Optimization Methods in Management Science / Operations Research 15.053/058

maximize $z = \sum_{j=1}^{n} c_j x_j$

subject to $\sum_{j=1}^{x} a_{ij} x_j = b_i$ for i = 1 to m

$x_j \geq 0$ for j = 1 to n

Algebraic Formulations

Algebraic Formulations

Usually in class, we describe linear programs by writing them out fully. This is fine for small linear programs, but it does not work when the linear programs are very large. In that case, it helps to use ***algebraic formulations***.

Algebraic formulations sound hard. But they are not so hard. However, they do take a while to get used to.

In this tutorial, we will explain algebraic formulations with some examples.

On Creating Algebraic Formulations

When we create algebraic formulations, we rely on substituting notations for some of the coefficients. Let's start with an example of a linear program.

min $500\, x_1 + 200\, x_2 + 250\, x_3 + 125\, x_4$

s.t. $50{,}000\, x_1 + 25{,}000\, x_2 + 20{,}000\, x_3 + 15{,}000\, x_4 \geq 1{,}500{,}000$

$$0 \leq x_1 \leq 20$$

$$0 \leq x_2 \leq 15$$

$$0 \leq x_3 \leq 10$$

$$0 \leq x_4 \leq 15$$

This is the MSR example which is described on the next slide.

MSR Marketing Inc.

adapted from Frontline Systems

- **Need to choose ads to reach at least 1.5 million people**
  - **Minimize Cost**
  - **Upper bound on number of ads of each type**

	TV	Radio	Mail	Newspaper
Audience Size	50,000	25,000	20,000	15,000
Cost/Impression	$500	$200	$250	$125
Max # of ads	20	15	10	15

Decision variables:

- x_1 is the number of TV ads.
- x_2 is the number of radio ads.
- x_3 is the number of mail ads.
- x_4 is the number of newspaper ads.

The LP Formulation again

$$\min \quad 500\,x_1 + 200\,x_2 + 250\,x_3 + 125\,x_4$$

$$\text{s.t.} \quad 50{,}000\,x_1 + 25{,}000\,x_2 + 20{,}000\,x_3 + 15{,}000\,x_4 \geq 1{,}500{,}000$$

$$0 \leq x_1 \leq 20$$

$$0 \leq x_2 \leq 15$$

$$0 \leq x_3 \leq 10$$

$$0 \leq x_4 \leq 15$$

Illustration of the objective function and constraints:

- The objective is to minimize the cost of advertising.
- The first constraint says that the number of people who see the ads is at least 1.5 million.
- The remaining four constraints give upper and lower bounds on the number of showings of each of the four ads.

Transforming into an algebraic problem

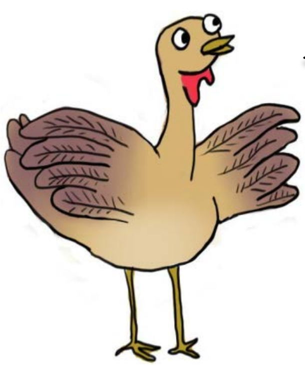
We'll transform this problem into an algebraic version in a couple of stages. Then we'll show how to do it all at once.

So, let's start with the four upper bound constraints. Suppose that we let d = (d1, d2, d3, d4) = (20, 15, 10, 15). We can then write the linear program as follows:

min $\quad 500\,x_1 + 200\,x_2 + 250\,x_3 + 125\,x_4$

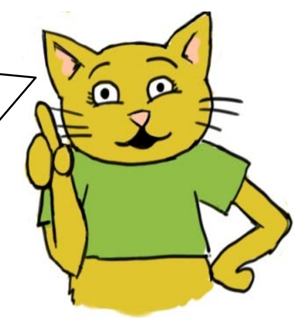
s.t. $\quad 50{,}000\,x_1 + 25{,}000\,x_2 + 20{,}000\,x_3 + 15{,}000\,x_4 \geq 1{,}500{,}000$

$$0 \leq x_j \leq d_j \text{ for } j = 1 \text{ to } 4.$$



Is d_j decision variable?

It looks like d_j is a variable, but it isn't. It's called a "*parameter*" and it means that there is an associated value stored for it somewhere, perhaps in a spreadsheet, perhaps in a database.

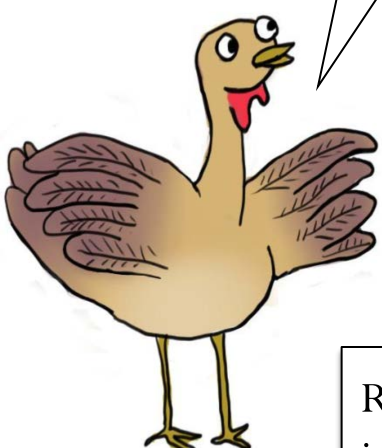


Parameters versus decision variables

I don't get it. The d's don't look like numbers to me.

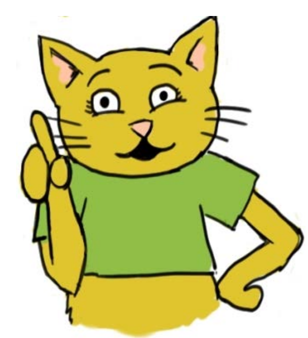
Most students (and others) find this confusing at the beginning. But after a while, one gets used to it.

For students who have seen linear algebra, it's pretty similar to when one first sees a system of linear equations expressed as $Ax = b$.



Remember that even in Algebra 1, the equation for a line is often represented as

"$ax + by = c$."

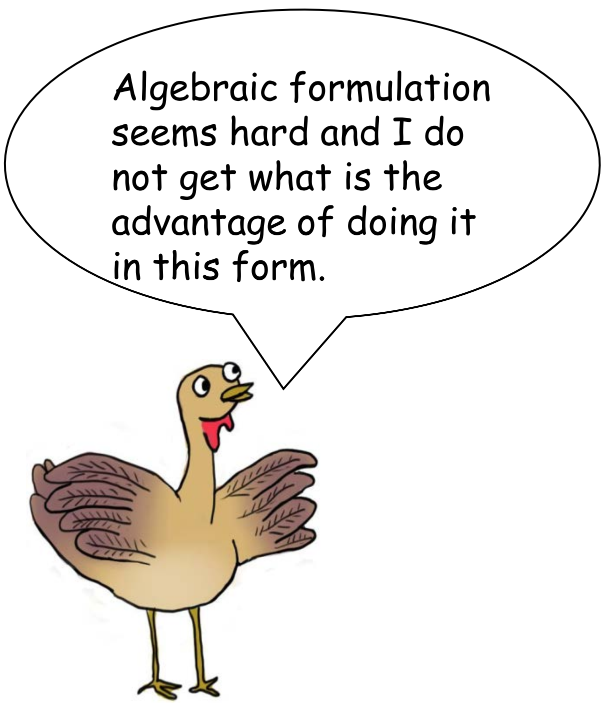


More on the algebraic formulation

min $500\,x_1 + 200\,x_2 + 250\,x_3 + 125\,x_4$

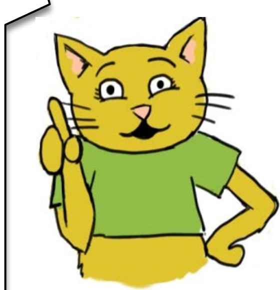
s.t. $50{,}000\,x_1 + 25{,}000\,x_2 + 20{,}000\,x_3 + 15{,}000\,x_4 \geq 1{,}500{,}000$

$$0 \leq x_j \leq d_j \text{ for } j = 1 \text{ to } 4.$$



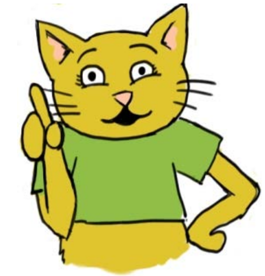
The key advantage of the algebraic formulation is that the formulation becomes "independent" of the data. For example, if we were to change the upper bounds on the x's, this more algebraic version would still be valid. You will see more advantages in the next slides.

Actually, it won't be the algebraic version until we get rid of almost all of the numbers. We will permit the number 0 at times, plus numbers for the indices. The above formulation is not yet the algebraic formulation. We next make the remaining constraints more algebraic.



Making the remaining constraint more algebraic

Let a_j be the audience size of the j-th ad type, which is the coefficient of x_j in the constraint. And let b denote the required number of people reached by the ads. We then can rewrite the constraint.



$$\min \quad 500\, x_1 + 200\, x_2 + 250\, x_3 + 125\, x_4$$

$$\text{s.t.} \quad a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_4 \geq b$$

$$0 \leq x_j \leq d_j \text{ for } j = 1 \text{ to } 4.$$

It doesn't look simpler than the old version. But it would if there were 1000 variables instead of just 4.

It is not yet an algebraic formulation, and we still need to write the objective function in an algebraic form.

Transforming the cost coefficients

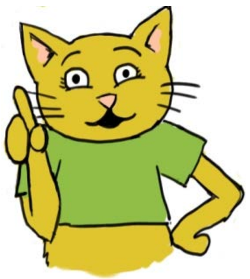
Let c_j be the cost of an ad of type j, which is the cost coefficient of x_j. We now rewrite the objective.

$$\begin{array}{ll} \textbf{min} & c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 x_4 \\ \textbf{s.t.} & a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_4 \geq b \\ & 0 \leq x_j \leq d_j \text{ for } j = 1 \text{ to } 4. \end{array}$$

This is a valid algebraic representation of the problem if we know that there are exactly four variables. But we can carry it a step further.

Using Summation Notation

Next we use summation notation and rewrite the LP formulation as follows:



min $\sum_{j=1}^{4} c_j x_j$

s.t. $\sum_{j=1}^{4} a_j x_j \geq b$

$0 \leq x_j \leq d_j \quad \text{for } j = 1 \text{ to } 4.$

Replacing the number of variables.

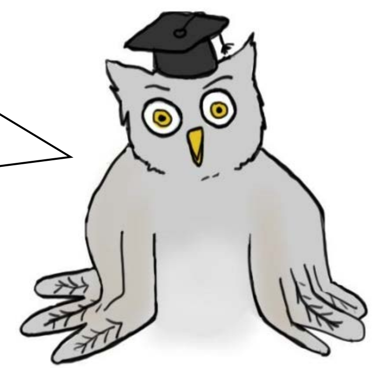
Minimize $$\sum_{j=1}^{n} c_j x_j$$

subject to $$\sum_{j=1}^{n} a_j x_j \geq b$$

$$0 \leq x_j \leq d_j \quad \text{for } j = 1 \text{ to } n.$$

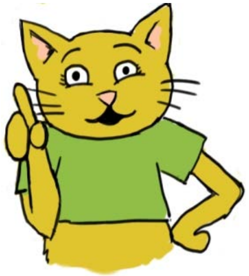
Next Finally, we use n to represent the number of variables

Yes. I know it looks much more abstract than the original formulation. But the abstraction means that this formulation is correct for many different possible choices of the data.



Summary of the transformation

- Let x_j be the number of ads purchased of type j for j = 1 to n.
- Let a_j be the number of persons who view one ad of type j
- Let b be the required number of viewers to see the ads. (That is, the total number of viewers must be at least b)
- Let d_j be an upper bound on the number of ads purchased of type j.



Minimize $\sum_{j=1}^{n} c_j x_j$

subject to $\sum_{j=1}^{n} a_j x_j \geq b$

$0 \leq x_j \leq d_j$ for $j = 1$ to n.

Minimize $500\ x_1 + 200\ x_2 + 250\ x_3 + 125\ x_4$

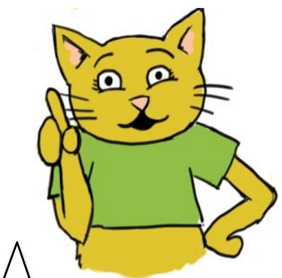
subject to $50{,}000\ x_1 + 25{,}000\ x_2 + 20{,}000\ x_3 + 15{,}000\ x_4 \geq 1{,}500{,}000$

$0 \leq x_1 \leq 20$; $0 \leq x_2 \leq 15$; $0 \leq x_3 \leq 10$; $0 \leq x_4 \leq 15$;

On the reason for algebraic formulations

- Remember that the advantage of algebraic formulations is in their ability to describe very large problems in a very compact manner. This is critical if one is to model large problems, involving thousands or perhaps millions of variables.
- For small problems, it seems unnecessarily cumbersome and difficult.

The notation is also very useful when we describe the simplex algorithm for linear programs.



We next formulate the problem for DTC, David's Tool Company

Formulation of the DTC Problem (David’s Tool Company)

We will write the linear program as if up the constraints are broken into two parts, the demand constraints and the resource constraints.

Maximize $z = 3K + 5S$

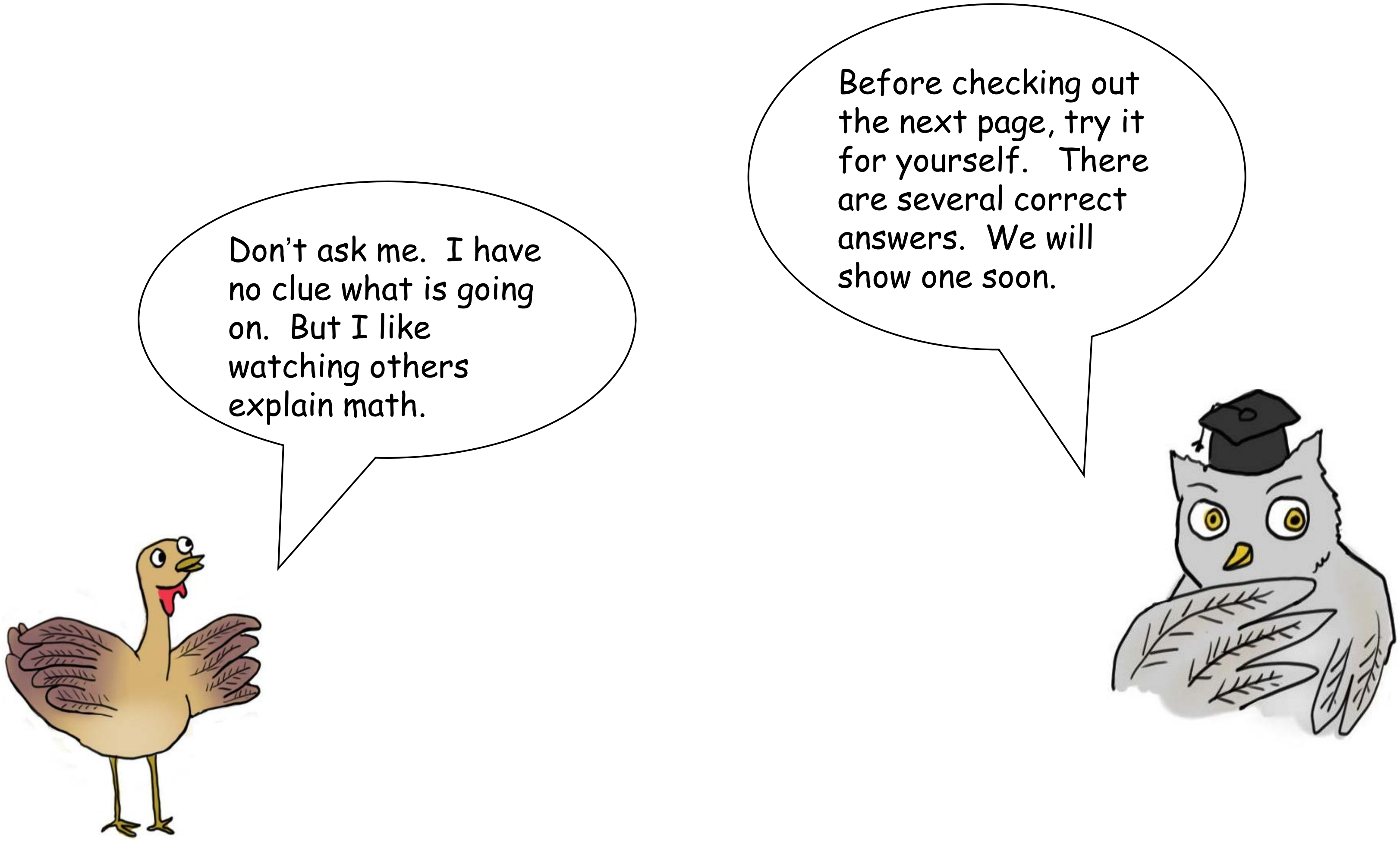
subject to

$2K + 3S$ δ 100	**Gathering time:**	Resource Constraints
$K + 2S$ δ 60	**Smoothing time:**	
$K + S$ δ 50	**Delivery time:**	

K δ 40	**Slingshot demand:**	Bounds on variables.
S δ 30	**Shield demand:**	
$K, S \geq 0$	**Non-negativity:**	

Decision Variables:

- K is number of kits made
- S is number of shields made


Don't ask me. I have no clue what is going on. But I like watching others explain math.
Before checking out the next page, try it for yourself. There are several correct answers. We will show one soon.

The algebraic formulation

Maximize $z = 3x_1 + 5x_2$

subject to

$2K + 3S \;\delta\; 100$	**Gathering time:**	Resource Constraints
$K + 2S \;\delta\; 60$	**Smoothing time:**	
$K + S \;\delta\; 50$	**Delivery time:**	

$K \;\delta\; 40$	**Slingshot demand:**	Bounds on variables.
$S \;\delta\; 30$	**Shield demand:**	
$K, S \geq 0$	**Non-negativity:**	

Let x_j be the number of items of type i that are produced. In the above formulation we have replaced K by x_1 and S by x_2. This will make it more easily described using algebraic notation.

Some hints

For now, we will keep the number of variables as 2. Later on, we will write the formulation so that the number of variables is n. This will be more general.

In linear programming, "n" is often used to represent the number of decision variables. And "m" usually represents the number of constraints (excluding the "≥ 0" constraints).

Also, the variables are often represented by letters near the end of the alphabet such as w, x, y, and z. This convention is not always followed, but it is used a lot.

Resource Constraints

There are three resources: gathering time, smoothing time, and delivery time. We will let the limits (upper bounds) on these three resources be denoted as b_1, b_2, and b_3.

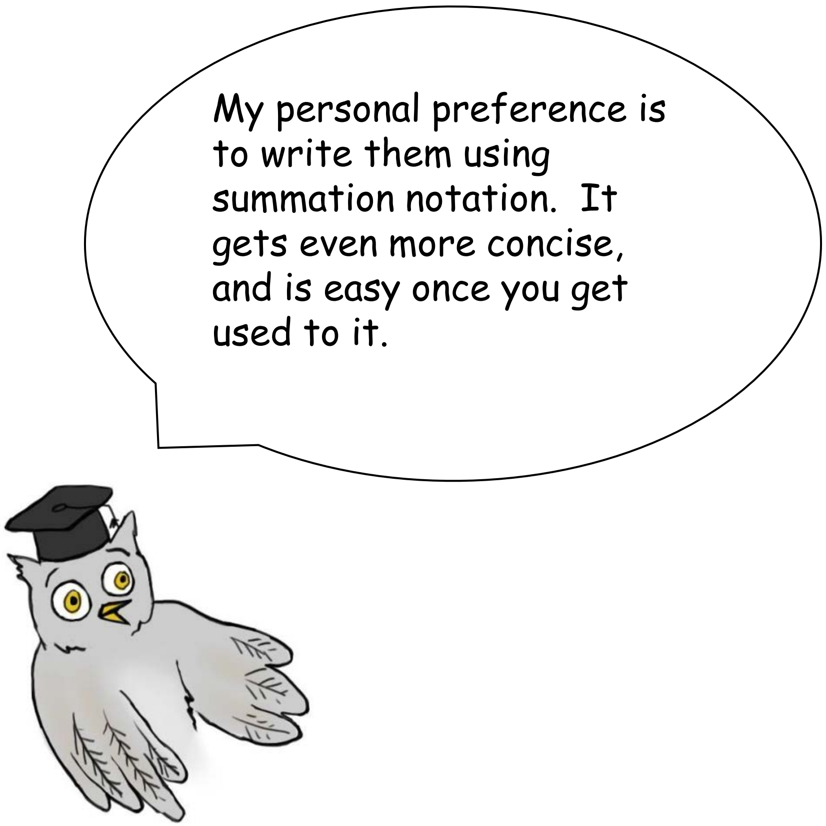
We let a_{i1} be the amount of resource i used in the making of one Kit. We let a_{i2} be the amount of resource i used in making one shield.

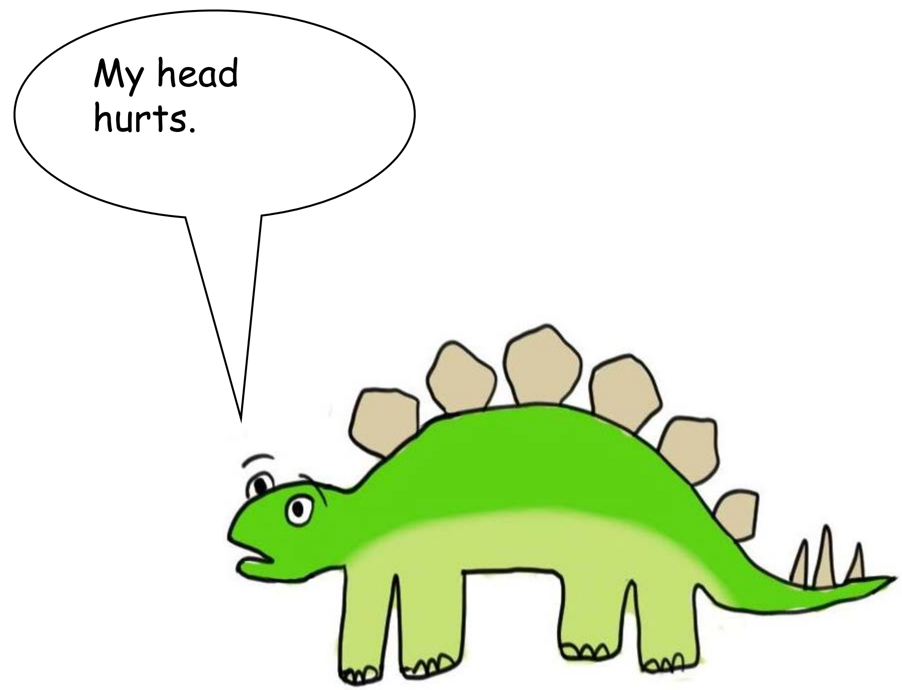
subject to

$a_{11} x_1 + a_{12} x_2 \leq b_1$	**Gathering time:**	Resource Constraints
$a_{21} x_1 + a_{22} x_2 \leq b_2$	**Smoothing time:**	
$a_{31} x_1 + a_{32} x_2 \leq b_3$	**Delivery time:**	

Resource Constraints

$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$ for $i = 1$ to m.	**Gathering time:** **Smoothing time:** **Delivery time:**	Resource Constraints





The complete algebraic formulation

Maximize $z = \sum_{j=1}^{n} p_j x_j$

subject to

$\sum_{j=1}^{n} a_{ij} x_j \le b_i$ for i = 1 to m.	**Gathering time:** **Smoothing time:** **Delivery time:**	Resource Constraints

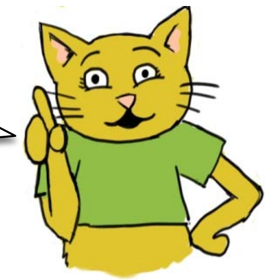
$0 \le x_j \le d_j$ for j = 1 to n.	**Slingshot demand:** **Shield demand:** **Non-negativity:**	Bounds on variables.

In this formulation:

d_j : an upper bound on the demand for item j.
n : the number of items.
a_{ij} : the amount of resource i used up by one unit of item j.
m : the number of different resources.
p_j : the profit from making one unit of item j

Another Practice Example

The following example called "Charging a Blast Furnace" is from Section 1.3 of Applied Mathematical Programming.



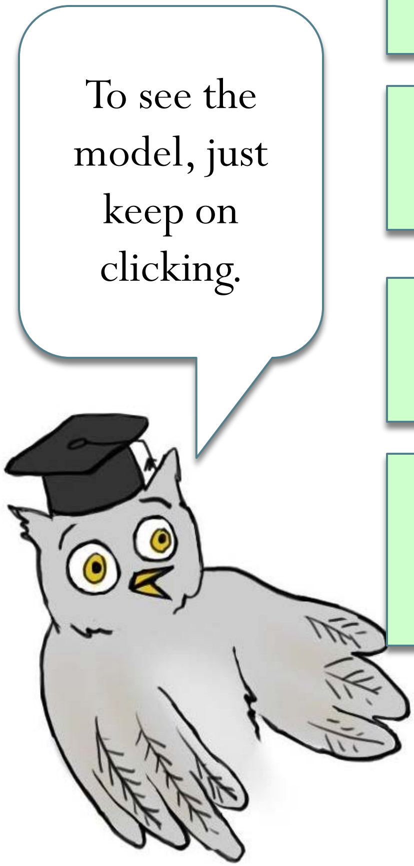
An iron foundry has a firm order to produce 1000 pounds of castings containing at least 0.45 percent manganese and between 3.25 percent and 5.50 percent silicon. As these particular castings are a special order, there are no suitable castings on hand. The castings sell for $0.45 per pound. The foundry has three types of pig iron available in essentially unlimited amounts, with the following properties:

Type of pig iron	A	B	C
Silicon	4 %	1 %	0.6%
Manganese	0.45%	0.5%	0.4%
Costs per 1000/lb.	$21	$25	$15

- Pure manganese can be purchased at $8/pound.
- The cost of melting pig iron is .5 cents per pound.
- Let x_A, x_B, x_C denote the amount of pig iron A, B, and C used.
- Let x_M be the amount of manganese used.

Before going to the next slide, try to formulate the linear program.

The LP formulation



Objective

Weight constraint

Manganese constraint

Silicon constraints

The constraints that must always be remembered

An Algebraic Version

Let's consider an algebraic version of the example.

An iron foundry has a firm order to produce P pounds of castings containing at least b_j pounds of material j and at most u_j pounds of the material j for j = 1 to m. The castings sell for \$d per pound. The foundry has n types of pig iron available in essentially unlimited amounts, with the following properties: Pig iron i costs c_i dollars per pound and the percentage of material j in the iron is a_{ij}. In addition, the firm can purchase material j in its pure form for m_i dollars per bound. The cost of melting pig iron is \$p per pound regardless of the type of pig iron.

Before going to the next slide, try to formulate the linear program.

- Let x_i be the amount of pig iron i used in the mixture.
- Let M_j be the amount of pure material j used that is purchased and used in the mixture

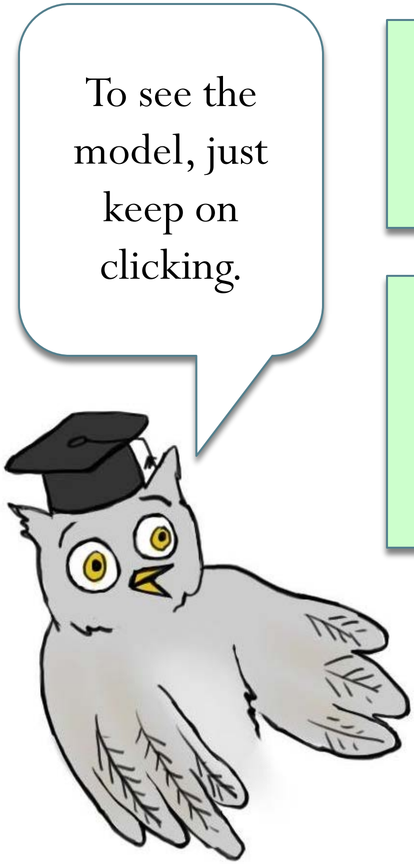
The LP formulation

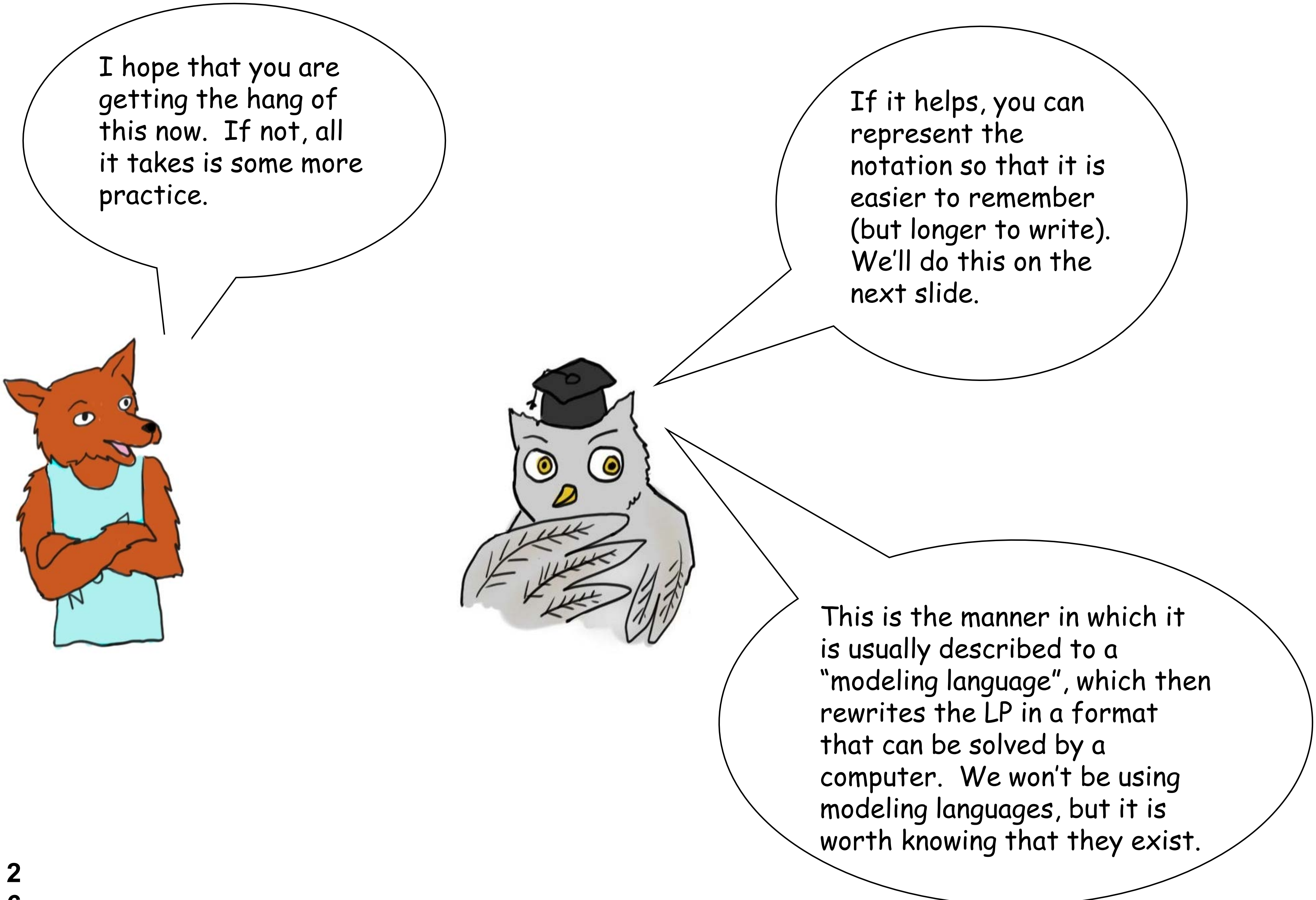
Objective

Weight constraint

Material constraints

The constraints that must always be remembered




I hope that you are getting the hang of this now. If not, all it takes is some more practice.
If it helps, you can represent the notation so that it is easier to remember (but longer to write). We'll do this on the next slide.
This is the manner in which it is usually described to a "modeling language", which then rewrites the LP in a format that can be solved by a computer. We won't be using modeling languages, but it is worth knowing that they exist.

SETS:

IRONS: Set of pig irons

MATERIALS: Set of materials

VARIABLES:

IronUsed(j): amount of iron i used, for j ∈ IRONS.

Purchased(i) : = amount of material j purchased, for i ∈ MATERIALS

PARAMETERS (Data)

CastingRevenue: The price per pound for selling castings

PigIronCost(j): Cost per pound of pig iron j for j ∈ IRONS

MaterialCost(i): Cost per pound of material i for i ∈ MATERIALS

MeltingCost = Cost per pound of melting any of the pig irons

TotalCastings = number of pounds of castings to be sold Materials_Per_Iron(i, j) = The amount of material i in pig iron j for i ∈ MATERIALS and j ∈ IRONS.

LowerLimit(i): The minimum fraction of Material i needed in the mixture

UpperLimit(i): The maximum fraction of material i allowedin the mixture

The LP formulation, for the last time

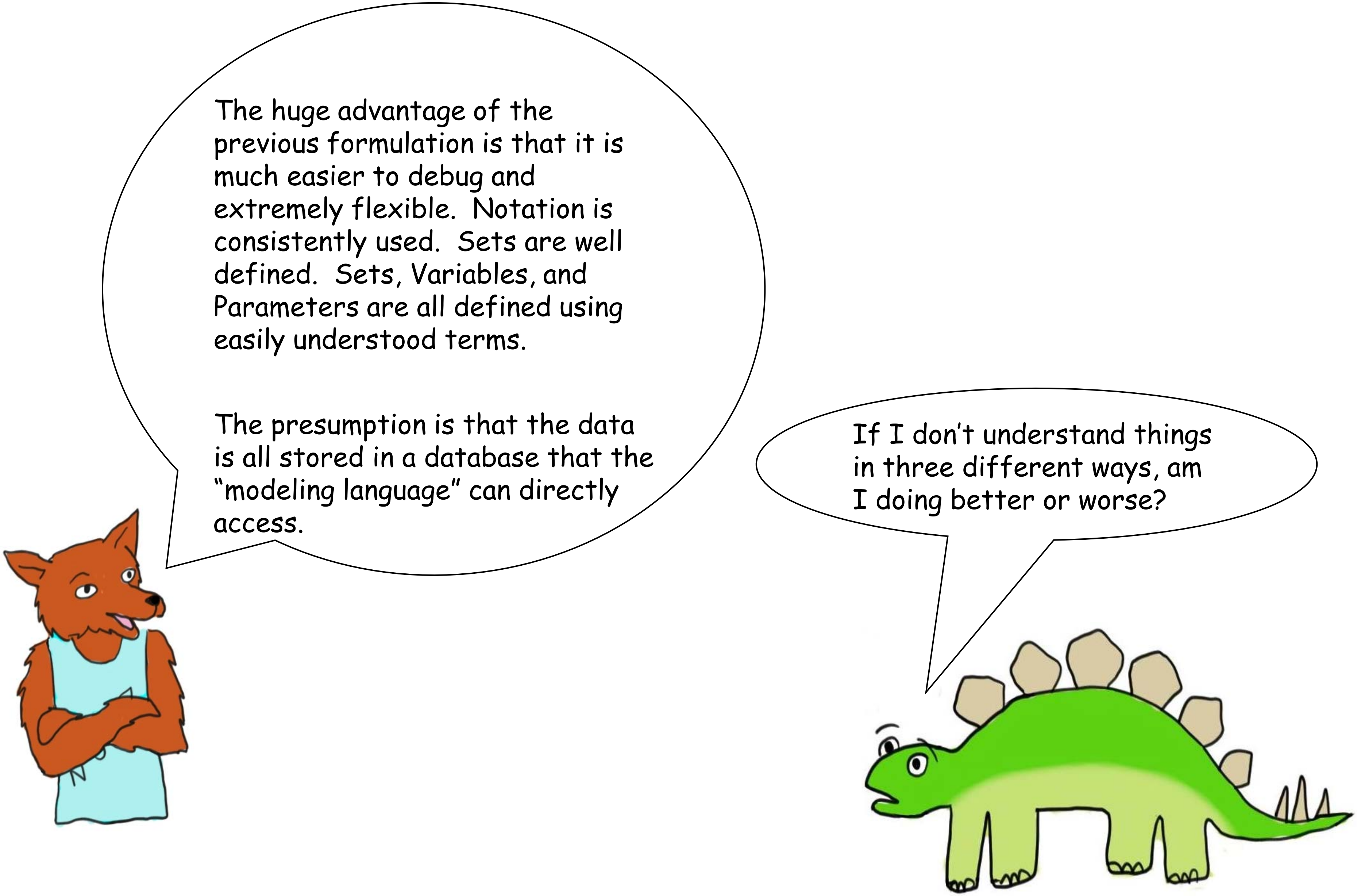
Objective

Weight constraint

Material constraints

The constraints that must always be remembered

28


The huge advantage of the previous formulation is that it is much easier to debug and extremely flexible. Notation is consistently used. Sets are well defined. Sets, Variables, and Parameters are all defined using easily understood terms.
The presumption is that the data is all stored in a database that the "modeling language" can directly access.
If I don't understand things in three different ways, am I doing better or worse?

Notation for linear programs in standard form

- Finally, we show some conventions that are used in describing a linear programming in standard form. The conventions are used in 15.053.

- There are usually n variables and m equality constraints
- The variables are usually $x_1, \ldots, x_n$.
- The cost coefficients are usually $c_1, \ldots, c_n$. (Objective function coefficients are often called cost coefficients even if one is maximizing profit. It is widely agreed that this is a weird convention, but it is commonly done in any case.)
- The coefficient for x_j in constraint i is a_{ij}. The RHS is b_i.
- Then the LP in "standard form" can be written as follows:

$$\text{maximize} \quad z = \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{x} a_{ij} x_j = b_i \qquad \text{for } i = 1 \text{ to } m$$

$$x_j \geq 0 \qquad \text{for } j = 1 \text{ to } n$$

Last slide

In case you were wondering, there are different ways of writing algebraic formulations. You can choose notation differently, and you can combine groups of constraints differently.

You will have a chance to practice algebraic formulations on the homework sets.

And that's the end of this tutorial. I hope it was of value to you. Bye!

MIT OpenCourseWare
http://ocw.mit.edu

15.053 Optimization Methods in Management Science
Spring 2013

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.