APPLIED MECHANISM DESIGN FOR SOCIAL GOOD

JOHN P DICKERSON

Lecture #8 – 02/20/2020

CMSC828M
Tuesdays & Thursdays
2:00pm – 3:15pm

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

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:

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

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

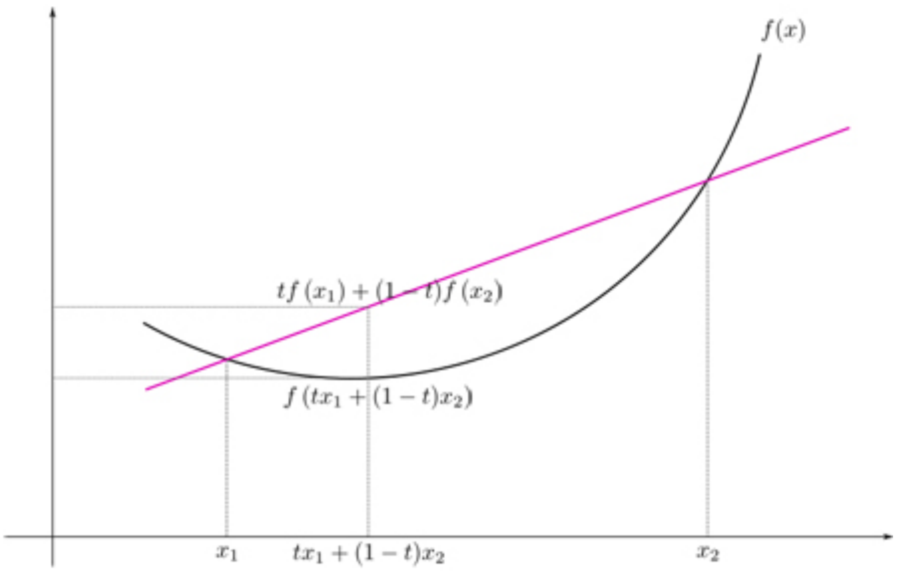
"A function is convex if the line segment between any two points on its graph lies above it."

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

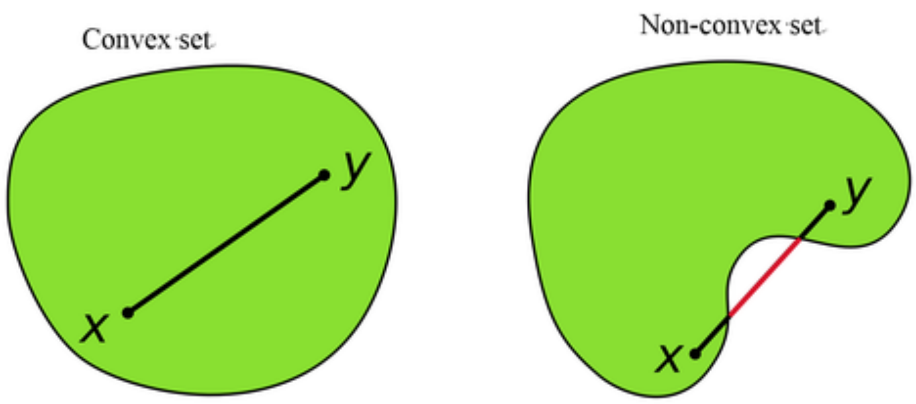
"A set is convex if, for every pair of points within the set, every point on the straight line segment that joins them is in the set."

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?

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



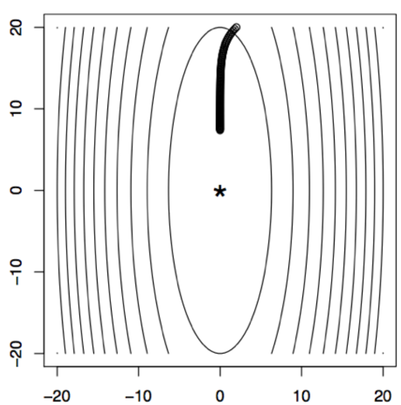
SO WHAT?

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!

There are 3 main parts that forms an optimization problem:

- **Decision variables** represent the decision that can be made
- **Objective function**: Each optimization problem is trying to optimize (maximize/minimize) some goal such as costs, profits, revenue.
- **Constraints**: Set of real restricting parameters that are imposed in real life or by the structure of the problem. Example for constraints can be:
  - Limited budget for a project
  - Limited manpower or resources
  - Being limited to choose only one option out of many options (Assignment)

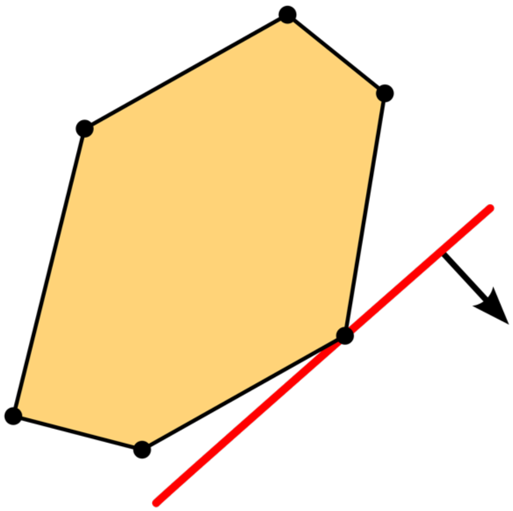
LINEAR PROGRAMS

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:

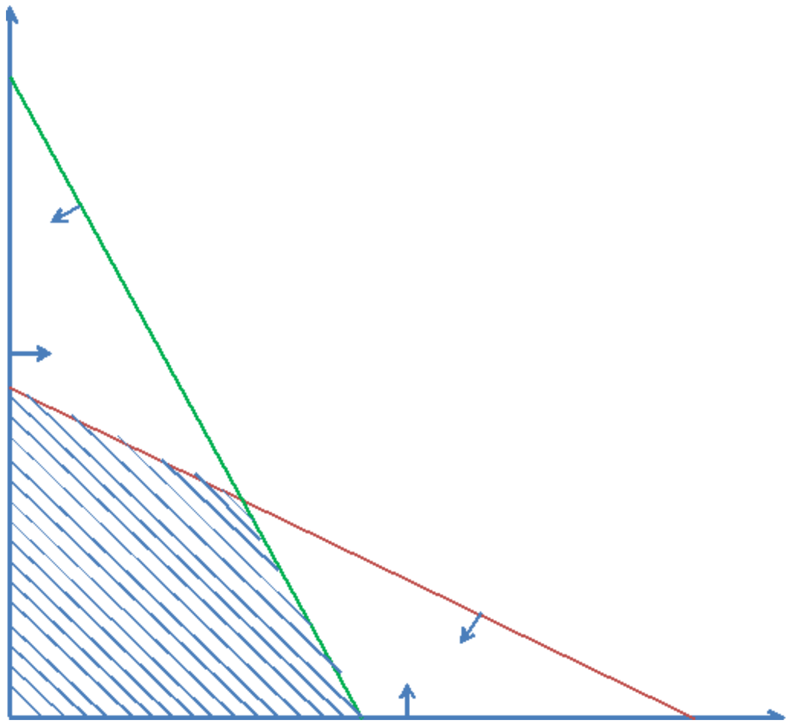
min/max	$\mathbf{c}^T\mathbf{x}$
subject to	$A\mathbf{x} \leq \mathbf{b}$
	$\mathbf{x} \geq 0$



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

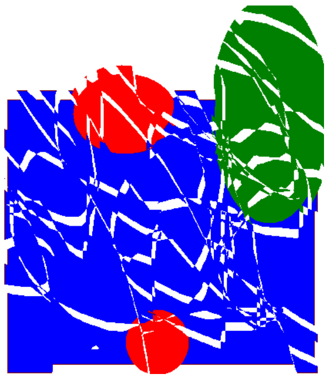
FEASIBLE REGION

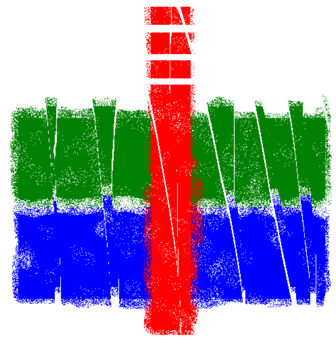
The feasible region is defined by the set of constraints of the problem, which is all the possible points that satisfy the all the constraints.



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

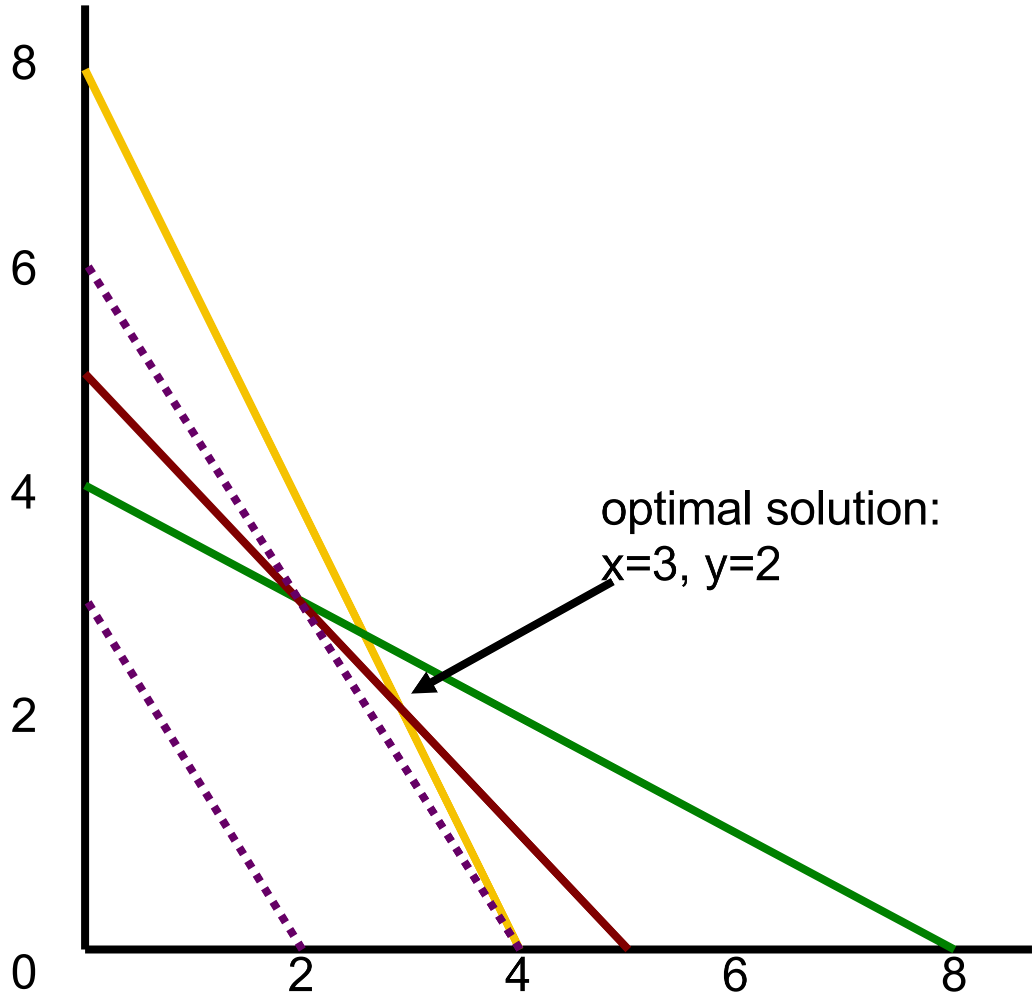
4x + 2y ≤ 16

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:

$$\langle x1, x2\rangle = \arg\max \min(3x_1 - 2x_2, -1x_1 + 1x_2)$$

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
	$\Sigma_i\, x_i = 1$			$\Sigma_j\, y_j = 1$	
	$x \geq 0$			$y \geq 0$	

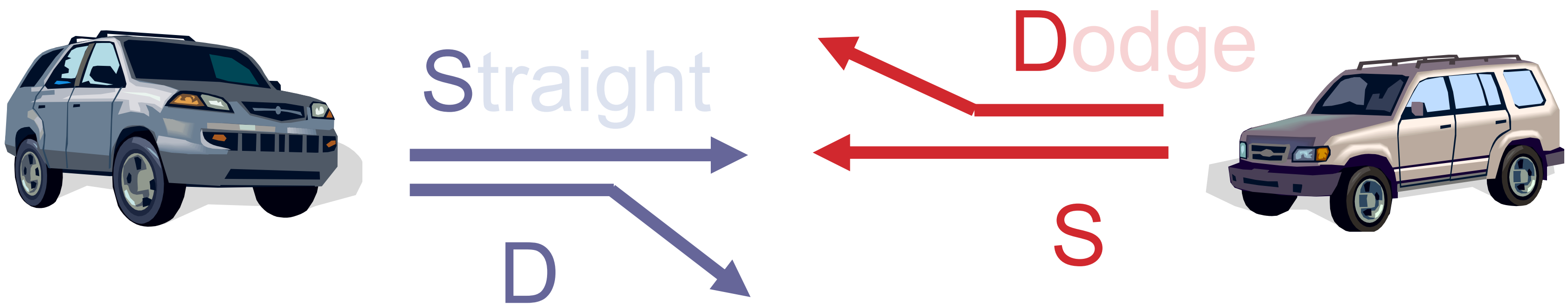
Know:

- 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

EXAMPLE: CHICKEN



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

????????

????????

- Thankfully, (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

CHICKEN

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

Is there an NE that uses mixed strategies?

- Say, where player 1 uses a mixed strategy?

- Note: 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

CRITICISMS OF NASH EQUILIBRIUM

Not unique in all games (like the example on Slide 31)

- Approaches for addressing this problem
  - Refinements (=strengthenings) of the equilibrium concept
    - Eliminate weakly dominated strategies first (IEDS)
    - Choose the Nash equilibrium with highest welfare
    - Subgame perfection … [see AGT book on course page]
  - Mediation, communication, convention, learning, …

Collusions amongst agents not handled well

- "No agent wants to deviate on her own"

Can be disastrous to "partially" play an NE

- (More) people may die!
- Correlated equilibria – strategies selected by an outsider, but the strategies must be stable (see Chp 2.7 of AGT)

CORRELATED EQUILIBRIUM

Suppose there is a trustworthy mediator who has offered to help out the players in the game

The mediator chooses a profile of pure strategies, perhaps randomly, then tells each player what her strategy is in the profile (but not what the other players' strategies are)

A correlated equilibrium is a distribution over pure-strategy profiles so that every player wants to follow the recommendation of the mediator (if she assumes that the others do so as well)

Every Nash equilibrium is also a correlated equilibrium

- Corresponds to mediator choosing players' recommendations independently

… but not vice versa

(Note: there are more general definitions of correlated equilibrium, but it can be shown that they do not allow you to do anything more than this definition.)

C.E. FOR CHICKEN

	D	S
D	0, 0 20%	-1, 1 40%
S	1, -1 40%	-5, -5 0%

Why is this a correlated equilibrium?

Suppose the mediator tells Row to Dodge

- From Row's perspective, the conditional probability that Col was told to Dodge is 20% / (20% + 40%) = 1/3
- So the expected utility of Dodging is (2/3)*(-1) = -2/3
- But the expected utility of Straight is (1/3)*1 + (2/3)*(-5) = -3
- So Row wants to follow the recommendation

If Row is told to go Straight, he knows that Col was told to Dodge, so again Row wants to follow the recommendation

Similar for Col

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.)

LINEAR ALGEBRA RECAP: POSITIVE DEFINITE MATRIX

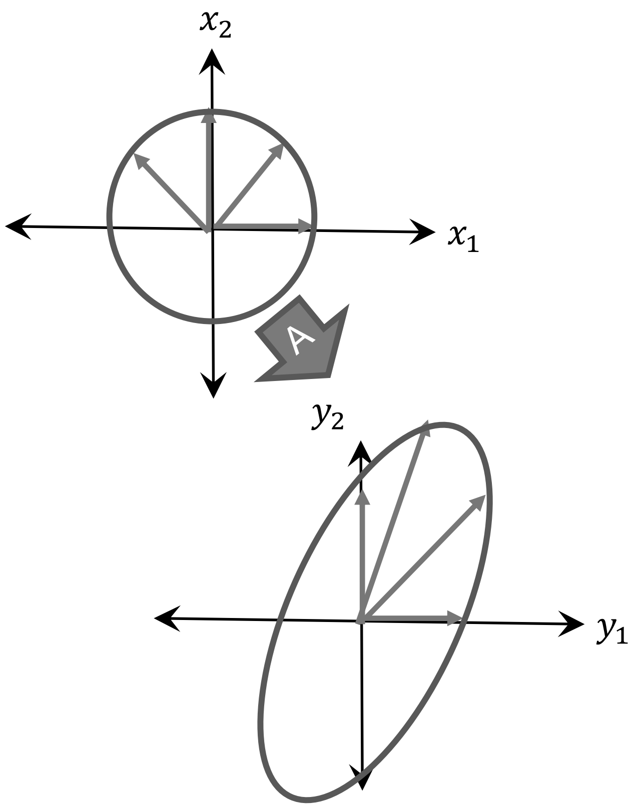
A linear transform $\vec{y} = A\vec{x}$ is called positive definite (written $A \succ 0$) if, for any vector $\vec{x}$,

$$\vec{x}^T A \vec{x} > 0$$

→ you can see that this means $\vec{x}^T \vec{y} > 0$.

→ this means that a matrix is positive definite if and only if the output of the transform, $\vec{y}$, is never rotated away from the input, $\vec{x}$, by 90 degrees or more! ← (useful geometric intuition)

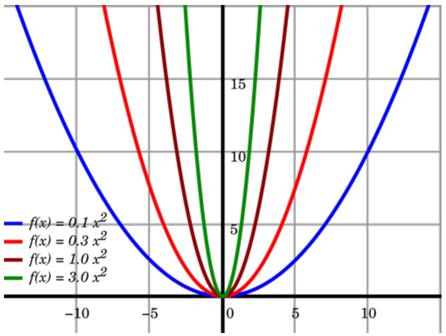
For example, the matrix $A = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$ is positive-definite.



Slide thanks to Mark Hasegawa-Johnson

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 …?

VC

INTEGER (LINEAR) PROGRAM

maximize **3x + 2y**

subject to

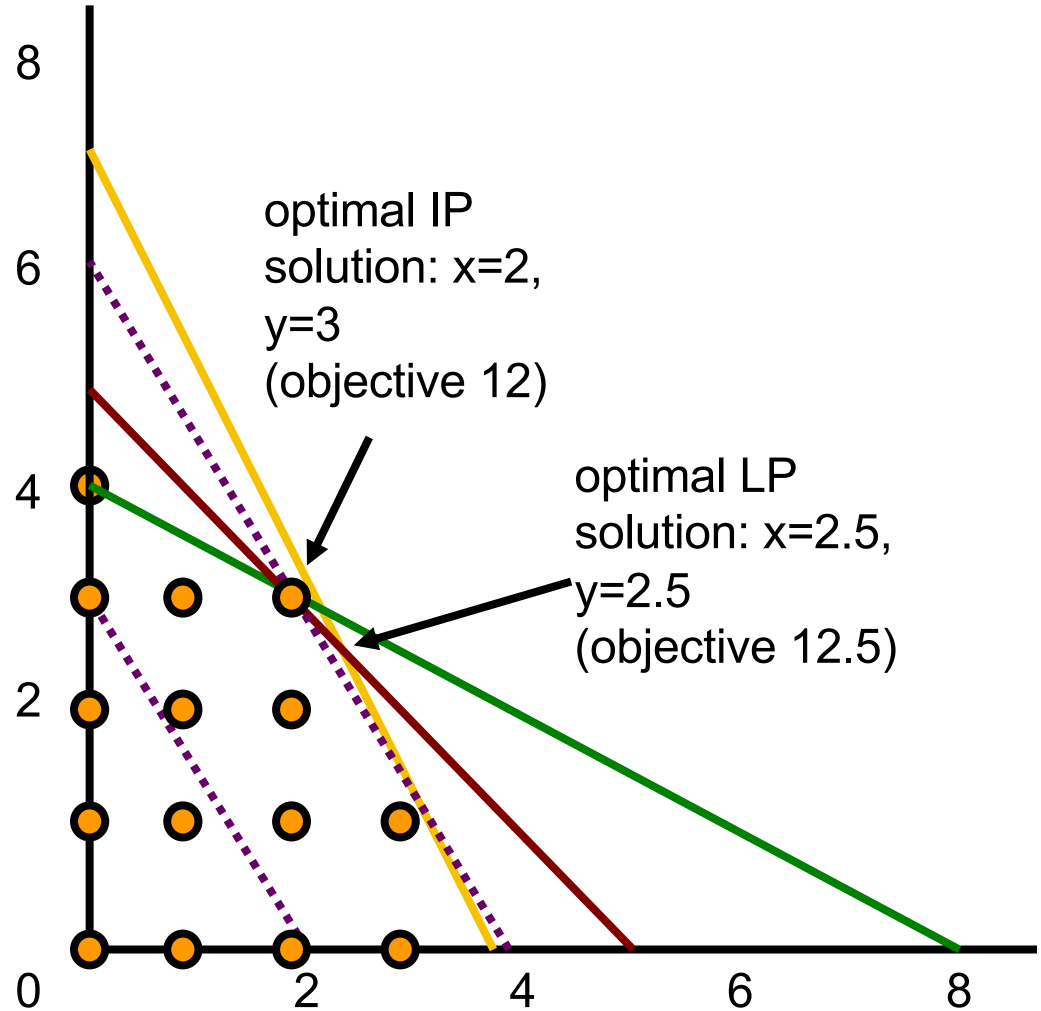
4x + 2y ≤ 15

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0, integer

y ≥ 0, integer



MIXED INTEGER (LINEAR) PROGRAM

maximize **3x + 2y**

subject to

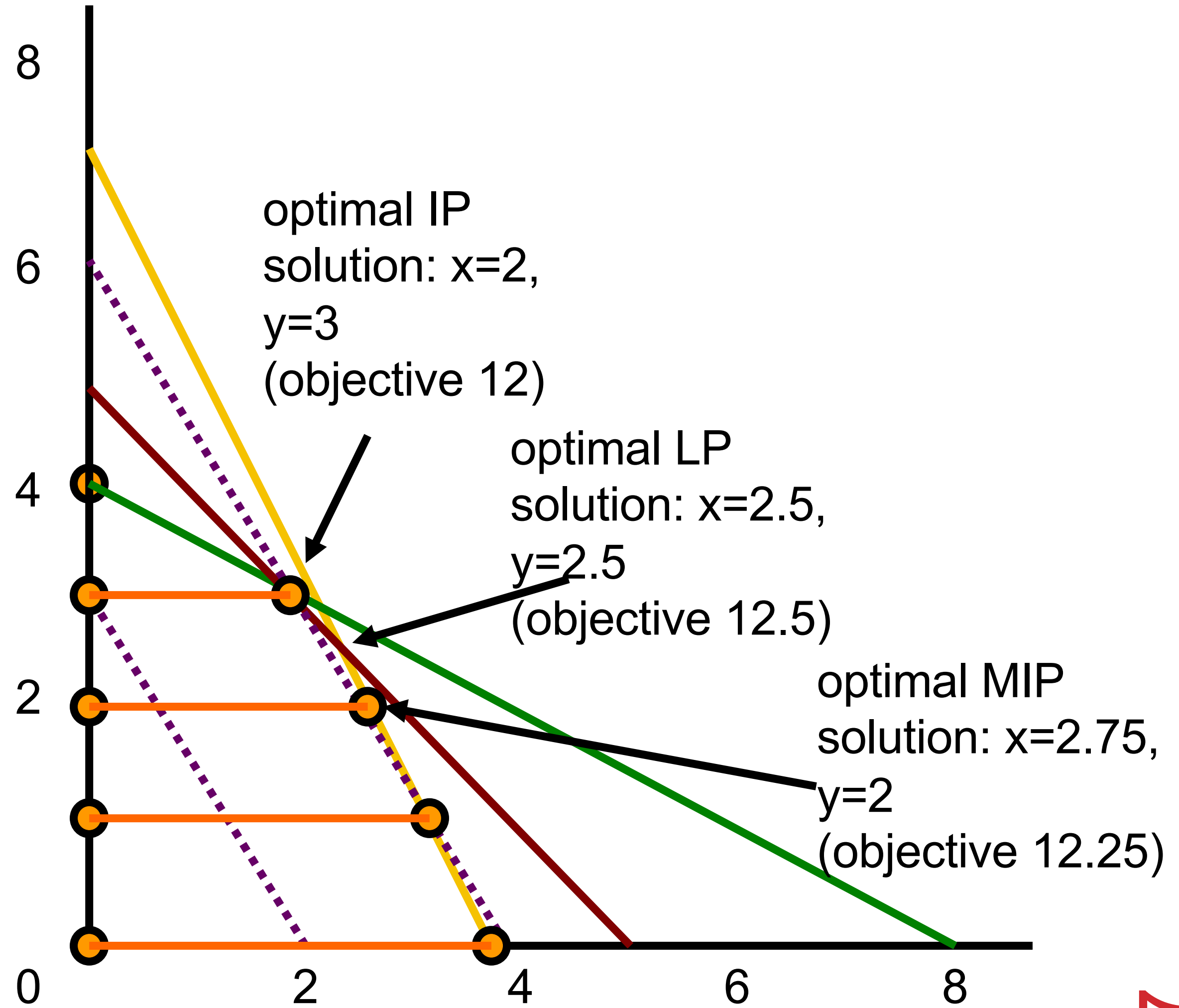
4x + 2y ≤ 15

x + 2y ≤ 8

x + y ≤ 5

x ≥ 0

y ≥ 0, integer



VC

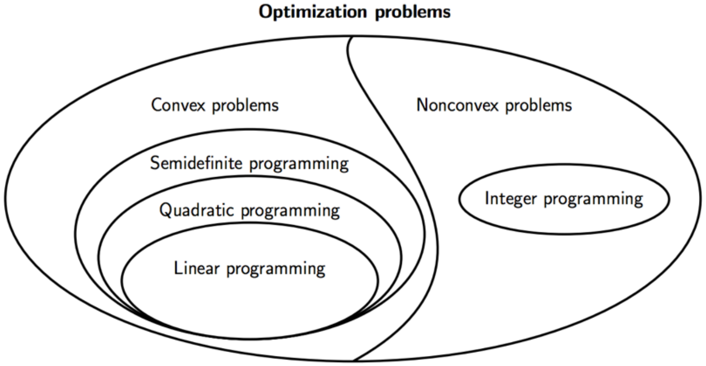
COMPLEXITY

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.

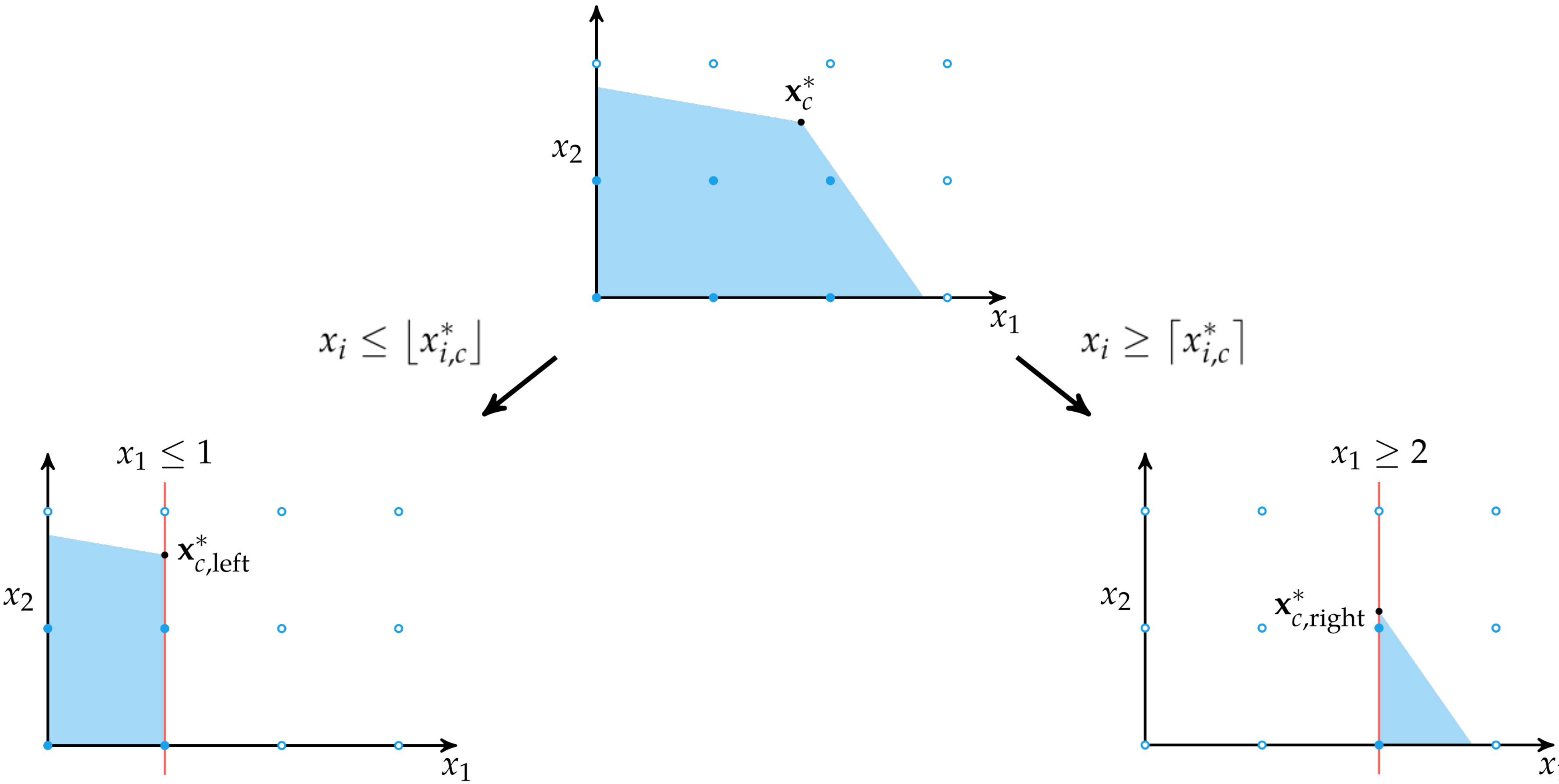
- In a maximization problem, LP OPT $\geq$ IP OPT. Why?
- So, we can use this as a PTIME upper bound during search

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

BRANCHING



$\mathbf{x}_c^*$
x_2
x_1
$x_i \leq \lfloor x_{i,c}^* \rfloor$
$x_i \geq \lceil x_{i,c}^* \rceil$
$x_1 \leq 1$
$\mathbf{x}_{c,\text{left}}^*$
x_2
x_1
$x_1 \geq 2$
$\mathbf{x}_{c,\text{right}}^*$
x_2
x_1

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

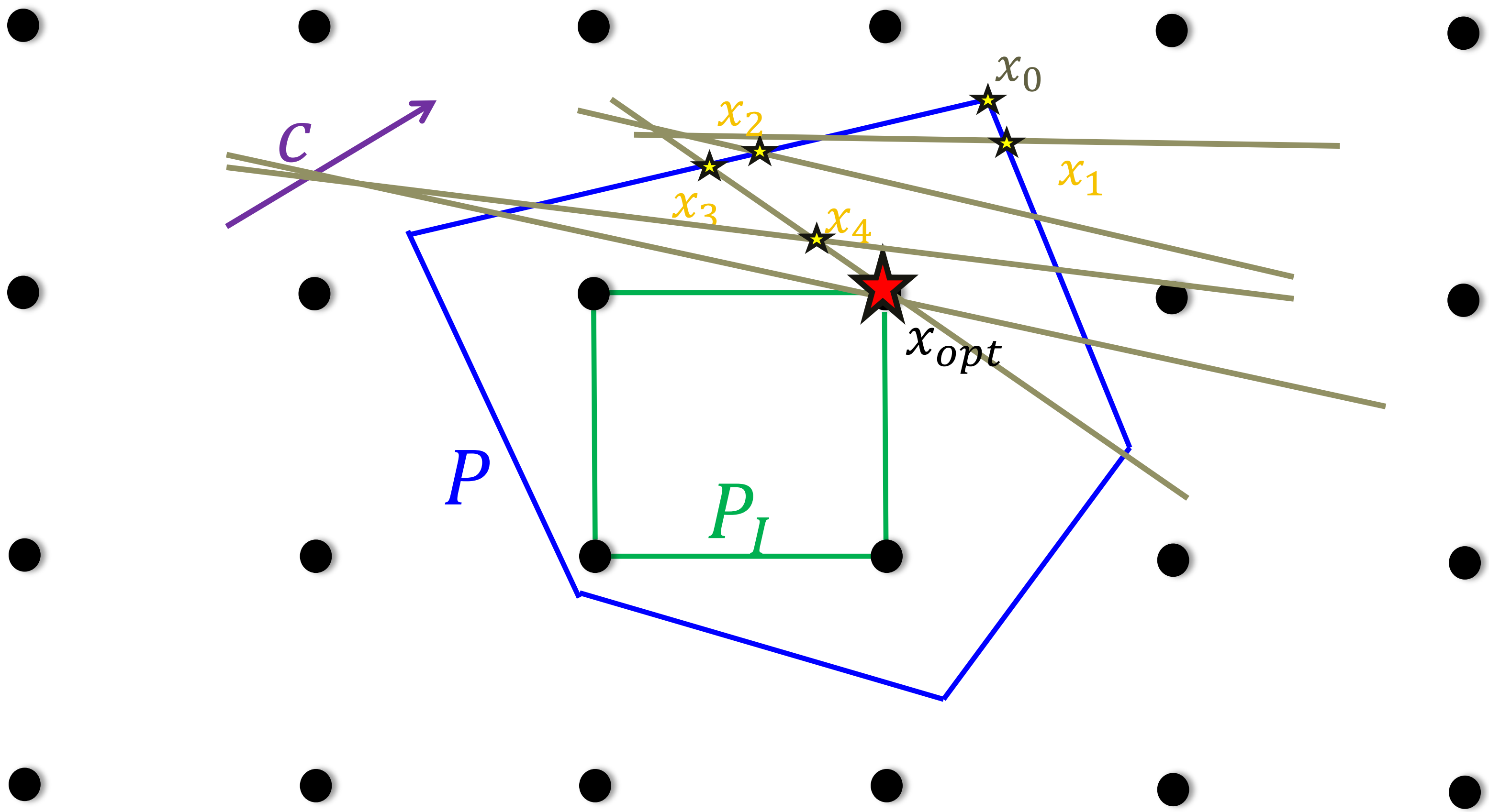
Lots of cuts! Which should we add?

Can cuts be computed quickly?

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

Sparse coefficients?

CUTTING PLANE METHOD



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

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

KC

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