Last time we saw linear programs
- definition and examples.

This time: algorithms and basic theory.

Recap: LP has (a) linear objective function (max/min)
(b) linear inequality constraints

E.g.: \[ \text{max } 5x_1 + x_2 \]
\[ x_1 + x_2 \leq 7 \]
\[ -2x_1 + 3x_2 \geq -5 \]
\[ x_1, x_2 \geq 0 \]

Feasible region is green
maximizer is one of the "corners"

In this case \((2\frac{6}{5}, 9\frac{1}{5})\) (by enumerating over all the corners)
Today's Menu (in more detail)

- Sketch of why the optimal solution of an LP is at the corners.

- Simplex Alg

- If time permits, idea of other algorithms: \[ \text{Ellipsoid} \quad \text{Interior Point} \]

---

Also notation:

General form for LP:

\[
\begin{align*}
\text{max } & \sum c_i x_i \\
\text{st. } & \langle a_i, x \rangle \geq b_i \\
& \langle a_m, x \rangle \geq b_m
\end{align*}
\]

\[
\begin{align*}
\text{max } & \langle c, x \rangle \\
\Rightarrow & \text{ st. } A x \geq b \quad \text{(component-wise greater than or equals)}
\end{align*}
\]

\[ A \text{ are } a_1, a_2, \ldots, a_m \]

All other LP forms can be converted to this form (see Lec #17 notes)
Warm up:

1-dimensional LP

Maximize $7x$

Subject to:
- $x \geq 2$
- $x \geq 3$
- $x \leq 5$
- $x \leq 6$
- $x \geq -2$
- $x \leq 13$

Where is optimum? At one of the endpoints (corners)

Why? Because value at any point is some
average of value at 2 endpoints (linear)

So look at just 2 endpoints.

Enumerate over both endpoints.

Important to note: boring, but gotta no started
So: did we need to enumerate over all corners?

Not really. Should use the geometry of the problem.

The objective is $5x_1 + x_2$

or $(5, 1)(x_1, x_2) = c^T x$

this vector $c$ is the direction we "want to go in"

Fact: draw this vector, and the line (in general the hyperplane) of points that are orthogonal to it (the orange one)

then all points on that hyperplane have objective value 0.

(by definition, because they all have zero dot product with $c$)

Now if you translate this plane, all points have same obj fn value.

(e.g. the pink ones)

And it increases the more you go in the direction of $c$. 
So: if you want to maximize $c^T x$ over a convex set.

Take this hyperplane $H = \{ x \mid c^T x = 0 \}$

And move it as far as you can in the direction of $c$

until you can't move it any more.

The value of those last points is the optimum

Great: for any LP now we can do this very visual hyperplane (line) shifting operation to find maximum —

or at least to figure out whereabouts to look!

(Not clear how to get efficient algorithm from it, though)
From now onwards.

$$\max c^T x$$

$$A x \geq b \quad \text{defines feasible region}$$

feasible region (convex "polyhedron")
BTW: LP may have unbounded feasible region

\[
\begin{align*}
    x_1 + x_2 &\leq 7 \\
    -2x_1 + 3x_2 &\geq -5 \\
    \text{max } 5x_1 + x_2
\end{align*}
\]

still have finite maximum.

Or:

\[
\begin{align*}
    x_1 &\geq 0, x_2 &\geq 0 \\
    -2x_1 + 3x_2 &\geq -5 \\
    \text{max } 5x_1 + x_2
\end{align*}
\]

may have unbounded objective function

here every point \((x_1, \frac{2x_1 - 5}{3})\) is feasible if \(x_1 \geq \frac{5}{2}\)

and obj fn = \(5x_1 + \frac{2x_1 - 5}{3} \to \infty\) as \(x_1 \to \infty\).

Or:

\[
\begin{align*}
    x_1 + x_2 &\geq 7 \\
    x_1 &\leq 3 \\
    x_2 &\leq 3
\end{align*}
\]

may have no feasible points

so no solution

these are all possible \(\{\text{finite obj value, infinite obj, infeasible LP} \}\)
Same ideas hold for higher dimensional LPs.

Example: \[ x_1 \geq 0, \ x_2 \geq 0, \ x_3 \geq 0 \]
\[
\begin{align*}
  x_1 + x_2 + 2x_3 & \leq 7 \\
  x_1 - 3x_2 & \leq 5 \\
  2x_2 + x_3 & \geq 2
\end{align*}
\]

\[
\max 2x_1 + x_2 + x_3
\]

Gives some 3d picture:

And then we want to go in the direction \((2, 1, 1)\) as far as poss.

Next steps: 1. Lemma that optimum @ corner

2. Algorithm to find optimum (Simplex also).
For now: assume feasible region is bounded (does not go off to infinity)
And I at least one feasible point.

Claim: the optimum always exists and it is achieved at a "corner".

Proof sketch:

1. Feasible region is convex polyhedron (each constraint is a halfspace, convex)

2. Feasible region is closed (remember: no strict inequalities allowed)

Def: corner. a point $x$ is a corner of feasible region if it cannot be written as $x = \lambda x_1 + (1-\lambda) x_2$

for $x_1, x_2$ also feasible $x_1 \neq x_2 \neq x$ and $\lambda \in [0,1]$.

In words, $x$ is not the average of two other points that are feasible.

Not allowed: $x_1 \quad \bar{x} \quad x_2 \quad \lambda = \frac{3}{4} x_1 + \frac{1}{4} x_2$
Take a feasible $x$.

- Now if you can move in the direction of $c$ then your objective improves.

Indeed $x' = x + t \cdot c$

then $c^T x' = c^T x + c^T (t c) = c^T x + t \|c\|^2 \geq c^T x$

- If you can move in any direction that has positive inner product with $c$ i.e. $\langle d, c \rangle > 0$

then you improve. (same argument).

- So now we have situation where for all direction $d$ of $\langle d, c \rangle > 0$ we cannot move in that direction

- then have that $x$ must be on the boundary of feasible region.
Next: Is \( z \) optimal?

Yes. Sps \( \exists y \text{ st } c^T y > c^T x \). Then \( c^T(y-x+z) > c^T x \) 

this is direct and st 

\[ c^T(x+d) > c^T x \]. 

Contradiction.

Is it a corner?

Maybe not. Eg. in picture on previous page (bottom)

But take the set of all points \( z = \frac{1}{\lambda} z : c^T z > c^T x, \) and \( z \) feasible \( \frac{1}{\lambda} \)

then \( c^T x_1, c^T x_2 \) are both optimal, and have

"fewer degrees of freedom". Move to one (say \( x_1 \))

And repeat until at a corner.

This is all hand-wavy, but can be made precise. (see a text).
To recap: LPs. \{ infeasible bounded optimum unbounded opt. \} \rightarrow\text{interesting case.}

Assume: Bounded optimum
then \exists a corner that is an optimal solution.

What next? why do we care?

Because we can get an algorithm from this:

\underline{The Simplex Algorithm}: \quad \leftarrow (\text{local search!})

1. Start at some corner \( x \)
2. At \( x \):
3. Look at the neighboring corners \( x_1, x_2, \ldots \)
4. If \exists a neighbor \( x_i \) s.t. \( \langle c, x_i \rangle > \langle c, x \rangle \), pick one \( x_i \)
5. Set \( x \leftarrow x_i \), go to 2.
6. Else (all neighbors worse than \( x \))
7. Return "\( x \) is optimal"

Very popular algorithm. Not always good in theory, but very good in many practical situations (with good engineering/implementation)
Many questions to be answered.

1. How do you find a corner to start? ← not today.
2. What are neighbors?
3. Which neighbor to choose to go to? ] related issues.
4. How long to terminate?

"Let's do it for 2dim first"

Neighbors: Easy to define geometrically. (esp in 2dim)

Also not not nbr

C = objective function to maximize.
Both neighbors are better. Say choose $x_1$, go to $x_1$.

Only $x_2$ is better. Go to it ($x \leftarrow x_2$).

Both neighbors are worse.

Declare $x = \text{optimum}$!
That's the algorithm in 2dim.

if no neighbor is strictly better, stop and say "x optimum", else move to some better nbr.

Fact: In 2d does not matter which neighbor you choose.

Fact: Algorithm stops in ≤ m steps, where m constraints.

Why? Each time if you strictly improve value then see new corner. And there are ≤ m corners in polygon with ≤ m sides.

Fact: x is indeed optimum @ end.

PF: $c^T x_1 \leq c^T x$
$c^T x_2 \leq c^T x$

$\Rightarrow c^T (\alpha x_1 + (1-\alpha) x_2) \leq c^T x$

$\Rightarrow c^T z \leq c^T x$

but z is an average of x and y as well $\Rightarrow c^T z = c^T (\beta x + (1-\beta) y) = \beta c^T x + (1-\beta) c^T y$
Things get more interesting in higher dimensions

- Neighbors are easy to define

Maximize \[ P = 20x_1 + 10x_2 + 15x_3 \]
Subject to:
\[
\begin{align*}
3x_1 &+ 2x_2 + 5x_3 &\leq 55 \\
2x_1 &+ x_2 + x_3 &\leq 26 \\
x_1 &+ x_2 + 3x_3 &\leq 30 \\
5x_1 &+ 2x_2 + 4x_3 &\leq 57 \\
x_1, x_2, x_3 &\geq 0
\end{align*}
\]

Formally: each corner is the intersection of \( d \) tight constraints.

Why? (Intuition only)

- Intersecting 2 lines gives a point (as long as not parallel).
- Intersecting 2 planes gives a line, and then intersecting it with another plane gives a point.
- Basically intersect \( d \) hyperplanes (which are linearly indep.) each intersection reduces dimension by 1.

So finally get 0-dim object, a point.

https://people.richland.edu/james/ictcm/2006/3dsimplex.html
2 corners neighbors if defined by sets $S_1$ and $S_2$ of $d$ tight constraints each, and $S_1, S_2$ differ in 1 constraint.

But now:

(a) May need to move to a neighbor even if values are equal.

May cause cycling - need to be careful.

(b) Choice of neighbor affects the performance a lot.

Many choices:

- best value neighbor
- steepest edge
- random improving edge
- lexicographically first improving edge

(c) Some other small issues (degeneracy) I am ignoring.

However, ideas remain same. Start from a corner, go to some corner that is improving, and do this until I cannot do this any more.
Summary of today’s lecture

- saw that optimal solution at corner of feasible region 
  (at least when polytope is bounded)

- simplex also walks from corner to corner, improving solution quality.

- problem with simplex: examples where most “pivoting” rules take exponential time. So not provably efficient. Although performs reasonably well on many instances

- can we get provably good algorithms?

  - Yes: Khachiyan 76 showed that the Ellipsoid also works in poly time (but not so good in practice)

  - Karmarkar and others (84-) developed interior point methods that are good in theory and in practice.

Sketch both next time. + Duality discussion.