Tuesday, Sept 6

1. Welcome Back!
2. Printing Notes
3. Questions?
4. LPs and their solutions.
5. Solvers
6. Outro
   - ↳ Small Work: look into solvers!
   - ↳ Keep an eye out for HW 1

LaTex workshop
9/8 11:45 - 12:45
OLRI 250

Last time we met, we talked about how optimization problems have a two parts, an objective function and a set of constraints. We did quite a bit of formulation as well. Let's go back to this as a sort of warm up.

Example: A farmer has 12 acres of land to plant either soybeans or corn. At least 7 acres have to be planted. Planting one acre of soybeans costs \$200 and one acre of corn costs \$100. Budget for planting is \$1500. The sale from one acre of soybeans is \$500 and from corn is \$300. How many acres of what should be planted to maximize profit?

We're going to have two variables s and c.
a.) What's our objective? Write a function for it.

$300s + 200c$

b.) What are our constraints? Write functions for them as well.

$200s + 100c \le 1500$
$7 \le s + c$
$s + c \le 12$

Problems that can be phrased like the one above are *linear programs*, and throughout our discussion of optimization, we'll see them a lot. To that end, let's get a solid definition:

Definition. *(affine, linear) A function $f : \mathbb{R}^n \to \mathbb{R}$ is an affine function if $f(x) = a^T x + \beta$. Where x, a are vectors in $\mathbb{R}^n$ and $\beta \in \mathbb{R}$. If $\beta = 0$, then f is a linear function.*

Note: lines are affine, through origin = linear.

Recall: take a minute to rewrite this definition in a more familiar manner (ie, without a transpose) and remind yourself of the three properties of linear functions.

$a = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$ $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ $f(x) = a_1x_1 + a_2x_2 + \cdots + a_nx_n + \beta$

$f(x+y) = f(x) + f(y)$
$f(cx) = cf(x)$
$f(0) = 0$

Definition. *(linear constraint) a linear constraint is an equation (or inequality) of the form $f(x) \le \beta$, $f(x) \ge \beta$ or $f(x) = \beta$, where $f(x)$ is a linear function and β is some real number.*

Note: no strict inequalities allowed!

Recall: We talked a bit about this last time, but what's a convenient way to write out a set of linear constraints?

matrix notation, provided all constraints match

Definition. *(linear program, LP) a linear program is defined by an affine objective function and a finite set of linear constraints. We'll often abbreviate this as an LP. Notationally, we will write:*

$$(LP)\begin{cases} \max & c^T x + d \\ \text{s.t.} & Ax \le b \\ & x \ge 0 \end{cases} \qquad \textit{or} \qquad (LP)\begin{cases} \min & c^T x + d \\ \text{s.t.} & Ax \ge b \\ & x \ge 0 \end{cases}$$

Example: Which of the following is a linear program? Explain why and why the others are not. For the linear program, write it with the notation above.

$$\begin{cases} \max & x_1 + x_3 + x_4 & \\ & x_1 + x_3 & \le 3 \\ \text{s. t.} & x_2 + x_4 & \ge 2 \\ & x_1 + x_2 & \le 4 \\ & x_i & \ge 0 \end{cases} \qquad \begin{cases} \min & x_2 + x_3 & \\ & x_1 & < 4 \\ \text{s. t.} & x_2 + x_4 & \ge 2 \\ & x_i & \ge 0 \end{cases} \qquad \begin{cases} \max & & x_1 x_3 \\ & x_3 + x_4 & \le 3 \\ \text{s. t.} & x_1 + x_2 & \ge 2 \\ & x_i & \ge 0 \end{cases}$$

$$\begin{cases} \max & c^T x \\ \text{st} & Ax \le b \\ & x \ge 0 \end{cases} \qquad c: \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad A: \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad b: \begin{bmatrix} 3 \\ -2 \\ 4 \end{bmatrix}$$

Example: State fair vendor Kebabs and Cobs makes two products: bell pepper kebabs and roasted corn on the cob. They make 4 dollars from every kebab, and 5 dollars for every cob. Kebabs take 2 minutes to make, and cobs take 3 minutes to roast.

They just got a group of 4H kids with the following requirements: they need at least 25 snacks, at least 5 of each type, and they have 1 hour.

Write a linear program for Kebabs and Cobs so that they maximize profits.

$$\begin{cases} \max & 4k + 5c \\ \text{s.t.} & k \ge 5 \\ & c \ge 5 \\ & k + c \ge 25 \\ & 2k + 3c \le 60 \end{cases}$$

Before we start talking about solving this problem, let's talk about 4 assumptions that linear programs make:

linear functions

- Proportionality: the contribution of any variable to the objective function is proportional to its value

  ↳ kebabs always profit \$4 and take 2 min.

  ↳ no discounts for bulk?

- Additivity: no interactions between decision variables.

  ↳ nothing multiplied

- Divisibility: we can take on fractional values for our variables

  ↳ variables can be real #

  ↳ problem with discrete objects!

- Correctness: we have all the appropriate terms for our program

  ↳ can tackle w/ sensitivity analysis.

Example: In penalty kicks in soccer, the kicker kicks the ball and usually tries to aim at one of the top corners of the goal. The goalie tries to guess which corner the kicker kicks and jumps towards one of the corners. Assume you are the kicker and you know that the goalie has a handicap that if you shoot to the left and the goalie jumps left, there is only 10% chance for you to score, but if you kick to the right and the goalie jumps to the right, there is 50% chance of scoring. If the goalie jumps in the opposite direction than yourkick, you have 95% chance of scoring.

Should you kick the ball to the left or to the right? If you always kick to the right, the goalie will always jump to the right and you score 0.5 goals per kick. It is better to pick left or right with some probability. What is the best left-right probability subject to the goalie picking his random jumps to counter your strategy as much as possible?

variables:
goals scored : s
goals shot L: l
goals shot R: r

$$\begin{cases} \max \quad s \\ \text{st} \quad 0.1l + 0.95r \geq s \\ \quad\quad 0.95l + 0.5r \geq s \\ \quad\quad l + r = 1 \\ \quad\quad l, r \geq 0 \end{cases}$$

kicked \ goalie	L	R
L	0.1	0.95
R	0.95	0.5

$s \approx 0.6557$

$l \approx 0.346 \quad r \approx .65$

So this is all well and good, but how do we *solve* these things? This is a big question, and it's going to get into the geometry of the spaces, and feasible regions and algorithms for efficient searches for solutions. Those will be our topics in the next few days.

For now, let's as a computer to do it for us. There are a lot of options for us on this, and you're more than welcome to explore alternate solvers, but I'm going to focus on 2: APMonitor and SAGE. We'll look at both with respect to our warm up question.

Code: APMonitor

```
Model farmer
   Variables
      soy = 0, >= 0          } initialize and constrain your variables.
      corn = 0, >= 0
End Variables
Equations
st maximize 500*soy + 300*corn - 200*soy - 100*corn      objective
      200*soy + 100*corn <= 1500     }
      soy + corn <= 12               } constraints.
      soy+corn >= 7                  }
   End Equations
End Model
```

Notice: AP Monitor has been giving me some weirdness with negative objective values that don't make sense... I'll look into it, but please be careful and double check for "realness". The nice part of AP Monitor is that it's online, and it can be added to other languages if you're interested.

Code: SAGE

```
p = MixedIntegerLinearProgram(maximization=True)] create an LP object
x = p.new_variable(nonnegative=True) add var.
p.set_objective( 500*x[0] + 300*x[1] - 200*x[0] - 100*x[1])] set.obj
p.add_constraint( 200*x[0] + 100*x[1] <= 1500 )  }
p.add_constraint( x[0] + x[1] <= 12 )            } add constraints
p.add_constraint( x[0] + x[1] >= 7 )             }
print("Profit $",p.solve()) -> find max profit
print("Soybeans",p.get_values(x[0]),"acres", Corn",p.get_values(x[1]),"acres")
                                    get where max happens
```

Notice: I love SAGE, and it's free to download and use. It also exists on the MCSS lab image. This is how I'll be doing LPs in class. If you want an online equivalent, cocalc is available, but it's often VERY slow.

Example: Try to work out a solution to our State Fair and soccer problems using a solver of your choice.

Hey wait... do you notice something weird about the state fair problem? What's wrong with it?

Let's dig into that weirdness with one more example.

Example: We are producing packages of two 15cm ropes and one 20cm rope (say for some kid's game). Suppose we have have four hundred 50cm ropes and one hundred 65cm ropes. How should we cut the ropes to maximize the number of produced packages?

Formulate the linear program and use a solver to complete the problem.

<u>variable</u>

p: packages

s: short

l: long.

max p

st (its hard) (no coffee...)

$p \leq l$

$p \leq \frac{1}{2} s$

$s = 2x_1 + 3x_2 + 4x_5 + x_6 + 3x_7$

$l = x_1 + 2x_3 + 3x_4 + 2x_6 + x_7$

$400 \geq x_1 + x_2 + x_3$

$100 \geq x_4 + x_5 + x_6 + x_7$

schemes

50: 15+15+20 x_1
15+15+15 x_2
20+20 x_3

65: 20+20+20 x_4
15+15+15+15 x_5
20+20+15 x_6
15+15+15+20 x_7

Clearly an LP is not the right tool for this problem. It misses the fact that...

enter, the *integer program!*

Definition. *(integer program, IP) An integer program, or IP, is a linear program where one or more variables is required to be integral. If not all variables are required to be integers, we sometimes call it a mixed integer program.*

It turns out these are much harder to solve, but they are often what we need when we're talking about, say, edges of a graph or making goods or job assignment. We'll be talking about them more at the end of our LPs unit, but be sure to consider if you need to add an integer constraint!