Lecture 20: Linear Programming

Michael Dinitz

November 4, 2021

601.433/633 Introduction to Algorithms

Introduction

Today: What, why, and juste a taste of how

- Entire course on linear programming over in AMS. Super important topic!
- Fast algorithms in theory and in practice.

Why: Even more general than max-flow, can still be solved in polynomial time!

- Max flow important in its own right, but also because it can be used to solve many other things (max bipartite matching)
- Linear programming: important in its own right, but also even more general than max-flow.
- Can model many, many problems!

Example: Planning Your Week (pre-COVID)

168 hours in a week. How much time to spend:

- Studying (**S**)
- Partying (**P**)
- Everything else (**E**)

Constraints:

- $\mathbf{E \geq 56}$ (at least 8 hours/day sleep, shower, etc.)
- $\mathbf{P + E \geq 70}$ (need to stay sane)
- $\mathbf{S \geq 60}$ (to pass your classes)
- $\mathbf{2S + E - 3P \geq 150}$ (too much partying requires studying or sleep)

Question: Is this possible? Is there a *feasible* solution?

- Yes! $\mathbf{S = 80}$, $\mathbf{P = 20}$, $\mathbf{E = 68}$

Question: Suppose "happiness" is $\mathbf{2P + 3E}$. Can we find a feasible solution maximizing this?

Linear Programming

Input (a "linear program"):

- $\mathbf{n}$ variables $\mathbf{x_1, \ldots, x_n}$ (take values in $\mathbb{R}$)
- $\mathbf{m}$ *non-strict linear inequalities* in these variables (constraints)
  - E.g.: $\mathbf{3x_1 + 4x_2 \leq 6}$, $\quad \mathbf{0 \leq x_1 \leq 3} \quad \mathbf{x_2 - 3x_3 + 2x_7 = 17}$
- Possibly a *linear* objective function
  - $\mathbf{\max 2x_3 - 4x_5}$, $\quad \mathbf{\min \frac{5}{2}x_4 + x_2}$, $\quad \ldots$

Goal:

- Feasibility: Find values for $\mathbf{x}$'s that satisfy all constraints
- Optimization: Find feasible solutions maximizing/minimizing objective function

Both achievable in polynomial time, reasonably fast!

Planning your week as an LP

Variables: $\mathbf{P}, \mathbf{E}, \mathbf{S}$

$$
\begin{aligned}
\textbf{max} \quad & 2P + E \\
\text{subject to} \quad & E \geq 56 \\
& S \geq 60 \\
& 2S + E - 3P \geq 150 \\
& P + E \geq 70 \\
& P + S + E = 168 \\
& P \geq 0 \\
& S \geq 0 \\
& E \geq 0
\end{aligned}
$$

When using an LP to model your problem, need to be sure that *all* aspects of your problem included!

Operations Research-style Example

Four different manufacturing plants for making cars:

	labor	materials	pollution
Plant 1	2	3	15
Plant 2	3	4	10
Plant 3	4	5	9
Plant 4	5	6	7

- Need to produce at least **400** cars at plant 3 (labor agreement)
- Have **3300** total hours of labor, **4000** units of material
- Environmental law: produce at most **12000** pollution
- Make as many cars as possible

OR example as an LP

Four different manufacturing plants for making cars:

	labor	materials	pollution
Plant 1	2	3	15
Plant 2	3	4	10
Plant 3	4	5	9
Plant 4	5	6	7

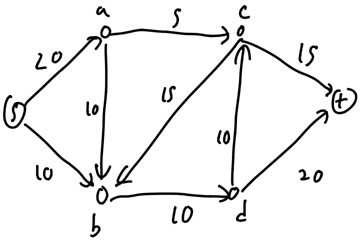
Variables: $\mathbf{x_i}$ = # cars produced at plant **i**, for $\mathbf{i} \in \{1, 2, 3, 4\}$

Objective: $\max x_1 + x_2 + x_3 + x_4$

Constraints:

$$x_3 \geq 400$$
$$2x_1 + 3x_2 + 4x_3 + 5x_4 \leq 3300$$
$$15x_1 + 10x_2 + 9x_3 + 7x_4 \leq 12000$$
$$x_i \geq 0 \qquad \forall i \in \{1, 2, 3, 4\}$$

Max Flow as LP



Variables: $\mathbf{f(e)}$ for all $\mathbf{e \in E}$

Objective: $\max \sum_v f(s,v) - \sum_v f(v,s)$

Constraints:

$$\sum_v f(v,u) - \sum_v f(u,v) = 0 \qquad \forall u \in V \setminus \{s,t\}$$

$$f(e) \le c(e) \qquad \forall e \in E$$

$$f(e) \ge 0 \qquad \forall e \in E$$

So can solve max-flow and min-cut (slower) by using generic LP solver

Multicommodity Flow

Generalization of max-flow with multiple commodities that can't mix, but use up same capacity

Setup:

- Directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$
- Capacities $\mathbf{c} : \mathbf{E} \to \mathbb{R}_{\geq 0}$
- $\mathbf{k}$ source-sink pairs $\{(\mathbf{s_i}, \mathbf{t_i})\}_{\mathbf{i} \in [\mathbf{k}]}$

Goal: send flow of commodity $\mathbf{i}$ from $\mathbf{s_i}$ to $\mathbf{t_i}$, max total flow sent across all commodities

Variables: $\mathbf{f_i(e)}$ for all $\mathbf{e} \in \mathbf{E}$ and for all $\mathbf{i} \in [\mathbf{k}]$. Flow of commodity $\mathbf{i}$ on edge $\mathbf{e}$

Objective: $\max \sum_{i=1}^{k} \left(\sum_v f_i(s_i, v) - \sum_v f_i(v, s_i) \right)$

Constraints:

$$\sum_v f_i(v,u) - \sum_v f_i(u,v) = 0 \qquad \forall i \in [k],\ \forall u \in V \setminus \{s_i, t_i\}$$

$$\sum_{i=1}^{k} f_i(e) \leq c(e) \qquad \forall e \in E$$

$$f_i(e) \geq 0 \qquad \forall e \in E,\ \forall i \in [k]$$

Concurrent Flow

Multicommodity flow, but:

- Also given *demands* $\mathbf{d} : [\mathbf{k}] \to \mathbb{R}_{\geq 0}$
- Question: Is there a multicommodity flow that sends at least $\mathbf{d(i)}$ commodity-$\mathbf{i}$ flow from $\mathbf{s_i}$ to $\mathbf{t_i}$ for all $\mathbf{i} \in [\mathbf{k}]$?

Variables: $\mathbf{f_i(e)}$ for all $\mathbf{e} \in \mathbf{E}$ and for all $\mathbf{i} \in [\mathbf{k}]$.

Constraints:

$$\sum_{v} f_i(v,u) - \sum_{v} f_i(u,v) = 0 \qquad \forall i \in [k],\ \forall u \in V \setminus \{s_i, t_i\}$$

$$\sum_{i=1}^{k} f_i(e) \leq c(e) \qquad \forall e \in E$$

$$f_i(e) \geq 0 \qquad \forall e \in E,\ \forall i \in [k]$$

$$\sum_{v} f_i(s_i,v) - \sum_{v} f_i(v,s_i) \geq d(i) \qquad \forall i \in [k]$$

Maximum Concurrent Flow

If answer is no: how much do we need to scale down demands so that there is a multicommodity flow?

Variables:

- $f_i(e)$ for all $e \in E$ and for all $i \in [k]$.
- λ

Objective: $\max \lambda$

Constraints:

$$\sum_v f_i(v,u) - \sum_v f_i(u,v) = 0 \qquad \forall i \in [k],\ \forall u \in V \setminus \{s_i, t_i\}$$

$$\sum_{i=1}^{k} f_i(e) \le c(e) \qquad \forall e \in E$$

$$f_i(e) \ge 0 \qquad \forall e \in E,\ \forall i \in [k]$$

$$\sum_v f_i(s_i,v) - \sum_v f_i(v,s_i) \ge \lambda d(i) \qquad \forall i \in [k]$$

Shortest $\mathbf{s}-\mathbf{t}$ path

Very surprising LP!

Variables: $\mathbf{d_v}$ for all $\mathbf{v} \in \mathbf{V}$: shortest-path distance from $\mathbf{s}$ to $\mathbf{v}$

$$\begin{aligned} \mathbf{max} \quad & \mathbf{d_t} \\ \text{subject to} \quad & \mathbf{d_s} = \mathbf{0} \\ & \mathbf{d_v} \le \mathbf{d_u} + \ell(\mathbf{u}, \mathbf{v}) \qquad \forall (\mathbf{u}, \mathbf{v}) \in \mathbf{E} \end{aligned}$$

Correctness Theorem: Let $\vec{\mathbf{d}}^*$ denote the optimal LP solution. Then $\mathbf{d_t^*} = \mathbf{d(s,t)}$

Proof Sketch: $\ge$: Let $\mathbf{d_v} = \mathbf{d(s,v)}$ for all $\mathbf{v} \in \mathbf{V}$. Feasible $\implies \mathbf{d_t^*} \ge \mathbf{d_t} = \mathbf{d(s,t)}$.

$\le$:Let $\mathbf{P} = (\mathbf{s} = \mathbf{v_0}, \mathbf{v_1}, \dots, \mathbf{v_k} = \mathbf{t})$ be shortest $\mathbf{s} \to \mathbf{t}$ path.

Prove by induction: $\mathbf{d^*_{v_i}} \le \mathbf{d(s, v_i)}$ for all $\mathbf{i}$

Base case: $\mathbf{i} = \mathbf{0}$ ✓

Inductive step: $\mathbf{d^*_{v_i}} \le \mathbf{d^*_{v_{i-1}}} + \ell(\mathbf{v_{i-1}}, \mathbf{v_i}) \le \mathbf{d(s, v_{i-1})} + \ell(\mathbf{v_{i-1}}, \mathbf{v_i}) = \mathbf{d(s, v_i)}$

Algorithms for LPs

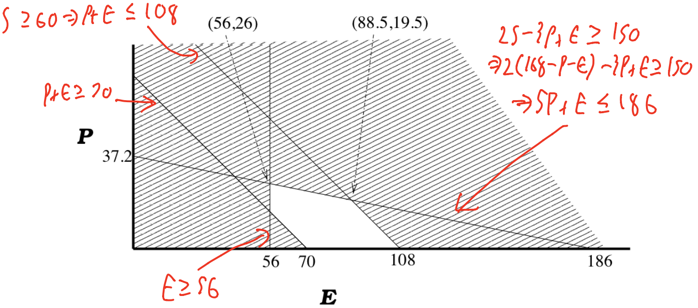
Geometry

To get intuition: think of LPs *geometrically*

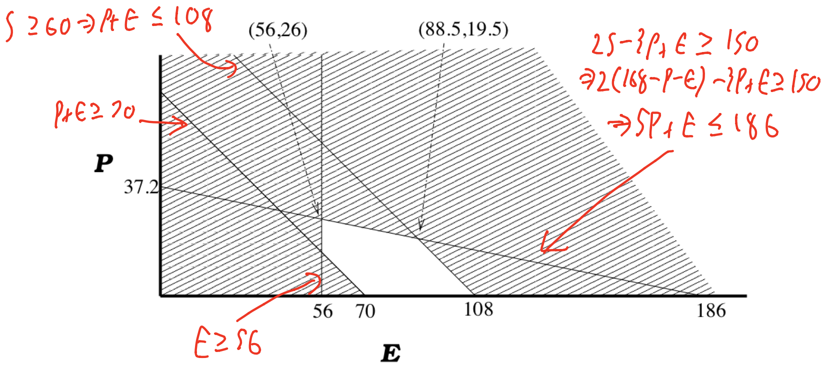
- Space: $\mathbb{R}^n$ (one dimension per variable
- Linear constraint: halfspace (one side of a hyperplane)
- Feasible region: intersection of halfspaces. *Convex Polytope* (usually just called a *polytope*)

Example: planning your week

- $\mathbf{3}$ variables $\mathbf{S}, \mathbf{P}, \mathbf{E}$ so $\mathbb{R}^3$
- But $\mathbf{S} + \mathbf{P} + \mathbf{E} = \mathbf{168} \implies \mathbf{S} = \mathbf{168} - \mathbf{P} - \mathbf{E}$
- Make this substitution, get $\mathbb{R}^2$



Geometry (cont'd)



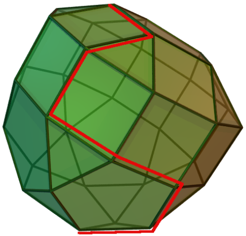
Objective: feasible solution "furthest" along specified direction

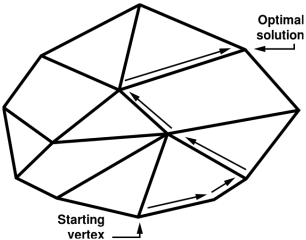
- **max P: (56, 26)**
- **max 2P + E: (88.5, 19.5)**

Main theorem: optimal solution is always at a "corner" (also called a "vertex")

Simplex Algorithm [Dantzig 1940's]

Initialize $\vec{\mathbf{x}}$ to an arbitrary corner
while(a neighboring corner $\vec{\mathbf{x}}'$ of $\vec{\mathbf{x}}$ has better objective value) {
    $\vec{\mathbf{x}} \leftarrow \vec{\mathbf{x}}'$
}
return $\vec{\mathbf{x}}$





Simplex Analysis

Theorem: Simplex returns the optimal solution.

Proof Sketch:

- Objective linear $\implies$ optimal solution at a corner
- Feasible set convex + linear objective $\implies$ any local opt is global opt

$\implies$ Once simplex terminates, at global opt

Problem: Exponential number of corners!

- Slow in theory
- Fast in practice!
  - Much of AMS LP course really about simplex: traditionally favorite algorithm of people who want to actually solve LPs
- Some theory to explain discrepancy ("smoothed analysis")

Ellipsoid Algorithm [Khachiyan 1980]

First polytime algorithm!

Designed to just solve feasibility question $\implies$ can also solve optimization

- Start with ellipsoid $\mathbf{E}$ containing feasible region $\mathbf{P}$ (if it exists)
- Let $\mathbf{x}$ be center of $\mathbf{E}$
- While($\mathbf{x}$ not feasible)
  - Find a hyperplane $\mathbf{H}$ through $\mathbf{x}$ such that all of $\mathbf{P}$ on one side
  - Let $\mathbf{E'}$ be the half-ellipsoid of $\mathbf{E}$ defined by $\mathbf{H}$
  - Find a new ellipsoid $\hat{\mathbf{E}}$ containing $\mathbf{E'}$ so that $\mathbf{vol}(\hat{\mathbf{E}}) \leq \left(\mathbf{1} - \frac{\mathbf{1}}{\mathbf{n}}\right) \mathbf{vol}(\mathbf{E})$
  - Let $\mathbf{E} = \hat{\mathbf{E}}$ and let $\mathbf{x}$ be center of $\hat{\mathbf{E}}$

Analysis

Extremely complicated!

Geometry of ellipsoids: can always find an ellipsoid containing a half-ellipsoid with at most $(1 - 1/n)$ of the volume of the original

- Using inequality from last time: after n iterations, volume drops by $\left(1 - \frac{1}{n}\right)^n \leq 1/e$ factor
- Crucial fact: if volume "too small", P must be empty

$\implies$ Polynomial time!

In practice: horrible.

Interior Point Methods (Karkmarkar's Algorithm)

Fast in both theory and practice!