Linear Programming and Zero Sum Games

Ron Parr
CPS 570

With thanks to Vince Conitzer for some content

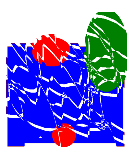
What are Linear Programs?

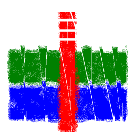
- Linear programs are ***constrained optimization problems***
- Constrained optimization problems ask us to maximize or minimize a function subject to mathematical constraints on the variables
  - Convex programs have convex objective functions and convex constraints
  - Linear programs (special case of convex programs) have linear objective functions and linear constraints

- LPs = generic language for wide range problems
- LP solvers = widely available hammers
- Entire classes and vast expertise invested in making problems look like nails

Linear programs: example

- Make reproductions of 2 paintings





- Painting 1:
  - Sells for $30
  - Requires 4 units of blue, 1 green, 1 red
- Painting 2
  - Sells for $20
  - Requires 2 blue, 2 green, 1 red
- We have 16 units blue, 8 green, 5 red

maximize $3x + 2y$

subject to

$4x + 2y \le 16$

$x + 2y \le 8$

$x + y \le 5$

$x \ge 0$

$y \ge 0$

Solving the linear program graphically

maximize $3x + 2y$

subject to

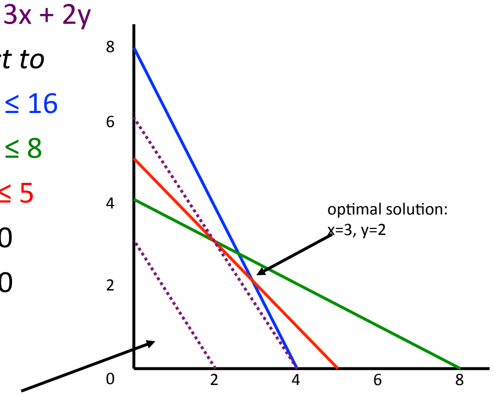
$4x + 2y \le 16$

$x + 2y \le 8$

$x + y \le 5$

$x \ge 0$

$y \ge 0$



Feasible region = region not violating constraints

Linear Programs in General

- Linear constraints, linear objective function
  - Maximize (minimize): $f(\mathbf{x})$ ⟵ Linear function of vector **x**
  - Subject to: $\mathbf{Ax} \leq \mathbf{b}$ ↖ Matrix **A**
- Can swap maximize/minimize, ≤/≥; can add equality
- View as search: Searches space of values of **x**
- Alternatively: Search for tight constraints w/high objective function value

Solving linear programs (1)

- Optimal solutions always exist at vertices of the feasible region
  - Why?
  - Assume you are not at a vertex, you can always push further in direction that improves objective function
- Dumb(est) algorithm:
  - Given n variables, k constraints
  - Check all k choose n = $O(k^n)$ possible vertices

Solving linear programs (2)

- Smarter algorithm (simplex)
  - Pick a vertex
  - Repeatedly hop to neighboring (one different tight constrain) vertices that improvement the objective function
  - Guaranteed to find solution (no local optima)
  - May take exponential time in worst case (though this rare)
- Still smarter algorithm
  - Move inside the interior of the feasible region, in direction that increases objective function
  - Stop when no further improvements possible
  - Tricky to get the details right, but weakly polynomial time

Solving LPs in Practice

- Use commercial products like cplex or gurobi
- Do not try to implement an LP solver yourself
- Do not use matlab’s linprog for anything other than small problems.

LP Trick (one of many)

- Suppose you have a huge number of constraints, but a small number of variables (k>>n)
- Constraint generation:
  - Start with a subset of the constraints
  - Find solution to simplified LP
  - Find most violated constraint, add back to LP
  - Repeat
- Why does this work?
  - If missing constraints are unviolated, then adding them back wouldn't change the solution
  - Sometimes terminates after adding in only a fraction of total constraints
  - No guarantees, but often helpful in practice

Duality

- For every LP there is an equivalent "Dual" probelm
- Solution to primal can be used to reconstruct solution to dual, and vice versa
- LP duality:

$$\begin{aligned} \text{minimize}: & \ c^T x \\ \text{subject to}: & \ \mathbf{A}x = b \\ : & \ x \geq 0 \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \text{maximize}: & \ b^T y \\ \text{subject to}: & \ \mathbf{A}^T y = c \\ : & \ y \geq 0 \end{aligned}$$

MDP Solved as an LP

$$V(s) = R(s,a) + \gamma \max_a \sum_{s'} P(s' \mid s,a) V(s')$$

Issue: Turn the non-linear max into a collection of linear constraints

$$\forall s,a : V(s) \geq \underbrace{R(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s')}$$

Optimal action has tight constraints

MINIMIZE: $\sum_s V(s)$

What is Game Theory?

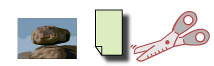
- Very general mathematical framework to study situations where multiple agents interact, including:
  - Popular notions of games
  - Everything up to and including multistep, multiagent, simultaneous move, partial information games
  - Can even including negotiating, posturing and uncertainty about the players and game itself
- von Neumann and Morgenstern (1944) was a major launching point for modern game theory
- Nash: Existence of equilibria in general sum games

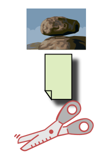
Covered Today

- 2 player, zero sum simultaneous move games
- Example: Rock, Paper, Scissors
- Linear programming solution

Rock, Paper, Scissors Zero Sum Formulation

- In zero sum games, one player's loss is other's gain
- Payoff matrix:





	R	P	S
R	0	–1	1
P	1	0	–1
S	–1	1	0

- Minimax solution maximizes worst case outcome

Rock, Paper, Scissors Equations

- R,P,S = probability that we play rock, paper, or scissors respectively (R+P+S = 1)
- U is our expected utility
- Bounding our utility:
  - Opponent rock case: $U \le P - S$
  - Opponent paper case: $U \le S - R$
  - Opponent scissors case: $U \le R - P$
- Want to maximize U subject to constraints
- Solution: (1/3, 1/3, 1/3)

Rock, Paper, Scissors LP Formulation

- Our variables are: $x=[U,R,P,S]^T$
- We want:
  - Maximize U
  - $U \le P - S$
  - $U \le S - R$
  - $U \le R - P$
  - $R+P+S = 1$
- How do we make this fit:

$$\begin{aligned} \text{maximize} &: c^T x \\ \text{subject to} &: \mathbf{A}x \le b \\ &: x \ge 0 \end{aligned}$$

?

Rock Paper Scissors LP Formulation

$x = [U,R,P,S]^T$

$$A = \begin{pmatrix} 1 & 0 & -1 & 1 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 \end{pmatrix}$$

$b = [0,0,0,1,-1]^T$

$c = [1,0,0,0]^T$

$$\begin{aligned} \text{maximize}: & \; c^T x \\ \text{subject to}: & \; \mathbf{A}x \le b \\ : & \; x \ge 0 \end{aligned}$$

Rock, Paper, Scissors Solution

- If we feed this LP to an LP solver we get:
  - R=P=S=1/3
  - U=0
- Solution for the other player is:
  - The same…
  - By symmetry
- This is the minimax solution
- This is also a Nash equilibrium

Tangent: Why is RPS Fun?

- OK, it's not…

- Why *might* RPS be fun?
  - Try to exploit non-randomness in your friends
  - Try to be random yourself

Minimax Solutions in General

- Minimax solutions for 2-player zero-sum games can always be found by solving a linear program
- The minimax solutions will also be Nash equilibria

- For general sum games:
  - Minimax does not apply
  - Equilibria may not be unique
  - Need to search for equilibria using more computationally intensive methods