.
   - The other basic variables are the current ones, $\mathbf{x}_b$, but without x_i. Their values are $\mathbf{x}_B^* - t\mathbf{d}$ —this will include the value zero for x_i, which won't be basic anymore.

16.2 Revised Simplex Method, version 2

In this version, we additionally keep track of the matrix A_B^{-1}. If we have A_B^{-1}, the steps where we compute $\mathbf{y}$ and $\mathbf{d}$ can be done by simple matrix multiplication instead of solving a linear system. Of course, it's not all roses: we need to figure out how to compute the *next* A_B^{-1}, that it is, the matrix $A_{B'}^{-1}$ where B' is the next basis.

To write down the A_B and $A_{B'}$ matrices you need to pick a specific order for the basic variables. Let's agree that the basis $\mathbf{x}_{B'}$ will have the same order as $\mathbf{x}_B$ with the exiting variable x_i replaced by the entering variable x_j. With that convention, if x_i occurs in position ℓ of $\mathbf{x}_B$, then $A_{B'}$ and A_B differ only in the ℓ-th column: A_B has A_i as ℓ-th column, while $A_{B'}$ has A_j.

This means that $A_{B'} = A_B E$ where E is the matrix whose ℓ-th column is $\mathbf{d} = A_B^{-1} A_j$ and otherwise agrees with the identity matrix. Why is that? Well, for any matrices M and N, if the columns of N are $\mathbf{v}_1, \ldots, \mathbf{v}_m$, it follows that the columns of the product $MN$are the vectors $M\mathbf{v}_1, \ldots, M\mathbf{v}_m$. In

particular, if the columns of the identity matrix are $\mathbf{e}_1, \ldots, \mathbf{e}_m$, then $M\mathbf{e}_i$ is simply the i-th column of the matrix M. For the matrix E, whose columns are $\mathbf{e}_1, \ldots, \mathbf{e}_{\ell-1}, \mathbf{d}, \mathbf{e}_{\ell+1}, \ldots, \mathbf{e}_m$ we see that the columns of $A_B E$ are the same as the columns of A_B except that the ℓ-th one is $A_B\mathbf{d} = A_j$; i.e., the columns of $A_B E$ are exactly those of $A_{B'}$!

From this we can easily state the *update rule* for A_B^{-1}, namely, $A_{B'}^{-1} = E^{-1}A_B^{-1}$. And this is extremely handy, because E^{-1} is very easy to calculate: namely, E^{-1} also agrees with the identity matrix except in the ℓ-th column, which is given by the following recipe: if $\mathbf{d}^\top = \begin{pmatrix} d_1 & d_2 & \ldots & d_m \end{pmatrix}$, then the ℓ-th column of E^{-1} is given by

$$\begin{pmatrix} -d_1/d_l & -d_2/d_l & \ldots & -d_{\ell-1}/d_l & 1/d_l & -d_{\ell+1}/d_l & \ldots & -d_m/d_l \end{pmatrix}^\top.$$

Notice that the ℓ-th entry of that vector is the one on the diagonal of the matrix E^{-1} and does *not* follow the pattern of the others.

For example, if $E = \begin{pmatrix} 1 & p & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & r & 1 & 0 \\ 0 & s & 0 & 1 \end{pmatrix}$, then $E^{-1} = \begin{pmatrix} 1 & -p/q & 0 & 0 \\ 0 & 1/q & 0 & 0 \\ 0 & -r/q & 1 & 0 \\ 0 & -s/q & 0 & 1 \end{pmatrix}$.

This is easy enough to check by multiplying out. But for completeness's sake let's prove it:

Proposition 1. *If E is a square matrix that only differs from the identity matrix in the ℓ-th column, where E has instead the column $\begin{pmatrix} d_1 & d_2 & \ldots & d_m \end{pmatrix}^\top$, then E^{-1} also only differs from the identity matrix in the ℓ-th column which is $\begin{pmatrix} -d_1/d_l & -d_2/d_l & \ldots & -d_{\ell-1}/d_l & 1/d_l & -d_{\ell+1}/d_l & \ldots & -d_m/d_l \end{pmatrix}^\top$.*

Proof. Let $\mathbf{e}_1, \ldots, \mathbf{e}_m$ be the columns of the identity matrix. We know that the columns of E are given by $E\mathbf{e}_k = \mathbf{e}_k$ for $k \neq l$ and $E\mathbf{e}_\ell = \sum_{k=1}^m d_k\mathbf{e}_k$. Multiplying both sides of these equations by E^{-1} gives us, first, that $\mathbf{e}_k = E^{-1}\mathbf{e}_k$ for $k \neq \ell$, and second, that $\mathbf{e}_\ell = \sum_{k=1}^m d_k E^{-1}\mathbf{e}_k$. So we've already obtained that all columns of E^{-1} other than the ℓ-th are the same as those of the identity matrix. Plugging that into the last equation we get, $\mathbf{e}_\ell = \sum_{k\neq\ell} d_k\mathbf{e}_k + d_\ell E^{-1}\mathbf{e}_\ell$. Solving, we get that $E^{-1}\mathbf{e}_\ell = \sum_{k\neq\ell}(-d_k/d_\ell)\mathbf{e}_k + (1/d_\ell)\mathbf{e}_\ell$, as desired. □

16.2.1 Recipe

To summarize, these are the steps you take given the list of basic variables $\mathbf{x}_B$, their values $\mathbf{x}_B^*$ in the associated solution, and the matrix A_B^{-1}:

1. Compute $\mathbf{c}_N^\top - \mathbf{c}^\top A_B^{-1} A_N$, which is the vector of coefficients in the z-row. Use it to choose the entering variable x_j by looking for the largest positive entry. If all entries of the coefficient vector are non-positive, then stop: you have reached optimality!

2. Compute $\mathbf{d} = A_B^{-1} A_j$.

3. Find the maximum value of t such that $\mathbf{x}_B^* - t\mathbf{d} \geq 0$. From now on, let t be that maximum.

4. The exiting variable x_i will be the one (with smallest i, in case of ties) for which the corresponding entry of $\mathbf{x}_B^* - t\mathbf{d}$ is zero.

5. Find the values of the basic variables in the next dictionary:

   - For x_j the value will be that maximum t.
   - The other basic variables are the current ones, $\mathbf{x}_b$, but without x_i. Their values are $\mathbf{x}_B^* - t\mathbf{d}$ —this will include the value zero for x_i, which won't be basic anymore.

6. Find the next A_B^{-1}: it is $E^{-1} A_B^{-1}$, where E is the matrix that differs from the identity matrix only in the column in the position of the entering/exiting variable, where E has the vector d.

16.3 Invertibility of A_B

The formula $A_{B'} = A_B E$ finally let's us repair a long standing omission: we haven't actually proved that A_B is invertible! To be clear, we're not saying that whenever you pick some columns of A^{aug} that form a square matrix that matrix is automatically invertible. What *is* true is that for any basis arising in the Simplex Method, the corresponding A_B will be invertible, as required for the matrix formulas for the dictionary to make sense!

At the very first step of the Simplex Method the basic variables are the slack variables and A_B is an identity matrix, which is, of course, invertible. Now, imagine that for some dictionary A_B is known to be invertible. Let's show the next matrix, namely $A_{B'} = A_B E$, is invertible too. It's enough to show E is invertible, since then $A_{B'}$ is a product of two invertible matrices and thus invertible itself.

So why *is* E invertible? Well, we found the inverse already! But for that inverse to make sense, we do need that $d_\ell \neq 0$[18]. Alternatively, expanding

[18] A point which was glossed over in the proof of the proposition!

by the ℓ-th row, we see that the determinant of E is exactly d_ℓ. At any rate, all we have left to show is that $d_\ell \neq 0$. This comes from the way pivots are chosen: d_ℓ is minus the coefficient of the exiting variable x_j in the formula for the entering variable x_i, so it has to be strictly positive (since the coefficient must be negative, otherwise x_i wouldn't be the exiting variable).

17 Sensitivity Analysis

When using linear programming to model real world situations we often need to solve new linear programs obtained by making small changes to problems we've already solved. This can happen for various reasons, among many others:

- The coefficients might have been estimates of key quantities, later with more information we might have better estimates.
- Even if the coefficients were completely accurate they might change over time, for example, if we needed to take into account the price of some resource, as that price changes we'd want to update our model.
- We might realize we forgot a constraint, or suddenly have an extra variable we want to know about.
- We might want to replace any coefficients that were estimated with more pessimistic estimates to compute a worst case scenario.

We can handle this problem by simply solving the new LP from scratch using the Simplex Method, but we can often reuse the work done in solving the original LP to solve the new one faster than we could from scratch. This ability of the Simplex Method to "reuse work" gives it an advantage when solving many related LPs over other methods that might be faster for single LPs but need to start over for every change. (We won't actually cover any other methods for solving LPs in this course, and you can take this as part of the justification for that.)

The study of how solutions of LPs change when you change the LP is called **sensitivity analysis**, and we've already seen some of it: the marginal values theorem tells us something about what happens in a standard form LP if you change the right-hand sides of the constraints. It gave us an upper bound for the new optimal solution and conditions under which that estimate was exactly right. Now we'll complete this analysis, indicating

what to do when those conditions don't hold; as well as analyze various other changes we can make to an LP.

Consider an LP in standard form:

$$(\mathsf{P})\ \text{Maximize } \mathbf{c} \cdot \mathbf{x}$$
$$\text{subject to } \begin{cases} A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases}$$

We can consider changes such as:

- Changing the objective function, $\mathbf{c}$.
- Changing the right-hand sides of the constraints, $\mathbf{b}$.[19]
- Changing (some entries of) the matrix of coefficients, A.
- Adding or removing a constraint.
- Adding or removing a variable.

For each type of change we'd like to know:

- Conditions for which the solution of the new LP is easy to get from the old optimal solution, say, for when the old optimal solution is still optimal, or at least the old optimal basis is still optimal even if the values of the variables need recomputing (this is what happens in the marginal values theorem).
- What to do if those conditions don't hold and we need to work more to solve the new LP. Ideally, this would be better than doing the Simplex Method from scratch on the new LP.

Each type of change needs separate analysis, and in each case we'll start by looking at the matrix formulas for the final dictionary of the original LP to see what changes.

[19] Notice that in a sense this is the same kind of change as the previous one: we can view this in the dual as changing the objective function. Here we'll stick with the point of view of the primal and treat this as a different case.

17.1 Changing the objective function

Let's consider changing $\mathbf{c}$ to $\hat{\mathbf{c}}$ in the problem (P) to get a problem $(\hat{\mathsf{P}})$. Say $\mathbf{x}_B$ is an optimal basis for (P). The matrix formulas say the dictionary is given by:

$$\begin{aligned}\mathbf{x}_B &= A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N \\ z &= \mathbf{c}_B^\top A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N\end{aligned}$$

Let's try to use the same basis for the new problem. The dictionary would be:

$$\begin{aligned}\mathbf{x}_B &= A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N \\ z &= \hat{\mathbf{c}}_B^\top A_B^{-1}\mathbf{b} + \left(\hat{\mathbf{c}}_N^\top - \hat{\mathbf{c}}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N\end{aligned}$$

Notice that the top part, the formulas for the basic variables, hasn't changed at all. In particular, this dictionary is feasible! And if the new coefficients in the z-row are non-positive, that is, if $\hat{\mathbf{c}}_N^\top - \hat{\mathbf{c}}_B^\top A_B^{-1}A_N \leq 0$, then this dictionary is even optimal!

And if $\hat{\mathbf{c}}_N^\top - \hat{\mathbf{c}}_B^\top A_B^{-1}A_N$ has some positive entries we can just use the above dictionary as the initial dictionary for the Simplex Method to solve $\hat{\mathbf{P}}$.

17.2 Adding a variable

Now let's consider adding a new variable x_{m+n+1} to an LP. The new variable should come, of course, with coefficients: we need a c_{m+n+1} for the objective functions and a new column A_{m+n+1} for the matrix of coefficients of the constraints.

Inspired by the initial dictionary (where we take the slack variables as basic and the non-slack variables as non-basic) we will make the guess that the new variable should be added as a non-basic variable to the final dictionary. Using the matrix formulas to compute the new column, we get this dictionary for the LP with the extra variable:

$$\begin{aligned}\mathbf{x}_B &= A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N - A_B^{-1}A_{m+n+1}x_{m+n+1} \\ z &= \mathbf{c}_B^\top A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N + (c_{m+n+1} - \mathbf{c}_B^\top A_B^{-1}A_{m+n+1})x_{m+n+1}\end{aligned}$$

This dictionary is feasible, because the first column hasn't changed, but the new coefficient in the z-row may have any sign. If we're lucky, $c_{m+n+1} -$

$\mathbf{c}_B^{-1} A_{m+n+1} \leq 0$ and this dictionary is already optimal! If not, since it is at least feasible, we can just use it as the initial dictionary for the Simplex Method.

17.3 Other types of changes

If you attempt this sort of analysis for a change of $\mathbf{b}$, or for adding a constraint (guessing the new slack variable is basic), you'll see that, while we can easily formulate a criterion for when the modified dictionary is optimal, when it isn't we won't be left with a feasible dictionary! Instead we'd get a dictionary that is not feasible but "looks optimal": all of the coefficients in the z-row are non-positive.

Such a dictionary can't be used in the Simplex Method: the Simplex Method pivots from *feasible* dictionary to *feasible* dictionary working towards non-positive coefficients in the z-row. To take advantage of this infeasible dictionary that "looks optimal", instead of the Simplex Method we'll use a very similar pivoting technique called the Dual Simplex Method. That's our next topic.

18 The Dual Simplex Method

The Simplex Method[20] pivots from *feasible* dictionary to *feasible* dictionary attempting to reach a dictionary whose z-row has all of its coefficients non-positive. In sensitivity analysis certain modifications of an LP will lead to dictionaries whose z-row "looks optimal" but that are not feasible. To take advantage of those those dictionaries, we will develop a dual version of the Simplex Method. Call a dictionary **dual feasible** if all the coefficients in its z-row are non-positive. The **Dual Simplex Method** will pivot from *dual* feasible dictionary to *dual* feasible dictionary working towards *feasibility*.

This new pivoting strategy is called the Dual Simplex Method because it really is the same as performing the usual Simplex Method on the dual linear problem. This also explains the term "dual feasible": each dictionary for the primal has a corresponding dictionary for the dual and a primal dictionary is dual feasible exactly when the corresponding dictionary for the dual is feasible (in the usual sense). We won't really take advantage of this correspondence, though: we won't directly talk about the dual LP instead

[20] More precisely, we're talking about Phase 2 —and the portion of Phase 1 after the first special pivot to feasibility.

explaining how to perform these dual pivots directly on a dual feasible dictionary for the primal.

But before that, let's show how certain kinds of sensitivity analysis naturally lead to dual feasible dictionaries.

18.1 Change of right-hand sides

Let's start, as usual, with the linear program:

$$(\mathsf{P})\ \text{Maximize } \mathbf{c}\cdot\mathbf{x} \quad \text{subject to } \begin{cases} A\mathbf{x} \le \mathbf{b} \\ \mathbf{x} \ge 0 \end{cases}$$

Now consider changing $\mathbf{b}$ to $\hat{\mathbf{b}} = \mathbf{b} + \Delta\mathbf{b}$ in the problem (P) to get a problem $(\hat{\mathsf{P}})$. Say $\mathbf{x}_B$ is an optimal basis for (P). The matrix formulas say the dictionary is given by:

$$\begin{aligned} \mathbf{x}_B &= A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N \\ z &= \mathbf{c}_B^\top A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N \end{aligned}$$

Let's try to use the same basis for the new problem. The dictionary would be:

$$\begin{aligned} \mathbf{x}_B &= A_B^{-1}\hat{\mathbf{b}} - A_B^{-1}A_N\mathbf{x}_N \\ z &= \mathbf{c}_B^\top A_B^{-1}\hat{\mathbf{b}} + \left(\mathbf{c}_N^\top - \mathbf{c}_B^\top A_B^{-1}A_N\right)\mathbf{x}_N \end{aligned}$$

Notice the coefficients of $\mathbf{x}_N$ in the z-row are unchanged, so this dictionary is automatically *dual* feasible. But this dictionary is not necessarily *feasible*: it is feasible precisely when $A_B^{-1}\hat{\mathbf{b}} \ge 0$. So we recover what we knew from the marginal values theorem: when $A_B^{-1}\hat{\mathbf{b}} \ge 0$ the same basis is still optimal for the modified primal $(\hat{\mathsf{P}})$.

And when that condition does not hold, we are left with a dictionary that is dual feasible but infeasible. The Dual Simplex Method will let us solve the modified LP using this as starting dictionary.

18.2 Adding a constraint

Now consider instead adding a new constraint to (P). The new constraint will require adding a new slack variable, say x_{m+n+1}. We need to add this new variable to the final dictionary as either a basic or non-basic. Once

again we take our inspiration from the way one writes an initial dictionary: we'll guess this new slack variable should be *basic*.

We can easily express the new slack variable in terms of the original (i.e., non-slack) variable of (P), indeed, if the new constraint is $a_{m+1,1}x_1 + a_{m+1,2}x_2 + \cdots + a_{m+1,n}x_n \leq b_{m+1}$, then the new slack variable is simply $x_{m+n+1} = b_{m+1} - a_{m+1,1}x_1 - a_{m+1,2}x_2 - \cdots - a_{m+1,n}x_n$. But to add a row to the dictionary for x_{m+n+1} we'll need a formula for it in terms of the basic variables. That's both easy and familiar: it's just like the step in the 2-phase Simplex Method right after phase 1 is over and we need to find a formula for the objective function in terms of the current basis variables. Here we do the same thing: in the formula for x_{m+n+1} above we plug in whatever the dictionary says for each basic variable occuring, to obtain a formula in terms of only non-basics.

The resulting dictionary for $(\hat{\mathsf{P}})$ just has an extra row compared to the final dictionary for P; in particular, it has exactly the same z-row, so it is automatically dual feasible. And it is "mostly feasible" too: all the basic variables except for the new one, namely x_{m+n+1}, have the same values as before, so they are all non-negative.

So, if the resulting value for x_{m+n+1} in the new row is non-negative, then the new dictionary is optimal. If not, it is at least dual feasible and the Dual Simplex Method can take it from there.

18.3 Performing dual pivots

OK, so now we have to explain how to perform a dual pivot. Remember we will start from a dictionary that is dual feasible and must maintain dual feasibility. For example, say we have the dictionary:

$$\begin{aligned} x_2 &= 5 - 2x_1 - x_3 + 2x_6 \\ x_3 &= -3 + 2x_1 + 4x_3 - 3x_6 \\ x_5 &= -4 + 2x_1 - 3x_3 + x_6 \\ z &= 2 - x_1 - 3x_3 - 2x_6 \end{aligned}$$

We first pick the *exiting* variable, for which we should take one of the variables with a negative value in the associated solution. We'll pick x_5 to exit, because it has the most negative value[21]. Now let's pick the entering

[21] This is in analogy with the standard pivoting rule. Recall that for the (usual) Simplex Method we can pick the entering variable to be any of the non-basic variables whose coefficient in the z-row is positive (all of those choices serve to increase the objective function), but specific pivoting rules tell you exactly which one to pick: the standard rule says to pick one with the largest positive coefficient, while Bland's rule says to pick among the variables

variable, keeping in mind that we want to preserve dual feasibility.

Once we decide on which variable x_j enters, the z-row in the next dictionary is obtained as follows: solve the x_5 equation for x_j and plug that into the equation for z. Another way to get the same effect is to add a multiple of the x_5-row to the z-row so that x_j cancels. If we just use an unknown coefficient t, we can do this calculation even before deciding which variable enters! Namely, the next z-row will come from the following equation:

$$z + tx_5 = (2 - x_1 - 3x_3 - 2x_6) + t(-4 + 2x_1 - 3x_3 + x_6),$$

so the next z-row will be

$$z = (2 - 4t) + (-1 + 2t)x_1 + (-3 - 3t)x_3 + (-2 + t)x_6 - tx_5,$$

where we need to choose the value of t so that all the coefficients are non-positive (to maintain dual feasibility), and so one of the variables x_1, x_3, x_6 disappears from the z-row —that variable will be the entering variable.

Since by design the coefficient of the exiting variable x_5 is $-t$, we see that $t \geq 0$. So then we need to choose the largest non-negative value of t so that $-1 + 2t, -3 - 3t, -2 + t \leq 0$. Let's see how each inequality constrains $t \geq 0$:

- From $-1 + 2t \leq 0$, we see $t \leq 1/2$.
- The second inequality, $-3 - 3t \leq 0$, is true for all $t \geq 0$ so it does not impose any restriction on t.
- From $-2 + t \leq 0$, we see $t \leq 2$.

For all three inequalities to hold we need $t \leq 1/2$ (i.e., that is the strong restriction). So we pick $t = 1/2$ and x_1 will be the entering variable. Now we can perform the pivot. As a small shortcut, to get the new z-row we can just plug $t = 1/2$ into the equation we obtained for z. The next dictionary works out to be:

$$\begin{aligned} x_2 &= 1 - 4x_3 + 3x_6 - x_5 \\ x_3 &= 1 + 7x_3 - 4x_6 + x_5 \\ x_1 &= 2 + \tfrac{3}{2}x_3 - \tfrac{1}{2}x_6 + \tfrac{1}{2}x_5 \\ z &= 0 - \tfrac{9}{2}x_3 - \tfrac{5}{2}x_6 - \tfrac{1}{2}x_5 \end{aligned}$$

with positive coefficient the one with the smallest index. For the Dual Simplex Method we won't insist on always following a standard pivoting rule as much as we did for the regular Simplex Method (since we'll typically use it in cases where only one pivot suffices).

We got lucky! This new dictionary turns out to be feasible, so we have reached optimality after only one dual pivot. Of course, in general, since we only worried about preserving dual feasibility, the dictionary we get after pivoting will be dual feasible, but not necessarily feasible. If it's not feasible, we just do more dual pivots.

We only described Dual Phase 2. We won't need the Dual version of Phase 1 because we will only use the Dual Simplex Method for some cases of sensitivity analysis, which provide us a dual feasible dictionary.

Also, notice that the step in dual pivoting where we pick the entering variable is very much like the step in the Revised Simplex Method where we pick the *exiting* variable by finding the maximum $t \geq 0$ such that $\mathbf{x}_B^* - t\mathbf{d} \geq 0$. There, $\mathbf{x}_B^*$ was the vector of values of the basic variables, and $-\mathbf{d}$ was the vector of coefficients of the entering variable in the top part of the dictionary. We can make the dual pivoting step look even more similar to (a transposed version of) this by writing it in terms of row vectors. In the example above, we were looking for $\max\{t \geq 0 : \begin{pmatrix}-1 & -3 & -2\end{pmatrix} + t\begin{pmatrix}2 & -3 & 1\end{pmatrix} \leq 0\}$. Both row vectors are vectors of coefficients of the non-basic variables x_1, x_3, x_6: $\begin{pmatrix}-1 & -3 & -2\end{pmatrix}$ is the vector of coefficients in the z-row, and $\begin{pmatrix}2 & -3 & 1\end{pmatrix}$ is the vector of coefficients in the x_5-row (and x_5 is the variable selected to exit the basis).

19 Some combinatorial optimization problems

Here we'll mention some examples of modeling problems in combinatorics using linear programs with **binary variables**, that is, variables that are restricted to be either 0 or 1.

19.1 The subset sum problem

Given a list of numbers $a_1, a_2, \ldots, a_n$ and a target sum s, choose some of the numbers so that their sum is as close as possible to s.

We can model this as an integer programming problem, with a **binary variable** x_i for each of the number. Here, "binary" means x_i is only allowed to take the values 0 or 1; which in LINDO is expressed by adding `INT xi` *after* the `END` of an LP. (Variables allowed to take any non-negative integral value are indicated with `GIN xi` —the `G` is for "general".)

The variable x_i will be 1 or 0 according to wheter the number a_i is among the chosen ones. Then the sum of the chosen numbers is $a_1x_1 + \ldots + a_nx_n$. So we can model a few variants of this question as integer programs:

The price is right If we want the sum to be as close as possible to s, *without going over*, we can solve:

Maximize $a_1x_1 + \ldots + a_nx_n$
subject to
$$a_1x_1 + \ldots + a_nx_n \leq s$$
$$x_1, \ldots, x_n \in \{0, 1\}$$

Can we get within k? Here we are only interested in whether the constraints are feasible, so we pick the objective function to be the constant 0: this makes sure the LP cannot be unbounded.

Maximize 0
subject to
$$a_1x_1 + \ldots + a_nx_n \leq s + k$$
$$a_1x_1 + \ldots + a_nx_n \geq s - k$$
$$x_1, \ldots, x_n \in \{0, 1\}$$

Notice that k is a parameter (i.e., a number we need to pick), not a variable in the LP.

What's the closest we can get? Just make k a variable too!

Minimize k
subject to
$$a_1x_1 + \ldots + a_nx_n \leq s + k$$
$$a_1x_1 + \ldots + a_nx_n \geq s - k$$
$$x_1, \ldots, x_n \in \{0, 1\}$$

19.2 The partition problem

Given a list of numbers $a_1, \ldots, a_n$, split them into two bunches such that the sums of the numbers in each bunch are as close as possible.

Minimize k
subject to
$$a_1x_1 + \ldots + a_nx_n \leq (a_1 + \cdots + a_n)/2 + k$$
$$a_1x_1 + \ldots + a_nx_n \geq (a_1 + \cdots + a_n)/2 - k$$
$$x_1, \ldots, x_n \in \{0, 1\}$$

19.3 Shortcomings of these examples

The point of these easy examples is that many questions that seem very combinatorial and have initially nothing to do with linear programming can be encoded as linear program with binary variables. Of course, some encodings can be more clever that others! The ones above aren't much good: one way to tell is that the relaxation of those problems is *too* easy, which is to say, the whole difficulty of the problem is packed into the constraint of the variables being 0 or 1.

For example, the LP for the partition problem always has the optimal solution $k = 0$, $x_1 = x_2 = \ldots = x_n = 0.5$, no matter what the a_i are!

Next we'll take a look at a better example of encoding a combinatorial question as an integer linear program. It has more interesting constraints, capturing more of the problem and it's relaxation is not trivial.

19.4 Graph coloring

A **graph** is simply a collection of **vertices** some pairs of which are connected by **edges**. A **proper coloring** of the graph is an assignment of a color to each vertex in such a way that any two vertices connected by an edge are assigned different colors (but vertices that are not connected directly by an edge may be assigned the same color). We can model proper colorings using linear constraints on binary variables as follows:

Let $v_1, v_2, \ldots, v_n$ the vertices of the graph and let $c_1, \ldots, c_m$ the available colors. Introduce a variable x_{ij} for every *pair* of a vertex v_i and color c_j: x_{ij} will be 1 if v_i is assigned color c_j and 0 if not. In terms of these variables we can express the constraints of a proper coloring:

Each vertex must be assigned a color For each i, we have $x_{i1} + x_{i2} + x_{im} = 1$.

Vertices connected by an edge must be assigned different colors For each pair of vertices v_{i_1} and v_{i_2} that are connected by an edge, and for every color c_j, we can express that v_{i_1} and v_{i_2} cannot both be color c_j via the constraint $x_{i_1 j} + x_{i_2 j} \leq 1$.

If we want to find out if a graph can be properly colored with some given number of colors, then these constraints are all we need. What if we want to know the minimum number of colors in a proper coloring? Well, if the graph has n vertices, and you have at least n colors available it is certainly possible to find a proper coloring: just give each vertex a different

color! And by cleverly choosing an objective function we can find out the minimum numbers of colors needed (the so-called **chromatic number** of the graph).

Say the graph has 9 vertices, for example. Then 9 colors certainly suffice. For every coloring of the graph imagine forming the number whose digits are simply the number of times each color was used. For example, the number $000120303 = 120303$ means that only four colors we actually used, one once, one twice, and two colors three times. Depending on how you number the colors, that same coloring could have gotten the numbers 312030, 2331, or 1233. Notice that 1233 is the smallest number that such a coloring (i.e., one using four colors, those numbers of times) can receive. Since numbers with more digits are always larger than numbers with fewer digits, if you look at the numbers associated to all colorings, the smallest one must correspond to a coloring with as few colors as possible! So, for our graph with 9 vertices, the objective function $\sum_{j=1}^{9} \left(\sum_{i=1}^{9} x_{ij} \right) 10^{j-1}$ is minimized when the number of colors actually used is as small as possible.

For a graph with n vertices the same thing works, replacing those base 10 numbers with base $n+1$ numbers, that is, $\sum_{j=1}^{n} \left(\sum_{i=1}^{n} x_{ij} \right) (n+1)^{j-1}$ will be minimized at the same time as the number of colors is minmized. Of course, the minimum value of that objective function is *not* the minimum number of colors need to properly color the graph, but given an optimal solution x_{ij} you can easily figure out how many colors it used: just count for how many j at least one of the x_{ij} is 1.

19.4.1 LINDO LP generator

If you are reading this on the website, below you'll a form that can generate these LPs in LINDO format, given a description of your graph. If you are reading this in PDF, the rest of this section will look empty.

20 The tollbooth staffing problem

```
min x1+x2+x3+x4+x5+x6+x7+x8+x9+x10
   +x11+x12+x13+x14+x15+x16+x17+x18+x19
   +x20+x21+x22+x23+x24
subject to
  12:00am)  x1+x17+x18+x19+x20+x22+x23+x24 > 2
   1:00am)  x1+ x2+x18+x19+x20+x21+x23+x24 > 2
   2:00am)  x1+ x2+ x3+x19+x20+x21+x22+x24 > 2
```

```
  3:00am)  x1+ x2+ x3+ x4+x20+x21+x22+x23 > 2
  4:00am)  x2+ x3+ x4+ x5+x21+x22+x23+x24 > 2
  5:00am)  x1+ x3+ x4+ x5+ x6+x22+x23+x24 > 2
  6:00am)  x1+ x2+ x4+ x5+ x6+ x7+x23+x24 > 8
  7:00am)  x1+ x2+ x3+ x5+ x6+ x7+ x8+x24 > 8
  8:00am)  x1+ x2+ x3+ x4+ x6+ x7+ x8+ x9 > 8
  9:00am)  x2+ x3+ x4+ x5+ x7+ x8+ x9+x10 > 8
 10:00am)  x3+ x4+ x5+ x6+ x8+ x9+x10+x11 > 4
 11:00am)  x4+ x5+ x6+ x7+ x9+x10+x11+x12 > 4
 12:00pm)  x5+ x6+ x7+ x8+x10+x11+x12+x13 > 3
  1:00pm)  x6+ x7+ x8+ x9+x11+x12+x13+x14 > 3
  2:00pm)  x7+ x8+ x9+x10+x12+x13+x14+x15 > 3
  3:00pm)  x8+ x9+x10+x11+x13+x14+x15+x16 > 3
  4:00pm)  x9+x10+x11+x12+x14+x15+x16+x17 > 6
  5:00pm) x10+x11+x12+x13+x15+x16+x17+x18 > 6
  6:00pm) x11+x12+x13+x14+x16+x17+x18+x19 > 5
  7:00pm) x12+x13+x14+x15+x17+x18+x19+x20 > 5
  8:00pm) x13+x14+x15+x16+x18+x19+x20+x21 > 5
  9:00pm) x14+x15+x16+x17+x19+x20+x21+x22 > 5
 10:00pm) x15+x16+x17+x18+x20+x21+x22+x23 > 3
 11:00pm) x16+x17+x18+x19+x21+x22+x23+x24 > 3
end
```

21 Matrix Games

Now we'll discuss some *game theory*. This is a broad topic and we'll only cover a small portion of it that has a close connection to linear programming. We'll talk about a particular kind of game we'll call a **matrix game**. These are two player games that proceed in a series of identical rounds, in each round the two players make a move and depending on the moves either a tie is declared or one player is declared the loser and has to pay the winner a certain amount (that can depend on the moves chosen). To analyze these games mathematically, we can ignore the nature of the moves: all we need is to know for each pair of moves for the two players, who wins and how much. This is recorded in the **payoff matrix**, a matrix $A = (a_{ij})$ where a_{ij} records the payoff in a round where player one makes move #i and player two makes move #j; this number a_{ij} will be:

- positive if player one wins the round,

- zero if the round is a tie, and
- negative if player two wins the round.

If we think of winning a negative amount as meaning you actually lost and you pay the absolute value of that negative amount to the other player, then we can just think of a_{ij} as the winnings *for player one* in around where player one makes move #i and player two makes move #j.

Take rock, paper, scissors for example. In each round each player picks one of those 3 objects and the rules are that rock beats scissors, scissors beats paper and, somewhat counter-intuitively, paper beats rock. Say that we just want to keep track of how many more times player one won than player two. We can say the winner of each round gets a dollar from the loser, then the net winnings will be simply $\#wins - \#loses$. The payoff matrix is:

	rock	paper	scissors
rock	0	-1	1
paper	1	0	-1
scissors	-1	1	0

These matrix games are examples of what are called **zero-sum** games in game theory: if you add the winnings (with loses counting as "negative winnings") of all the players the net result is zero! Indeed, for our games, in every round the amount one player wins is exactly the amount the other player loses!

Notice that we've made the choice to record the payoff for player one, i.e., we decided positive numbers mean player one wins and negative numbers mean player two wins. We'll describe this by saying the payoff matrix is "from the point of view of player one". What's the payoff matrix "from the point of view of player two"? Well, its (i, j)-th entry should be the payoff for player *two* when player *two* makes move #i and player *one* makes move #j; that's $-a_{ji}$, so the payoff matrix for player two is $-A^\top$.

When the rules of the game treat both players exactly the same, the game is called **symmetric**, which may be a little confusing because it corresponds to the payoff matrix being **anti-symmetric**, which simply means $A = -A^\top$, i.e., the payoff matrix from the point of view of player two is the *same* as the one from the point of view of player one. Rock, paper, scissors is a symmetric game. Below we'll see some examples of games that aren't symmetric.

21.1 Mixed strategies

We want to figure out how to maximize your winnings when playing a matrix game (or at least minimize your losses if the game is stacked against you!). That sounds like a maximization problem, but what are we maximizing over? We want to pick the best *strategy* so we'll need a mathematical model of a playing strategy.

One strategy you could follow is to always pick the same move in every round. We'll call these **pure strategies** and there is one for every choice of move. These are typically poor strategies: for example, if you use the strategy *always play rock* in rock, paper, scissors, your opponent will soon catch on and start using the *paper 4ever* strategy.

You can do much better by mixing it up unpredictably! In rock, paper, scissors playing each move one third of the time chosen randomly[22] will protect you against any possible strategy of player two: you will on average win a third of the rounds, tie a third of the rounds and lose a third of the rounds. That's the best you can hope for, since if player two uses that same uniformly random strategy you won't win more than a third of the time on average.

In rock, paper, scissors all the moves are basically equivalent: each move beats one of your opponents moves, ties with one and loses against one. So in a random strategy it makes sense to pick them all with the same probability. In general matrix games this need not be true, so we should consider playing randomly, but where each move is assigned a possibly different probability of being chosen.

This is what we will call a **mixed strategy**: a strategy described by assigning each move a certain probability and then playing randomly according to those probabilities. If player one has n possible moves, a mixed strategy for player one is given by choosing probabilities $x_1, x_2, \ldots, x_n$. The vector of probabilities $\begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix}^\top$ is called a **stochastic vector**, meaning a vector whose entries are non-negative and satisfy $x_1 + x_2 + \cdots + x_n = 1$.

Notice that pure strategies are a special case of a mixed strategy: they are the case where one of the $x_i = 1$ and all the other are 0. So maybe a better name would be "potentially mixed" strategies, but of course, that's just too long! In fact we'll mostly just say "strategies" instead of "mixed strategies".

[22] For example, you could roll a die and pick rock if it lands 1 or 2, paper for 3 or 4, and scissors for 5 or 6

How do we measure how good a mixed strategy is? As a first step, let's see how to measure how good a mixed strategy when played against a specific mixed strategy for the other player. For example, say in rock, paper, scissors the two players decided to use the following strategies:

Player	Rock	Paper	Scissors
One	1/2		1/2
Two	1/2	1/3	1/6

How do those strategies fare against each other? If the game goes on for N rounds, we'd expect it to go approximately like this (where P_i means player #i):

# of rounds	P_1	P_2	P_1's Payoff
$N/4$	rock	rock	0
$N/6$	rock	paper	-1
$N/12$	rock	scissors	1
$N/4$	scissors	rock	-1
$N/6$	scissors	paper	1
$N/12$	scissors	scissors	0

So the net winnings for player one over the course of those N rounds will be $-N/6 + N/12 - N/4 + N/6 = -N/6$, for an average of $-1/6$ per round. Notice that the N just "comes along for the ride" and cancels in the end. We could find the average winnings just thinking about probabilities: for example, with probability $(1/2)(1/3) = 1/6$, player one picks rock and player two picks paper, and since with those moves player one loses 1, this contributes a $-1/6$ to the average winnings.

In general, assuming player one will follow the mixed strategy $\mathbf{x}$ and that player two will follow the mixed strategy $\mathbf{y}$, we can compute the expected[23] winnings for player one as follows: with probability $x_i y_j$ player one will make move #i and player two will make move #j; that combination results in player one winning a_{ij}, so in total the expected winnings of player one are $\sum_{i,j} a_{ij} x_i y_j = \mathbf{x}^\top A \mathbf{y}$.

21.2 The value of a game

Now that we know how to evaluate any pair of mixed strategies against each other, let's think about how to find optimal strategies. Let's think

[23] "Expected" is a term from probability theory, you can think of these expected winnings as the average winnings per round of player one if both players used those strategies over the course of many, many rounds.

from the point of view of player one. Player one has no control over what strategy player two will use, so to play it safe, player one should pick one that maximizes her winnings assuming player two responds as well as possible against the strategy. If player one use strategy $\mathbf{x}$, the best player two can do against that would be to pick a strategy $\mathbf{y}$ that *minimizes* $\mathbf{x}^\top A\mathbf{y}$ — remember that that expression computes the expected winnings for player *one*, so player two wants to minimize it! So if player one uses strategy $\mathbf{x}$, she can't guarantee she'll win more than $\min_{\text{stoch. }\mathbf{y}} \mathbf{x}^\top A\mathbf{y}$ (where the minimum is taken over all stochastic vectors $\mathbf{y}$) on average per round. To be prepared for player two playing as smart as possible, player one should choose the strategy $\mathbf{x}$ that maximizes that number, i.e., player one should compute:

$$v(A) = \max_{\text{stoch. }\mathbf{x}} \left(\min_{\text{stoch. }\mathbf{y}} \mathbf{x}^\top A\mathbf{y} \right).$$

We'll call that number the **value of the game**. It is the largest number v so that player one has a mixed strategy that guarantees an expected winnings per round of at least v. To compute $v(A)$ we will phrase the maximization as a linear program, which can then solved via the Simplex Method.

The key step is that the inner minimization is actually very simple! For a fixed stochastic vector $\mathbf{x}$ we want to find the minimum value of $\mathbf{x}^\top A\mathbf{y}$ where y varies over all stochastic vectors (of the appropriate size). We can think of this in terms of the game: player one is committed to using strategy $\mathbf{x}$, player two knows this and is trying to figure out how to best respond. In that situation, player two could simply figure out which pure strategy is best against $\mathbf{x}$ and just use that! For example, if in rock, paper, scissors player one proclaims: "I shall play rock 70% of the time, paper 20% and scissors 10%!", player two should respond by always playing paper!

In more algebraic terms, what we are saying is that

$$\min_{\text{stoch. }\mathbf{y}} \mathbf{x}^\top A\mathbf{y} = \min \text{entry of } \mathbf{x}^\top A.$$

This is not hard to prove formally: if $\mathbf{x}^\top A = (s_1, s_2, \ldots, s_m)$ and s_j is the smallest of those entries, we have for any stochastic vector $\mathbf{y}$:

$$\begin{aligned} \mathbf{x}^\top A\mathbf{y} &= s_1y_1 + s_2y_2 + \cdots + s_my_m \\ &\geq s_jy_1 + s_jy_2 + \cdots s_jy_m = s_j(y_1 + \cdots + y_m) = s_j. \end{aligned}$$

Since when we take y to be the pure strategy for move #j equality is attained, we can conclude that $\min_{\text{stoch. }\mathbf{y}} \mathbf{x}^\top A\mathbf{y} = s_j$.

Using this we can easily turn the computation of the value of the game into a linear program. We'll add a variable z to stand for the minimum entry of $\mathbf{x}^\top A$. Of course, we can't simply put "$z = \min$ entry of $\mathbf{x}^\top A$" as a constraint in the linear program. But if we put in constraints that say "$z \leq j$th entry of $\mathbf{x}^\top A$" for each j, then those already tells us z is *at most* the minimum entry. Since z is exactly what we wish to maximize, in any optimal solution z will be equal to that minimum entry. Putting it all together this is the LP we get:

Maximize z

subject to

$$\begin{aligned} \begin{pmatrix} z & z & \dots & z \end{pmatrix} &\leq \mathbf{x}^\top A \\ x_1 + x_2 + \cdots + x_n &= 1 \\ x_1, x_2, \dots, x_n &\geq 0 \end{aligned}$$

The optimal value of this LP is $v(A)$, the value of the game with matrix A. Notice that this LP always has an optimal value:

- It is feasible because we can get a feasible solution by taking $\mathbf{x}$ to be *any* stochastic vector and z to be the minimum entry of $\mathbf{x}^\top A$.
- It is bounded because as $\mathbf{x}$ varies over all stochastic vectors, the entries of $\mathbf{x}^\top A$ are bounded (by $\sum_{i,j} |a_{ij}|$, for example), and thus so is the objective function z.

21.3 The minimax theorem

The analysis above is written from the point of view of player one. Player two could also carry out similar reasoning and reach the conclusion that the smallest number v such that player *two* has strategy that guarantees that player *one* can't find a strategy against it with more than v expected winnings is equal to

$$\min_{\text{stoch. } \mathbf{y}} \left(\max_{\text{stoch.} \mathbf{x}} \mathbf{x}^\top A \mathbf{y} \right).$$

We can also repeat the argument in the last section: first, to show that

$$\max_{\text{stoch. } \mathbf{x}} \mathbf{x}^\top A \mathbf{y} = \max \text{entry of } A\mathbf{y},$$

and second, to find an LP whose optimal value is that minimum that player two wants:

Minimize w

subject to

$$\begin{aligned} \begin{pmatrix} w & w & \dots & w \end{pmatrix}^\top &\geq A\mathbf{y} \\ y_1 + y_2 + \cdots + y_m &= 1 \\ y_1, y_2, \dots, y_m &\geq 0 \end{aligned}$$

If you look at this LP carefully you'll notice something wonderful: it's the dual of the LP for player one![24] Let's figure out what duality theory for these LP's means in terms of the games. For that, notice that there is a strong connection between feasible solutions and mixed strategies. Take the LP for player one, for instance. Its variables are the x_i and an extra variable z. A feasible solution is precisely a stochastic vector $\mathbf{x}$ together with a number z that is at most the minimum entry of the vector $\mathbf{x}^\top A$. Given any stochastic vector $\mathbf{x}$ we can get a corresponding feasible solution simply by setting $z :=$ min entry of $\mathbf{x}^\top A$; and this value of z is the maximum we can take for a feasible solution that includes $\mathbf{x}$.

Using that correspondenc we can translate results from duality theory for LPs to results about matrix games.

Strong duality tells us the optimal values of primal and dual must agree (when those LPs have optimal solutions, which as we saw above, is guaranteed in this case). This translates to:

Theorem 7. *(**Minimax**) For any matrix A the value of the game with payoff matrix A can be computed in the folowing two equivalent ways:*

$$v(A) = \max_{\text{stoch. } \mathbf{x}} \left(\min_{\text{stoch. } \mathbf{y}} \mathbf{x}^\top A\mathbf{y} \right) = \min_{\text{stoch. } \mathbf{y}} \left(\max_{\text{stoch.}\mathbf{x}} \mathbf{x}^\top A\mathbf{y} \right).$$

We can also use duality theory to get a very convenient way to check whether or not strategies are optimal. The fact that given feasible primal and dual solutions they are optimal if and only if their objective functions agree translates to:

Theorem 8. *Let A be the payoff matrix of a game. Then two stochastic vectors $\mathbf{x}^*$ and $\mathbf{y}^*$ are optimal strategies for players one and two if and only if*

$$\min \textit{entry of } {\mathbf{x}^*}^\top A = \max \textit{entry of } A\mathbf{y},$$

in which case the common value of those expressions is the value of the game.

[24] Exercise: carefully check this!

If you have a good guess as to what optimal strategies for players one and two are then the above theorem is certainly the best way to prove your guess is correct. But even if you only have a guess for player one it might be that the easiest way to show you're right is to make a guess for an optimal strategy for player two and check using this theorem.

21.4 Symmetric games are fair

A game is called **fair** is its value is zero. This means that on average the wins and loses for each player balance out. Remember that a game is *symmetric* if "its rules are the same for player one and player two", or mor formally, if its payoff matrix is antisymmetric: $A = -A^\top$. It seems obvious that a symmetric game is fair: if the rules are the same for both player, neither player should have an advantage! Let's show that.

Lemma 1. *For any payoff matrix A, $v(A) = -v(-A^\top)$.*

Notice that this is for an arbitrary matrix, we are not assuming that $A = -A^\top$.

Proof. We can argue directly in terms of games: remember that $-A^\top$ is the payoff matrix from the point of view of player two, so $v(-A^\top)$ is the value of the game from the point of view of player two. This of course is just $-v(A)$.

Alternatively, we can compute from the formula for the value:

$$\begin{aligned}
v(-A^\top) &= \max_{\text{stoch. } \mathbf{x}} \left(\min_{\text{stoch. } \mathbf{y}} \mathbf{x}^\top (-A^\top) \mathbf{y} \right) \\
&=^{(1)} \max_{\text{stoch. } \mathbf{x}} \left(\min_{\text{stoch. } \mathbf{y}} -\mathbf{y}^\top A \mathbf{x} \right) \\
&=^{(2)} \max_{\text{stoch. } \mathbf{x}} \left(- \max_{\text{stoch. } \mathbf{y}} \mathbf{y}^\top A \mathbf{x} \right) \\
&=^{(2)} - \min_{\text{stoch. } \mathbf{x}} \left(\max_{\text{stoch. } \mathbf{y}} \mathbf{y}^\top A \mathbf{x} \right) \\
&=^{(3)} - \min_{\text{stoch. } \mathbf{y}} \left(\max_{\text{stoch. } \mathbf{x}} \mathbf{x}^\top A \mathbf{y} \right) = -v(A).
\end{aligned}$$

Where the equalities are true for the following reasons:

1. Since $\mathbf{x}^\top A^\top \mathbf{y}$ is a 1×1 matrix it is automatically equal to its tranpose $\mathbf{x}^\top A^\top \mathbf{y} = \left(\mathbf{x}^\top A^\top \mathbf{y}\right) = \mathbf{y}^\top A \mathbf{x}$

2. The minimum of minus something is equal to minus the maximum of that something.

3. The names of the variables are irrelevant, we can rename **x** to **y** and viceversa.

□

Now if the game is symmetric, that is, if $A = -A^\top$, the lemma tells us that $v(A) = -v(-A^\top) = -v(A)$, from which we conclude that $v(A) = 0$.

21.5 Dominated moves can be removed

Certain moves are never a good idea. If player one has two moves, say #*k* and #*i* such that against every move of player two the payoff using move #*k* is at least as much as it is for move #*i*, then player one can simply decide to never use move #*i* and doesn't miss out any potential payoff. We'll say that move #*k* **dominates** move #*i*.

We can state that dominated moves are irrelevant as follows:

Proposition 2. *If a payoff matrix A satisfies that $\text{row}_k(A) \geq \text{row}_i(A)$, then there exists an optimal strategy $\mathbf{x}^*$ for player one in which $x_i^* = 0$.*

Before we prove this, a couple of remarks:

- Notice that we can't claim that in *every* optimal strategy for player one we have $x_i^* = 0$, i.e., that move #*i* is never used in an optimal: for one thing rows *k* and *i* could be equal, but even if they are not, they could differ only in entries occurring in columns that an optimal strategy for player two doesn't need to use, in which case the difference is not relevant. The best we can say is that it is *possible* to avoid using move #*i* in an optimal strategy.

- By symmetry there is an analogous proposition for player two: if $\text{col}_k(A) \geq \text{col}_i(A)$, then there is an optimal strategy $\mathbf{y}^*$ for player two in which $y_k^* = 0$. Notice that with that inequality, move #*k* is the one that player two can avoid: it always give a payoff *for player one* that is at least as large as the one from move #*i*.

Proof. We'll argue that given any optimal strategy $\mathbf{x}^*$ for player one, if we shift all the probability from move #*i* to move *k* (to get a new strategy that gives move #*i* a probability of 0), the new strategy is also optimal.

Let $\mathbf{e}_i$ be the stochastic vector that has a 1 in spot i and 0s everywhere else. Then the strategy obtained from $\mathbf{x}^*$ by shifting all the weight from move #i to move #k is $\mathbf{x}^\circ := \mathbf{x}^* - x_i^*\mathbf{e}_i + x_i^*\mathbf{e}_k$. To check this new mixed strategy is still optimal, we just need to show that for the corresponding feasible solution of the player one LP the objective function is at least as large as it was for $\mathbf{x}^*$. Since $\mathbf{x}^*$ achieved the maximum, that would imply the objective functions are actually equal and the strategy $\mathbf{x}^\circ$ is optimal too.

So we need to show that $\min$ entry of ${\mathbf{x}^\circ}^\top A \geq \min$ entry of ${\mathbf{x}^*}^\top$. This follows from:

$$\begin{aligned}
{\mathbf{x}^\circ}^\top A &= \left(\mathbf{x}^* + x_i^*(\mathbf{e}_k - \mathbf{e}_i)\right)^\top A \\
&= {\mathbf{x}^*}^\top A + x_i^*(\mathbf{e}_k^\top A - \mathbf{e}_i^\top A) \\
&= {\mathbf{x}^*}^\top A + x_i^*(\mathrm{row}_k(A) - \mathrm{row}_i(A)) \\
&\geq {\mathbf{x}^*}^\top A.
\end{aligned}$$

□

Once we know that a move *can* be avoided, we can just remove that row or column from the matrix and get a smaller game *with the same value*! And sometimes after removing a move there appears a new dominated move that can be removed in turn and so on. For example, consider the game with payoff matrix:

$$A = \begin{pmatrix} 3 & 4 & 1 & -5 & -2 \\ 1 & -3 & 10 & -1 & 2 \\ 7 & -1 & 8 & 5 & 6 \\ 8 & -2 & 9 & 4 & 3 \end{pmatrix}.$$

Columns 1 and 3 are both dominated by column 4, so we can reduce to:

$$A' = \begin{pmatrix} 4 & -5 & -2 \\ -3 & -1 & 2 \\ -1 & 5 & 6 \\ -2 & 4 & 3 \end{pmatrix}.$$

Now rows 2 and 4 are dominated by row 3 (and this wasn't true in A!), so we can reduce to:

$$A'' = \begin{pmatrix} 4 & -5 & -2 \\ -1 & 5 & 6 \end{pmatrix}.$$

Finally, column 3 is dominated by column 2, so we can reduce to:

$$A''' = \begin{pmatrix} 4 & -5 \\ -1 & 5 \end{pmatrix}.$$

At this point no move dominates another move so if we want to compute $v(A''')$, we need to do something else. We could of course just use the Simplex Method on the LP for player one:

Maximize z

subject to

$$\begin{aligned} z &\leq 4x_1 - x_2 \\ z &\leq -5x_1 + 5x_2 \\ x_1 + x_2 &= 1 \\ x_1, x_2 &\geq 0 \end{aligned}$$

But maybe for such a small problem it's not worth it to bust out the Simplex Method. Just let $t = x_1$ so $x_2 = 1 - t$. The LP becomes:

Maximize z

subject to

$$\begin{aligned} z &\leq 5t - 1 \\ z &\leq 5 - 10t \\ 0 &\leq t \leq 1 \end{aligned}$$

Now, $5t - 1 \leq 5 - 10t$ when $t \leq 2/5$. So for $0 \leq t \leq 2/5$ we have $z \leq 5t - 1$ and the maximum is for $t = 2/5, z = 1$. For $2/5 \leq t \leq 1$ we have $z \leq 5 - 10t$ and the maximum is again $t = 2/5, z = 1$.

So we get that $v(A''') = 1$ and that an optimal strategy is given by $\begin{pmatrix} 2/5 & 3/5 \end{pmatrix}^\top$. This in turn means that $v(A) = 1$ with optimal strategy $\begin{pmatrix} 0 & 2/5 & 0 & 3/5 & 0 \end{pmatrix}^\top$.

21.6 Example games

21.6.1 Battleship

Let's look at an overly simplified version of the game of Battleship. In Battleship each player hides some rectangular ships on a board and then the two players take turns picking locations on the board to shoot, try to sink the other player's ships. Our simplified version pares this down a lot:

There is a rectangular board of size $m \times n$. Player one moves by placing a ship of length ℓ somewhere on the board (either horizontally as an $\ell \times 1$ rectangle, or vertically as a $1 \times \ell$ rectangle). Player two picks one of the mn squares on the board to shoot at. Player two wins the round if he hits the

ship that player one hid, otherwise player one wins the round. The payoff is 1 to the winner of the round.

Let analyze the example of a board of size 2×3 with player one placing a ship of length 2.

1. LINDO LP generator

   If you are reading this in PDF form, the rest of this section is empty, but in the webpage version, there is an interactive Battleship LP generator that will write the LP to compute the value of the game given the size of the board and the size of the ship.

21.6.2 Morra

The game of Morra is played as follows. In each round each player hides either one or two francs and also guesses how many francs the other player hid (they guess in secret and reveal their guesses simultaneously). If either both players guess incorrectly or both players guess correctly the round is a tie and no money changes hands. But if exactly one player guesses correctly, then that player gets to keep the francs hidden by *both* players!

So the moves can be described by a pair of numbers: how many francs you hide and how many francs you guess the other player hid. We'll use, for instance, H1G2 to indicate you hide 1 and guess the other player hid 2. The payoff matrix is then:

	H1G1	H1G2	H2G1	H2G2
H1G1	0	2	-3	0
H1G2	-2	0	0	3
H2G1	3	0	0	-4
H2G2	0	-3	4	0

This matrix is anti-symmetric, of course: the game's rules as the same for both players. So the value of the game is zero. You can confirm this in LINDO *and* find a pair of optimal strategies with the following LP:

```
max z
subject to
  z+2x2-3x3 < 0
  z-2x1+3x4 < 0
  z+3x1-4x4 < 0
  z-3x2+4x3 < 0
```

```
  x1+x2+x3+x4 = 1
end
free z
```

The LINDO output is:

```
LP OPTIMUM FOUND AT STEP      4

       OBJECTIVE FUNCTION VALUE

       1)     0.0000000E+00

 VARIABLE        VALUE          REDUCED COST
        Z         0.000000          0.000000
       X2         0.600000          0.000000
       X3         0.400000          0.000000
       X1         0.000000          0.142857
       X4         0.000000          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         0.000000          0.000000
       3)         0.000000          0.571429
       4)         0.000000          0.428571
       5)         0.200000          0.000000
       6)         0.000000          0.000000

NO. ITERATIONS=       4
```

This tells us that the following are optimal strategies:

- For player one: 60% of the time hide 1, guess 2; 40% of the time hide 2, guess 1.
- For player two: 57.1429% = 4/7 of the time hide 1, guess 2; 42.8571% = 3/7 of the time hide 2, guess 1.

Of course, there are other optimal strategies, the Simplex Method is content with finding one.

Now imagine that after a while of playing player two gets bored and cocky and proposes a variant to player one: both players hide 1 or 2 francs

as before, then player two make a guess *and reveals it*, then player one guesses (player one is allowed to change her guess based on player two's guess, but not to change the number of francs she hides). Player two figures that this won't make a difference: she's not revealing the number of francs she hid, only the guess, and player one can't change the number of francs she hid after hearing player two's guess, only guess differently than she would have. It seems to player two that this will mess with player one's mind without actually conferring any advantage, i.e., the value of the game will still be 0. Is she right?

Let's see! Player one now has four guessing options: as before she can guess 1 or guess 2, but now she also has the option of guessing the *same* as player two (we'll denote this by "S" for same), or guessing *differently* than player one. The new payoff matrix is:

	H1G1	H1G2	H2G1	H2G2
H1G1	0	2	-3	0
H1G2	-2	0	0	3
H2G1	3	0	0	-4
H2G2	0	-3	4	0
H1GS	0	0	-3	3
H1GD	-2	2	0	0
H2GS	3	-3	0	0
H2GD	0	0	4	-4

Let's put this new game into LINDO:

```
max z
subject to
  z+2x2-3x3+2x6-3x7 < 0
  z-2x1+3x4-2x6+3x7 < 0
  z+3x1-4x4+3x5-4x8 < 0
  z-3x2+4x3-3x5+4x8 < 0
  x1+x2+x3+x4+x5+x6+x7+x8=1
end
free z
```

The result may be a little surprising:

```
LP OPTIMUM FOUND AT STEP      7

        OBJECTIVE FUNCTION VALUE
```

```
       1)     0.4040404E-01

 VARIABLE        VALUE          REDUCED COST
        Z         0.040404          0.000000
       X2         0.565657          0.000000
       X3         0.404040          0.000000
       X6         0.020202          0.000000
       X7         0.000000          0.101010
       X1         0.000000          0.070707
       X4         0.000000          0.101010
       X5         0.000000          0.070707
       X8         0.010101          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000          0.282828
        3)         0.000000          0.303030
        4)         0.000000          0.212121
        5)         0.000000          0.202020
        6)         0.000000          0.040404

 NO. ITERATIONS=       7
```

The value of the game is 4/99! So player two was wrong: revealing her guess first does confer a slight advantage to player one!

22 All problems are basically the same

We've explained how to use the Simplex Method to solve systems of linear inequalities and to find optimal strategies for matrix games. In both cases this was accomplished by taking the problem we wanted to solve and using it to create a related linear program whose solution would also tell us how to solve the starting problem. This means that not only the Simplex Method, but *any* method for solving LPs can be used to for both those tasks. In particular, linear programming is at least as hard as those other problems: an easy method for solving linear programs would give us an easy method for these other problems. But is linear programming equally hard or is it harder?

We'll compare the following three problems:

1. **LP**: Solving linear programs.

2. **Ineq**: Solving systems of linear inequalities (and you can allow some equations thrown in too).

3. **Games**: Finding optimal strategy for matrix games.

To show two such problems, say A and B, are "equally hard" we have to **reduce** A to B *and* vice versa. Reducing A to B means explaining:

- Given P, an example of a problem of type A (called an **instance** of A) how to construct an instance Q of B whose solution will helps us solve P.

- How to interpret a solution for Q to produce a solution of P.

If we can do both of these things any method of solving problems of type B can be used to solve problems of type A too. So solving problems of type A can't really be harder than solving problems of type B.

22.1 Warm up: LP and LP_{std}

As an easy warm up let's consider the tasks LP of solving linear problems and LP_{std} of solving linear programs in standard form. Let's be clear about what a solution to an LP means in this context: it means that we determine if the LP is infeasible, unbounded or has an optimal solution; if it is unbounded we find a 1-parameter unbounded family of feasible solutions, and if it has optimal solutions we find one.

Clearly it's not harder to solve standard form LPs than to solve arbitrary LPs! So LP_{std} reduces to LP trivially: given a standard form LP you don't have to do anything to convert it to an LP, and if you have a solution for it thought of as an LP that is obviously also a solution for it thought of as a standard form LP.

The interesting reduction is to go the other way: reduce solving arbitrary LPs to solving standard form LPs. This is one of the first things we studied: converting LPs to standard form. We gave a simple procedure for doing so, that came with a corresponding procedure to interpret the solution of the standard form LP as a solution to the original LP.

The only reason I bring this simple example of comparing two problems up is to point out that when you convert an LP, say P, to standard form, say P_{std}, you really do have to do *something* to interpret the solution of P_{std} as a

solution to P. It's not hard, but it's not nothing either. Here's a sketch of the procedure:

If P_{std} is infeasible, unbounded or has an optimal solution so does P. In the infeasible case that's all there is to say, in the other cases we need to know how the variables of the two LPs correspond. If P had any free variables, say x, then P_{std} has in place of x two variables x' and x''; a solution for P_{std} will include values for x' and x'', and to get the value of x in the corresponding solution of P we set $x := x' - x''$. Also, if the objective function of P had a constant term, for example, if P started with "maximize $x_1 + 2x_2 - x_3 + 7$", then P_{std} had objective function $x_1 + 2x_2 - x_3$, without the 7, so the optimal value of P is the optimal value of P_{std} plus 7. Even if P had no constant term, if it was a minimization question, the optimal value of P is not the same as the optimal value of P_{std}: they differ in sign.

OK, enough about that let's pass to more interesting comparisons.

22.2 *LP* and *Ineq*

Let's start by being clear about what it should mean to solve a system of linear inequalities: we should be able to decide whether they have a simultaneous solution and if they do, to find one.

We already discussed a reduction in one direction between *LP* and *Ineq*. Pause a bit to make sure you know which direction and what we called it.

The reduction of *Ineq* to *LP* is basically Phase 1 of the Simplex Method! Not quite, because we also need to convert to standard form to use the exact construction we used in Phase 1. Let's review it:

To solve a system of linear inequalities, you first convert it to standard form $A\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq 0$, just like you do for the constraints of an LP. Once it is in that form you make the auxiliary LP:

$$\begin{aligned} &\text{Maximize } -x_0 \\ &\text{subject to } \begin{cases} A\mathbf{x} - \begin{pmatrix} x_0 & x_0 & \dots x_0 \end{pmatrix}^\top \leq \mathbf{b} \\ x_0, \mathbf{x} \geq 0 \end{cases} \end{aligned}$$

How do you interpret the solution to this LP as a solution to the system of inequalities? Well, this LP always has an optimal solution and all optimal solutions have the same value of x_0[25]. If $x_0 = 0$ then the system of inequalities has solutions and, for the standard form version, a solution

[25] Exercise: Why was that again?

is given by the vector $\mathbf{x}$. If $x_0 > 0$ then the system of inequalities has no solution.

What about the other direction? Does is sound plausible that you can reduce *LP* to *Ineq*? That having the ability to solve systems of inequalities means you also have the ability to solve any linear program? It does, by the following clever trick!

Take any LP. We may as well assume it is in standard form, say

$$\text{Maximize } \mathbf{c} \cdot \mathbf{x}$$
$$\text{subject to } \begin{cases} A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases}$$

We need to find out if it is infeasible, unbounded or has an optimal solution, and we are assuming we have the ability to solve systems of linear inequalities. We can first check if the LP is feasible simply by checking if the system of inequalities given by the constraints has a solution. If not, we are done: the LP is infeasible.

So now assume the LP is feasible. Look at the dual:

$$\text{Minimize } \mathbf{b} \cdot \mathbf{y}$$
$$\text{subject to } \begin{cases} A^\top \mathbf{y} \geq \mathbf{c} \\ \mathbf{y} \geq 0 \end{cases}$$

We can use our powers of inequality solving to find out if the dual is feasible. If not, the primal was unbounded (by strong duality).

So now assume both primal and dual are feasible. At this stage we know the primal must *have* an optimal solution, but we still need to *find* one. To do that we can form the following system of inequalities:

$$\begin{aligned} A\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq 0 \\ A^\top \mathbf{y} &\geq \mathbf{c} \\ \mathbf{y} &\geq 0 \\ \mathbf{c} \cdot \mathbf{x} &\geq \mathbf{b} \cdot \mathbf{y} \end{aligned}$$

We put in the constraints of both the primal and the dual, and also the constraint that the objective function of the primal is greater than or equal to that of the dual. If this system of inequalities has a solution, then $\mathbf{x}$ is feasible for the primal, $\mathbf{y}$ is feasible for the dual and the objective functions

agree: indeed, the system includes the inequality $\mathbf{c} \cdot \mathbf{x} \geq \mathbf{b} \cdot \mathbf{y}$ and the opposite, $\mathbf{c} \cdot \mathbf{x} \leq \mathbf{b} \cdot \mathbf{y}$ is true by weak duality. Then, again by duality theory, $\mathbf{x}$ is optimal for the primal.

Notice that in this reduction we reduced solving an LP to solving three systems of inequalities. We can easily save one: just try the last system of inequalities first. That'll discover if the LP has an optimal solution. If not, try the constraints of the LP: that'll tells us if the LP was infeasible or unbounded.

If we had defined "solution of an LP" to mean only "find an optimal solution if there is one, otherwise just say there is no optimal solution"; call that modified problem LP_{opt}. Then we would only need the last system of inequalities, the one with both $\mathbf{x}$ and $\mathbf{y}$ in it.

22.3 *LP* and *Games*

We've already seen how to use linear programming to find optimal strategies of games, that is we saw a reduction of *Games* to *LP*. Can we go the other way? Almost, but again the way to do it is a clever trick.

Say you have an LP, that we may as well assume is in standard form:

$$\begin{aligned} &\text{Maximize } \mathbf{c} \cdot \mathbf{x} \\ &\text{subject to } \begin{cases} A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \end{aligned}$$

We'll build a payoff matrix out of it as follows:

$$M = \begin{pmatrix} 0 & -\mathbf{c}^\top & \mathbf{b}^\top \\ \mathbf{c} & 0 & -A^\top \\ -\mathbf{b} & A & 0 \end{pmatrix}$$

Notice that $M = -M^\top$ so that $v(M) = 0$. We won't quite be able to interpret an arbitrary optimal strategy for M as a solution for the LP we started with, so this technically won't reduce *LP* to *Games*. Here's what we can prove:

Theorem 9. *The LP above has an optimal solution if and only there is an optimal strategy for M that uses the first move with positive probability. If $\mathbf{u} = \begin{pmatrix} s \\ \mathbf{x}' \\ \mathbf{y}' \end{pmatrix}$ is such an optimal strategy with $s > 0$, then $\mathbf{x} := \mathbf{x}'/s$ is an optimal primal solution and $\mathbf{y} := \mathbf{y}'/s$ is an optimal dual solution.*

Before we get to the proof: why does this talk about optimal strategy without specifying the player? Well, since $M = -M^\top$, we have $(\mathbf{u}^\top M)^\top = M^\top \mathbf{u} = -M\mathbf{u}$, so the minimum entry of $\mathbf{u}^\top M$ is $v(M) = 0$ if and only if the maximum entry of $M\mathbf{u}$ is 0. This shows that $\mathbf{u}$ is optimal for player one if and only if it is optimal for player two.

Proof. Assume first that $\mathbf{u} = \begin{pmatrix} s \\ \mathbf{x}' \\ \mathbf{y}' \end{pmatrix}$ is an optimal strategy. Then we know that the maximum entry of $M\mathbf{u} = \begin{pmatrix} -\mathbf{c}\cdot\mathbf{x}' + \mathbf{b}\cdot\mathbf{y}' \\ s\mathbf{c} - A^\top\mathbf{y}' \\ -s\mathbf{b} + A\mathbf{x}' \end{pmatrix}$ is $v(M) = 0$. In particular all of the entries of that vector must be ≤ 0. If $s > 0$, we can divide all of those inequalities by s to get that $\mathbf{x} := \mathbf{x}'/s$ is a feasible primal solution, that $\mathbf{y} := \mathbf{y}'/s$ is a feasible dual solution and that $\mathbf{c}\cdot\mathbf{x} \geq \mathbf{b}\cdot\mathbf{y}$, which implies both are optimal.

Conversely, if the LP has an optimal solution $\mathbf{x}$ and an optimal solution $\mathbf{y}$ we can set $s := \sum_j x_j + \sum_i y_i + 1$, $\mathbf{x}' := s\mathbf{x}$, $\mathbf{y}' := s\mathbf{y}$ and $\mathbf{u} := \begin{pmatrix} s \\ \mathbf{x}' \\ \mathbf{y}' \end{pmatrix}$. It is easy to check that $\mathbf{u}$ is stochastic and, reversing the work above, that $M\mathbf{u} \leq 0$ and that the first entry is $-\mathbf{c}\cdot\mathbf{x}' + \mathbf{b}\cdot\mathbf{y}' = s(-\mathbf{c}\cdot\mathbf{x} + \mathbf{b}\cdot\mathbf{y})$, which is 0 by strong duality. So the maximum entry of $M\mathbf{u}$ is $v(M) = 0$ and thus $\mathbf{u}$ is an optimal strategy in which $s > 0$. □

So we didn't quite reduce *LP* to *Games* for two reasons:

- We can only tell if the LP has an optimal solution or not. In case it doesn't, this game doesn't seem to tells us whether the LP is infeasible or unbounded.
- We weren't using just the ability of finding optimal strategies for matrix games, but rather the souped up ability to tell whether or not some optimal strategy uses the first move and if so to find one.

So, to be more precise we reduced LP_{opt} to *Games'*, where:

- LP_{opt} is the problem of determining whether an LP has an optimal solution and finding one if it does, and
- *Games'* is the problem of determining whether or not a game has an optimal strategy for player one that assigns the first move a non-zero probability and finding such an optimal strategy if it does.

23 The cutting stock problem

Chapter 13 of Chvátal's *Linear Programming* explains that paper is produced in big, wide rolls called *raws*. Customers can order rolls of different widths which are formed by cutting the raws; the resulting rolls are called *finals*. For example, a manufacter that makes raws 200cm wide might receive orders for several finals of 47cm and 33cm and decide to cut a raw into two rolls of width 47cm and three rolls of width 33cm with 7cm left over as waste (200 = 47×2 + 33×3 + 2).

We can use linear programming to find the most economical way of cutting raws into finals. This is known as the *cutting-stock* problem.

23.1 Specification

Width of raws:

23.2 Cutting Patterns

23.3 Linear Program