

Linear Programming I

David Woodruff

Outline

- Definition of linear programming and examples
- A linear program to solve max flow and min-cost max flow
- A linear program to solve minimax-optimal strategies in games
- Algorithms for linear programming

Example

- There are 168 hours in a week. Want to allocate our time between
 - studying (S)
 - going to parties (P)
 - everything else (E)
- To survive: $E \geq 56$
- For sanity: $P + E \geq 70$
- To pass courses: $S \geq 60$
- If party a lot, need to study or eat more: $2S + E - 3P \geq 150$
- Is there a *feasible* solution? Yes, $S = 80$, $P = 20$, $E = 68$
- Happiness is $2P + E$. Find a feasible solution maximizing this *objective function*

Linear Program

- This is called a *linear program (LP)*
- All constraints are linear in our variables
- Objective function is linear
- Don't allow $S \cdot E \geq 100$, that's a polynomial program. Much harder.

Formal Definition

- Given:
 - n variables x_1, \dots, x_n
 - m linear inequalities in these variables
 - E.g., $3x_1 + 4x_2 \leq 6, 0 \leq x_1, x_1 \leq 3$
- Goal:
 - Find values for the x_i 's that satisfy constraints and maximize objective
 - In the feasibility problem just satisfy the constraints
 - What would happen if we allowed strict inequalities $x_1 < 3$?
 - $\max x_1$

Time Allocation Problem

- Variables: S, P, E
- Objective: Maximize $2P + E$ subject to
- Constraints: $S + P + E = 168$

$$E \geq 56$$

$$S \geq 60$$

$$2S + E - 3P \geq 150$$

$$P + E \geq 70$$

$$P \geq 0$$

Operations Research Problem

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

What are the variables?

x_1, x_2, x_3, x_4 denote the number of cars at plant i

What's our objective?

maximize $x_1 + x_2 + x_3 + x_4$

- Required to make at least 400 cars at plant 3
- Have 3300 hours of labor and 4000 units of material
- At most 12000 units of pollution
- Maximize number of cars made

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

What are the variables?

x_1, x_2, x_3, x_4 denote the number of cars at plant i

What's our objective?

maximize $x_1 + x_2 + x_3 + x_4$

Make at least 400 cars at plant 3
 3300 hours of labor and 4000 units of material
 At most 12000 units of pollution
 Maximize number of cars made

Note: linear programming does not give an integral solution (NP-hard)

Constraints: $x_i \geq 0$ for all i

$$x_3 \geq 400$$

$$2x_1 + 3x_2 + 4x_3 + 5x_4 \leq 3300$$

$$3x_1 + 4x_2 + 5x_3 + 6x_4 \leq 4000$$

$$15x_1 + 10x_2 + 9x_3 + 7x_4 \leq 12000$$

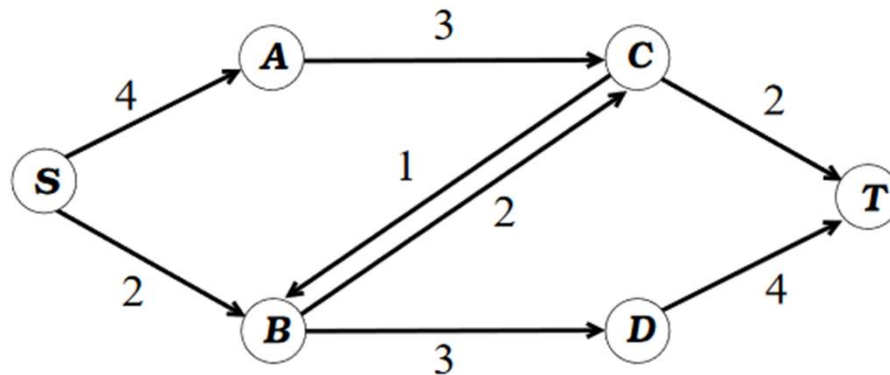
Modeling Network Flow

Variables: f_{uv} for each edge (u,v) , representing positive flow

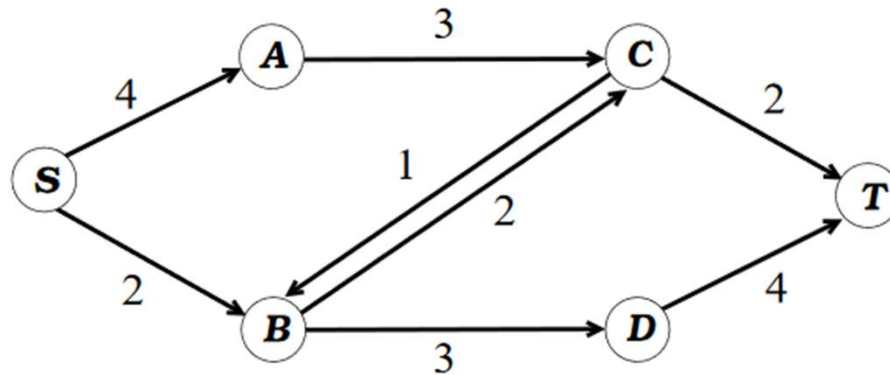
Objective: maximize $\sum_u f_{ut}$ (or maximize $\sum_v f_{sv}$)

Constraints: For all edges (u,v) $0 \leq f_{uv} \leq c(u,v)$ (capacity constraints)

For all $v \notin \{s, t\}$, $\sum_u f_{uv} = \sum_u f_{vu}$ (flow conservation)



Modeling Network Flow



In this case, our LP is: maximize $f_{ct} + f_{dt}$ subject to the constraints:

$$0 \leq f_{sa} \leq 4, 0 \leq f_{ac} \leq 3, \text{ etc.}$$

$$f_{sa} = f_{ac}, f_{sb} + f_{cb} = f_{bc} + f_{bd}, f_{ac} + f_{bc} = f_{cb} + f_{ct}, f_{bd} = f_{dt}.$$

Min Cost Max Flow

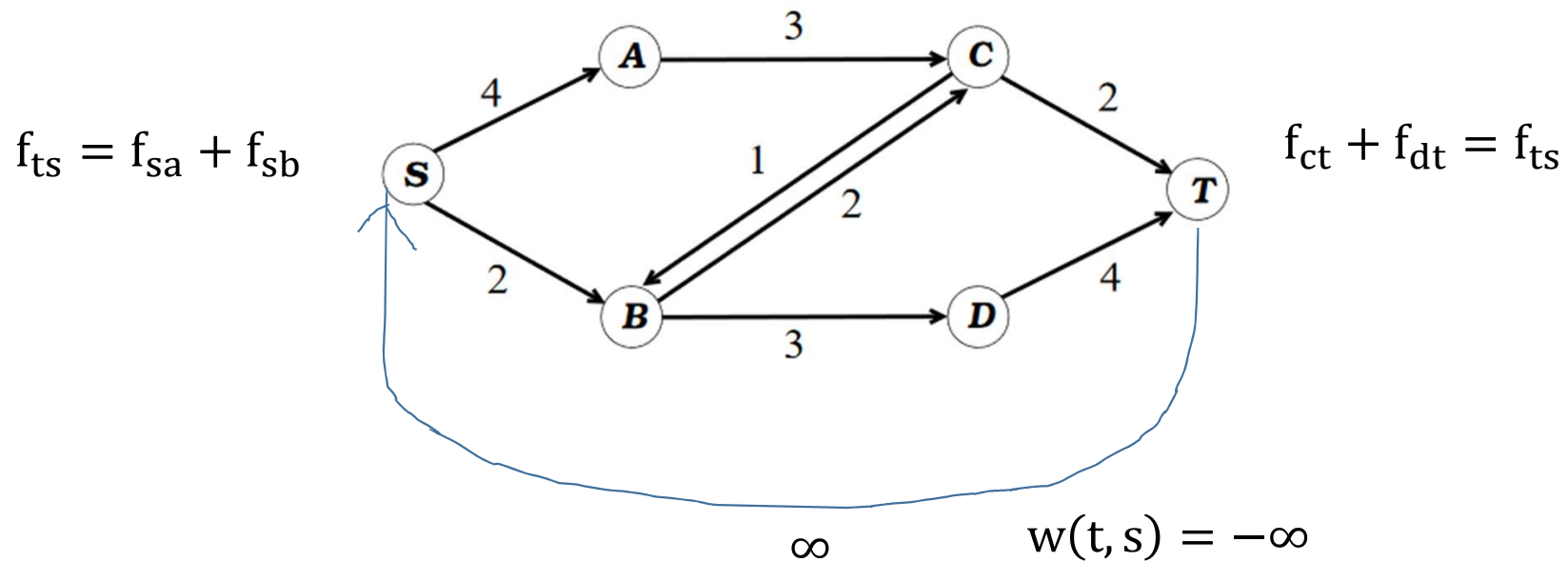
- Edge (u,v) has a capacity $c(u,v)$ and a cost $w(u,v)$
- Find a max s-t flow of least total cost, where the cost of flow f is

$$\sum_{(u,v) \in E} w(u,v) f_{uv}$$

- *How to solve this?*
- **Solution 1:** Solve for a maximum flow f
 - Add a constraint that flow must equal the flow of f
 - Minimize $\sum_{(u,v) \in E} w(u,v) f_{uv}$ also subject to original constraints
- **Solution 2:** Add an edge (t,s) of infinite capacity and very negative cost
 - Minimizing cost automatically maximizes flow

Min Cost Max Flow

$$\min \sum_{(u,v) \in E} w(u,v) f_{uv}$$



Zero Sum Games

Row payoffs:

20	-10	5
5	10	-10
-5	0	10

- Given a zero-sum game with n rows and n columns, compute a minimax optimal strategy for row player
- *What are the variables?*
 - Probabilities p_1, \dots, p_n on our actions
 - Linear constraints: $\sum_{i=1, \dots, n} p_i = 1$ and $p_i \geq 0$ for all i
 - Maximize the minimum expected payoff, over all column pure strategies
- *How to maximize a minimum with a linear program?*
- Create new “dummy variable” v to represent minimum

Zero Sum Games

- $R_{i,j}$ represents payoff to row player with row player action i and column player action j
- Variables: p_1, \dots, p_n and v
- Objective: maximize v
- Constraints:
 - $p_i \geq 0$ for all i , and $\sum_i p_i = 1$
 - For all columns j , $\sum_i p_i R_{ij} \geq v$

Linear Programs in Standard Form?

- Many different ways to write the same LP
- Use vector notation, so $c^T x = \sum_{i=1, \dots, d} c_i x_i$ if there are d variables
- Any LP can be written in the following form:
- Max $c^T x$

$$\begin{aligned} \text{Subject to } Ax &\leq b \\ x &\geq 0 \end{aligned}$$

How to handle equality constraints $d^T x = e$?

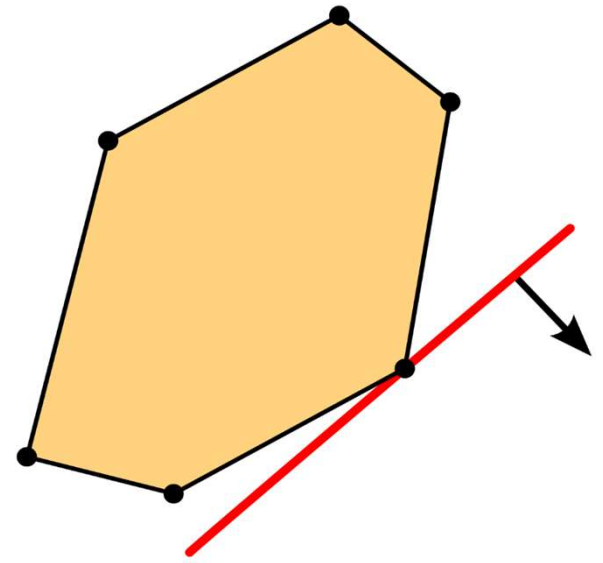
How to convert $\min c^T x$ to a maximization?

How to handle an unconstrained variable x_i which could be positive or negative?

Substitute $x_i = y_i - z_i$, $y_i \geq 0$, $z_i \geq 0$, everywhere

Facts about Linear Programs

- Consider the LP
- Max $c^T x$
Subject to $Ax \leq b$
 $x \geq 0$
- Think of maximizing $c^T x$ over the set $Ax \leq b, x \geq 0$
- What does the set $Ax \leq b, x \geq 0$ look like?
 - Each row is a *halfspace*, cutting \mathbb{R}^d into two pieces by a hyperplane
 - The intersection of halfspaces could be empty
 - Then the LP is *infeasible*
 - Could be unbounded
 - Could be bounded and then we call it the *feasible region*
- Maximizing $c^T x$ moves the hyperplane with normal vector c until it is tangent to the feasible region



Convexity Properties

- Feasible region $Ax \leq b, x \geq 0$ is convex
 - If p and q are in the feasible region, then so is the line segment joining p and q . **Why?**
- Proof by pictures, e.g., convex polygon in two dimensions
- Formally, since $Ap \leq b$ and $Aq \leq b$, for any $\lambda \in [0,1]$,
 - $\lambda Ap \leq \lambda b$ and $(1 - \lambda)Aq \leq (1 - \lambda)b$
 - So $A(\lambda p + (1 - \lambda)q) \leq b$
 - Also $\lambda p \geq 0$ and $(1 - \lambda)q \geq 0$ since $p \geq 0$ and $q \geq 0$
- More generally, intersections of convex sets are convex
- Max $c^T x$ occurs at a vertex. **Can we just enumerate all vertices?**

Algorithms for Linear Programming

- Simplex Algorithm
 - Practical, but exponential time in the worst-case
- Ellipsoid Algorithm
 - First polynomial time algorithm, but slow in practice
- Karmarkar's Algorithm (interior point)
 - Polynomial time algorithm and competitive in practice
- Software: LINDO, CPLEX, Solver (in Excel)

Time Allocation Problem

- Variables: S, P, E
- Objective: Maximize $2P + E$ subject to
- Constraints: $S + P + E = 168$

$$E \geq 56$$

$$S \geq 60$$

$$2S + E - 3P \geq 150$$

$$P + E \geq 70$$

$$P \geq 0$$

Substitute $S = 168 - P - E$, so two variables P and E, want to maximize $2P + E$.

Intuition for Linear Programming

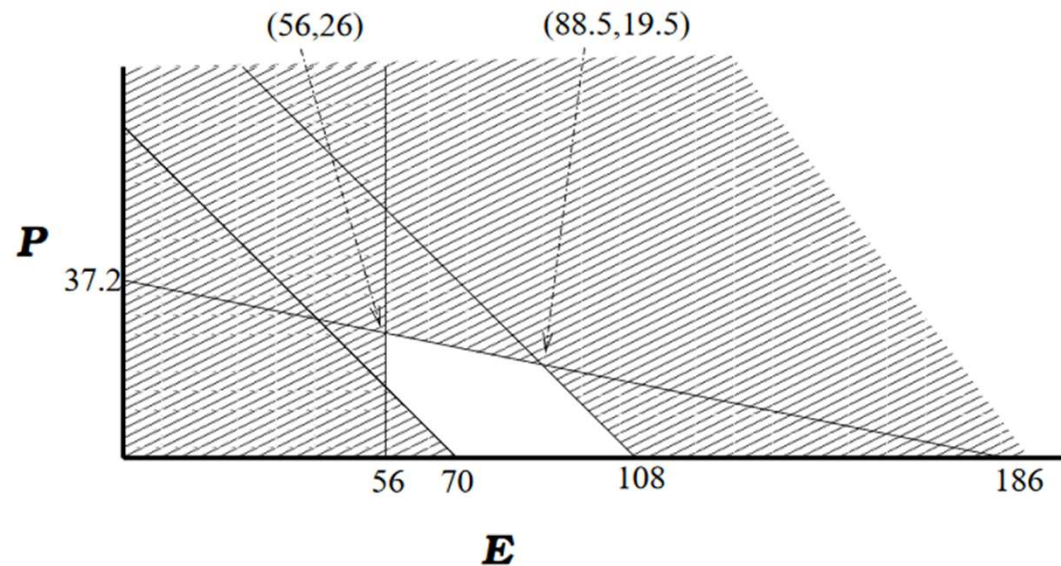
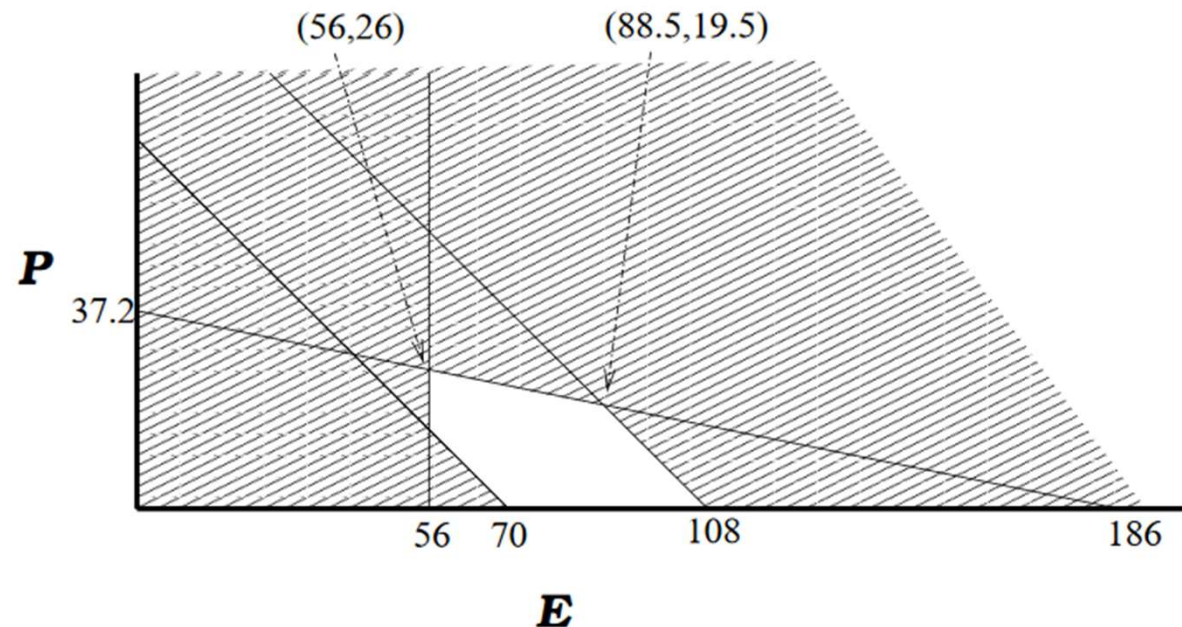


Figure 13.1: Feasible region for our time-planning problem. The constraints are: $E \geq 56$; $P + E \geq 70$; $P \geq 0$; $S \geq 60$ which means $168 - P - E \geq 60$ or $P + E \leq 108$; and finally $2S - 3P + E \geq 150$ which means $2(168 - P - E) - 3P + E \geq 150$ or $5P + E \leq 186$.

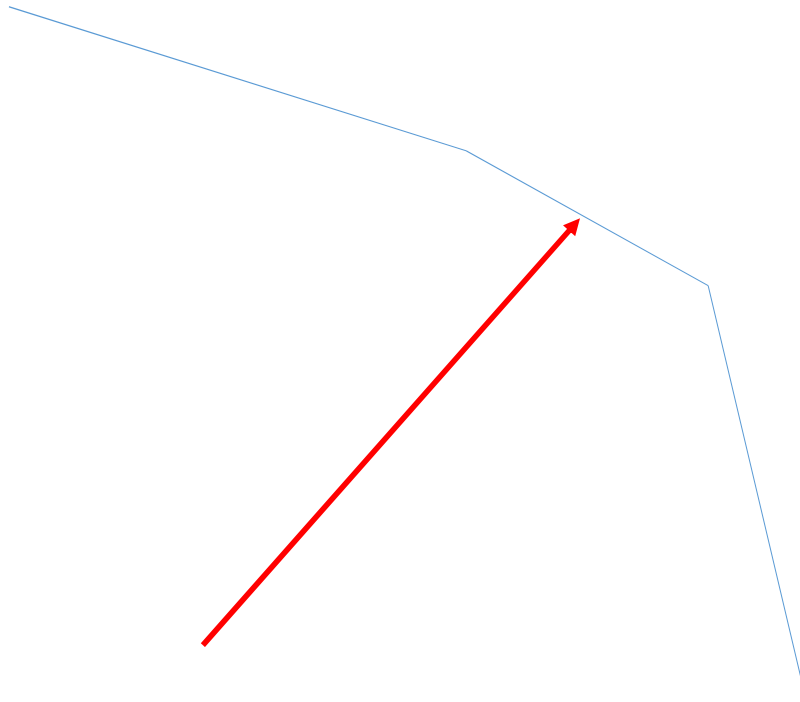
Maximizing P occurs at $(56, 26)$. Maximizing $2P + E$ occurs at $(88.5, 19.5)$

Simplex Algorithm



- Start at vertex of the feasible region (polyhedron in high dimensions)
- Look at cost of objective function at each neighbor
- Move to neighbor of maximum cost
- Always make progress, but could take exponential time (in high dimensions)

Simplex Algorithm



Get stuck in local maximum?

No, since feasible set is convex

Other Annoyances I

- How to start at a vertex of the feasible region?
- $Ax \leq b$
 $x \geq 0$
- What if it's not even feasible?
- Introduce "slack" variable s . Consider:
- $\min s$
subject to $Ax \leq b + s \cdot 1^m$
 $x \geq 0, s \geq 0, s \leq \max_i -b_i$
- Feasible. Can run simplex starting at $x = 0^n$ and $s = \max_i -b_i$
- If original LP is feasible, minimum achieved when $s = 0$, and x that is output is a vertex in the feasible region of original LP

Other Annoyances II

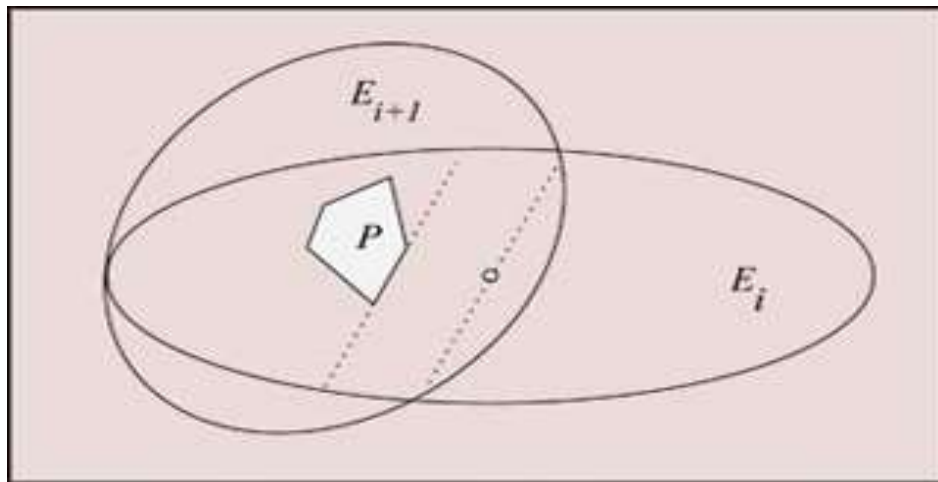
- What if the feasible region is unbounded?
 - Ok, as long as objective function is bounded
- What if objective function is unbounded?
 - Output ∞ , how to detect this?
- Many ways
 - see one based on duality in a few lectures
 - include constraints $-M \leq x_i \leq M$ for all i , for a very large value M
 - can efficiently find M to ensure if solution is finite, still find the optimum

Ellipsoid Algorithm

Solves feasibility problem

Replace objective function with constraint, do binary search

Replace “minimize $x_1 + x_2$ ” with $x_1 + x_2 \leq \lambda$



Can handle exponential number of constraints if there's a separation oracle

Karmarkar's Algorithm

- Works with feasible points but doesn't go corner to corner
- Moves in interior of the feasible region – “interior point method”

