3. Quadratically Constrained Quadratic Programming

- Quadratic programming
- MAXCUT
- Boolean optimization
- Primal and dual SDP relaxations
- Randomization
- Interpretations
- Examples
- LQR with binary inputs
- Rounding schemes

Quadratic Programming

A *quadratically constrained quadratic program* (QCQP) has the form

$$\begin{array}{rll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & \text{for all } i = 1, \ldots, m \end{array}$$

where the functions $f_i : \mathbb{R}^n \to \mathbb{R}$ have the form

$$f_i(x) = x^T P_i x + q_i^T x + r_i$$

- A *very* general problem
- If all the f_i are convex then the QCQP may be solved by SDP; but specialized software for *second-order cone programming* is more efficient

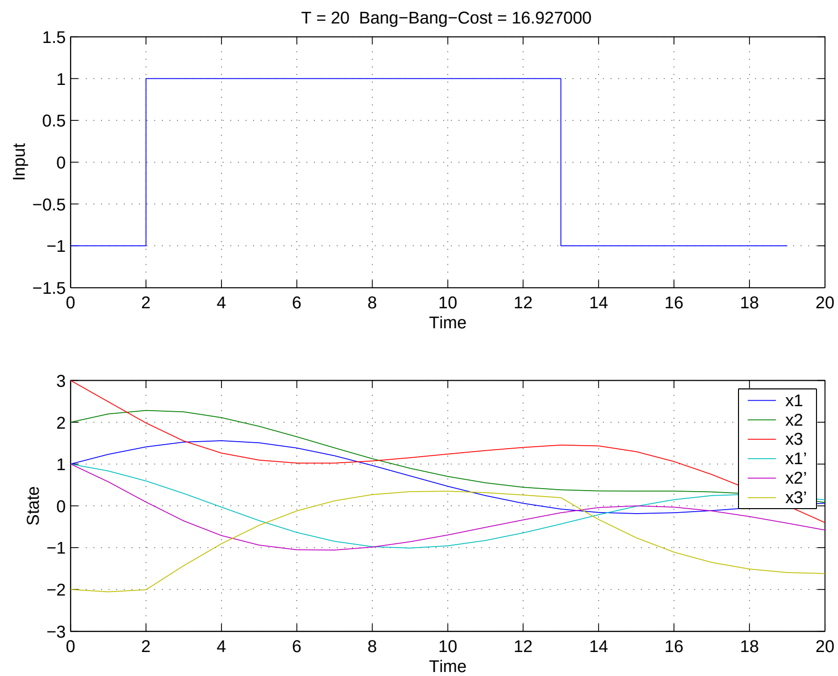
Example: LQR with Binary Inputs

Consider the discrete-time LQR problem

$$\text{minimize} \quad \|y(t) - y_r(t)\|^2 \qquad \text{subject to} \quad \begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

where y_r is the reference output trajectory, and the input $u(t)$ is constrained by $|u(t)| = 1$ for all $t = 0, \ldots, N$.

An open-loop LQR-type problem, but with a *bang-bang* input.



LQR with Binary Inputs

The objective $\|y(t) - y_r(t)\|^2$ is a *quadratic* function of the input u:

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \vdots \\ CA^tB & CA^{t-1}B & \dots & CB & 0 \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(t) \end{bmatrix}$$

So the problem can be written as:

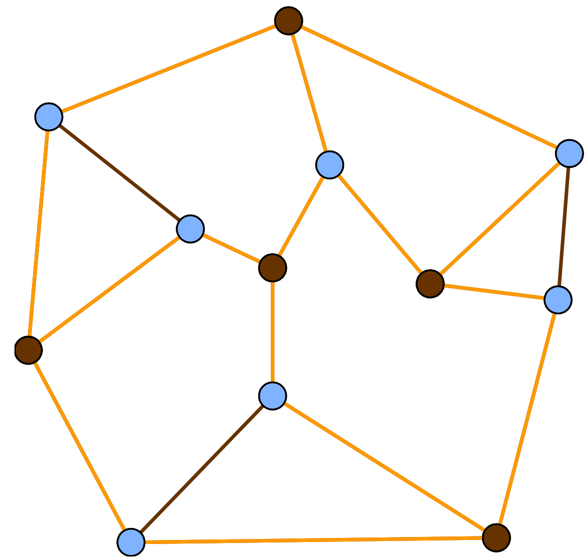
$$\begin{array}{ll} \text{minimize} & \begin{bmatrix} u \\ 1 \end{bmatrix}^T \begin{bmatrix} Q & r \\ r^T & s \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \\ \text{subject to} & u_i \in \{+1, -1\} \text{ for all } i \end{array}$$

where Q, r, s are functions of the problem data.

This is a quadratic *boolean optimization* problem.

MAXCUT

given an undirected graph, with no self-loops



- vertex set $V = \{1, \ldots, n\}$
- edge set $E \subset \Big\{ \{i, j\} \mid i, j \in V,\ i \neq j \Big\}$

For a subset $S \subset V$, the *capacity* of S is the number of edges connecting a node in S to a node not in S

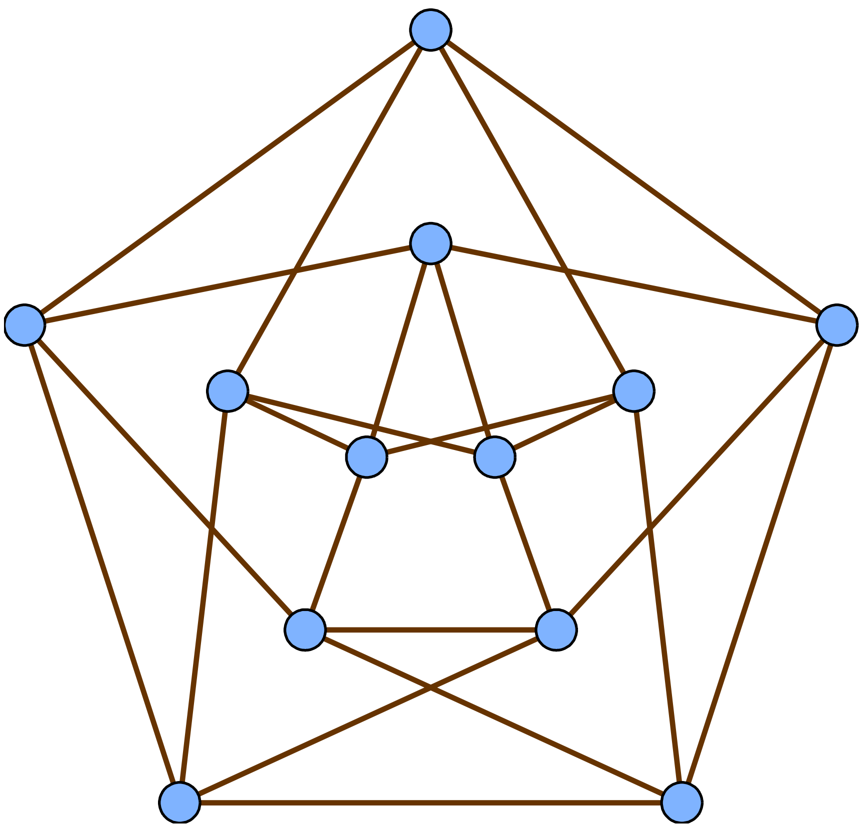
the MAXCUT problem

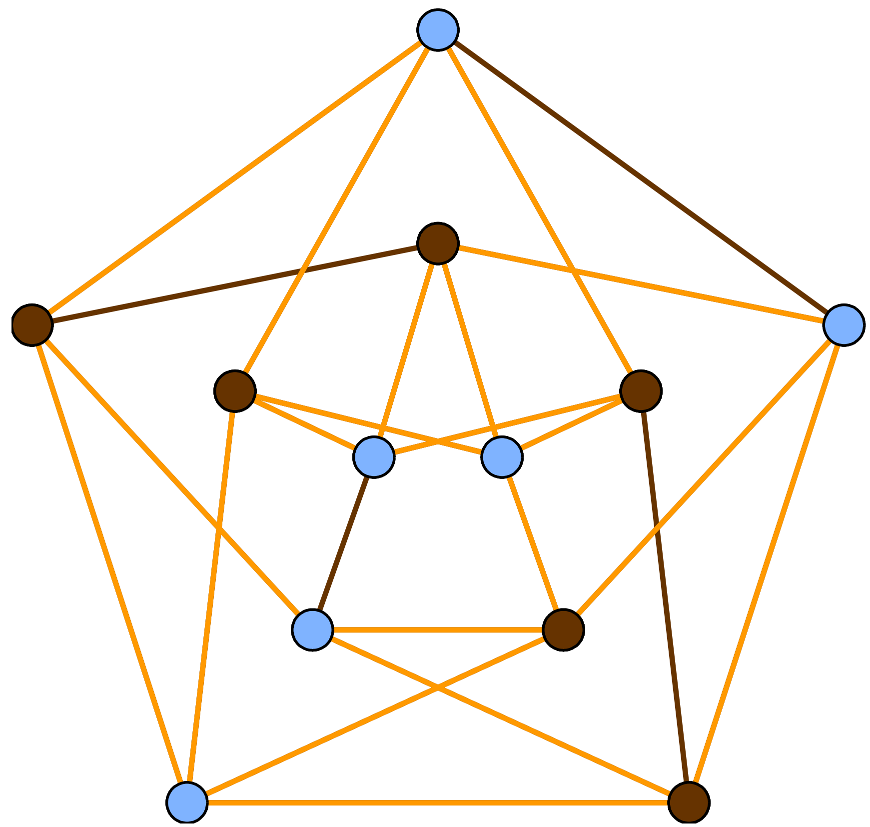
find $S \subset V$ with maximum capacity

the example above shows a cut with capacity 15; this is the maximum

Example

a graph with 12 nodes, 24 edges; the maximum capacity $c_{\mathsf{max}} = 20$





Problem Formulation

the graph is defined by its adjacency matrix

$$Q_{ij} = \begin{cases} 1 & \text{if } \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

and specify a cut S by a vector $x \in \mathbb{R}^n$

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ -1 & \text{otherwise} \end{cases}$$

then $1 - x_i x_j = 2$ if $\{i,j\}$ is a cut, so the capacity of x is

$$c(x) = \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} (1 - x_i x_j) Q_{ij}$$

Optimization Formulation

so we'd like to solve

$$\begin{aligned} \text{minimize} \quad & x^T Q x \\ \text{subject to} \quad & x_i \in \{-1, 1\} \qquad \text{for all } i = 1, \ldots, n \end{aligned}$$

call the optimal value $p^\star$, then the maximum cut is

$$c_{\mathrm{max}} = \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n Q_{ij} - \frac{1}{4} p^\star$$

Boolean Optimization

A classic combinatorial problem:

$$\begin{array}{rl} \text{minimize} & x^T Q x \\ \text{subject to} & x_i \in \{-1, 1\} \end{array}$$

- Many other examples; knapsack, LQR with binary inputs, etc.
- Can model the constraints with quadratic equations:

$$x_i^2 - 1 = 0 \quad \Longleftrightarrow \quad x_i \in \{-1, 1\}$$

- An exponential number of points. Cannot check them all!
- The problem is *NP-complete* (even if $Q \succeq 0$).

Despite the hardness of the problem, there are some very good approaches. . .

SDP Relaxations

We can find a lower bound via the dual; the primal is

$$\begin{array}{ll} \text{minimize} & x^T Q x \\ \text{subject to} & x_i^2 - 1 = 0 \end{array}$$

Let $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$, then the Lagrangian is

$$L(x, \lambda) = x^T Q x - \sum_{i=1}^{n} \lambda_i (x_i^2 - 1) = x^T (Q - \Lambda) x + \mathbf{trace}\,\Lambda$$

The dual is therefore the SDP

$$\begin{array}{ll} \text{maximize} & \mathbf{trace}\,\Lambda \\ \text{subject to} & Q - \Lambda \succeq 0 \end{array}$$

SDP Relaxations

From this SDP we obtain a *primal-dual pair of SDP relaxations*

$$\begin{array}{ll} \text{minimize} & \mathbf{trace}\, QX \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1 \end{array} \qquad \begin{array}{ll} \text{maximize} & \mathbf{trace}\, \Lambda \\ \text{subject to} & Q \succeq \Lambda \\ & \Lambda \ \text{diagonal} \end{array}$$

- We derived them from Lagrangian and SDP duality
- But, these SDP relaxations arise in *many* other ways
- Well-known in combinatorial optimization, graph theory, etc.
- Several interpretations

SDP Relaxations: Dual Side

Gives a simple *underestimator* of the objective function.

$$\begin{array}{ll} \text{maximize} & \mathbf{trace}\,\Lambda \\ \text{subject to} & Q \succeq \Lambda \\ & \Lambda \ \text{diagonal} \end{array}$$

Directly provides a *lower bound* on the objective: for any feasible x:

$$x^T Q x \geq x^T \Lambda x = \sum_{i=1}^{n} \Lambda_{ii} x_i^2 = \mathbf{trace}\,\Lambda$$

- The first inequality follows from $Q \succeq \Lambda$
- The second equation from Λ being diagonal
- The third, from $x_i \in \{+1, -1\}$

SDP Relaxations: Primal Side

The original problem is:

$$\begin{array}{rl} \text{minimize} & x^T Q x \\ \text{subject to} & x_i^2 = 1 \end{array}$$

Let $X := xx^T$. Then

$$x^T Q x = \mathbf{trace}\, Q x x^T = \mathbf{trace}\, Q X$$

Therefore, $X \succeq 0$, has *rank one*, and $X_{ii} = x_i^2 = 1$.

Conversely, any matrix X with

$$X \succeq 0, \quad X_{ii} = 1, \quad \mathbf{rank}\, X = 1$$

necessarily has the form $X = xx^T$ for some ± 1 vector x.

Primal Side

Therefore, the original problem can be exactly rewritten as:

$$\begin{array}{rl} \text{minimize} & \mathbf{trace}\, QX \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1 \\ & \mathbf{rank}(X) = 1 \end{array}$$

Interpretation: *lift* to a higher dimensional space, from $\mathbb{R}^n$ to $\mathbb{S}^n$.
Dropping the (nonconvex) rank constraint, we obtain the relaxation.

If the solution X has rank 1, then we have solved the original problem.

Otherwise, *rounding schemes* to project solutions. In some cases, approximation guarantees (e.g. Goemans-Williamson for MAX CUT).

Feasible Points and Certificates

$$\begin{array}{ll} \text{minimize} & \operatorname{trace} QX \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1 \end{array} \qquad \begin{array}{ll} \text{maximize} & \operatorname{trace} \Lambda \\ \text{subject to} & Q \succeq \Lambda \\ & \Lambda \ \text{diagonal} \end{array}$$

- Dual relaxations give *certified* bounds.
- Primal relaxations give information about possible *feasible* points.
- Both are solved *simultaneously* by primal-dual SDP solvers

Example

$$\begin{aligned} \text{minimize} \quad & 2x_1x_2 + 4x_1x_3 + 6x_2x_3 \\ \text{subject to} \quad & x_i^2 = 1 \end{aligned}$$

The associated matrix is $Q = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \end{bmatrix}$. The SDP solutions are:

$$X = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -5 \end{bmatrix}$$

We have $X \succeq 0$, $X_{ii} = 1$, $Q - \Lambda \succeq 0$, and

$$\mathbf{trace}\, QX = \mathbf{trace}\, \Lambda = -8$$

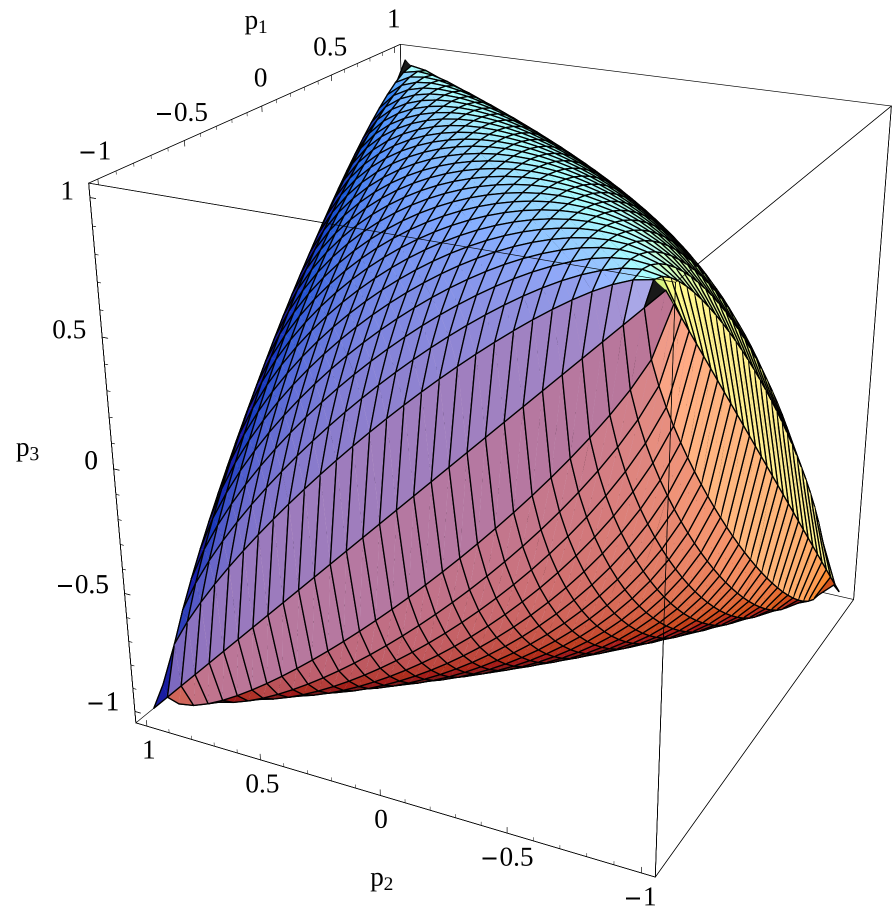
Since X is rank 1, from $X = xx^T$ we recover the optimal $x = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}^T$,

Spectrahedron

We can visualize this (in 3×3):

$$X = \begin{bmatrix} 1 & p_1 & p_2 \\ p_1 & 1 & p_3 \\ p_2 & p_3 & 1 \end{bmatrix} \succeq 0$$

in (p_1, p_2, p_3) space.



When optimizing the linear objective function

$$\mathbf{trace}\, QX = 2p_1 + 4p_2 + 6p_3,$$

the optimal solution is at the *vertex* $(1, -1, -1)$.

Primalization

After solving the SDP for $X \in \mathbb{S}^n$, we'd like to map back to $x \in \{-1, 1\}^n$

There may not exist an $x \in \{-1, 1\}^n$ such that $X = xx^T$

We can interpret this

- *algebraically*: $\mathbf{rank}\, X \neq 1$
- *geometrically*: X is not a *lifted point*

We need a procedure for finding a good x given X; called rounding, primalization, or projection.

This is hard in general, but for MAXCUT good methods are known

Randomization

Suppose we solve the primal relaxation

$$\begin{array}{ll} \text{minimize} & \mathbf{trace}\, QX \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1 \qquad \text{for all } i = 1, \ldots, n \end{array}$$

and the optimal X is not rank 1. Goemans and Williamson developed the following randomized algorithm for finding a feasible point

- Factorize X as $X = V^T V$, where $V = \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix} \in \mathbb{R}^{r \times n}$
- Then $X_{ij} = v_i^T v_j$, and since $X_{ii} = 1$ this factorization gives n vectors on the unit sphere in $\mathbb{R}^r$
- Instead of assigning either 1 or -1 to each vertex, we have assigned a point on the unit sphere in $\mathbb{R}^r$ to each vertex

Randomized Slicing

Pick a random vector $q \in \mathbb{R}^r$, and choose cut

$$S = \{\, i \mid v_i^T q \geq 0 \,\}$$

Then the probability that $\{i, j\}$ is a cut edge is

$$\frac{\text{angle between } v_i \text{ and } v_j}{\pi} = \frac{1}{\pi} \arccos v_i^T v_j = \frac{1}{\pi} \arccos X_{ij}$$

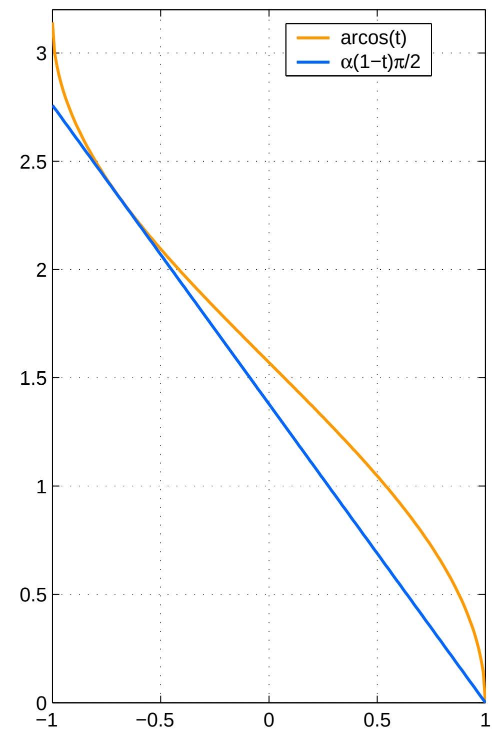
So the expected cut capacity is

$$c_{\text{sdp-expected}} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{\pi} Q_{ij} \arccos X_{ij}$$

Randomization (MAXCUT only)

SDP gives an upper bound on the cut capacity

$$c_{\text{sdp-upper-bound}} = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{4}(1 - X_{ij})Q_{ij}$$



With $\alpha = 0.878$, we have

$$\alpha(1 - t)\frac{\pi}{2} \leq \arccos(t) \qquad \text{for all } t \in [-1, 1]$$

So we have

$$c_{\text{sdp-upper-bound}} \leq \frac{1}{2\alpha\pi} \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} \arccos X_{ij}$$

$$= \frac{1}{\alpha} c_{\text{sdp-expected}}$$

Randomization

So far, we have

- $c_{\text{sdp-upper-bound}} \leq \frac{1}{\alpha} c_{\text{sdp-expected}}$
- Also clearly $c_{\text{sdp-expected}} \leq c_{\text{max}}$
- And $c_{\text{max}} \leq c_{\text{sdp-upper-bound}}$

After solving the SDP, we *slice randomly* to generate a random family of feasible points.

We can *sandwich* the expected value of this family as follows. $(\alpha = 0.878)$

$$\alpha c_{\text{sdp-upper-bound}} \leq c_{\text{sdp-expected}} \leq c_{\text{max}} \leq c_{\text{sdp-upper-bound}}$$

Coin-Flipping Approach

Suppose we just randomly assigned vertices to S with probability $\frac{1}{2}$; then

$$c_{\text{coinflip-expected}} = \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij}$$

A trivial upper bound on the maximum cut is just the total number of edges

$$c_{\text{trivial-upper-bound}} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij}$$

and so $c_{\text{coinflip-expected}} = \frac{1}{2} c_{\text{trivial-upper-bound}}$

Coin-Flipping Approach

We have

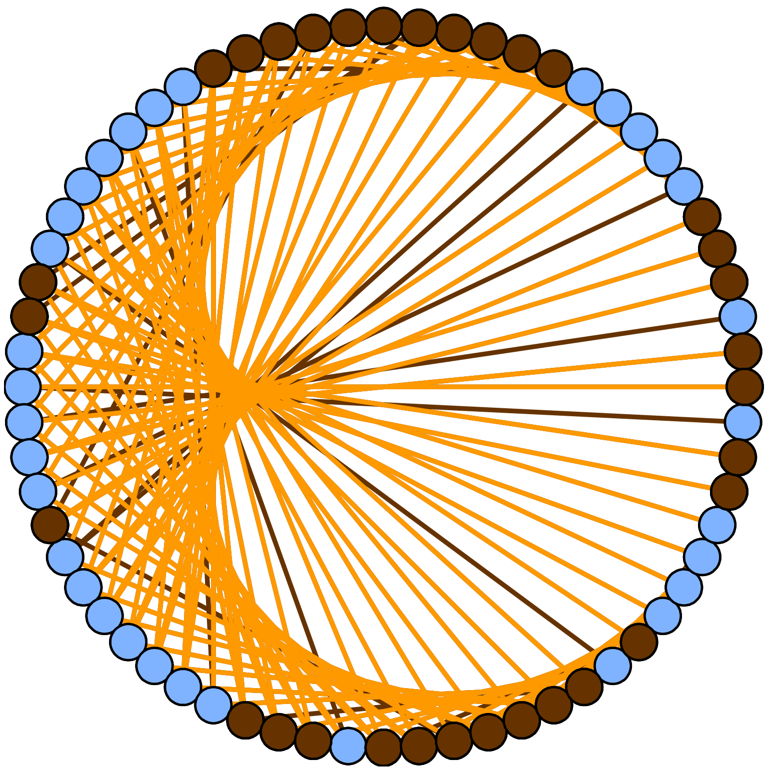
- $c_{\text{coinflip-expected}} = \frac{1}{2} c_{\text{trivial-upper-bound}}$
- $c_{\text{coinflip-expected}} \leq c_{\text{max}}$
- $c_{\text{max}} \leq c_{\text{trivial-upper-bound}}$

Again, we have a sandwich result

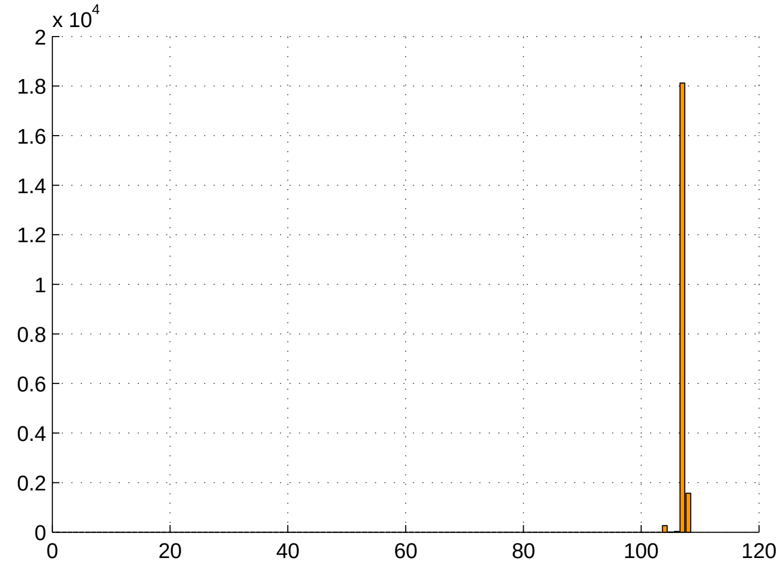
$$\frac{1}{2} c_{\text{trivial-upper-bound}} = c_{\text{coinflip-expected}} \leq c_{\text{max}} \leq c_{\text{trivial-upper-bound}}$$

Example

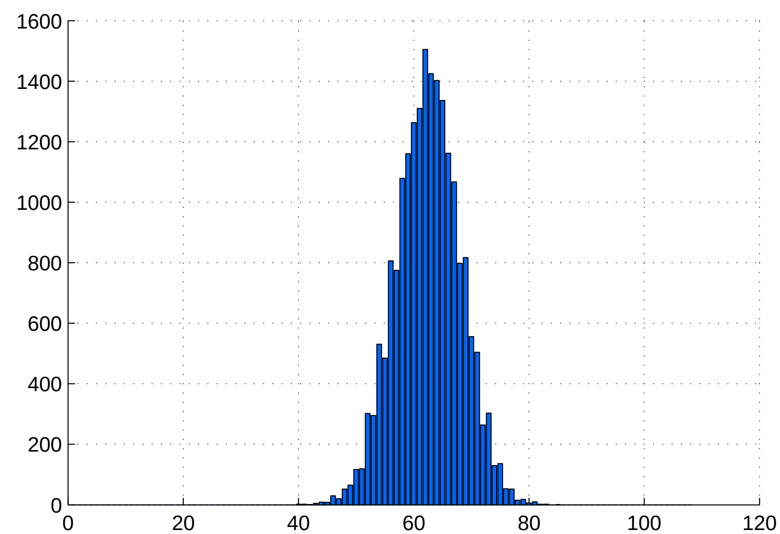
- 64 vertices, 126 edges
- SDP upper bound 116



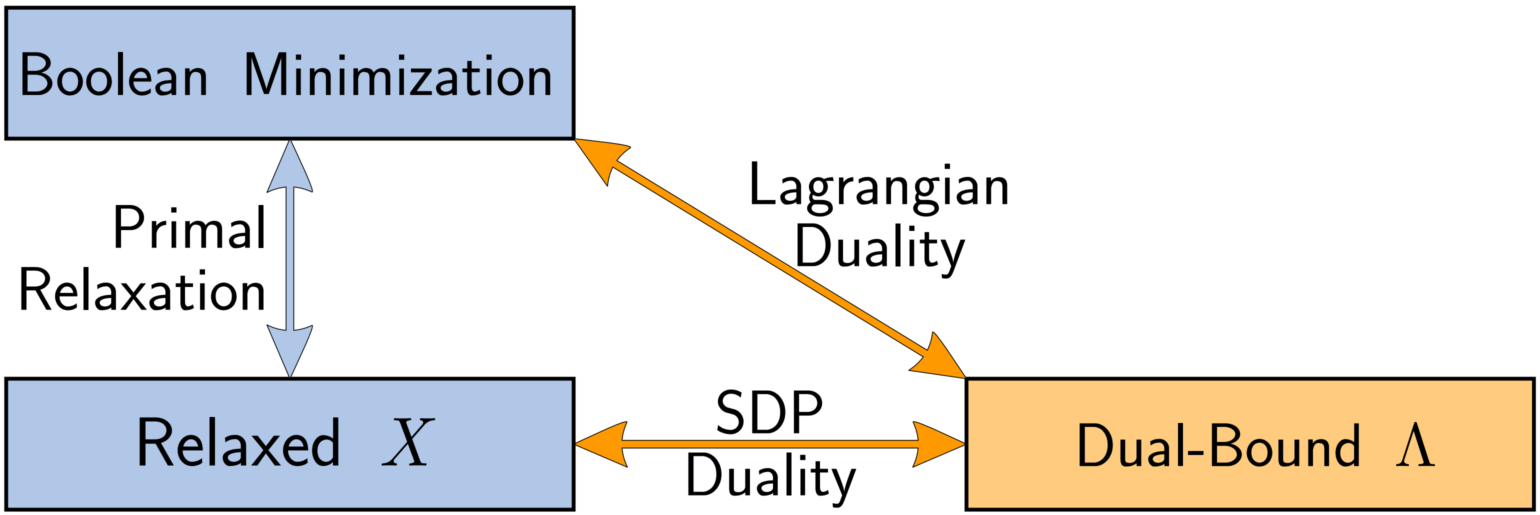
histogram of SDP capacities



histogram of coin-flip capacities



A General Scheme



- The *relaxed* X suggests candidate points.
- The diagonal matrix Λ *certifies* a lower bound.

Ubiquitous scheme in optimization (convex hulls, fractional colorings, etc. . .)

We will learn systematic ways of constructing these relaxations, and more. . .

LQR with Binary Inputs

$$\begin{array}{ll} \text{minimize} & \begin{bmatrix} u \\ 1 \end{bmatrix}^T \begin{bmatrix} Q & r \\ r^T & s \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \\ \text{subject to} & u_i \in \{+1, -1\} \text{ for all } i \end{array}$$

for some matrices (Q, r, s) function of the problem data (A, B, C, N).

An *SDP dual bound*:

$$\begin{array}{ll} \text{maximize} & \mathbf{trace}(\Lambda) + \mu \\ \text{subject to} & \begin{bmatrix} Q - \Lambda & r \\ r^T & s - \mu \end{bmatrix} \succeq 0, \qquad \Lambda \text{ diagonal} \end{array}$$

Let q^*, q_* be the optimal value of both problems. Then, $q^* \geq q_*$:

$$\begin{bmatrix} u \\ 1 \end{bmatrix}^T \begin{bmatrix} Q & r \\ r^T & s \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \geq \begin{bmatrix} u \\ 1 \end{bmatrix}^T \begin{bmatrix} \Lambda & 0 \\ 0 & \mu \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} = \mathbf{trace}\, \Lambda + \mu$$

LQR with Binary Inputs

$$\begin{array}{ll} \text{maximize} & \mathbf{trace}(\Lambda) + \mu \\ \text{subject to} & \begin{bmatrix} Q - \Lambda & r \\ r^T & s - \mu \end{bmatrix} \succeq 0, \qquad \Lambda \text{ diagonal} \end{array}$$

Since $(\Lambda, \mu) = (0, 0)$ is always feasible, $q_* \geq 0$.

Furthermore, the bound is never worse than the LQR solution obtained by dropping the ± 1 constraint, since

$$\Lambda = 0, \quad \mu = s - r^T Q^{-1} r$$

is a feasible point.

Example:

N	LQR cost	SDP bound	Optimal
10	14.005	15.803	15.803
15	15.216	16.698	16.705
20	15.364	16.905	16.927

The S-procedure

A *sufficient* condition for the infeasibility of quadratic inequalities:

$$\{x \in \mathbb{R}^n \quad | \quad x^T A_i x \geq 0\}$$

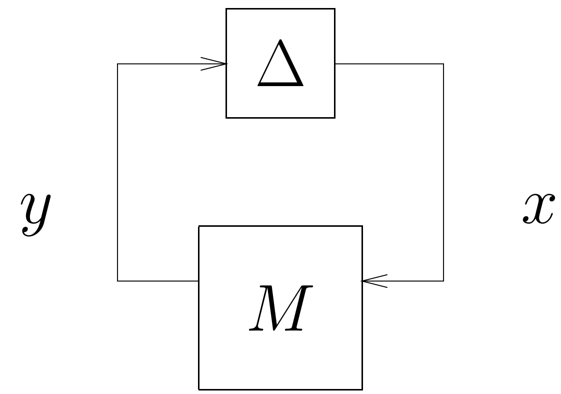
Again, a primal-dual pair of SDP relaxations:

$$\begin{aligned} X &\succeq 0 \\ \textbf{trace}\, X &= 1 \\ \textbf{trace}\, A_i X &\geq 0 \end{aligned} \qquad\qquad \begin{aligned} \textstyle\sum_i \lambda_i A_i &\preceq -I \\ \lambda_i &\geq 0 \end{aligned}$$

The basis of many important results in control theory.

Structured Singular Value

- A central paradigm in robust control.
- μ is a measure of robustness: how big can a structured perturbation Δ be, without losing stability.



Do the loop equations admit nontrivial solutions?

$$y = Mx, \quad y_i^2 - x_i^2 \geq 0$$

Applying the standard SDP relaxation:

$$\sum_i d_i(y_i^2 - x_i^2) = x^T(M^T D M - D)x < 0, \qquad D = \mathbf{diag}(d_i),\ d_i \geq 0$$

We obtain the standard μ upper bound:

$$M^T D M - D \prec 0, \qquad D \quad \text{diagonal}, \quad D \succeq 0$$