EE464: SDP Relaxations for QP

convex quadratic constraints

suppose P is symmetric, and $P \succeq 0$; we can represent the convex quadratic constraint

$$x^T P x + q^T x + r < 0$$

as a semidefinite programming constraint as follows

write P as the product $P = A^T A$ via Cholesky or eigenvalue decomposition, then

$$x^T P x + q^T x + r < 0 \qquad \Longleftrightarrow \qquad \begin{bmatrix} -I & Ax \\ x^T A^T & q^T x + r \end{bmatrix} \prec 0$$

Quadratic programming

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

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

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

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

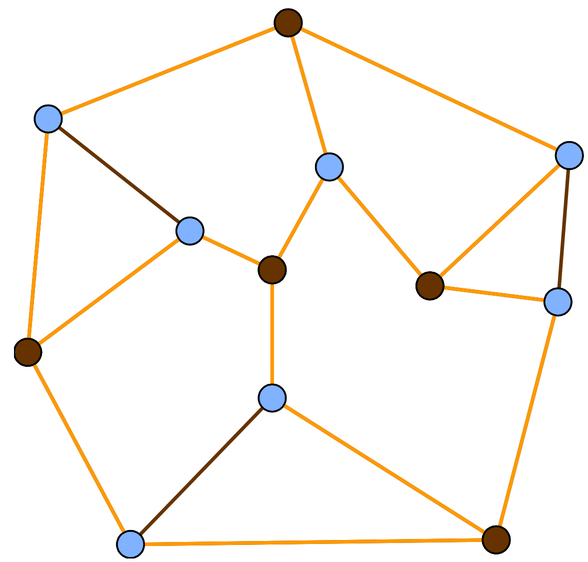
If $P_i \succeq 0$ then f_i is a convex function

- if all the f_i are convex then the QCQP may be solved by semidefinite programming
- but specialized software for *second-order cone programming* is more efficient

MAXCUT

given an undirected graph, with no self-loops

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



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

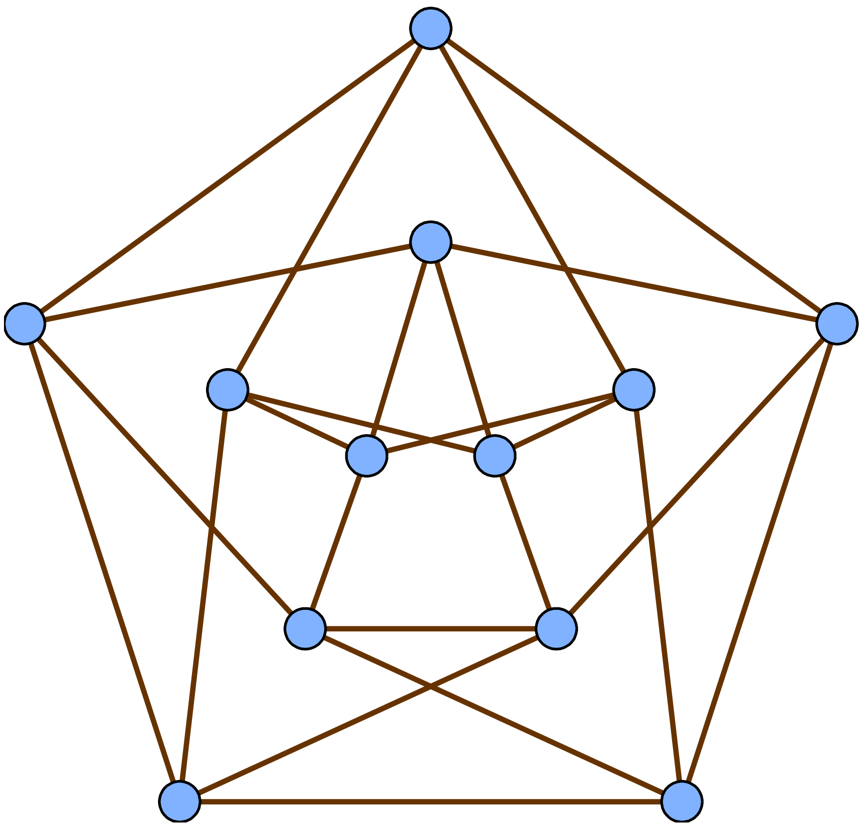
the MAXCUT problem

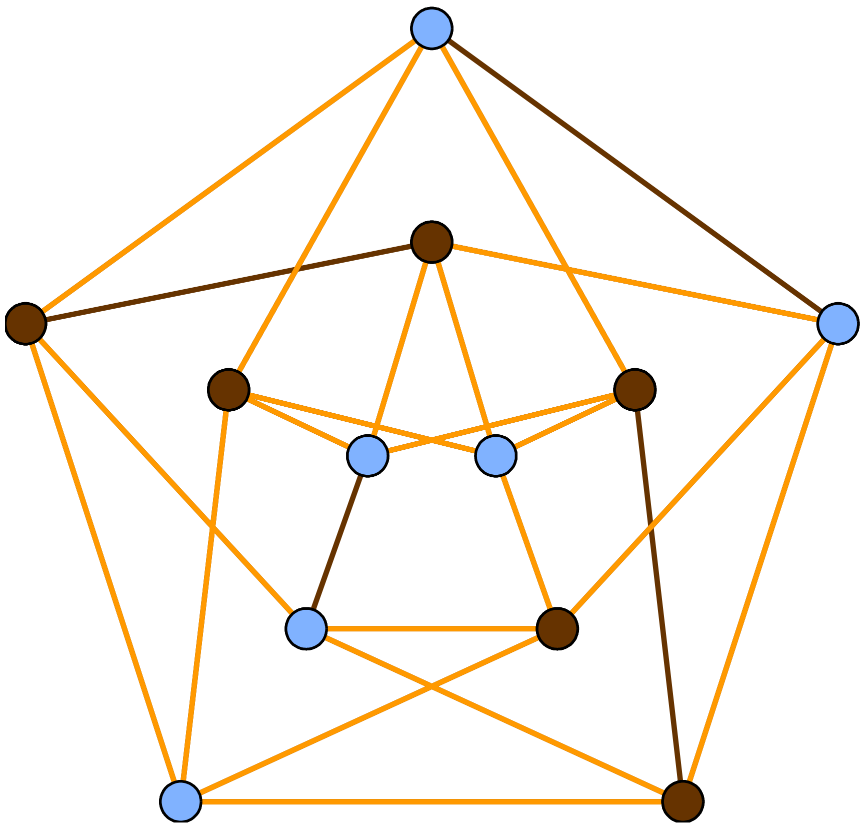
find $S \subset V$ with maximum capacity

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

example

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





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

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

the extra factor of $\frac{1}{2}$ arises because A is symmetric

optimization formulation

so we'd like to solve

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

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

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

Boolean optimization

A classic combinatorial problem:

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

- 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 on the minimum of this QP, (and hence an upper bound on MAXCUT) using the dual problem; the primal is

minimize	x^TQx
subject to	$x_i^2 - 1 = 0$

the Lagrangian is

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

where $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$; the Lagrangian is bounded below w.r.t. x if $Q - \Lambda \succeq 0$

The dual is therefore the SDP

maximize	$\mathbf{tr}\,\Lambda$
subject to	$Q - \Lambda \succeq 0$

SDP Relaxations

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

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

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

- 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.

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

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

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

- 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:

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

Let $X := xx^T$. Then

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

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:

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

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

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

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

Example

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

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

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

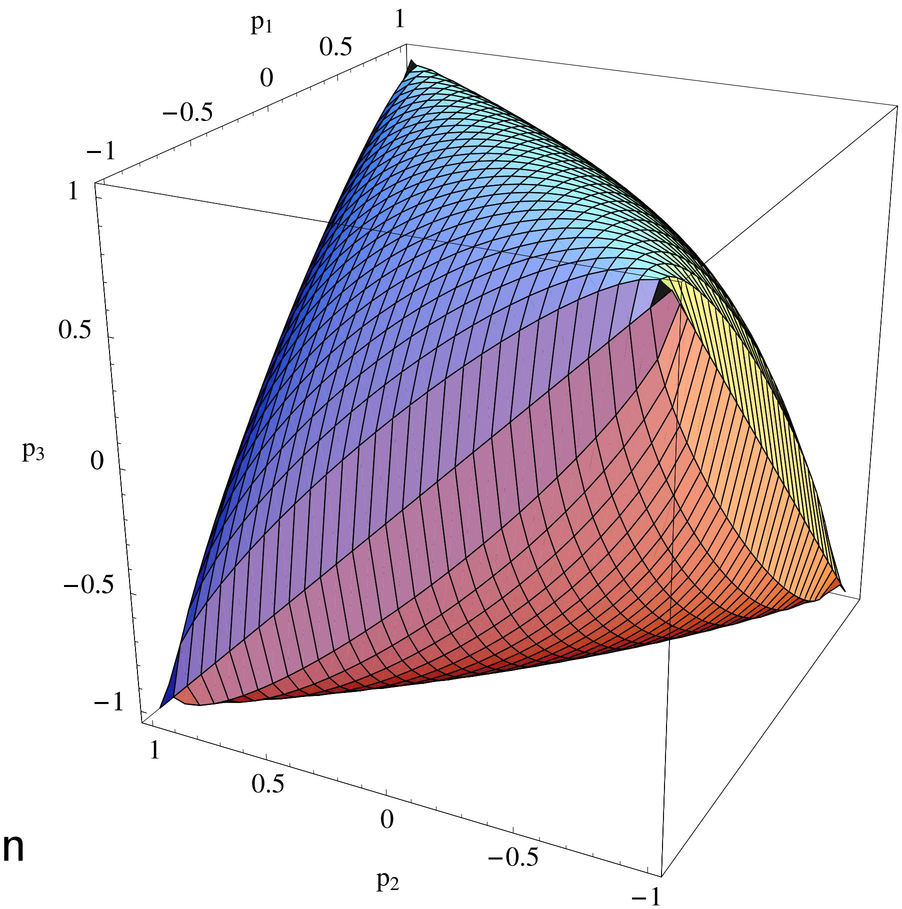
Since X is rank 1, from $X = xx^T$ we recover the optimal $x = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}^T$,

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

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

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

Randomization

suppose we solve the primal relaxation

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

and the optimal X is not rank 1

the following randomized algorithm gives 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 $\mathbb{R}^r$

interpretation; 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

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

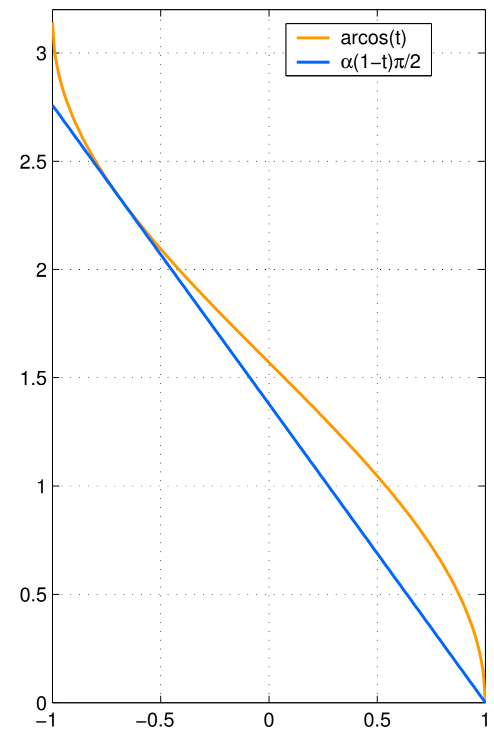
so the expected cut capacity is

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

randomization

the upper bound on the cut capacity obtained from the SDP is

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



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

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

Randomization

So far, we have

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

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. ($\alpha = 0.878$)

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

coin-flipping approach

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

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

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

again, since $c_{\text{coinflip-expected}} \leq c_{\text{max}}$, we have

Coin-Flipping Approach

We have

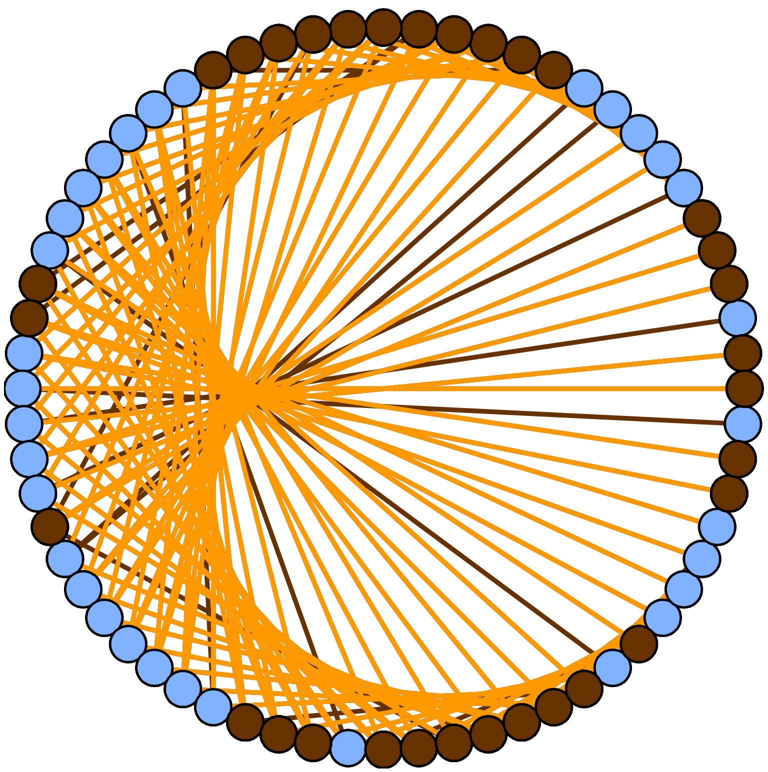
- $c_{\text{coinflip-expected}} = \frac{1}{2}c_{\text{trivial-upper-bound}}$
- $c_{\text{coinflip-expected}} \leq c_{\text{max}}$
- $c_{\text{max}} \leq c_{\text{trivial-upper-bound}}$

Again, we have a sandwich result

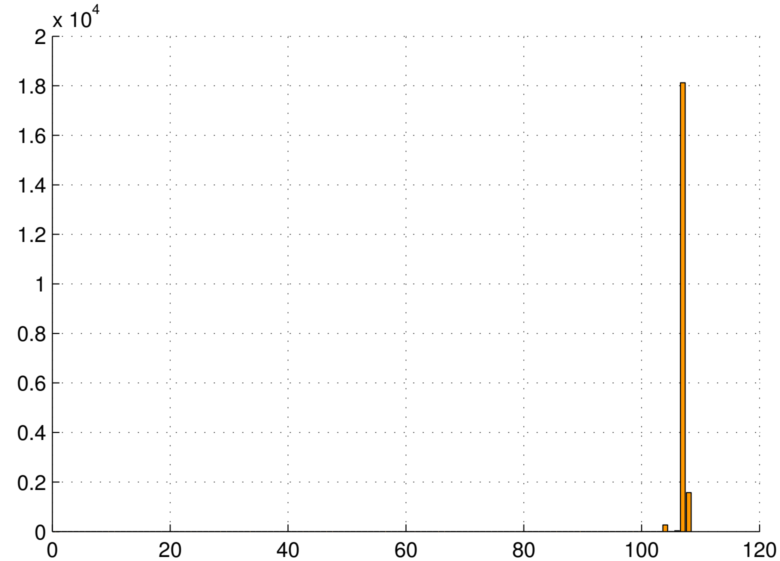
$$\boxed{\frac{1}{2}c_{\text{trivial-upper-bound}} = c_{\text{coinflip-expected}} \leq c_{\text{max}} \leq c_{\text{trivial-upper-bound}}}$$

Example

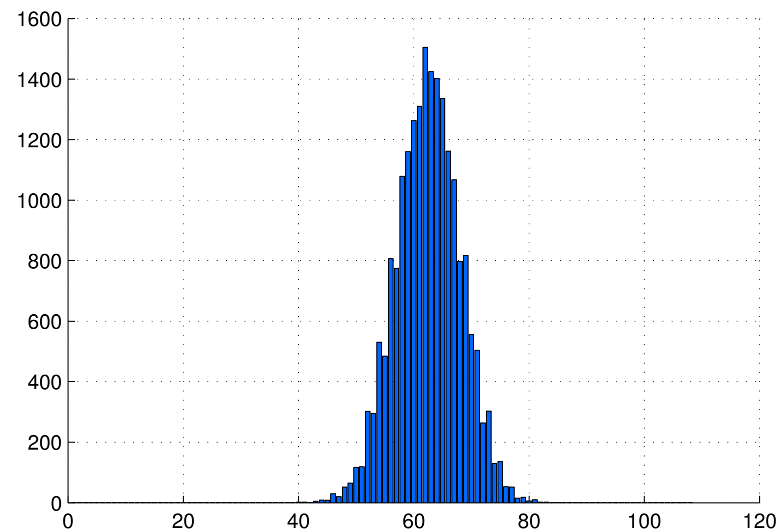
- 64 vertices, 126 edges
- SDP upper bound 116



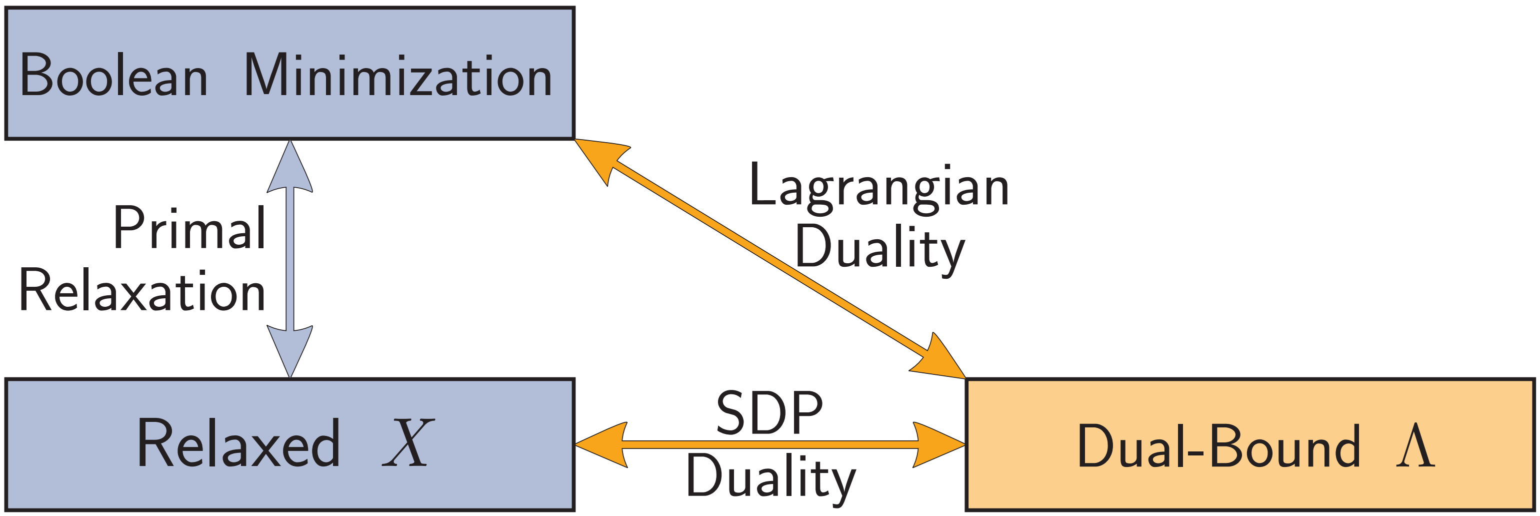
histogram of SDP capacities



histogram of coin-flip capacities



A General Scheme



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

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