CSCI 3100

Linear Programming
Today:
- No class in 1 week
- Reading assignment
- HW due by end of day on Wednesday
In a linear program, we are given a set of variables. The goal is to give these real values so that:

1. We satisfy some set of linear equations or inequalities.

2. We maximize or minimize some linear objective function.
An example: Maximize profit

A chocolate shop produces 2 products
- Type 1, worth $1 each
- Type 2, worth $6 each

Constraints:
- Can only produce 200 of type 1 per day
- And at most 300 of type 2
- Total output per day of both is ≤ 400

LP:

maximize: \( X_1 + 6X_2 \)

s.t. \( X_1 \leq 200 \)
\( X_2 \leq 300 \)
\( X_1 + X_2 \leq 400 \)
\( X_1, X_2 \geq 0 \)
LP:
These go up in dimension new chocolate!

Maximize \( x_1 + 6x_2 + 13x_3 \)

s.t.

1. \( x_1 \leq 200 \)
2. \( x_2 \leq 300 \)
3. \( x_1 + x_2 + x_3 \leq 400 \)
4. \( x_2 + 3x_3 \leq 600 \)
5. \( x_1 \geq 0 \)
6. \( x_2 \geq 0 \)
7. \( x_3 \geq 0 \)
Another (more general)

in foods, m nutrients

Let \( a_{i,j} \) = amount of nutrient \( i \) in food \( j \)

\( r_i \) = requirement of nutrient \( i \)

\( x_j \) = amount of food \( j \) purchased

\( c_j \) = cost of food \( j \)

Goal: Buy food so you satisfy nutrients while minimizing cost.

\[
\min \sum_{j=1}^{n} c_j x_j \\
\sum_{j=1}^{n} a_{i,j} x_j \geq r_i \quad (i=1, \ldots, m)
\]

\[
A \vec{x} \geq \vec{r}
\]
The LP

\[
\begin{align*}
\min \quad & \sum_{i} x_i \\
\text{subject to} \quad & A x \geq r \\
& a_{ij_1} x_1 + a_{ij_2} x_2 + \cdots + a_{ijn} x_n \geq r_j \\
& \vdots \\
& \vdots
\end{align*}
\]
In general, get systems like this:

\[
\begin{align*}
\text{maximize} & \quad \sum_{j=1}^{d} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{d} a_{ij} x_j \leq b_i \quad \text{for each } i = 1..p \\
& \quad \sum_{j=1}^{d} a_{ij} x_j = b_i \quad \text{for each } i = p+1..p+q \\
& \quad \sum_{j=1}^{d} a_{ij} x_j \geq b_i \quad \text{for each } i = p+q+1..n
\end{align*}
\]

Geometric picture:

A two-dimensional polyhedron (white) defined by 10 linear inequalities.
Canonical Form:

Avoid having both $\leq$ and $\geq$.

So get something more like our first example:

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

subject to $\sum_{j=1}^{d} a_{ij} x_j \leq b_i$ for each $i = 1..n$

$x_j \geq 0$ for each $j = 1..d$

Or, given a vector $\vec{c}$ and matrix $A$:

maximize $\vec{c} \cdot \vec{x}$

s.t. $A\vec{x} \leq \vec{b}$
Anything can be put into canonical forms.

1. Avoid $= \text{eqn}: \sum a_i x_i = b$

   $c \Rightarrow \sum a_i x_i \leq b$

   $+ \sum a_i x_i \geq b$

2. Avoid $\geq$

   \[
   \left[ \sum a_i x_i \geq b \right] - 1
   \]

   $- \sum a_i x_i \leq -b$

   $\Rightarrow \quad Ax \leq b$
How could these not have a solution?

2 ways:

Feasible

or

Unbounded
Better pictures (still 2d):

maximize $x - y$
subject to $2x + y \leq 1$
$x + y \geq 2$
$x, y \geq 0$
Note:

1. Multiplying by -1 turns any maximization problem into a minimization one:

\[ \text{max } \mathbf{c} \mathbf{x} \]
\[ \text{s.t. } \mathbf{A} \mathbf{x} \leq \mathbf{b} \]
\[ \text{⇒ } \min \ -\mathbf{c} \mathbf{x} \]
\[ \text{s.t. } -\mathbf{A} \mathbf{x} = -\mathbf{b} \]

2. Can turn inequalities into equalities via slack variables:

\[ \sum_{i=0}^{n} a_i x_i \leq b \]

\[ \downarrow \]

\[ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n + s = b \]
\[ s \geq 0 \]
(3) Can change equalities into inequalities, also!

$$\sum_{i=1}^{n} a_i x_i \geq b$$

(already saw)
Solving LP's:
The simplex algorithm

“ball dropping”

Start on boundary (some linear constraint)
If not best possible, improve by moving along this constraint

This will stop at optimal solution