PRELIMINARIES & TECHNIQUES

DAS, DICKERSON, & WILDER

WHAT'S USED IN MARKET DESIGN & RESOURCE ALLOCATION?

We want the best outcome from a set of outcomes.

Convex optimization:

- Linear programming
- Quadratic programming

Nonconvex optimization:

- (Mixed) integer linear programming
- (Mixed) integer quadratic programming

Incomplete heuristic & greedy methods

Care about maximization (social welfare, profit), minimization (regret, loss), or simple feasibility (does a stable matching with couples exist?)

"PROGRAMMING?"

It's just an optimization problem.

Blame this guy:

- **George Dantzig (Maryland alumnus!)**
- **Focused on solving US military logistic scheduling problems aka programs**

Solving (un)constrained optimization problems is much older:

- **Newton (e.g., Newton's method for roots)**
- **Gauss (e.g., Gauss-Newton's non-linear regression)**
- **Lagrange (e.g., Lagrange multipliers)**

GENERAL MODEL

General math program:

$$
\begin{aligned}
&\text{min/max} && f(\boldsymbol{x}) \\
&\text{subject to} && g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\
& && h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, k \\
& && \mathbf{x} \in X \subset \Re^n \\
& && f, g_i, h_j : \Re^n \rightarrow \Re
\end{aligned}
$$

Linear programming: all of f, g_i, h_j are linear (affine) functions

Nonlinear programming: at least part of f, g_i, h_j is nonlinear

Integer programming: Feasible region constrained to integers

Convex, quadratic, etc …

CONVEX FUNCTIONS

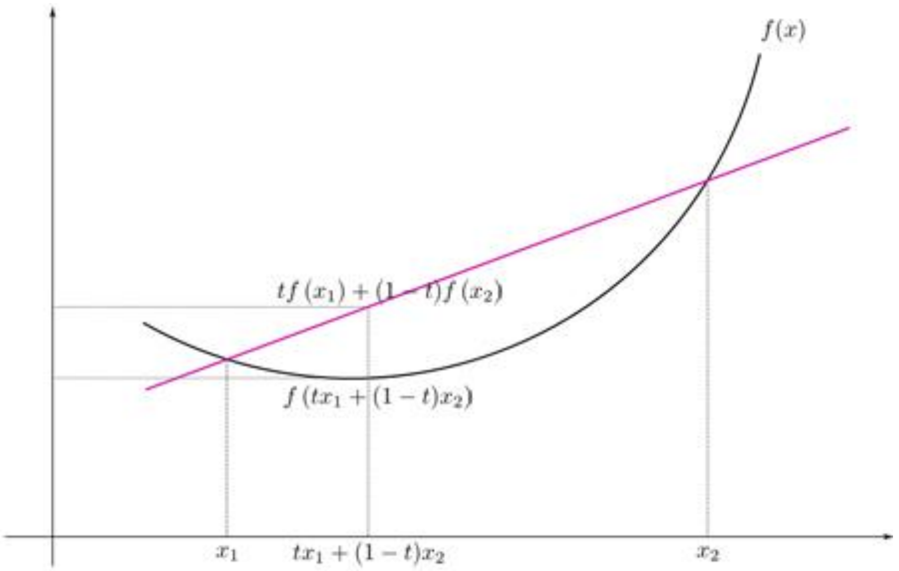
"A function is convex if the line segment between any two points on its graph lies above it."

Formally, given function *f* and two points x, y:

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}) \quad \forall \lambda \in [0,1]$$

Convex or non-convex?

- $\mathbf{a}^T\mathbf{x} + b$
- e^x, e^{-x}
- $\mathbf{x}^T\mathbf{Q}\mathbf{x}, \quad \mathbf{Q} \succeq \mathbf{0}$
- $\mathbf{x}^T\mathbf{Q}\mathbf{x}, \quad \mathbf{Q}$ indefinite
- $\|\mathbf{x}\|$
- $\log x, \ \sqrt{x}$



CONVEX SETS

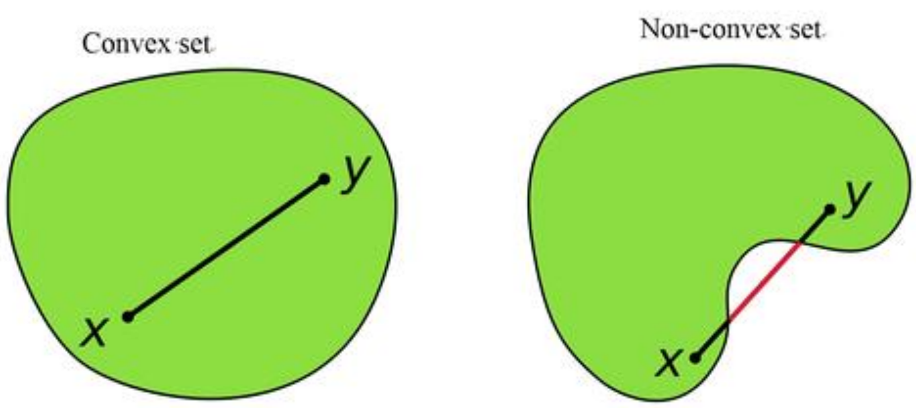
“A set is convex if, for every pair of points within the set, every point on the straight line segment that joins them is in the set.”

Formally, give a set *S* and two points x, y in S:

$$\mathbf{x} \in S, \mathbf{y} \in S \Rightarrow \lambda \mathbf{x} + (1 - \lambda)\mathbf{y} \in S$$

Convex or non-convex sets?

- $\{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$
- $\mathbb{R}^n_+$
- $\{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}\}$
- $\{(\mathbf{x}, t) : \|\mathbf{x}\| \leq t\}$



SO WHAT?

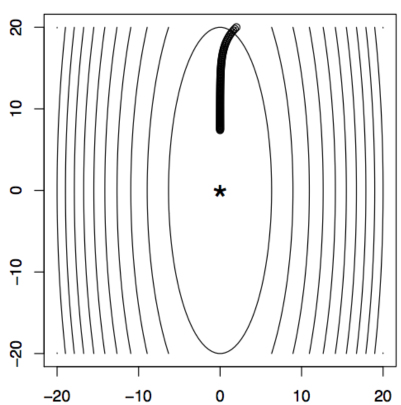
An optimization (minimization) problem with a convex objective function and a convex feasible region is solved via convex programming.

Lets us use tools from convex analysis

- Local minima are global minima
- The set of global mimina is convex
- There is a unique global minimum if strictly convex

Lets us make statements like gradient descent converges to a global minimum (under some assumptions w.r.t local Lipschitz and step size)

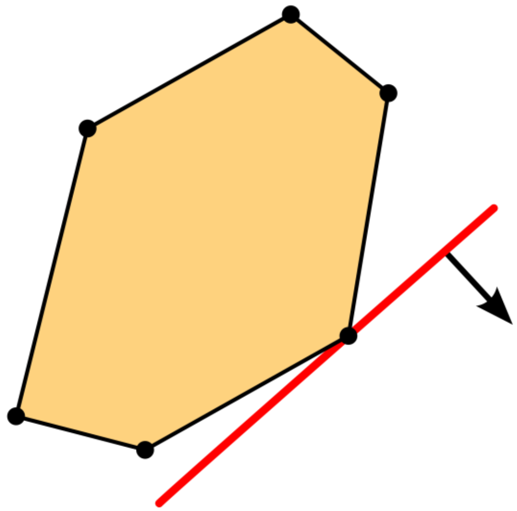
But let's start even simpler …



LINEAR PROGRAMS

An "LP" is an optimization problem with a linear objective function and linear constraints.

- **A line drawn between any two points *x*, *y* on a line is on the line → clearly convex**
- **Feasible region aka polytope also convex**



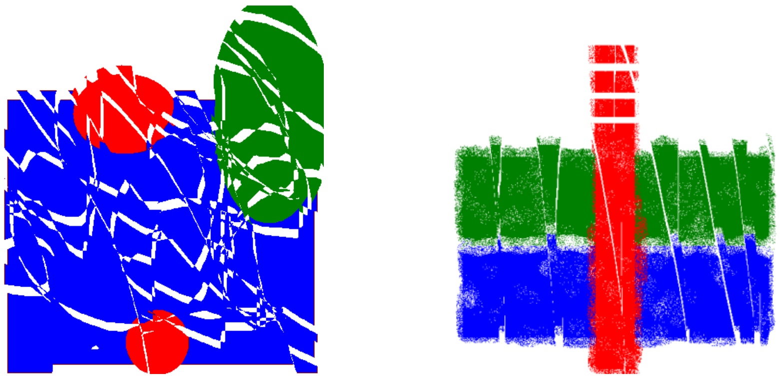
General LP:

$$\text{min/max} \quad \mathbf{c}^T\mathbf{x}$$
$$\text{subject to} \quad A\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq 0$$

Where c, *A*, b are known, and we are solving for x.

LP: EXAMPLE

We make reproductions of two paintings:



Painting 1 sells for \$30, painting 2 sells for \$20

Painting 1 requires 4 units of blue, 1 green, 1 red

Painting 2 requires 2 blue, 2 green, 1 red

We have 16 units blue, 8 green, 5 red

maximize $3x + 2y$

subject to

$4x + 2y \leq 16$

$x + 2y \leq 8$

$x + y \leq 5$

$x \geq 0$

$y \geq 0$

Objective ???????   Constraints ???????

SOLVING THE LINEAR PROGRAM GRAPHICALLY

***maximize* 3x + 2y**

subject to

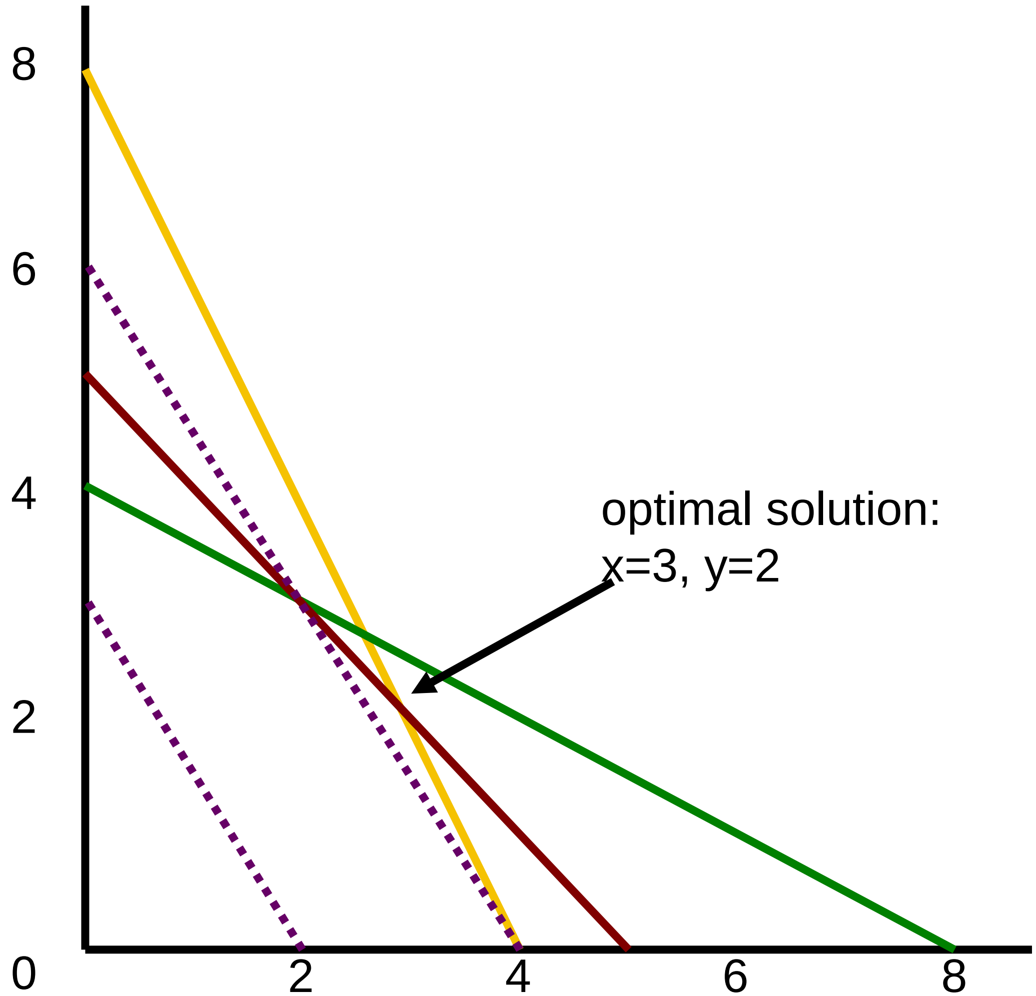
4x + 2y ≤ 16

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0

y ≥ 0



VC

LP EXAMPLE: SOLVING FOR 2-P ZERO-SUM NASH

Recall:

- Mixed Nash Equilibrium always exists
- Even if I know your strategy, in equilibrium I don't deviate

Given a payoff matrix A:

	Morality	Tax-Cuts
Economy	+3, -3	-1, +1
Society	-2, +2	+1, -1

[Example from Daskalakis]

If Row announces strategy $<x_1, x_2>$, then Col gets expected payoffs:

E["Morality"] = $-3x_1 + 2x_2$

E["Tax-Cuts"] = $1x_1 - 1x_2$

So Col will best respond with $\max(-3x_1 + 2x_2, 1x_1 - 1x_2)$ …

LP EXAMPLE: SOLVING FOR 2-P ZERO-SUM NASH

But if Col gets max(-3x_1 + 2x_2, 1x_1 – 1x_2),
then Row gets -max(-3x_1 + 2x_2, 1x_1 – 1x_2) = min(…)

So, if Row must announce, she will choose the strategy:

<x1, x2> = arg max min(3x_1 - 2x_2, -1x_1 + 1x_2)

This is just an LP:

maximize z

such that $3x_1 - 2x_2 \geq z$

$-1x_1 + 1x_2 \geq z$

$x_1 + x_2 = 1$

$x_1, x_2 \geq 0$

So Row player is guaranteed to get at least z

LP EXAMPLE: SOLVING FOR 2-P ZERO-SUM NASH

Can set up the same LP for the Col player, to get general LPs:

max	z_R	**min**	z_C
s.t.	$(xA)_j \geq z_R$ **for all *j***	**s.t.**	$(Ay)_i \leq z_C$ **for all *i***
	$\Sigma_i x_i = 1$		$\Sigma_j y_j = 1$
	$x \geq 0$		$y \geq 0$

Know:

- Row gets at least z_R, and exactly z_R if Col plays equilibrium response to announced strategy (has no incentive to deviate, loses exactly $z_R = z^*$)
- Col gets at most z_C, and exactly z_C if Row plays equilibrium response to announced strategy (has no incentive to deviate, gains exactly $z_C = z^*$)

So these form an equilibrium: $z_R = z^* = z_C$, since:

- Row cannot increase gain due to Col being guaranteed max loss z_C
- Col cannot decrease loss due to Row being guaranteed min gain z_R

LP EXAMPLE: CORRELATED EQUILIBRIA FOR N PLAYERS

Recall:

- A correlated equilibrium is a distribution over pure-strategy profiles so that every player wants to follow the recommendation of the arbitrator

Variables are now p_s where s is a profile of pure strategies

- **Can enumerate! E.g., $p_{\{Row=Dodge,\ Col=Straight\}} = 0.3$**

maximize whatever you like (e.g., social welfare)

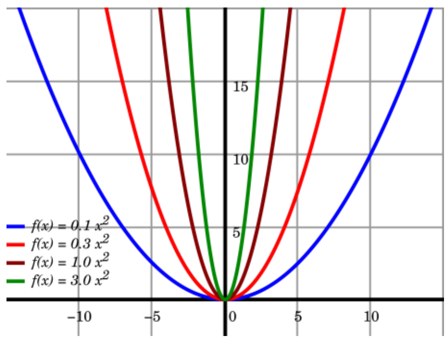
subject to

- for any i, s_i, s_i', $\Sigma_{s_{-i}}\ p_{(s_i,\ s_{-i})}\ u_i(s_i, s_{-i}) \geq \Sigma_{s_{-i}}\ p_{(s_i,\ s_{-i})}\ u_i(s_i', s_{-i})$
- $\Sigma_s\ p_s = 1$

(Minor aside: this has #variables exponential in the input; the dual just has #constraints exponential, though, so ellipsoid solves in PTIME.)

QUADRATIC PROGRAMMING

A "QP" is an optimization problem with a quadratic objective function and linear constraints.



- Quadratic functions → convex ("looks like a cup")
- Feasibility polytope also convex

Can also have quadratically-constrained QPs, etc

General objective: min/max $\mathbf{x}Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$

Sometimes these problems are easy to solve:

- If Q is positive definite, solvable in polynomial time

Sometimes they're not:

- If Q is in indefinite, the problem is non-convex and NP-hard

SO, WHAT IF WE'RE NOT CONVEX?

Global optimization problems deal with (un)constrained optimization of functions with many local optima:

- Solve to optimality?
- Try hard to find a good local optimum?

Every (non-trivial) discrete problem is non-convex:

- (Try to draw a line between two points in the feasible space.)

Combinatorial optimization: an optimization problem where at least some of the variables are discrete

- Still called "linear" if constraints are linear functions of the discrete variables, "quadratic," etc …

MODIFIED LP FROM EARLIER …

maximize **3x + 2y**

subject to

4x + 2y ≤ 15

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0

y ≥ 0

Optimal solution: x = 2.5, y = 2.5

Solution value: 7.5 + 5 = 12.5

Partial paintings …?

INTEGER (LINEAR) PROGRAM

maximize **3x + 2y**

subject to

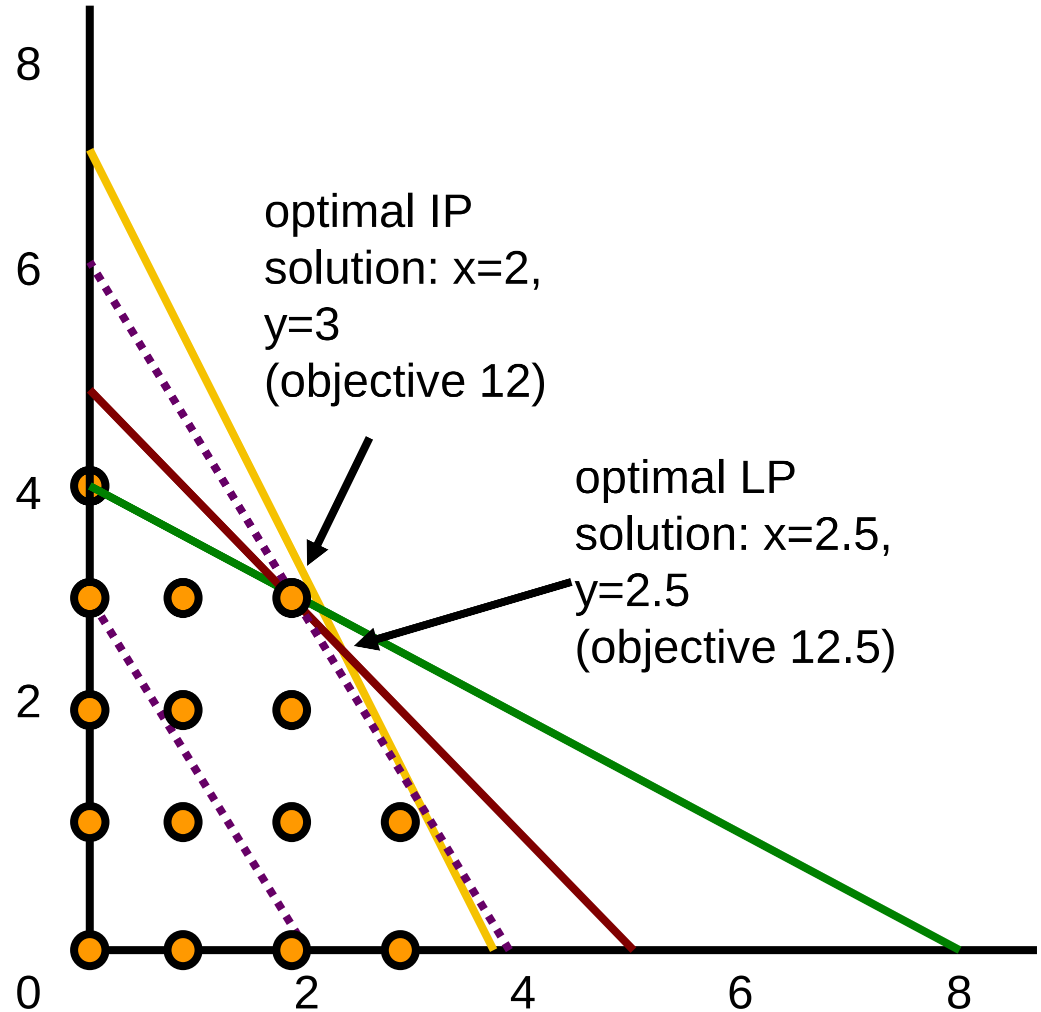
4x + 2y ≤ 15

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0, integer

y ≥ 0, integer



MIXED INTEGER (LINEAR) PROGRAM

maximize **3x + 2y**

subject to

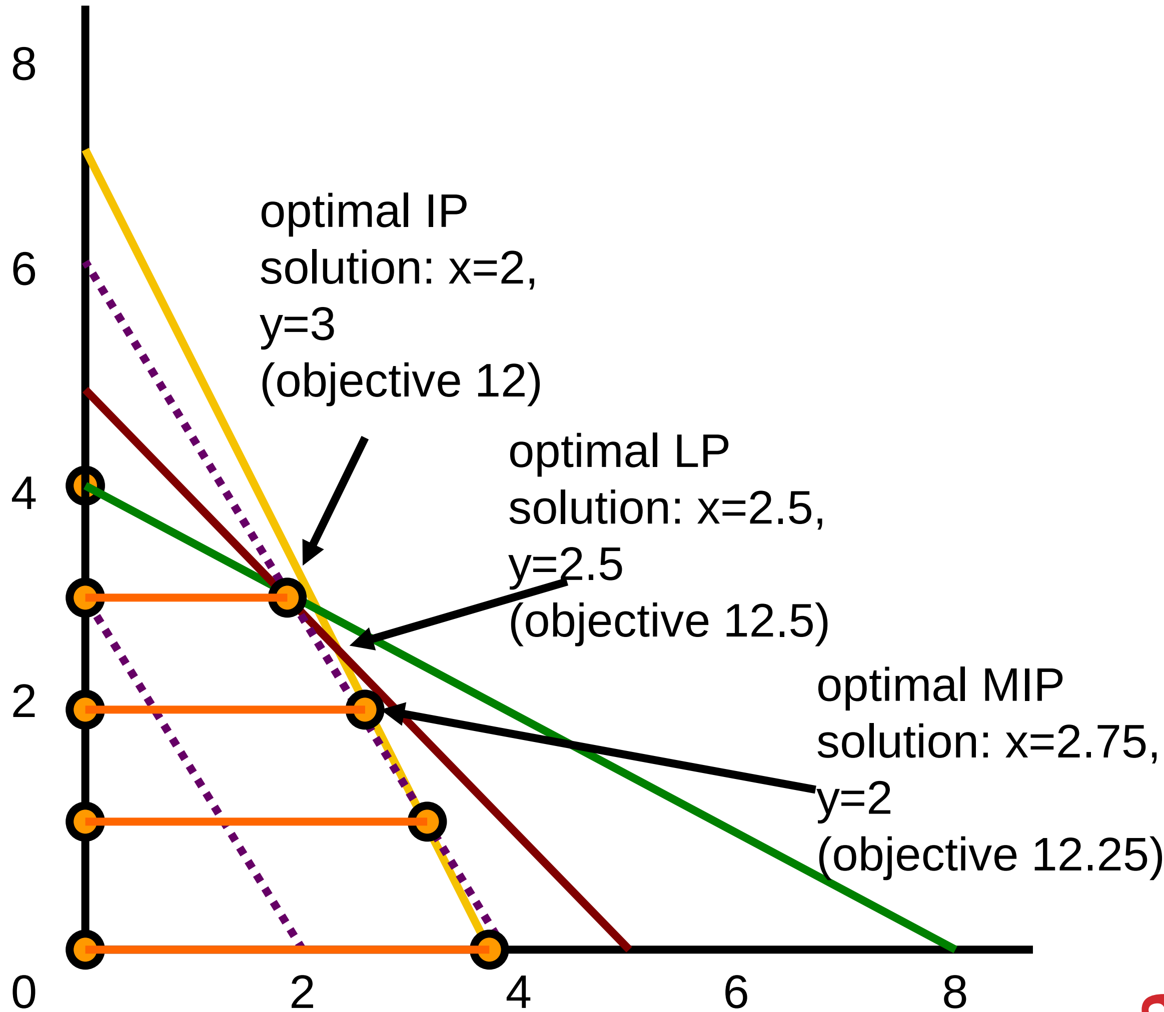
4x + 2y ≤ 15

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0

y ≥ 0, integer



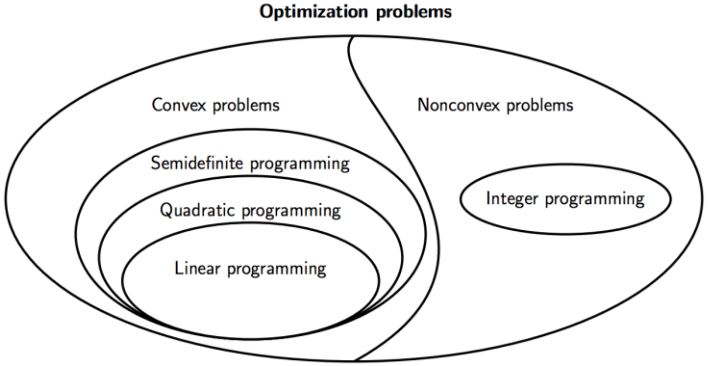
COMPLEXITY

Linear programs can be solved in polynomial time

- **If we can represent a problem as a compact LP, we can solve that problem in polynomial time**
- **2-player zero-sum Nash equilibrium computation**

General (mixed) integer programs are NP-hard to solve

- **General Nash equilibrium computation**
- **Computation of (most) Stackelberg problems**
- **Many general allocation problems**



[Thanks Zico Kolter]

LP RELAXATION, B&B

Given an IP, the LP relaxation of that IP is the same program with any integrality constraints removed.

- In a maximization problem, LP OPT $\geq$ IP OPT. Why?
- So, we can use this as a PTIME upper bound during search

Branch and bound (for maximization of binary IPs):

- **Start with no variable assignments at the root of a tree**
- **Split the search space in two by branching on a variable. First, set it to 0, see how that affects the objective:**
  - If upper bound (LPR) of branch is worse than incumbent best solution, prune this branch and backtrack (aka set var to 1)
  - Otherwise, possibly continue branching until all variables are set, or until all subtrees are pruned, or until LP = IP

Tighter LP relaxations → aggressive pruning → smaller trees

CUTTING PLANES

"Trimming down" the LP polytope – while maintaining all feasible IP points – results in tighter bounds:

- Extra linear constraints, called cuts, are valid to add if they remove no integral points

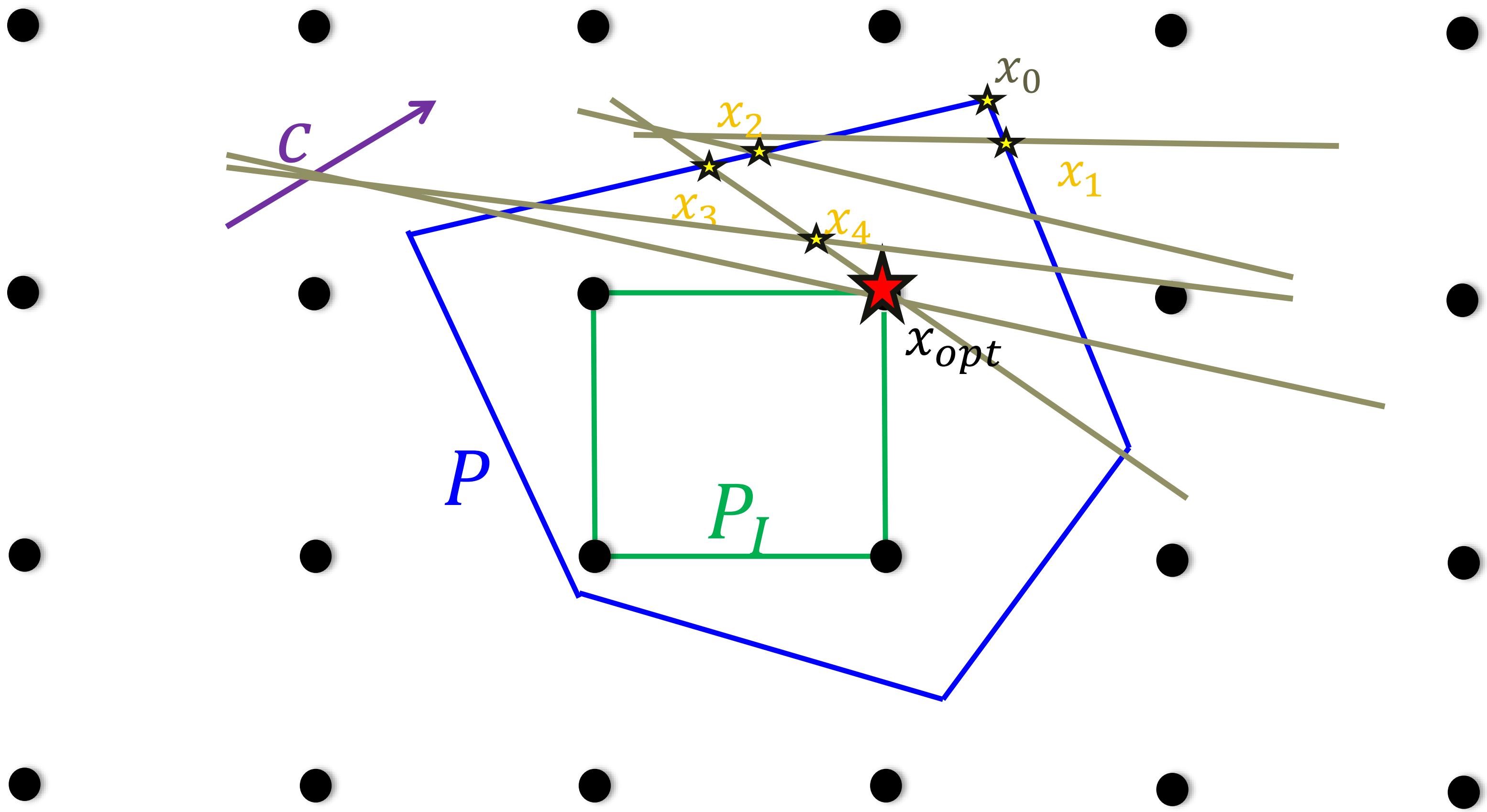
Lots of cuts! Which should we add?

Can cuts be computed quickly?

- Some families of cuts can be generated quickly
- Often just generate and test separability

Sparse coefficients?

CUTTING PLANE METHOD



$P = \{x \in R^n : Ax \leq b\}$

$P_I = \text{conv-hull}(P \cap Z^n)$

CUTTING PLANE METHOD

Starting LP. Start with the LP relaxation of the given IP to obtain basic optimal solution x

Repeat until x is integral:

- **Add Cuts.** Find a linear inequality that is valid for the convex hull of integer solutions but violated by **x** and add it to the LP
- **Re-solve LP.** Obtain basic optimal solution **x**

Can integrate into branch and bound ("branch and cut") – cuts will tighten the LP relaxation at the root or in the tree.

PRACTICAL STUFF

{CPLEX, Gurobi, SCIP, COIN-OR}:

- Variety of problems: LPs, MIPs, QPs, QCPs, CSPs, …
- CPLEX and Gurobi are for-profit, but will give free, complete copies for academic use (look up “Academic Initiative”)
- SCIP is free for non-commercial use, COIN-OR project is free-free
- Bindings for most of the languages you’d use

cvxopt:

- Fairly general convex optimization problem solver
- Lots of reasonable bindings (e.g., http://www.cvxpy.org/)

{Matlab, Mathematica, Octave}:

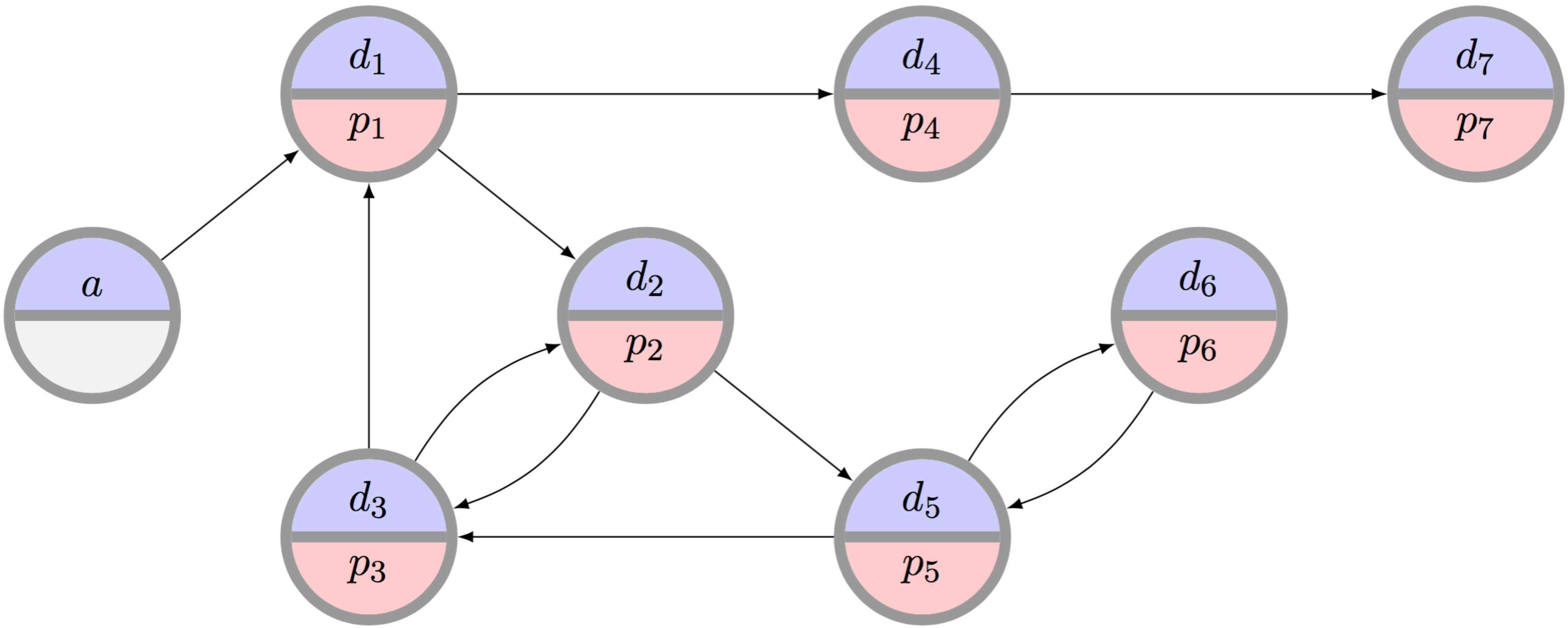
- Built in LP solvers, toolkits for pretty much everything else
- If you can hook into a specialized toolkit from here (CPLEX, cvxopt), do it

Bonmin:

- If your problem looks truly crazy – very nonlinear, but with some differentiability – look at global solvers like Bonmin

RUNNING EXAMPLE: METHODS FOR OPTIMIZING KIDNEY EXCHANGE

RECALL!



BASIC APPROACH #1: THE EDGE FORMULATION

Binary variable x_{ij} for each edge from *i* to *j*

Maximize

$u(M) = \Sigma\, w_{ij}\, x_{ij}$

Subject to

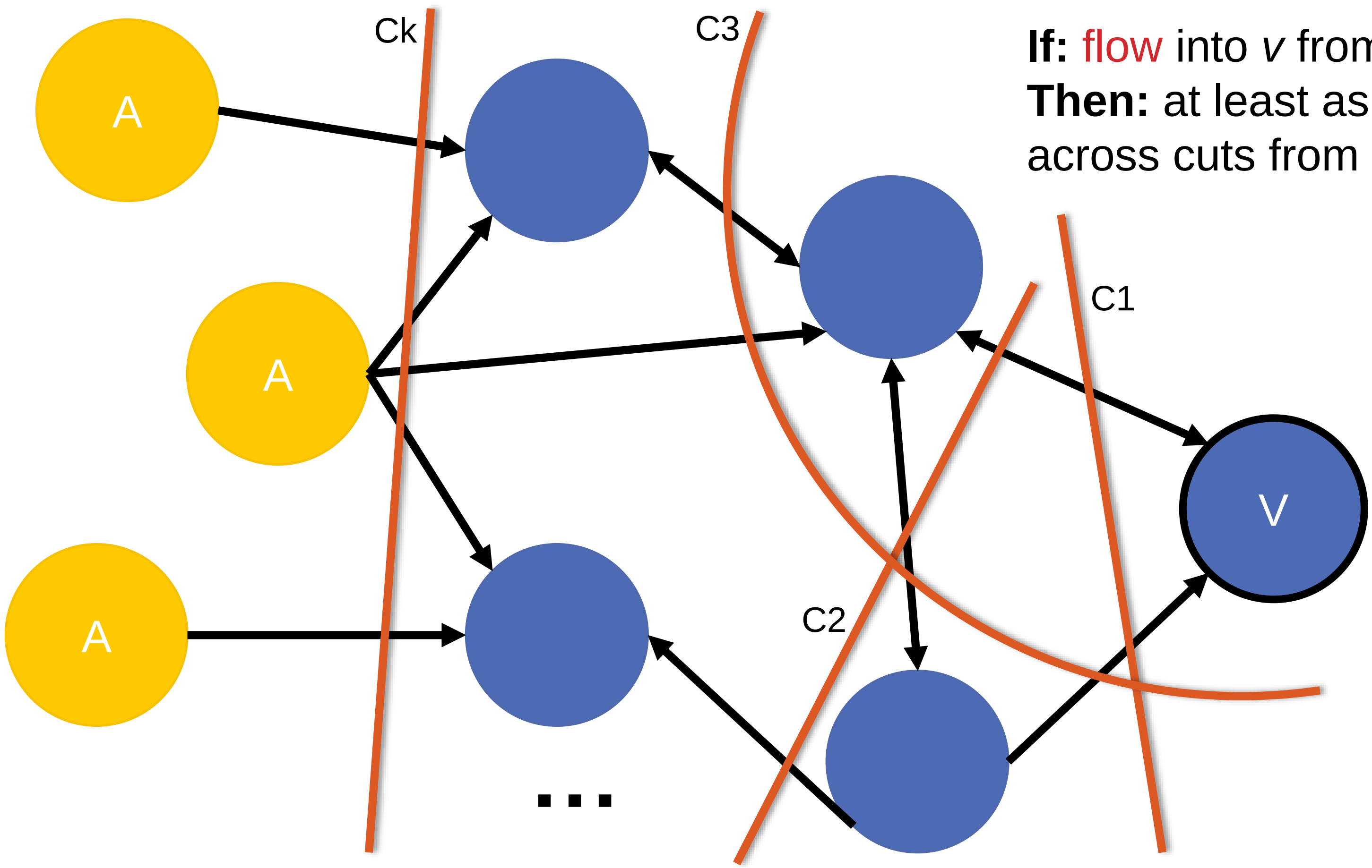
$\Sigma_j\, x_{ij} = \Sigma_j\, x_{ji}$ for each vertex *i*

$\Sigma_j\, x_{ij} \leq 1$ for each vertex *i*

$\Sigma_{1 \leq k \leq L}\, x_{i(k)i(k+1)} \leq L-1$ for paths i(1)…i(L+1)

(no path of length L that doesn't end where it started – cycle cap)

GENERATING CUTS FOR THE EDGE FORMULATION



If: flow into *v* from a chain
Then: at least as much flow across cuts from {A}

BASIC APPROACH #2: THE CYCLE FORMULATION

Binary variable x_c for each feasible cycle or chain c

Maximize

$$u(M) = \Sigma\, w_c\, x_c$$

Subject to

$$\Sigma_{c\,:\,i \text{ in } c}\, x_c \leq 1 \text{ for each vertex } i$$

A HYBRID MODEL I

In practice, cycle cap L is small and chain cap K is large

Old idea: enumerate all cycles but not all chains

- (Slide 30) required $O(|V|^K)$ constraints in the worst case
- Can reduce to $O(K|V|) = O(|V|^2)$ constraints

M
A
I
N

I
D
E
A

Track not just if an edge is used in a chain, but where in a chain an edge is used.

For edge (i,j) in graph: $K'(i,j) = \{1\}$ if i is an altruist

$K'(i,j) = \{2, \ldots, K\}$ if i is a pair

A HYBRID MODEL II

Maximize

$$u(M) = \Sigma_{ij \text{ in } E} \Sigma_{k \text{ in } K'(i,j)} w_{ij} y_{ijk} + \Sigma_{c \text{ in } C} w_c z_c$$

Subject to

$\Sigma_{ij \text{ in } E} \Sigma_{k \text{ in } K'(i,j)} y_{ijk} + \Sigma_{c\,:\,i \text{ in } c} z_c \leq 1$ for every *i* in *Pairs*

Each pair can be in at most one chain or cycle

$\Sigma_{ij \text{ in } E} y_{ij1} \leq 1$ for every *i* in *Altruists*

***Each altruist can trigger at most one chain via outgoing edge at position* 1**

$\Sigma_{j:ij \text{ in } E} y_{ijk+1} - \Sigma_{j:ji \text{ in } E \;\wedge\; k \text{ in } K'(j,i)} y_{jik} \leq 0$ for every *i* in *Pairs* and *k* in {1, …, *K*-1}

Each pair can be have an outgoing edge at position k+1 in a chain iff it has an incoming edge at position k in a chain

PRELIMINARIES & TECHNIQUES

DAS, DICKERSON, & WILDER

WHAT'S USED IN MARKET DESIGN & RESOURCE ALLOCATION?

We want the best outcome from a set of outcomes.

Convex optimization:

- Linear programming
- Quadratic programming

Nonconvex optimization:

- (Mixed) integer linear programming
- (Mixed) integer quadratic programming

Incomplete heuristic & greedy methods

Care about maximization (social welfare, profit), minimization (regret, loss), or simple feasibility (does a stable matching with couples exist?)

"PROGRAMMING?"

It's just an optimization problem**.**

Blame this guy:

- **George Dantzig (Maryland alumnus!)**
- **Focused on solving US military logistic scheduling problems aka** programs

Solving (un)constrained optimization problems is much older:

- **Newton (e.g., Newton's method for roots)**
- **Gauss (e.g., Gauss-Newton's non-linear regression)**
- **Lagrange (e.g., Lagrange multipliers)**

GENERAL MODEL

General math program:

$$\begin{aligned} &\text{min/max} && f(\boldsymbol{x}) \\ &\text{subject to} && g_i(\mathbf{x}) \le 0, \quad \mathrm{i} = 1, \ldots, m \\ & && h_j(\mathbf{x}) = 0, \quad \mathrm{j} = 1, \ldots, k \\ & && \mathbf{x} \in X \subset \Re^{\mathrm{n}} \\ & && f, g_i, h_j : \Re^n \rightarrow \Re \end{aligned}$$

Linear programming: all of f, g_i, h_j are linear (affine) functions

Nonlinear programming: at least part of f, g_i, h_j is nonlinear

Integer programming: Feasible region constrained to integers

Convex, quadratic, etc …

CONVEX FUNCTIONS

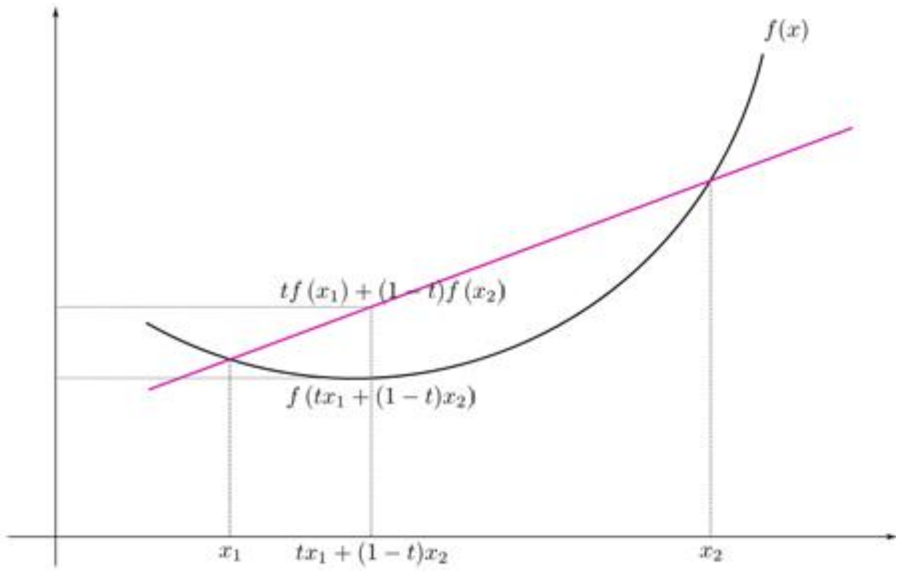
"A function is convex if the line segment between any two points on its graph lies above it."

Formally, given function *f* and two points x, y:

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}) \quad \forall \lambda \in [0,1]$$

Convex or non-convex?

- $\mathbf{a}^T \mathbf{x} + b$
- e^x, e^{-x}
- $\mathbf{x}^T \mathbf{Q} \mathbf{x}, \quad \mathbf{Q} \succeq \mathbf{0}$
- $\mathbf{x}^T \mathbf{Q} \mathbf{x}, \quad \mathbf{Q}$ indefinite
- $\|\mathbf{x}\|$
- $\log x, \ \sqrt{x}$



CONVEX SETS

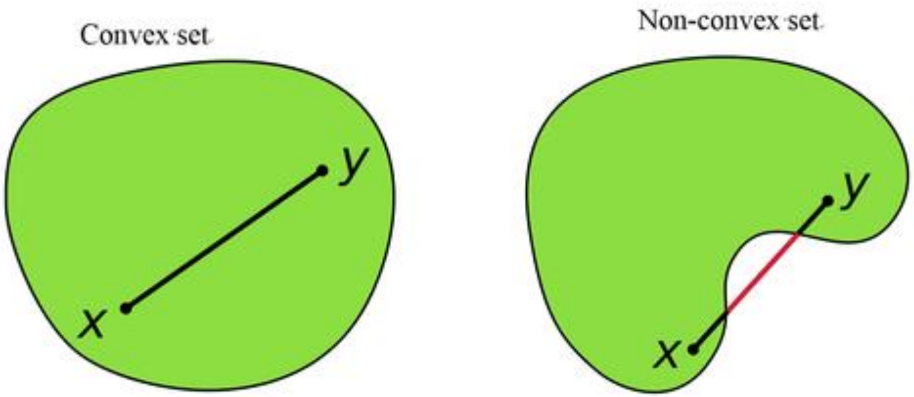
"A set is convex if, for every pair of points within the set, every point on the straight line segment that joins them is in the set."

Formally, give a set *S* and two points x, y in S:

$$\mathbf{x} \in S, \mathbf{y} \in S \Rightarrow \lambda \mathbf{x} + (1-\lambda)\mathbf{y} \in S$$

Convex or non-convex sets?

- $\{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$
- $\mathbb{R}^n_+$
- $\{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}\}$
- $\{(\mathbf{x}, t) : \|\mathbf{x}\| \leq t\}$



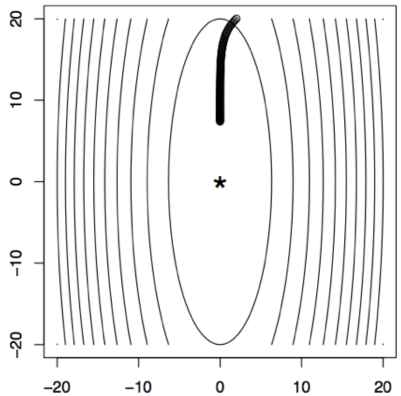
SO WHAT?

An optimization (minimization) problem with a convex objective function and a convex feasible region is solved via convex programming.

Lets us use tools from convex analysis

- Local minima are global minima
- The set of global mimina is convex
- There is a unique global minimum if strictly convex

Lets us make statements like gradient descent converges to a global minimum (under some assumptions w.r.t local Lipschitz and step size)

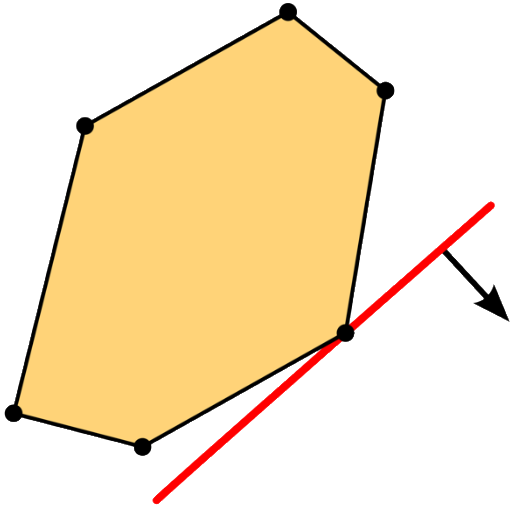


But let's start even simpler …

LINEAR PROGRAMS

An "LP" is an optimization problem with a linear objective function and linear constraints.

- **A line drawn between any two points *x*, *y* on a line is on the line → clearly convex**
- **Feasible region aka polytope also convex**



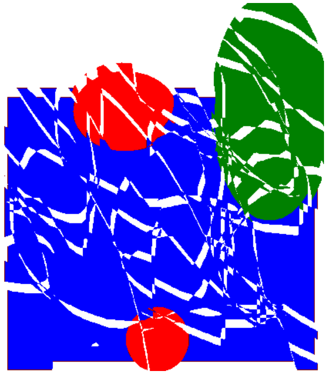
General LP:

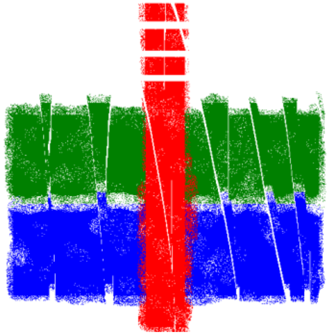
$$\text{min/max} \quad \mathbf{c}^T\mathbf{x}$$
$$\text{subject to} \quad A\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq 0$$

Where c, *A*, b are known, and we are solving for x.

LP: EXAMPLE

We make reproductions of two paintings:





Painting 1 sells for $30, painting 2 sells for $20

Painting 1 requires 4 units of blue, 1 green, 1 red

Painting 2 requires 2 blue, 2 green, 1 red

We have 16 units blue, 8 green, 5 red

maximize $3x + 2y$

subject to

$4x + 2y \leq 16$

$x + 2y \leq 8$

$x + y \leq 5$

$x \geq 0$

$y \geq 0$

Objective ???????

Constraints ???????

SOLVING THE LINEAR PROGRAM GRAPHICALLY

maximize **3x + 2y**

subject to

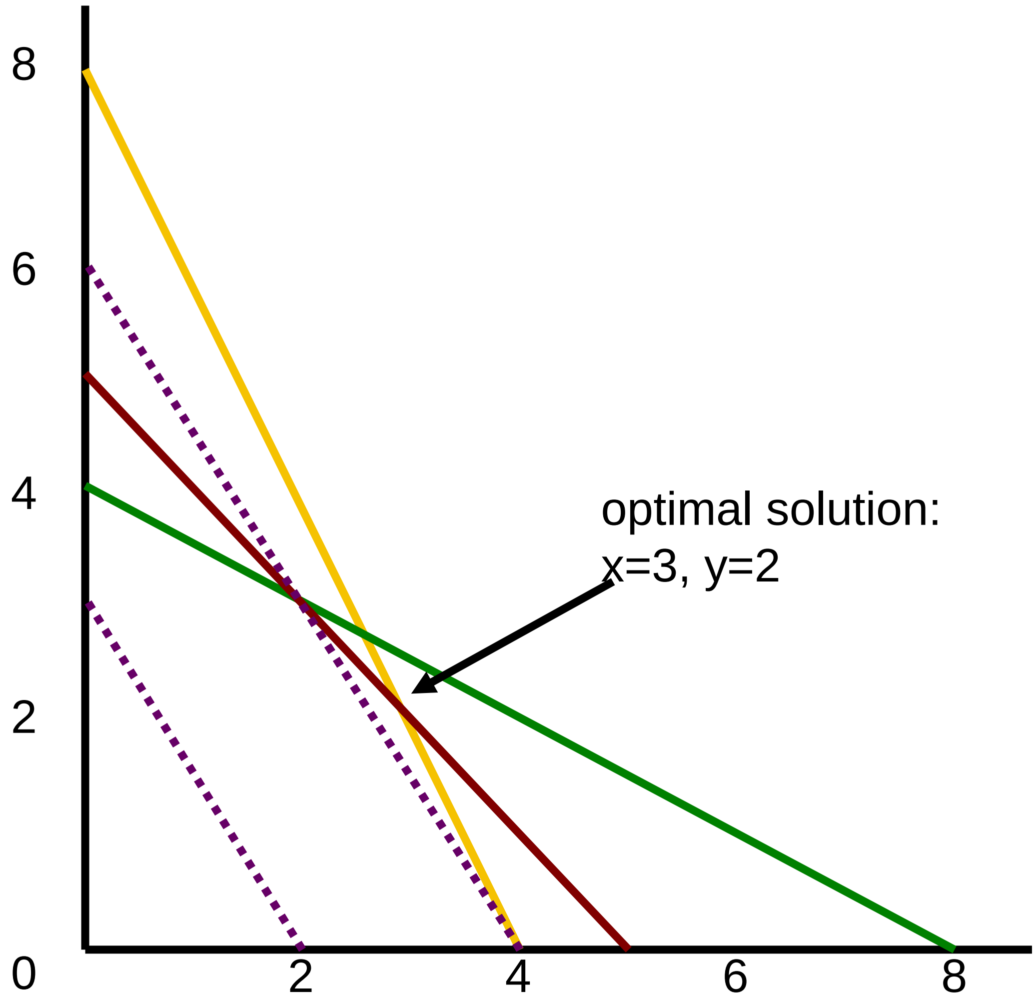
4x + 2y ≤ 16

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0

y ≥ 0



LP EXAMPLE: SOLVING FOR 2-P ZERO-SUM NASH

Recall:

- Mixed Nash Equilibrium always exists
- Even if I know your strategy, in equilibrium I don't deviate

Given a payoff matrix A:

	Morality	Tax-Cuts
Economy	+3, -3	-1, +1
Society	-2, +2	+1, -1

[Example from Daskalakis]

If Row announces strategy $<x_1, x_2>$, then Col gets expected payoffs:

E["Morality"] = $-3x_1 + 2x_2$

E["Tax-Cuts"] = $1x_1 - 1x_2$

So Col will best respond with max($-3x_1 + 2x_2$, $1x_1 - 1x_2$) …

LP EXAMPLE: SOLVING FOR 2-P ZERO-SUM NASH

But if Col gets max(-3x_1 + 2x_2, 1x_1 – 1x_2),
then Row gets -max(-3x_1 + 2x_2, 1x_1 – 1x_2) = min(…)

So, if Row must announce, she will choose the strategy:

<x1, x2> = arg max min(3x_1 - 2x_2, -1x_1 + 1x_2)

This is just an LP:

maximize z

such that 3x_1 - 2x_2 $\geq$ z

-1x_1 + 1x_2 $\geq$ z

$x_1 + x_2 = 1$

$x_1, x_2 \geq 0$

So Row player is guaranteed to get at least z

LP EXAMPLE: SOLVING FOR 2-P ZERO-SUM NASH

Can set up the same LP for the Col player, to get general LPs:

max	z_R	**min**	z_C
s.t.	$(xA)_j \geq z_R$ **for all** ***j***	**s.t.**	$(Ay)_i \leq z_C$ **for all** ***i***
	$\Sigma_i \, x_i = 1$		$\Sigma_j \, y_j = 1$
	$x \geq 0$		$y \geq 0$

Know:

- Row gets at least z_R, and exactly z_R if Col plays equilibrium response to announced strategy (has no incentive to deviate, loses exactly $z_R = z^*$)
- Col gets at most z_C, and exactly z_C if Row plays equilibrium response to announced strategy (has no incentive to deviate, gains exactly $z_C = z^*$)

So these form an equilibrium: $z_R = z^* = z_C$, since:

- Row cannot increase gain due to Col being guaranteed max loss z_C
- Col cannot decrease loss due to Row being guaranteed min gain z_R

LP EXAMPLE: CORRELATED EQUILIBRIA FOR N PLAYERS

Recall:

- A correlated equilibrium is a distribution over pure-strategy profiles so that every player wants to follow the recommendation of the arbitrator

Variables are now p_s where s is a profile of pure strategies

- **Can enumerate! E.g., $p_{\{Row=Dodge,\ Col=Straight\}} = 0.3$**

maximize whatever you like (e.g., social welfare)

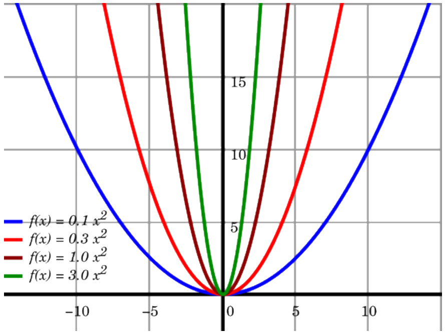
subject to

- for any i, s_i, s_i', $\Sigma_{s_{-i}}\ p_{(s_i,\ s_{-i})}\ u_i(s_i, s_{-i}) \geq \Sigma_{s_{-i}}\ p_{(s_i,\ s_{-i})}\ u_i(s_i', s_{-i})$
- $\Sigma_s\ p_s = 1$

(Minor aside: this has #variables exponential in the input; the dual just has #constraints exponential, though, so ellipsoid solves in PTIME.)

QUADRATIC PROGRAMMING

A "QP" is an optimization problem with a quadratic objective function and linear constraints.



- Quadratic functions → convex ("looks like a cup")
- Feasibility polytope also convex

Can also have quadratically-constrained QPs, etc

General objective: min/max $\mathbf{x}Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$

Sometimes these problems are easy to solve:

- If Q is positive definite, solvable in polynomial time

Sometimes they're not:

- If Q is in indefinite, the problem is non-convex and NP-hard

SO, WHAT IF WE'RE NOT CONVEX?

Global optimization problems deal with (un)constrained optimization of functions with many local optima:

- Solve to optimality?
- Try hard to find a good local optimum?

Every (non-trivial) discrete problem is non-convex:

- (Try to draw a line between two points in the feasible space.)

Combinatorial optimization: an optimization problem where at least some of the variables are discrete

- Still called "linear" if constraints are linear functions of the discrete variables, "quadratic," etc …

MODIFIED LP FROM EARLIER …

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: $7.5 + 5 = 12.5$

Partial paintings …?

INTEGER (LINEAR) PROGRAM

maximize **3x + 2y**

subject to

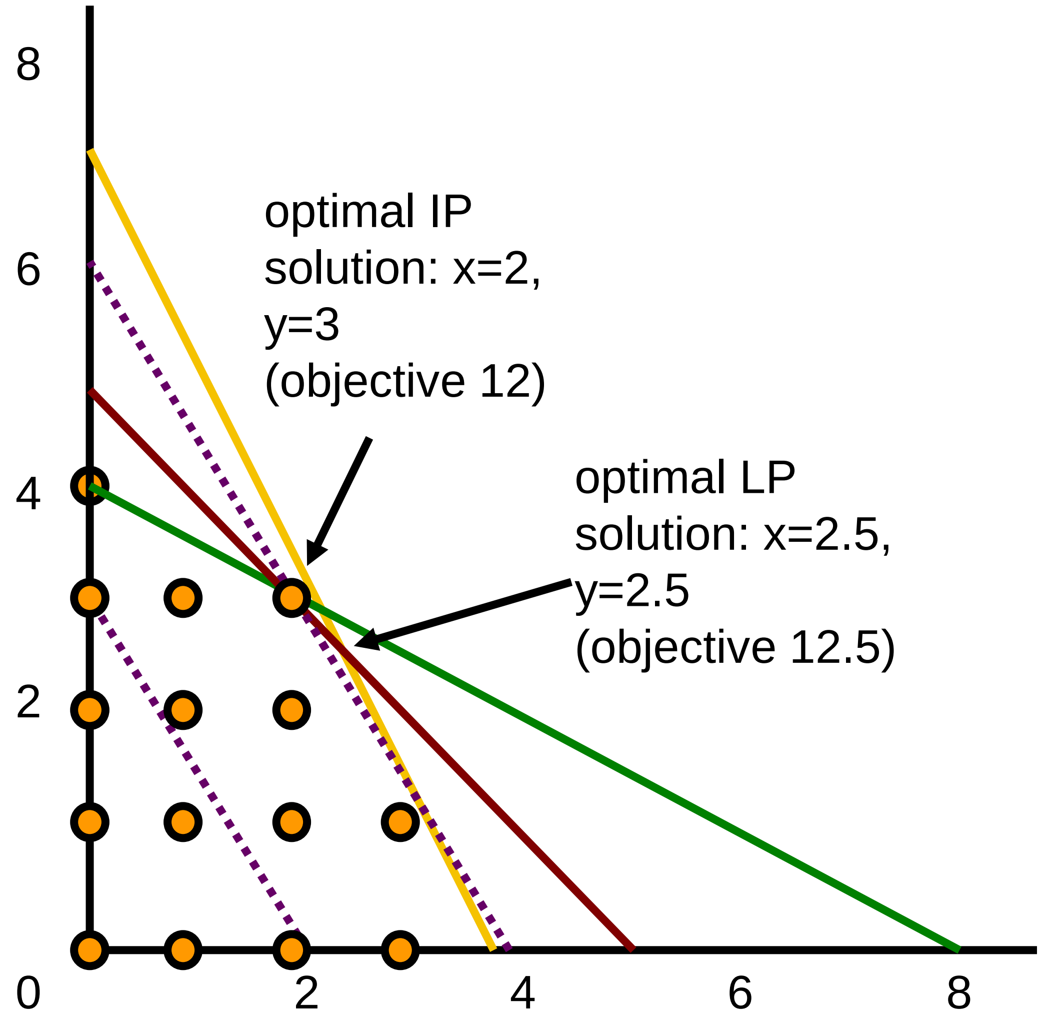
4x + 2y ≤ 15

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0, integer

y ≥ 0, integer



MIXED INTEGER (LINEAR) PROGRAM

maximize **3x + 2y**

subject to

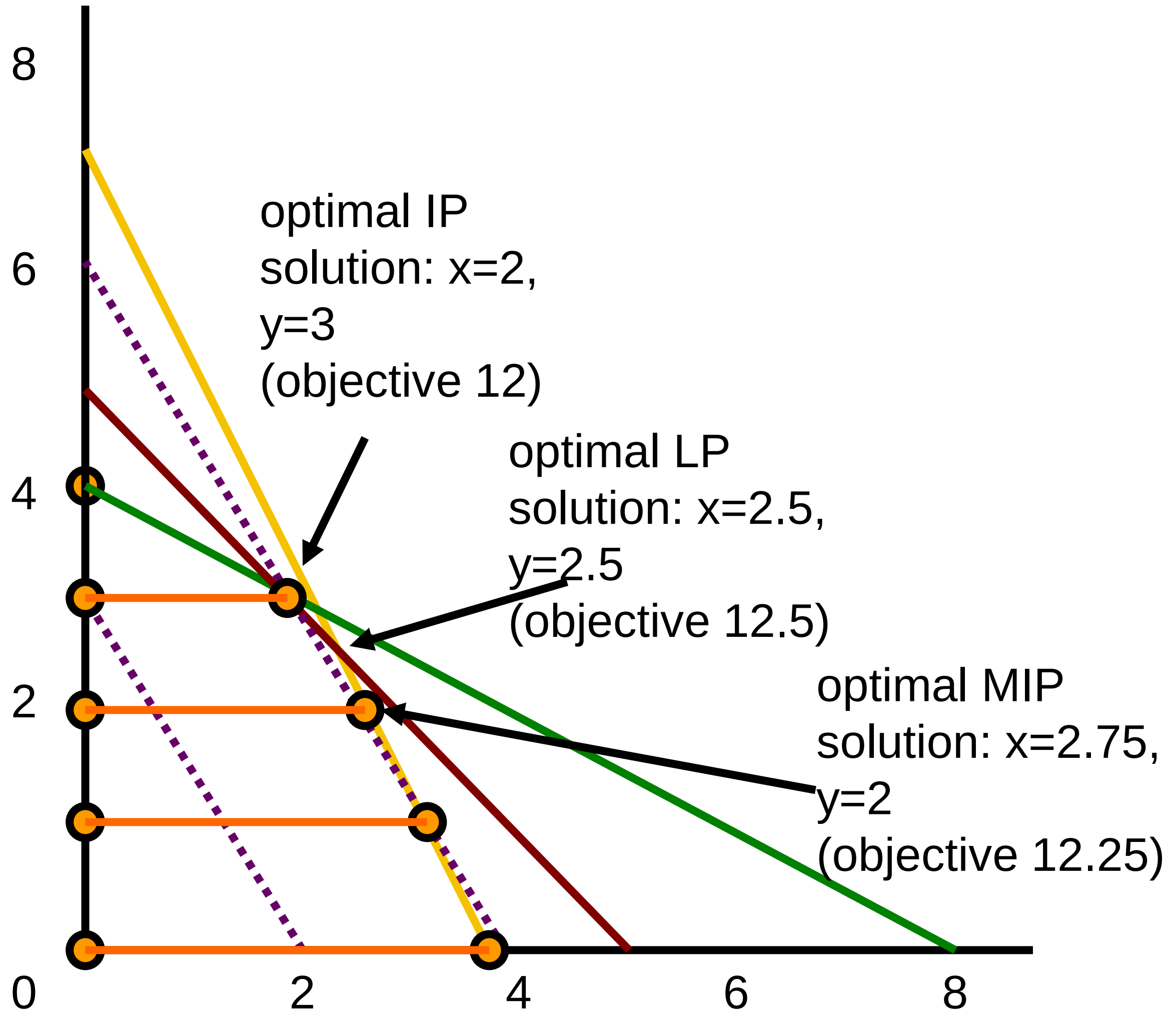
4x + 2y ≤ 15

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0

y ≥ 0, integer



VC

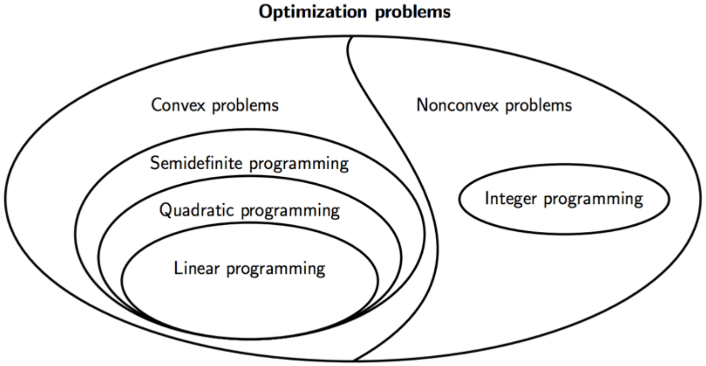
COMPLEXITY

Linear programs can be solved in polynomial time

- **If we can represent a problem as a compact LP, we can solve that problem in polynomial time**
- **2-player zero-sum Nash equilibrium computation**

General (mixed) integer programs are NP-hard to solve

- **General Nash equilibrium computation**
- **Computation of (most) Stackelberg problems**
- **Many general allocation problems**



[Thanks Zico Kolter]

LP RELAXATION, B&B

Given an IP, the LP relaxation of that IP is the same program with any integrality constraints removed.

- In a maximization problem, LP OPT $\geq$ IP OPT. Why?
- So, we can use this as a PTIME upper bound during search

Branch and bound (for maximization of binary IPs):

- **Start with no variable assignments at the root of a tree**
- **Split the search space in two by branching on a variable. First, set it to 0, see how that affects the objective:**
  - If upper bound (LPR) of branch is worse than incumbent best solution, prune this branch and backtrack (aka set var to 1)
  - Otherwise, possibly continue branching until all variables are set, or until all subtrees are pruned, or until LP = IP

Tighter LP relaxations → aggressive pruning → smaller trees

CUTTING PLANES

“Trimming down” the LP polytope – while maintaining all feasible IP points – results in tighter bounds:

- Extra linear constraints, called cuts, are valid to add if they remove no integral points

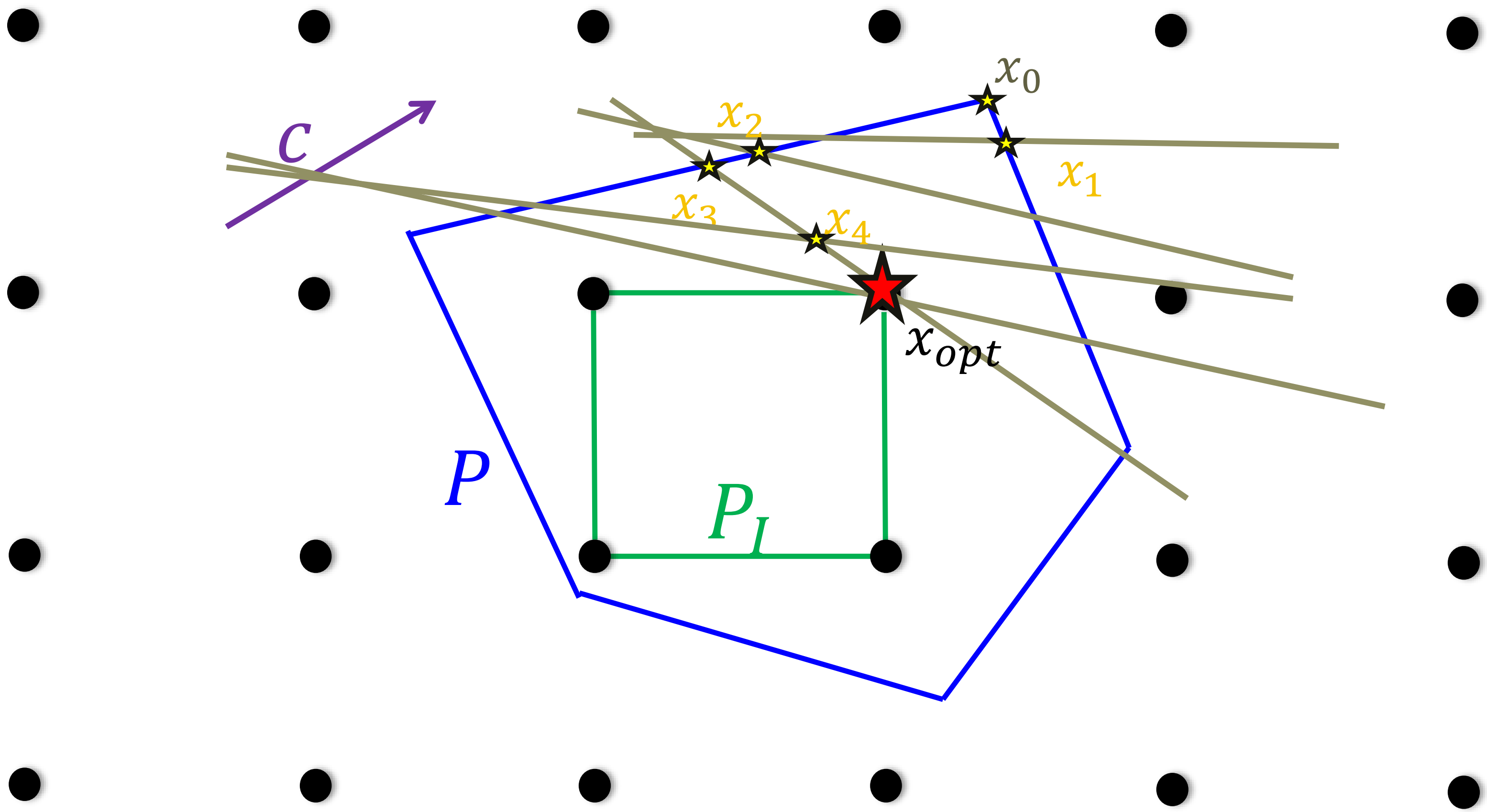
Lots of cuts! Which should we add?

Can cuts be computed quickly?

- Some families of cuts can be generated quickly
- Often just generate and test separability

Sparse coefficients?

CUTTING PLANE METHOD



$P = \{x \in R^n : Ax \leq b\}$

$P_I = \text{ conv-hull}(P \cap Z^n)$

CUTTING PLANE METHOD

Starting LP. Start with the LP relaxation of the given IP to obtain basic optimal solution x

Repeat until x is integral:

- **Add Cuts.** Find a linear inequality that is valid for the convex hull of integer solutions but violated by **x** and add it to the LP
- **Re-solve LP.** Obtain basic optimal solution **x**

Can integrate into branch and bound ("branch and cut") – cuts will tighten the LP relaxation at the root or in the tree.

PRACTICAL STUFF

{CPLEX, Gurobi, SCIP, COIN-OR}:

- Variety of problems: LPs, MIPs, QPs, QCPs, CSPs, …
- CPLEX and Gurobi are for-profit, but will give free, complete copies for academic use (look up "Academic Initiative")
- SCIP is free for non-commercial use, COIN-OR project is free-free
- Bindings for most of the languages you'd use

cvxopt:

- Fairly general convex optimization problem solver
- Lots of reasonable bindings (e.g., http://www.cvxpy.org/)

{Matlab, Mathematica, Octave}:

- Built in LP solvers, toolkits for pretty much everything else
- If you can hook into a specialized toolkit from here (CPLEX, cvxopt), do it

Bonmin:

- If your problem looks truly crazy – very nonlinear, but with some differentiability – look at global solvers like Bonmin

Social networks

Applications

- Widely used in preventative health and other fields
- Substance abuse, microfinance adoption, HIV prevention, childhood nutrition, smoking prevention, cancer screening…

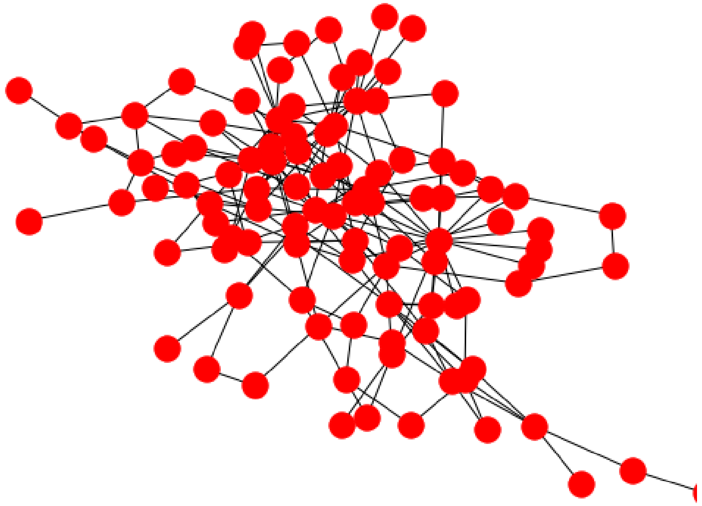
Example: HIV and homelessness

- 6,000 homeless youth
- 10x HIV prevalence vs general population

Mayor Eric Garcetti: "the moral and humanitarian crisis of our time"

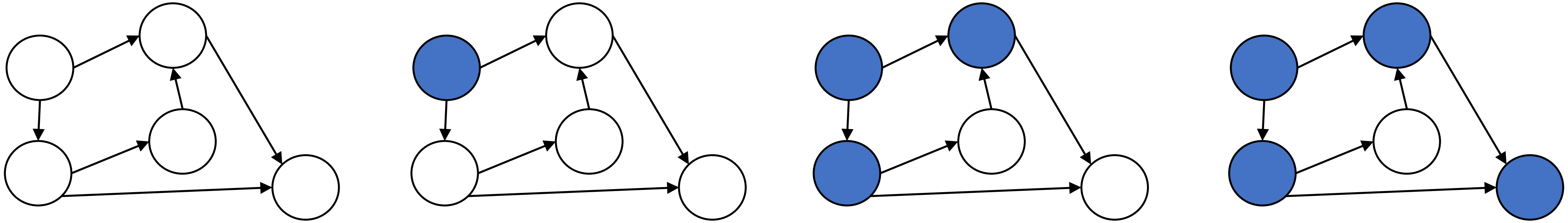
Example: HIV and homelessness

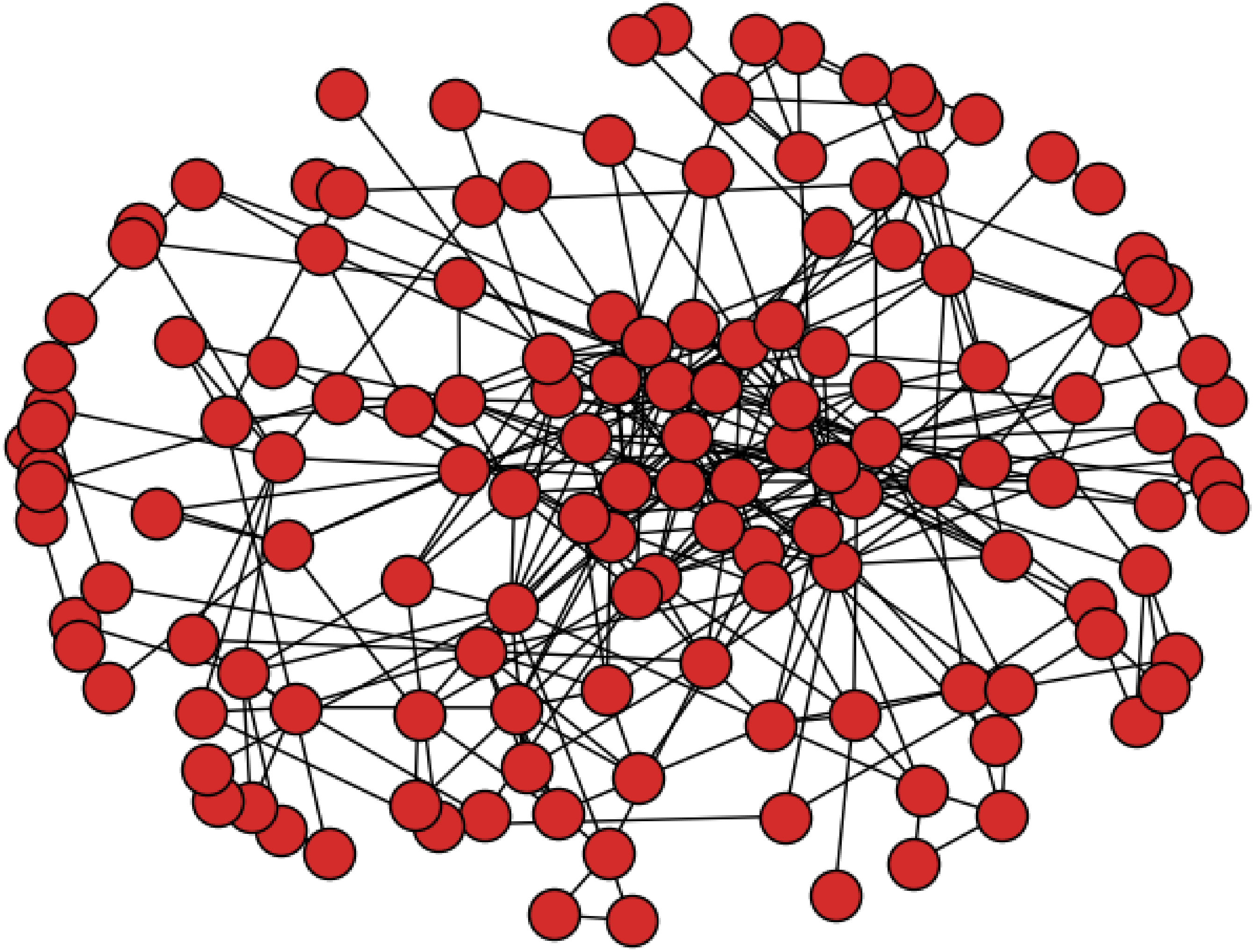
- Shelters conduct educational interventions
- Resource constraints: work with 4-6 youth at a time
- *Peer leaders*: spread message through social network

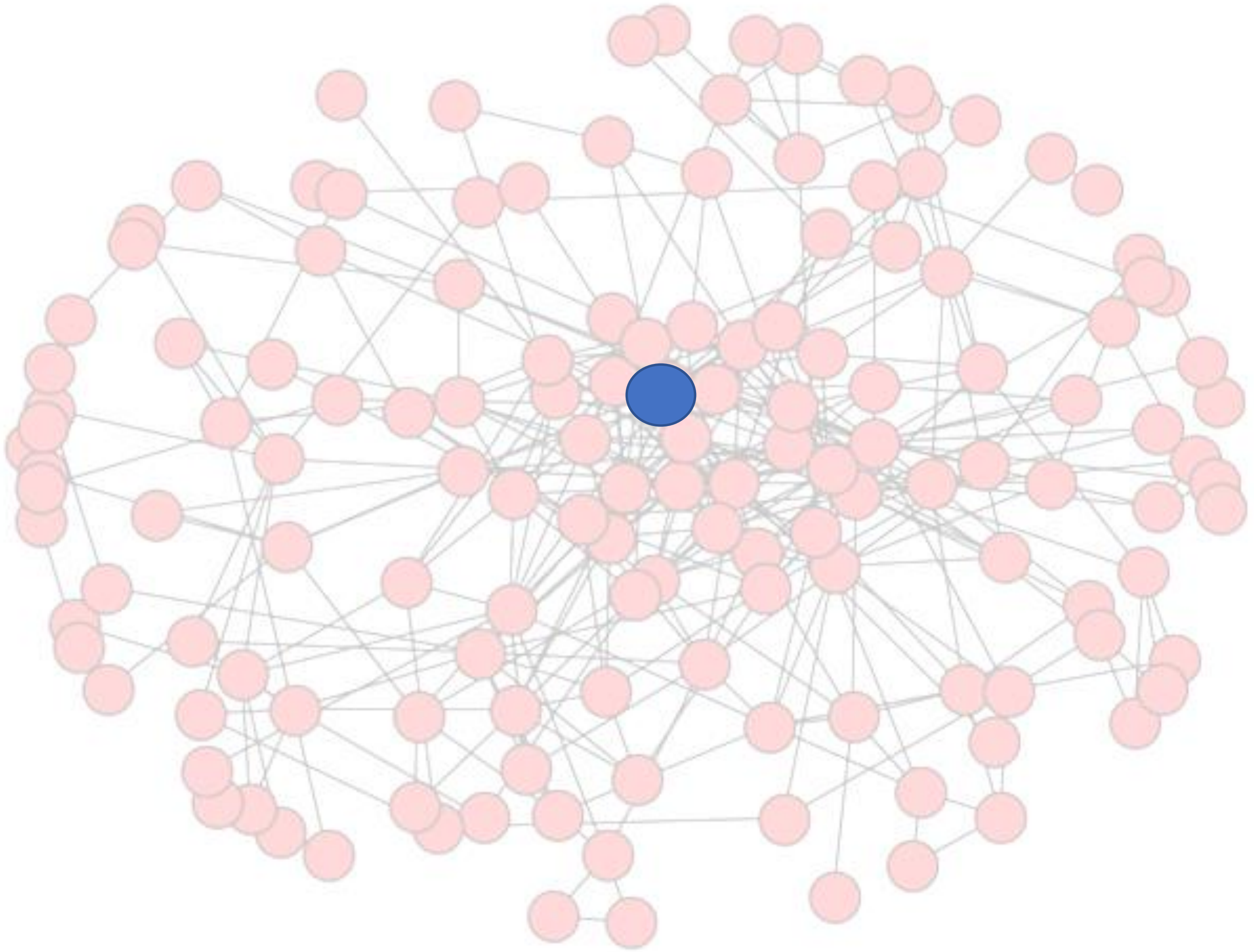


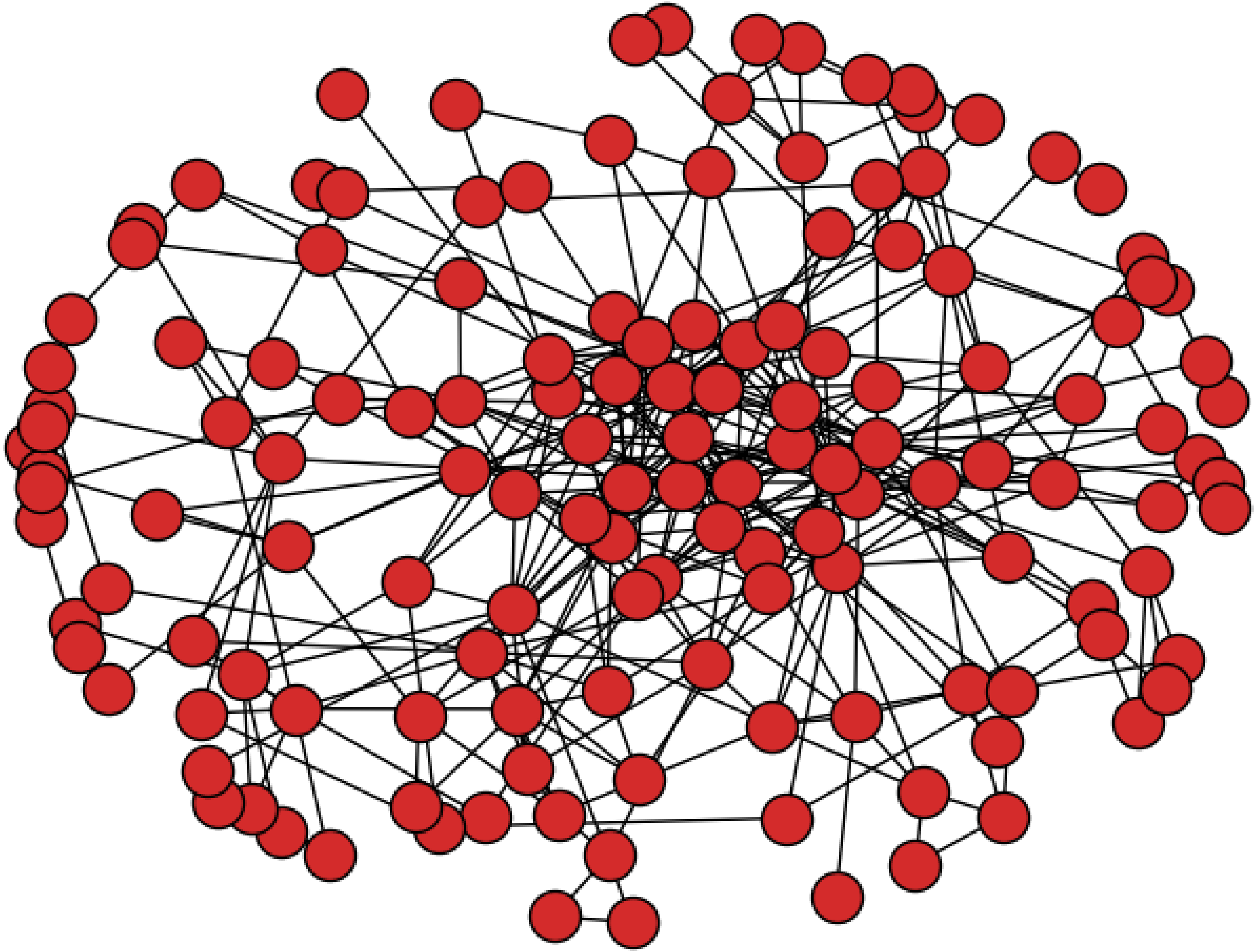
Example: HIV and homelessness

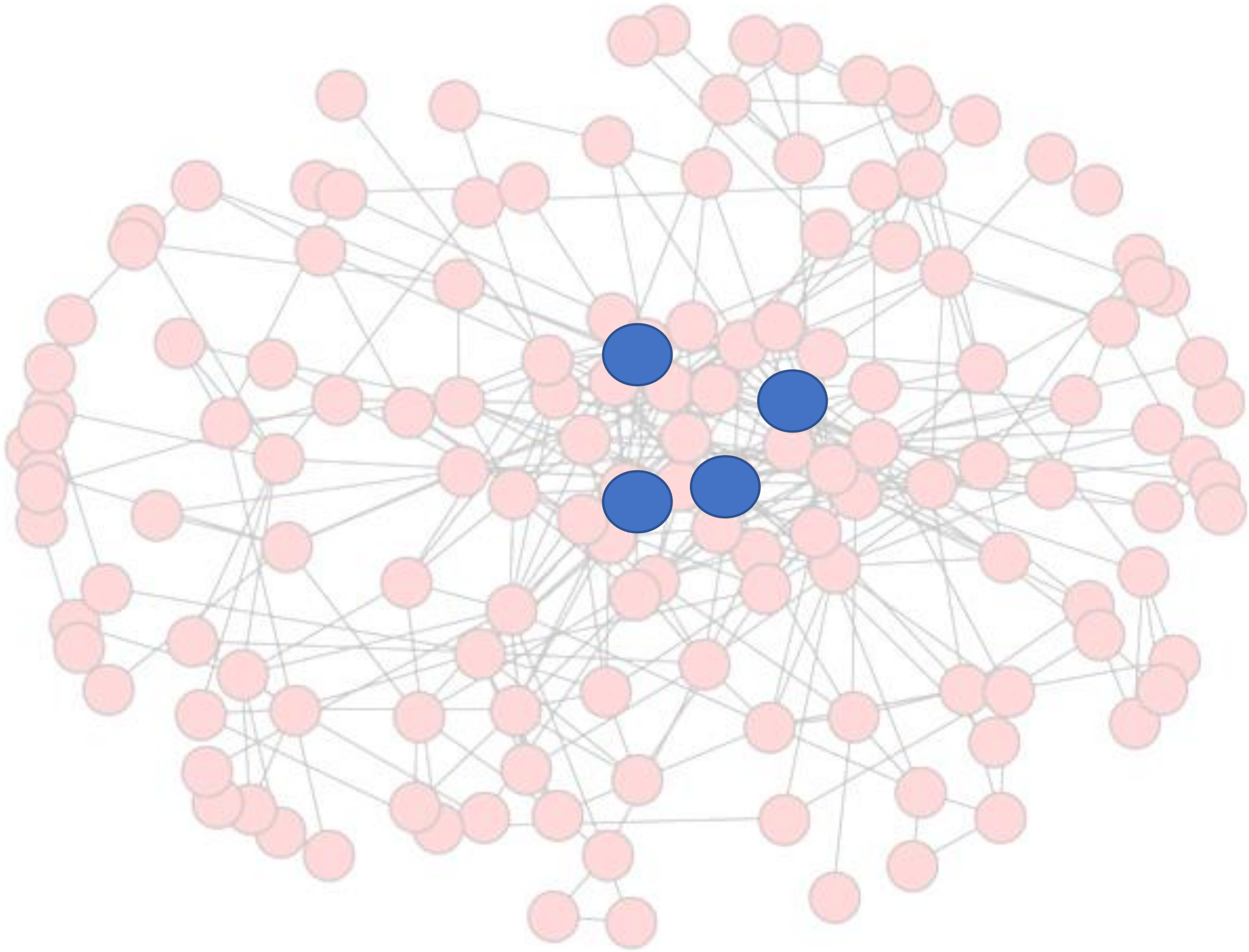
- Limited budget for total peer leaders trained
- Which nodes lead to greatest influence spread?
- Influence maximization problem





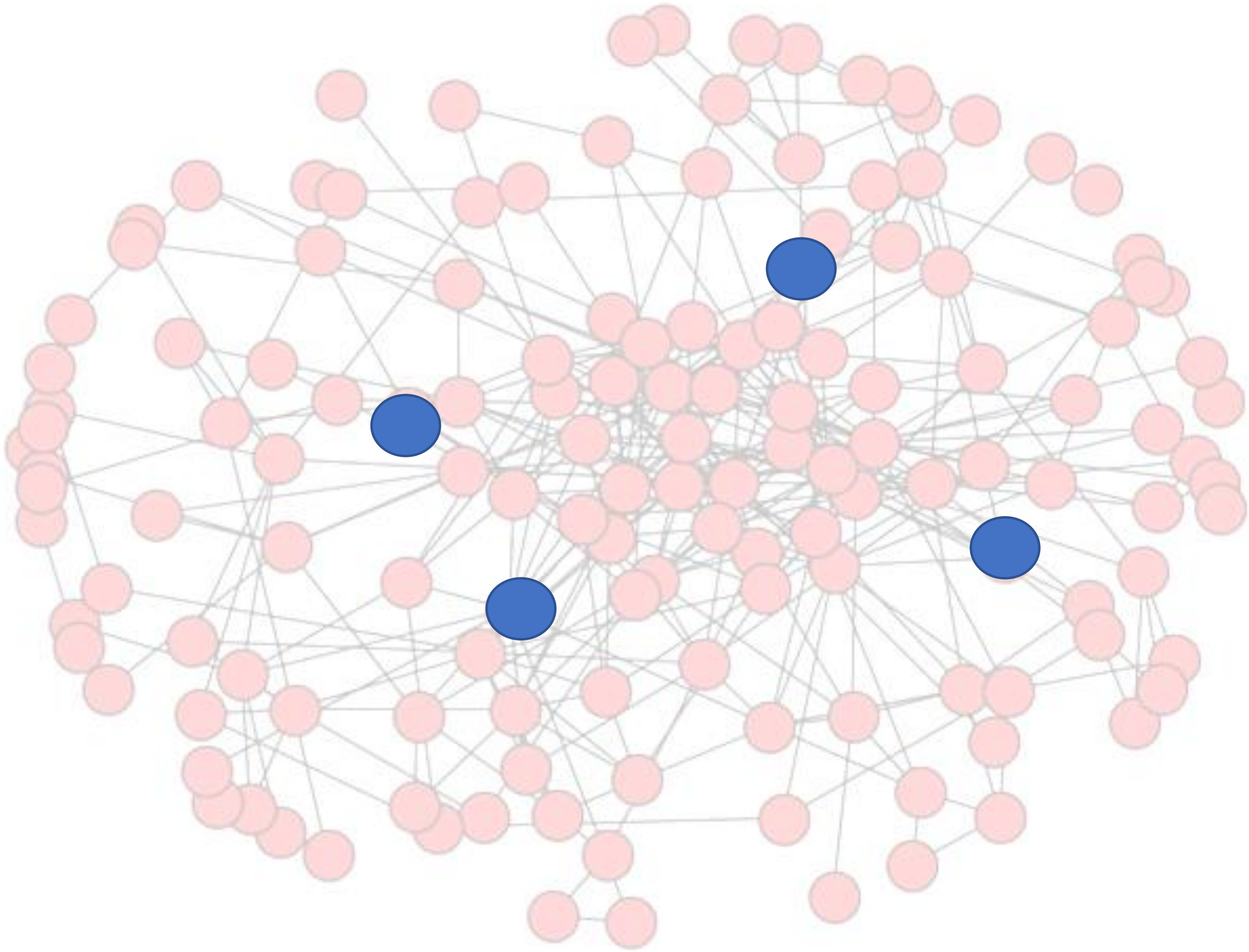




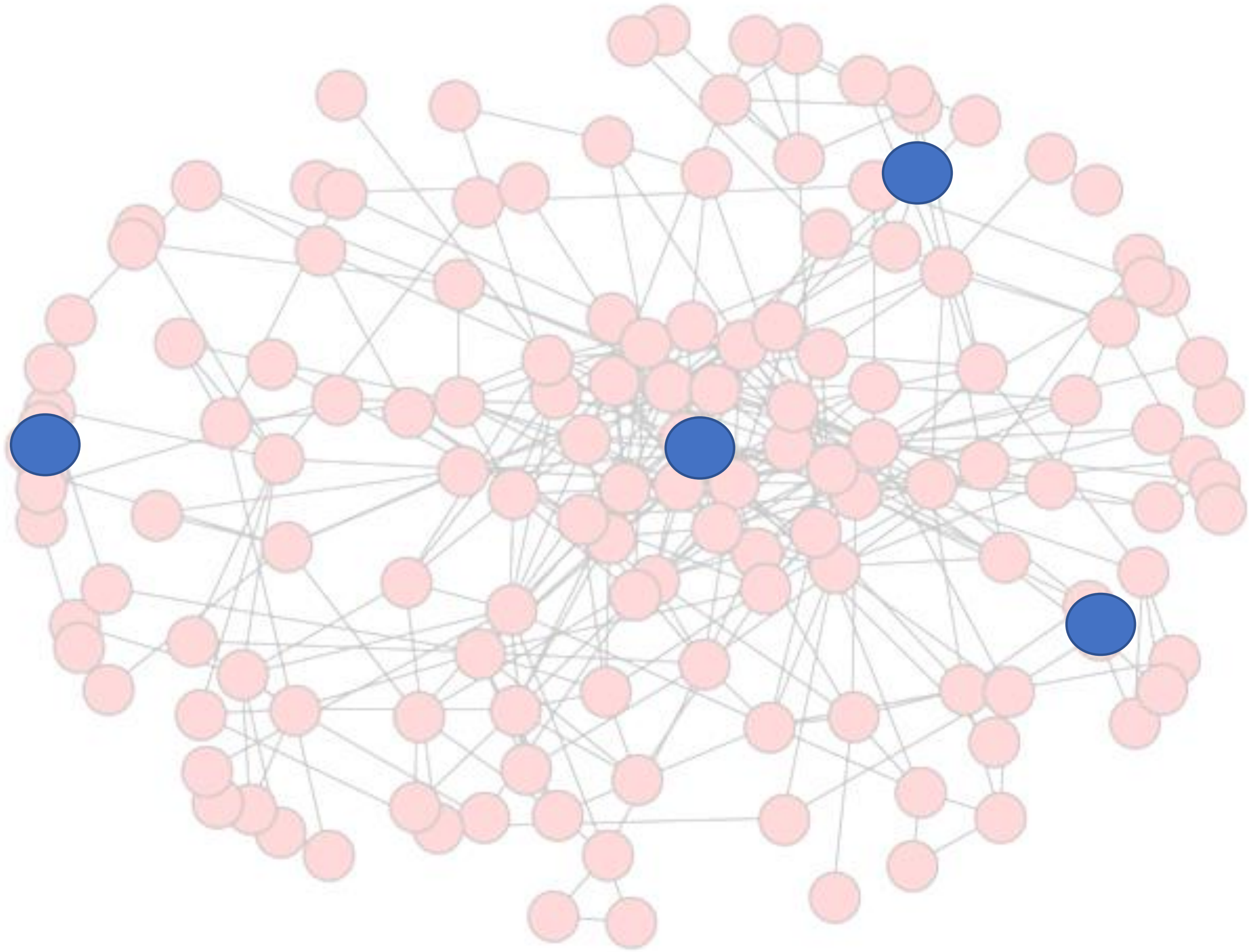


Central/popular nodes?

(degree: # of connections)



"Bridge" nodes?



A mix?

Computational problem

- Limited budget of seed nodes to recruit from a graph $G = (V, E)$
- For $S \subseteq V$, let $f(S)$ be the expected number of nodes reached when S is recruited as seeds
- Problem:

$$\max_{|S| \leq k} f(S)$$

Models of influence spread

- Where does f come from?
- Need some theory about how influence spreads on a network
- Many different models, appropriate for different situations

Independent cascade model

- Most common model in the literature
- Each edge (u, v) has a propagation probability $p_{u,v}$
- When u is influenced, v is influenced w.p. $p_{u,v}$
- All activations are independent

Linear threshold model

- Also common
- Each node v draws a threshold $t_v \sim U[0,1]$
- Each edge (u, v) has a weight $w_{u,v}$; $\sum_{u \to v} w_{u,v} = 1$
- v activates when total weight of activated neighbors exceeds t_v

Non-progressive models

- ICM/LTM: once activated, stay activated
- Makes sense for information diffusion
- Sometimes, want to model behavior that can “relapse”
  - E.g., obesity-interventions

Non-progressive models

- Voter model:
  - Each node has discrete state $x_v \in \{0,1\}$
  - At each step, each node copies a random neighbor
- DeGroot model:
  - Each node has a continuous state $x_v \in [0,1]$
  - At each step, take the average of its neighbors
- These amount to same thing: long-run behavior governed by eigenvalues of adjacency matrix

Non-progressive models

- Here: focus on progressive models (ICM/LTM)
- Motivation: information diffusion (awareness/education)

Computational problem

- Limited budget of seed nodes to recruit from a graph $G = (V, E)$
- For $S \subseteq V$, let $f(S)$ be the expected number of nodes reached when S is recruited as seeds
- Problem:

$$\max_{|S| \leq k} f(S)$$

How to solve?

Submodular optimization

Optimizing set functions

- Particular kind of combinatorial optimization problem
- Ground set of items V
- Choose a subset S
- Objective: $f(S)$
- Constraints: $S \in I, I \subseteq 2^V$
  - E.g., $|S| \leq k$

Optimizing set functions

- Without any additional structure, clearly impossible (NP-hard to do anything)
- Can probably encode as a MIP, but solving may be intractable
- What if objective function f and feasible set I have nice structure?
- Discrete equivalent of convexity?

Key property: submodularity

- Property of set functions which enables efficient optimization
- *Diminishing returns:*

$$f(A \cup \{v\}) - f(A) \leq f(B \cup \{v\}) - f(B) \quad \forall v, \qquad B \subseteq A$$

- Sometimes, also assume monotone: $f(A \cup \{v\}) - f(A) \geq 0$

Key property: submodularity

- When f is submodular, many optimization problems become tractable
- For “nice” constraint families, like budget constraint
  - More generally, matroid constraints

Submodular optimization: greedy

- Simplest possible algorithm

$S = \emptyset$

while $|S| \leq k$:

$$v^* = \arg\max_{v \notin S} f(S \cup \{v\}) - f(S)$$

$$S = S \cup \{v^*\}$$

Submodular optimization: greedy

- Simplest possible algorithm
- Bottleneck: evaluating f
- Some tricks to speed up
  - "Accelerated/Lazy" greedy

$S = \emptyset$

while $|S| \leq k$:

$$v^* = \arg\max_{v \notin S} f(S \cup \{v\}) - f(S)$$

$$S = S \cup \{v^*\}$$

Submodular optimization: greedy

***Theorem** [Nemhauser, Wolsey, Fisher 1978]: The greedy algorithm obtains a $\left(1 - \frac{1}{e}\right)$-approximation for maximizing a monotone submodular function subject to cardinality constraint.*

[Feige 1998]: This is the best possible unless P = NP.

Submodular optimization

- More complicated in many real world settings
  - E.g., handling uncertainty
- Still useful starting point for addressing more complex problems