Linear Programming and Game Theory

Ron Parr

CPS 570

With thanks to Vince Conitzer for some content

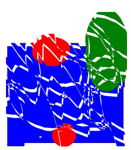
What are Linear Programs?

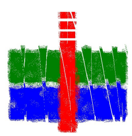
- Linear programs are ***constrained optimization problems***
- Constrained optimization problems ask us to maximize or minimize a function subject to mathematical constraints on the variables
  - Convex programs have convex objective functions and convex constraints
  - Linear programs (special case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

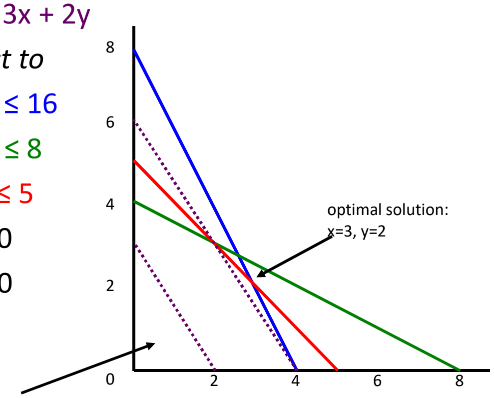
$4x + 2y \le 16$

$x + 2y \le 8$

$x + y \le 5$

$x \ge 0$

$y \ge 0$

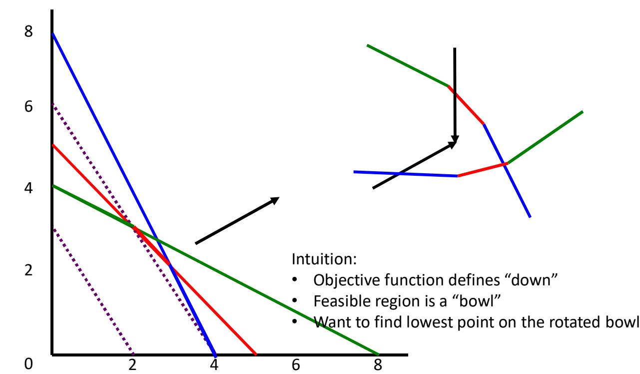


Feasible region = region not violating constraints

Linear Programs in General

- Linear constraints, linear objective function
  - Maximize (minimize): $f(\mathbf{x})$ ← Linear function of vector **x**
  - Subject to: $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ ← 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

What Happens In Higher Dimensions (1) Understanding the Feasible Region



What Happens In Higher Dimensions (2) lines->hyperplanes

- Inequality w/2 variables -> one side of a line
- 3 variables -> one side of a plane
- k variables -> one side of hyperplane
- Physical intuition:

http://www.rubylane.com/item/623546-4085/Orrefors-x22Zenithx22-Pattern-Crystal-Bowl

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 (or at least doesn't hurt)
  - How many vertices does a kxn matrix imply?
- 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 improve the objective function
  - Guaranteed to find solution (no local optima)
  - May take exponential time in worst case (though rarely)
- 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. Really. No – REALLY!

Modified LP

maximize $3x + 2y$

subject to

$4x + 2y \leq 15$

$x + 2y \leq 8$

$x + y \leq 5$

$x \geq 0$

$y \geq 0$

Optimal solution: x = 2.5, y = 2.5

Solution value = 7.5 + 5 = 12.5

Half paintings?

Integer (linear) program

maximize $3x + 2y$

subject to

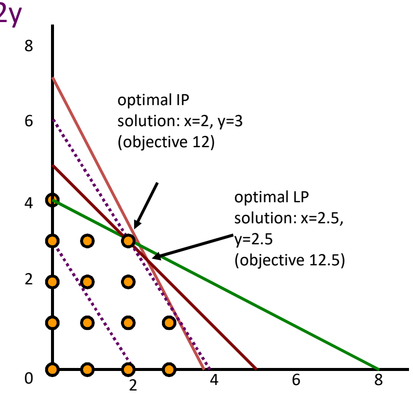
$4x + 2y \leq 15$

$x + 2y \leq 8$

$x + y \leq 5$

$x \geq 0$, integer

$y \geq 0$, integer



Mixed integer (linear) program

maximize 3x + 2y

subject to

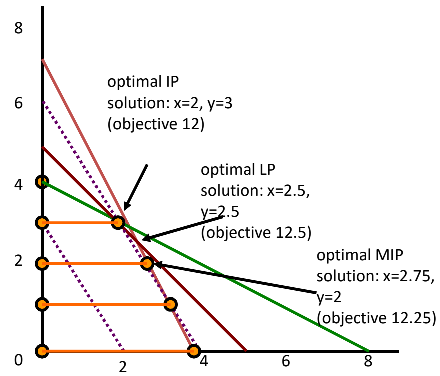
4x + 2y ≤ 15

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0

y ≥ 0, integer



Solving (M)IPs

- (Mixed) Integer programs are NP-hard to solve
- Intuition: Constraint surface is jagged; no obvious way to avoid checking exponential number of assignments to integer variables
- In practice:
  - Constraints often give clues on how to restrict number of solutions considered
  - Smart solvers (cplex, gurobi) can ***sometimes*** find solutions to large (M)IPs surprisingly quickly (and surprisingly slowly)

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:

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

MDP Solved as an LP

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

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

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

MINIMIZE: $\sum_s V(s)$

Optimal action has tight constraints

What is Game Theory? I

- 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
  - Example Duke CS research: Aiming sensors to catch hiding enemies, assigning guards to posts
  - Can even include 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

(wikipedia)

What is game theory? II

- Study of settings where multiple agents each have
  - Different preferences (utility functions),
  - Different actions
- Each agent's utility (potentially) depends on all agents' actions
  - What is optimal for one agent depends on what other agents do
  - Can be circular
- Game theory studies how agents can rationally form beliefs over what other agents will do, and (hence) how agents should act
- Useful for acting and (potentially) predicting behavior of others
- Not necessarily descriptive

Real World Game Theory Examples

- War
- Auctions
- Animal behavior
- Networking protocols
- Peer to peer networking behavior
- Road traffic
- Mechanism design:
  - Suppose we want people to do X?
  - How to engineer situation so they will act that way?

Covered Today

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

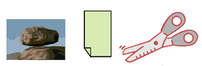
Linear Programs (max formulation)

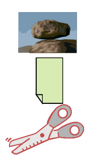
$$\text{maximize}: c^T x$$
$$\text{subject to}: \mathbf{A}x \leq b$$
$$: x \geq 0$$

- Note: min formulation also possible
  - Min: $c^T x$
  - Subject to: $Ax \geq b$
- Some use equality as the canonical representation (introducing slack variables)
- LP tricks
  - Multiply by -1 to reverse inequalities
  - Can easily introduce equality constraints, or arbitrary domain constraints

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 ≤ P – S
  - Opponent paper case: U ≤ S – R
  - Opponent scissors case: U ≤ 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 ≤ P – S
  - U ≤ S – R
  - U ≤ 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 an equilibrium
  - No player has an incentive to deviate
  - (Defined more precisely later)

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

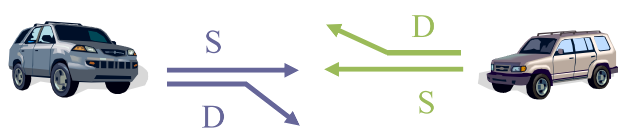
- What do we know about minimax solutions?
  - Can a suboptimal opponent trick minimax?
  - When should we abandon minimax?
- Minimax solutions for 2-player zero-sum games can always be found by solving a linear program
- The minimax solutions will also be equilibria
- For general sum games:
  - Minimax does not apply
  - Equilibria may not be unique
  - Need to search for equilibria using more computationally intensive methods

Outline

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

"Chicken"

- Two players drive cars towards each other
- If one player goes straight, that player wins
- If both go straight, they both die



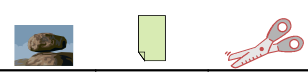
Source: wikipedia

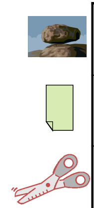
	D	S
D	0, 0	-1, 1
S	1, -1	-5, -5

not zero-sum

Rock-paper-scissors – Seinfeld variant

MICKEY: All right, rock beats paper!
(Mickey smacks Kramer's hand for losing)
KRAMER: I thought paper covered rock.
MICKEY: Nah, rock flies right through paper.
KRAMER: What beats rock?
MICKEY: (looks at hand) Nothing beats rock.





	0, 0	1, -1	1, -1
	-1, 1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

Dominance

- Player i's strategy s_i strictly dominates s_i' if
  - for any s_{-i}, $u_i(s_i, s_{-i}) > u_i(s_i', s_{-i})$
- s_i weakly dominates s_i' if
  - for any s_{-i}, $u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$; and
  - for some s_{-i}, $u_i(s_i, s_{-i}) > u_i(s_i', s_{-i})$

-i = "the player(s) other than i"

strict dominance

weak dominance

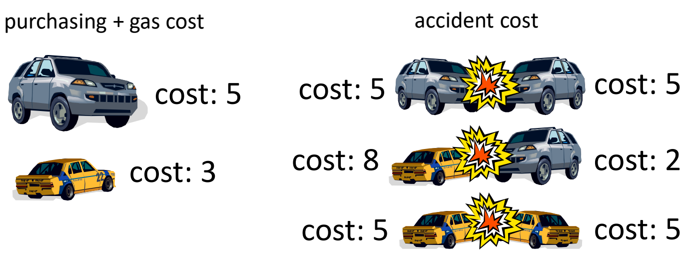
	0, 0	1, -1	1, -1
	-1, 1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

Prisoner's Dilemma

- Pair of criminals has been caught
- District attorney has evidence to convict them of a minor crime (1 year in jail); knows that they committed a major crime together (3 years in jail) but cannot prove it
- Offers them a deal:
  - If both confess to the major crime, they each get a 1 year reduction
  - If only one confesses, that one gets 3 years reduction

	confess	don't confess
confess	-2, -2	0, -3
don't confess	-3, 0	-1, -1

"Should I buy an SUV?"



	-10, -10	-7, -11
	-11, -7	-8, -8

"2/3 of the average" game

- Everyone writes down a number between 0 and 100
- Person closest to 2/3 of the average wins
- Example:
  - A says 50
  - B says 10
  - C says 90
  - Average(50, 10, 90) = 50
  - 2/3 of average = 33.33
  - A is closest (|50-33.33| = 16.67), so A wins

Iterated dominance

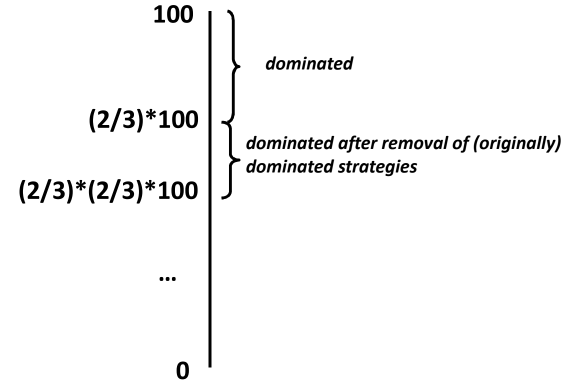
- Iterated dominance: remove (strictly/weakly) dominated strategy, repeat
- Iterated strict dominance on Seinfeld's RPS:

0, 0	1, -1	1, -1
-1, 1	0, 0	-1, 1
-1, 1	1, -1	0, 0

→

0, 0	1, -1
-1, 1	0, 0

"2/3 of the average" game revisited



Mixed strategies

- Mixed strategy for player i = probability distribution over player i's (pure) strategies
- E.g. 1/3 , 1/3 , 1/3
- Example of dominance by a mixed strategy:

1/2	3, 0	0, 0
1/2	0, 0	3, 0
	1, 0	1, 0

Best Responses

- Let A be a matrix of player 1's payoffs
- Let σ_2 be a mixed strategy for player 2
- $A\sigma_2$ = vector of expected payoffs for each strategy for player 1
- Highest entry indicates **best response** for player 1
- Any mixture of ties is also BR
- Generalizes to >2 players

0, 0	-1, 1
1, -1	-5, -5

σ_2

Nash equilibrium [Nash 50]

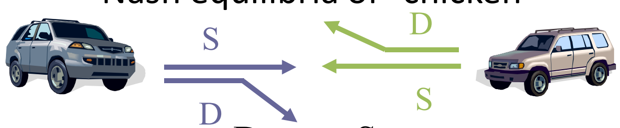
- A vector of strategies (one for each player) = a strategy profile
- Strategy profile $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a Nash equilibrium if each σ_i is a best response to σ_{-i}
  - That is, for any i, for any σ_i', $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma_i', \sigma_{-i})$
- Does not say anything about multiple agents changing their strategies at the same time
- In any (finite) game, at least one Nash equilibrium (possibly using mixed strategies) exists [Nash 50]
- (Note - singular: equilibrium, plural: equilibria)

Equilibrium Strategies vs. Best Responses

- equilibrium strategy -> best response?
- best response -> equilibrium strategy?
- Consider Rock-Paper-Scissors
  - Is (1/3, 1/3, 1/3) a best response to (1/3, 1/3, 1/3)?
  - Is (1, 0, 0) a best response to (1/3, 1/3, 1/3)?
  - Is (1, 0, 0) a strategy for any equilibrium?

	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

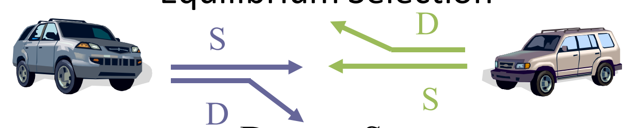
Nash equilibria of "chicken"



	D	S
D	0, 0	-1, 1
S	1, -1	-5, -5

- (D, S) and (S, D) are Nash equilibria
  - They are pure-strategy Nash equilibria: nobody randomizes
  - They are also strict Nash equilibria: changing your strategy will make you strictly worse off
- No other pure-strategy Nash equilibria

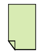
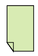
Equilibrium Selection



	D	S
D	0, 0	-1, 1
S	1, -1	-5, -5

- (D, S) and (S, D) are Nash equilibria
- Which do you play?
- What if player 1 assumes (S, D), player 2 assumes (D, S)
- Play is (S, S) = (-5, -5)!!!
- This is the **equilibrium selection** problem

Rock-paper-scissors

			
	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

- Any pure-strategy Nash equilibria?
- It has a mixed-strategy Nash equilibrium:

  Both players put probability 1/3 on each action

Nash equilibria of "chicken"…

	D	S
D	0, 0	-1, 1
S	1, -1	-5, -5

- Is there a Nash equilibrium that uses mixed strategies -- say, where player 1 uses a mixed strategy?
- *If a mixed strategy is a best response, then all of the pure strategies that it randomizes over must also be best responses*
- So we need to make player 1 indifferent between D and S
- Player 1's utility for playing D = $-p^c_S$
- Player 1's utility for playing S = $p^c_D - 5p^c_S = 1 - 6p^c_S$
- So we need $-p^c_S = 1 - 6p^c_S$ which means $p^c_S = 1/5$
- Then, player 2 needs to be indifferent as well
- Mixed-strategy Nash equilibrium: ((4/5 D, 1/5 S), (4/5 D, 1/5 S))
  - People may die! Expected utility -1/5 for each player

$-p^c_S$ = probability that column player plays s

Computational Issues

- Zero-sum games - solved efficiently as LP
- General sum games may require exponential time (in # of actions) to find a single equilibrium (no known efficient algorithm and good reasons to suspect that none exists)
- Some better news: Despite bad worst-case complexity, many games can be solved quickly

Game Theory Issues

- How descriptive is game theory?
  - Some evidence that people play equilibria
  - Also, some evidence that people act irrationally
  - If it is computationally intractable to solve for equilibria of large games, seems unlikely that people are doing this
- How reasonable is (basic) game theory?
  - Are payoffs known?
  - Are situations really simultaneous move with no information about how the other player will act?
  - Are situations really single-shot? (repeated games)
  - How is equilibrium selection handled in practice?

Extensions

- Partial information
- Uncertainty about the game parameters, e.g., payoffs (Bayesian games)
- Repeated games: Simple learning algorithms can converge to equilibria in some repeated games
- Multistep games with distributions over next states (game theory + MDPs = stochastic games)
- Multistep + partial information (Partially observable stochastic games)

- Game theory is so general, that it can encompass essentially all aspects of strategic, multiagent behavior, e.g., negotiating, threats, bluffs, coalitions, bribes, etc.