• DISCLAIMER: These notes are not necessarily an accurate representation of what I said during the class. They are mostly what I intend to say, and have not been carefully edited.

• Main topics:
  – Return Test 3
  – Linear programming: what it is, and how to formulate it.

• Test 1:
  – Question 1: minimize maximum, instead of total.
  – Partial points on question 4.

• Office hour changes this week:
  – Ashwin’s office hours this Wednesday are moved to 10-11am.
  – Richard’s office hours this week are moved to Wednesday 4-6pm (instead of Thursday).

The main idea of linear programming is to formulate a problem as a set of linear inequalities, and invoke a high powered algorithm (whose details we won’t go into) to solve. There are two main primitives in linear programs:

1. variables $x_1 \ldots x_n$.

2. linear inequalities of the form

$$\sum_j a_{ij}x_j \leq b_j.$$

The goal is to minimize/maximize some objective subject to the constraints. The objective is also linear,

$$\sum_j c_jx_j.$$
Here the $i$ in $a_{ij}$ indexes into constraints, whose number we will denote with $m$. A linear program in standard form is formally given $a_{ij}$ for $1 \leq i \leq m, 1 \leq j \leq n$, and costs $c_j$ for $1 \leq j \leq n$, and solve for $x_1 \ldots x_n$ that:

$$\begin{align*}
\text{maximize} & \quad \sum_j c_j x_j \\
\text{subject to} & \quad \sum_j a_{ij} x_j \leq b_i \quad \forall 1 \leq i \leq m
\end{align*}$$

The number of dimensions, $n$, is often referred to as the dimension of the linear program. This analogy is fairly direct: in two dimensions, the pair of variables $(x_1, x_2)$ correspond precisely to points on the 2-D plane.

It can also be checked that constraints of the form

$$x_1 + x_2 \leq 1$$

are precisely the points above/below lines on the plane. In particular, the set of constraints

$$\begin{align*}
x_1 & \geq 0 \\
x_2 & \geq 0 \\
x_1 + x_2 & \leq 1
\end{align*}$$

defines a triangle near the origin.

The reason that these spaces form lines is that if $x^{(1)}$ and $x^{(2)}$ are both on the boundary, aka

$$\sum_{j=1}^n a_{ij} x^{(1)}_j = \sum_{j=1}^n a_{ij} x^{(2)}_j = b,$$

then any linear combination of them $\theta x^{(1)} + (1 - \theta) x^{(2)}$ also satisfies

$$\sum_j a_{ij} \left( \theta x^{(1)} + (1 - \theta) x^{(2)} \right) = b_i$$

The reason constraints are useful is that they allow us to express a variety of objectives. Suppose there is a time budget of 3 hours, each of which can be spent on doing 2 units of task $A$ plus 1 unit of task $B$, or 1 unit of task $A$ and plus 3 units of task $B$. And the goal is to maximize the amount of task $B$ done. These constraints can be captured using the following linear program:

$$\begin{align*}
\text{max} & \quad x_2 \\
\text{subject to} & \quad x_1 + x_2 \leq 3 \quad \text{(total time)} \\
& \quad 2x_1 + x_2 \geq 4 \quad \text{task } A \\
& \quad x_1 + 3x_2 \geq 4 \quad \text{task } B \\
& \quad x_1 \geq 0 \\
& \quad x_2 \geq 0
\end{align*}$$
These constraints can be interpreted as half spaces: \( x_1 + x_2 = 3 \) is a line, so \( x + y \leq 3 \) is the set of points below this line. Incorporating in \( 2x_1 + x_2 \geq 4 \) and \( x_1 + 3x_2 \geq 4 \) leads the following region of the 2-D plane in Figure 1.

![Figure 1: Feasible Solutions of the scheduling linear program above](image)

A major reason that linear inequalities are a core primitive in optimization is that they can capture a variety of other objectives, and there are very good algorithms (e.g. CVX, CPLEX, coordinate/gradient descent) for optimizing over them. Many constraints/objectives that don’t immediately look linear by introducing variables / more inequalities.

1. Linear equality: the condition

   \[
   \sum_j a_{ij}x_j = b_j,
   \]

   can be expressed as:

   \[
   \sum_j a_{ij}x_j \leq b_j \quad \text{(6)}
   \]

   \[
   \sum_j a_{ij}x_j \geq b_j \quad \text{(7)}
   \]

2. Absolute value: the condition of \( |x_i - x_j| \leq b_i \) can be formulated as:

   \[
   x_i - x_j \leq b_i \quad \text{(9)}
   \]

   \[
   -x_i + x_j \leq b_i \quad \text{(10)}
   \]
3. Objectives of the form

\[
\text{minimize} \quad |x_1 - 1|,
\]

can be done by introducing another variable \( y_1 \), changing the objective to

\[
\text{minimize} \quad y_1,
\]

and adding the constraints

\[
\begin{align*}
x_1 & \geq 1 - y_1 \\
x_1 & \leq 1 + y_1
\end{align*}
\]