Advanced Algorithms

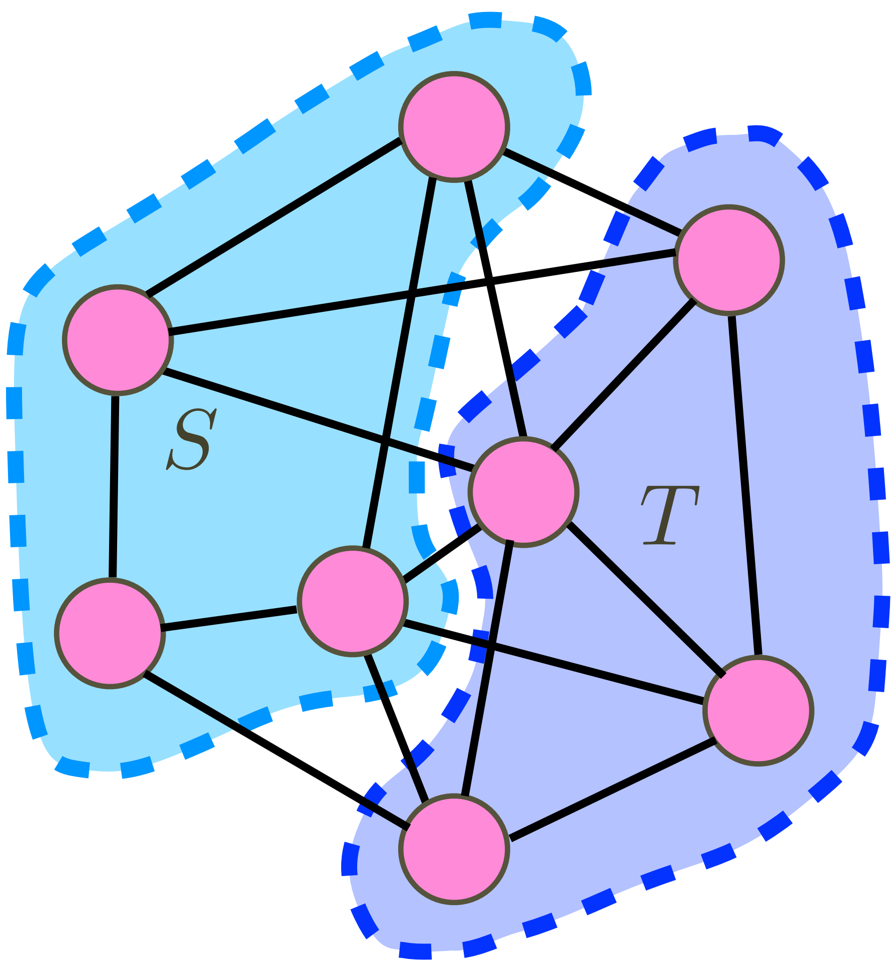
南京大学

尹一通

Max-Cut

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.

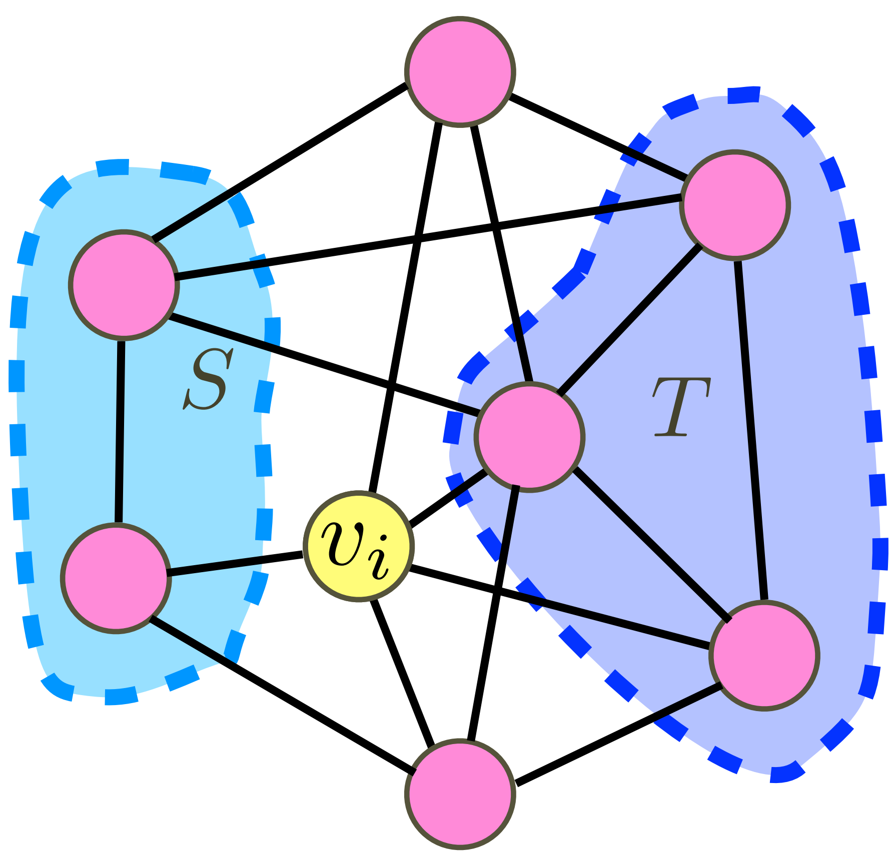


- One of Karp's 21 **NPC** problems.
- Typical Max-CSP.
- *Greedy* is 1/2-approximate.
- *Random selection* is 1/2-approximate.
- *Local search* is 1/2-approximate.

Greedy Algorithm

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.



GreedyMaxCut

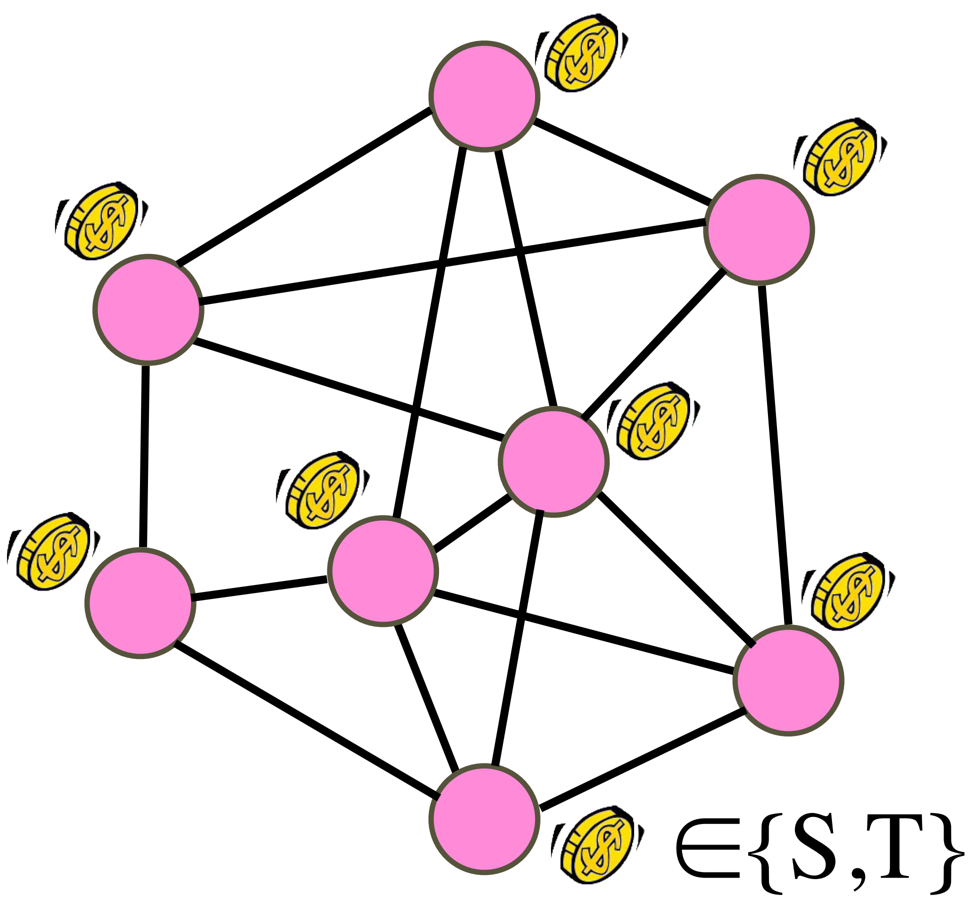
$V = \{v_1, v_2, \ldots, v_n\}$;
initially, $S=T=\emptyset$;
for $i = 1,2, \ldots, n$
  v_i joins one of S, T
  to maximize *current* $E(S,T)$

1/2-approximate

Random Selection

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.



i.i.d.

u.a.r.

RandomSelection

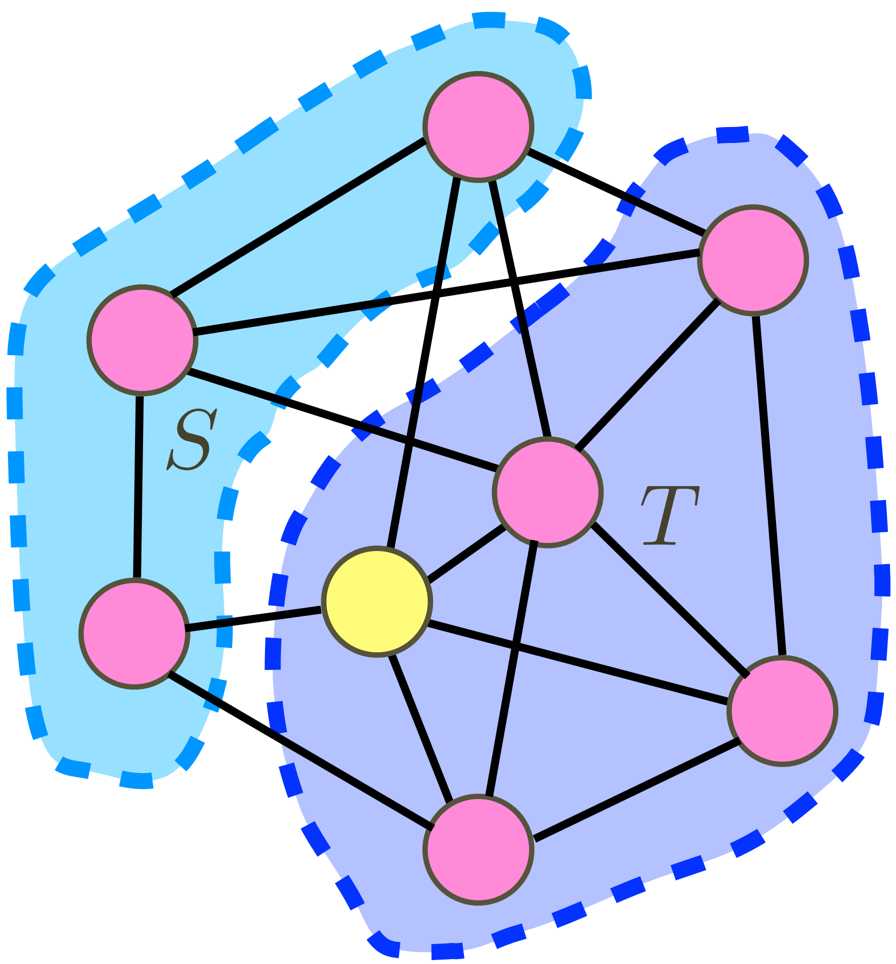
for each $v \in V$
  v joins one of S, T *uniformly*
  and *independently* at random;

1/2-approximate

Local Search

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.



LocalSearch

Start with an *arbitrary* bipartition;
repeat until nothing changed:
  if $\exists v$ flipping side will increase cut
    v moves to the other side;

1/2-approximate

Proof of Local Search

Given a simple undirected graph $G(V, E)$, for any $v \in V$ define

$\delta(v)$:=number of edges incident to v.

When the algorithm terminates, $\forall v \in S \ |\delta(v) \cap (S \times S)| \leq |\delta(v) \cap (S \times T)|$

$\forall v \in T \ |\delta(v) \cap (T \times T)| \leq |\delta(v) \cap (S \times T)|$

$\Rightarrow \sum_{v \in S} |\delta(v) \cap (S \times S)| + \sum_{v \in T} |\delta(v) \cap (T \times T)| \leq \sum_{v} |\delta(v) \cap (S \times T)|$

$\Rightarrow \sum_{v \in S} |\delta(v) \cap (S \times S)| + \sum_{v \in T} |\delta(v) \cap (T \times T)| + \sum_{v} |\delta(v) \cap (S \times T)| \leq 2 \sum_{v} |\delta(v) \cap (S \times T)|$

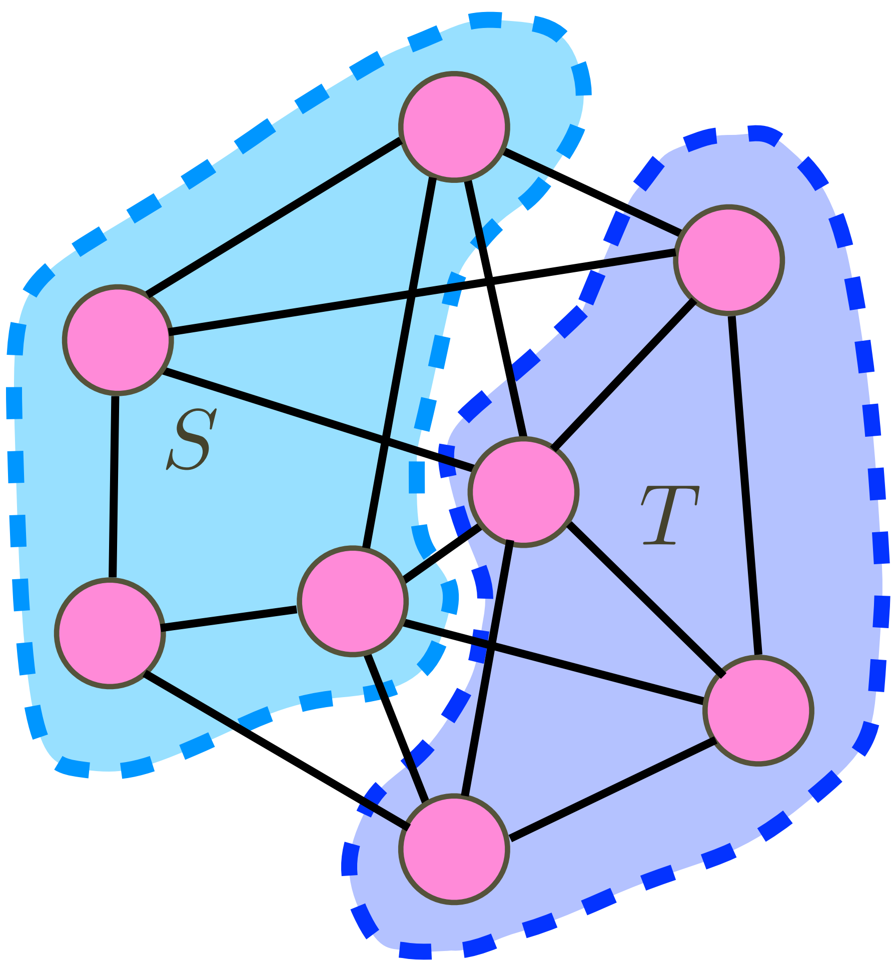
$\Rightarrow 2|E| \leq 4|S \times T|$

$\Rightarrow \ |S \times T| \geq \frac{1}{2}|E|.$

LP for Max-Cut

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.



$$\max \quad \sum_{uv \in E} y_{uv}$$

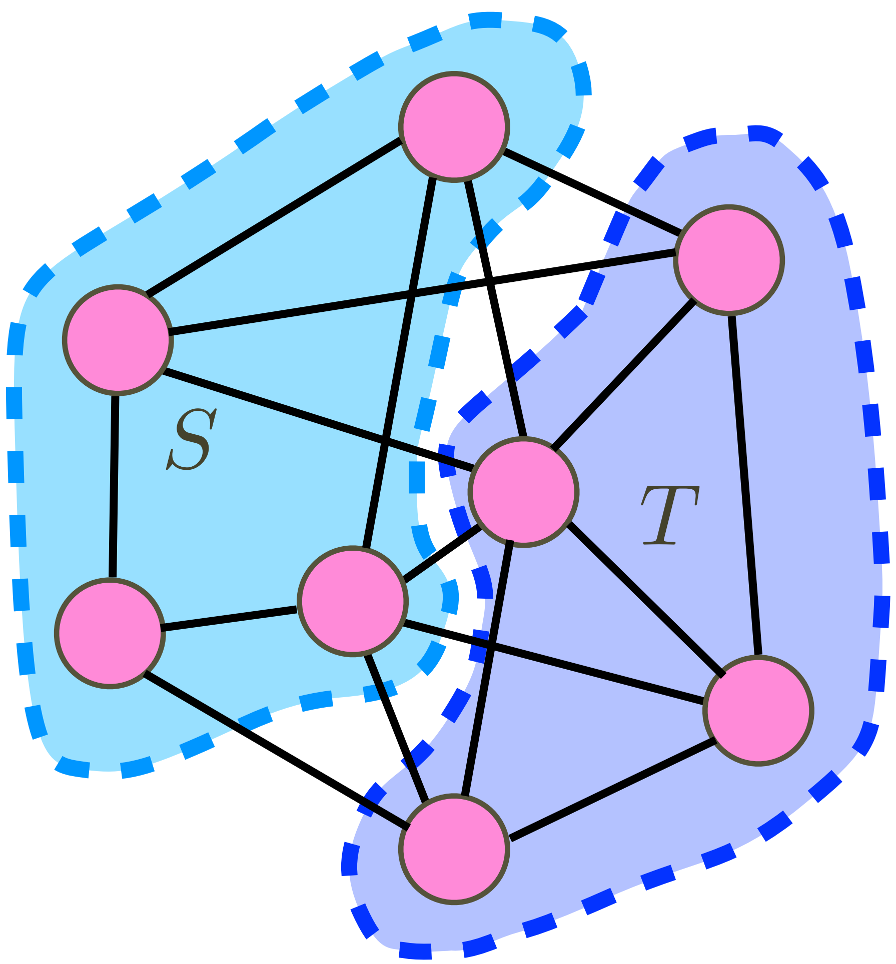
$$\text{s.t.} \quad y_{uv} \le |x_u - x_v|, \qquad \forall uv \in E$$

$$x_v \in \{0,1\}, \qquad \forall v \in V$$

LP for Max-Cut

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.



max $\sum_{uv \in E} y_{uv}$

not linear

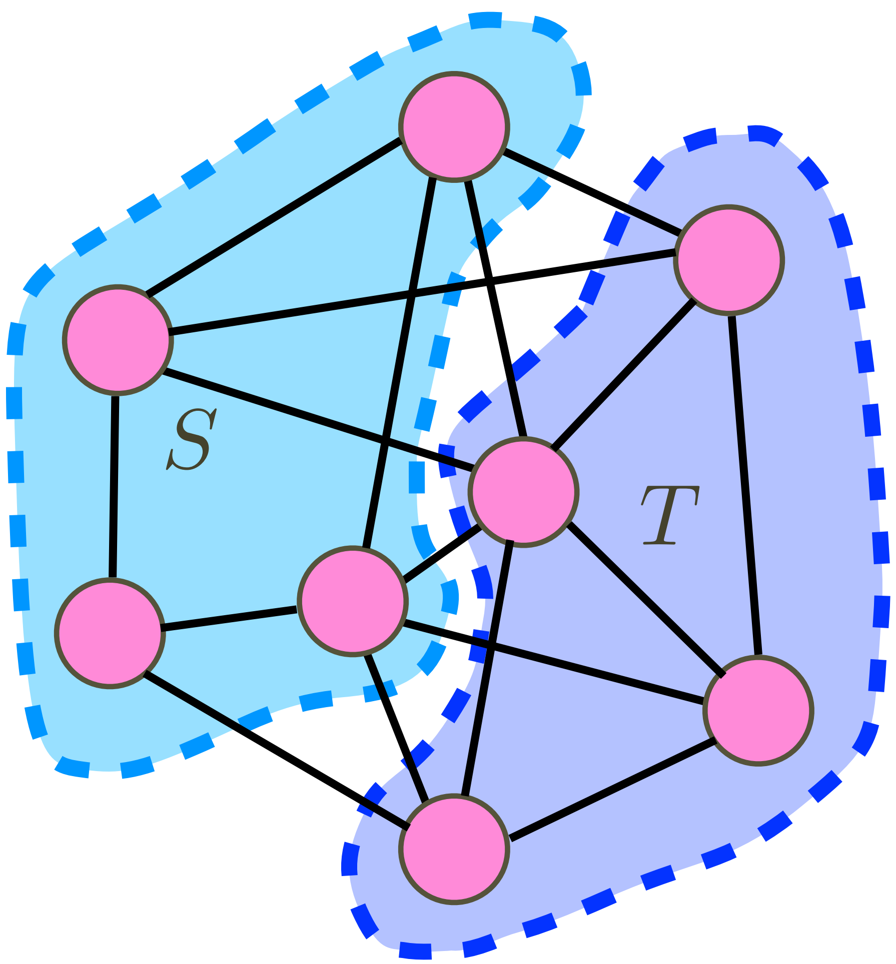
s.t. $y_{uv} \leq |x_u - x_v|, \quad \forall uv \in E$

$x_v \in \{0, 1\}, \quad \forall v \in V$

LP for Max-Cut

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.



$$\max \sum_{uv \in E} y_{uv}$$

s.t.

$$y_{uv} \le y_{uw} + y_{wv}, \qquad \forall u, v, w \in V$$

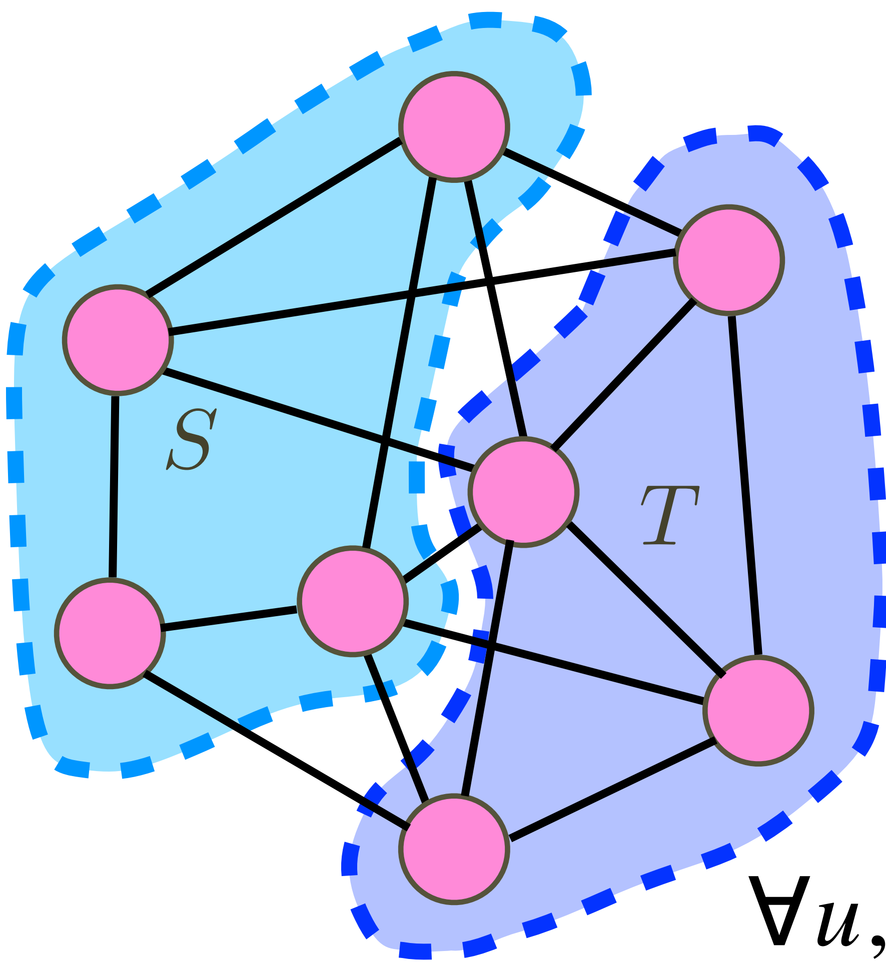
$$y_{uv} + y_{uw} + y_{wv} \le 2, \qquad \forall u, v, w \in V$$

$$y_{uv} \in \{0, 1\}, \qquad \forall u, v \in V$$

LP for Max-Cut

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.



$$\max \sum_{uv \in E} y_{uv}$$

$$\text{s.t.} \quad y_{uv} \le y_{uw} + y_{wv}, \qquad \forall u,v,w \in V$$

$$y_{uv} + y_{uw} + y_{wv} \le 2, \qquad \forall u,v,w \in V$$

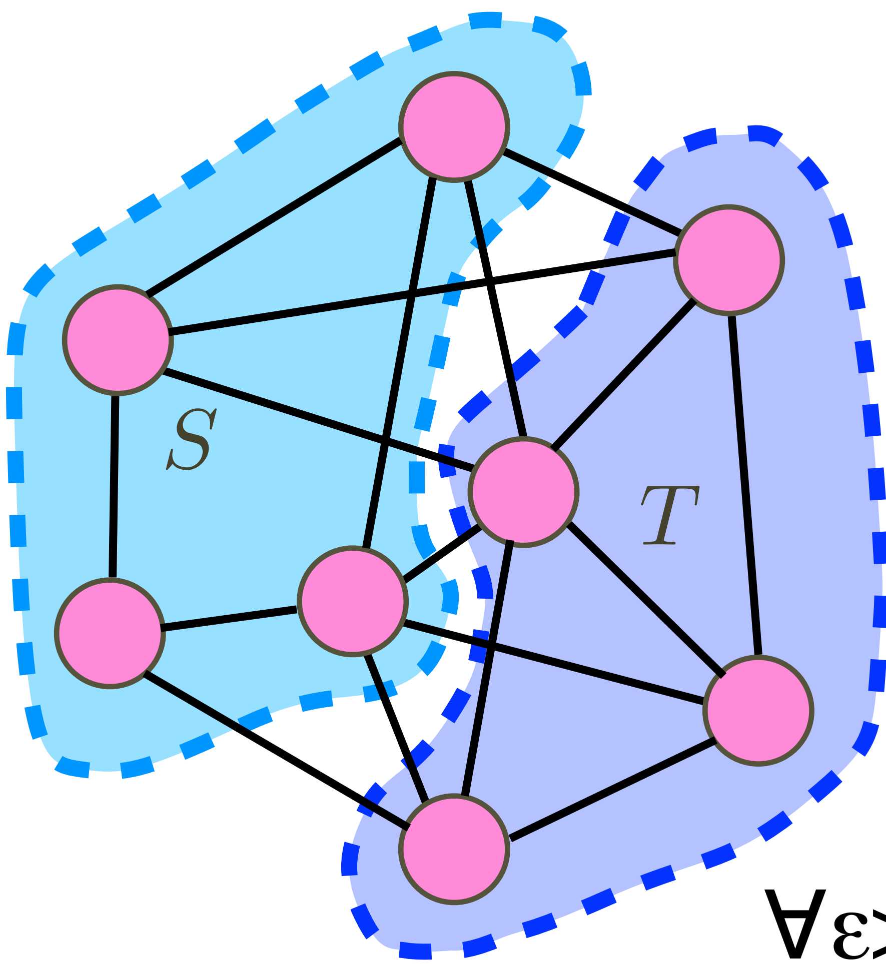
$$y_{uv} \in \{0,1\}, \qquad \forall u,v \in V$$

$\forall u,v,w \in V$: 0 or 2 of $\{u,v\}, \{v,w\}, \{u,w\}$ are "crossing pairs"

LP for Max-Cut

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.



$$\max \quad \sum_{uv \in E} y_{uv}$$

integrality gap ≥ 2

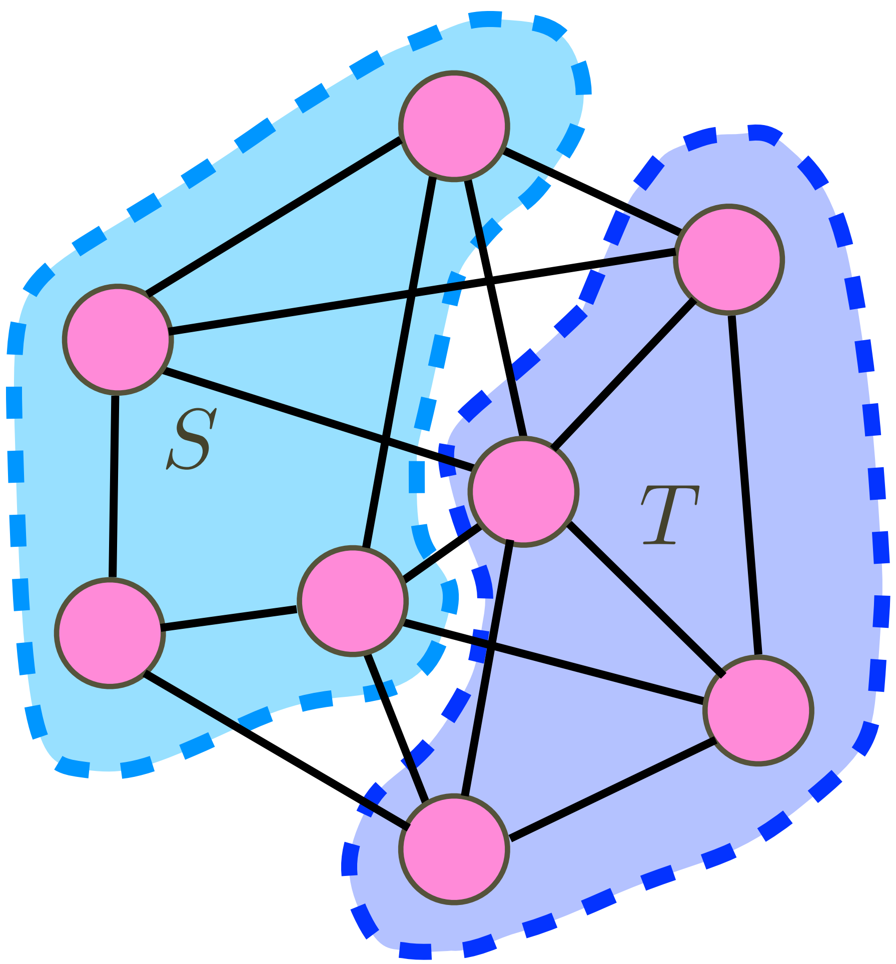
$$\text{s.t.} \quad y_{uv} \le y_{uw} + y_{wv}, \qquad \forall u, v, w \in V$$

$$y_{uv} + y_{uw} + y_{wv} \le 2, \qquad \forall u, v, w \in V$$

$$y_{uv} \in \{0, 1\}, \qquad \forall u, v \in V$$

$\forall \varepsilon > 0$: $\exists G$ s.t. $\mathrm{OPT}_{\mathrm{LP}}(G)/\mathrm{OPT}_{\mathrm{IP}}(G) > 2-\varepsilon$

Quadratic Program for Max-Cut



$$\max \sum_{uv \in E} y_{uv}$$

$$\text{s.t.} \quad y_{uv} \leq |x_u - x_v|, \qquad \forall uv \in E$$

$$x_v \in \{0, 1\}, \qquad \forall v \in V$$

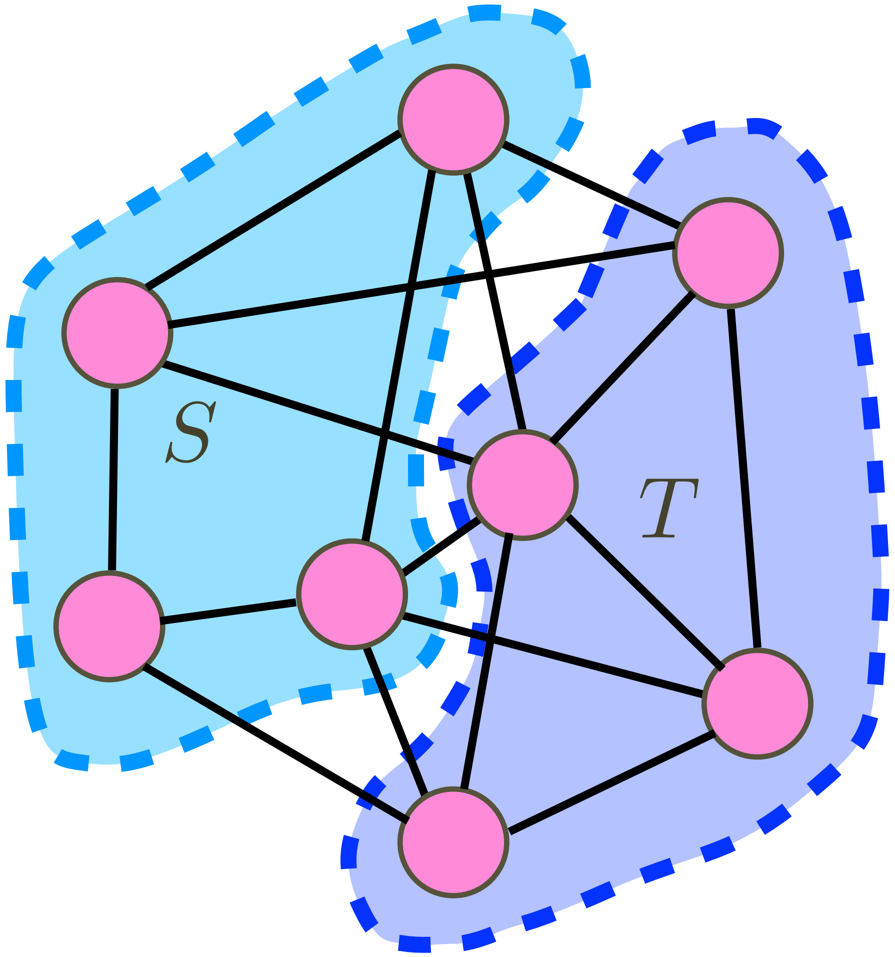
quadratic program:

$$\max \sum_{uv \in E} y_{uv}$$

$$\text{s.t.} \quad y_{uv} \leq \tfrac{1}{2}(1 - x_u x_v), \qquad \forall uv \in E$$

$$x_v \in \{-1, 1\}, \qquad \forall v \in V$$

Quadratic Program for Max-Cut



max $\sum_{uv \in E} y_{uv}$

s.t. $y_{uv} \leq |x_u - x_v|, \quad \forall uv \in E$

$x_v \in \{0, 1\}, \quad \forall v \in V$

strictly quadratic program:

max $\sum_{uv \in E} \frac{1}{2}(1 - x_u x_v)$

s.t. $x_v^2 = 1, \quad \forall v \in V$

Nonlinear, non-convex!

Relaxation

strictly quadratic program:

$$
\begin{aligned}
\max \quad & \frac{1}{2}\sum_{uv\in E}(1-x_u x_v) \\
\text{s.t.} \quad & x_v^2 = 1, \qquad \forall v\in V \\
& x_v \in \mathbb{R}, \qquad \forall v \in V
\end{aligned}
$$

relax to vector program: semidefinite program (SDP)

$$
\begin{aligned}
\max \quad & \frac{1}{2}\sum_{uv\in E}(1-\langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle) \\
\text{s.t.} \quad & \langle \boldsymbol{x}_v, \boldsymbol{x}_v\rangle = 1, \qquad \forall v\in V \\
& \boldsymbol{x}_v \in \mathbb{R}^n, \qquad \forall v \in V
\end{aligned}
$$

inner-products:

$$\langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle = \sum_{i=1}^{n} x_v(i)x_u(i)$$

$n = |V|$

Positive Semidefiniteness

Definition

A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is *positive semidefinite*, denoted $A \succcurlyeq 0$, if $\forall\, \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x} \geq 0$.

Theorem

For symmetric matrix $A \in \mathbb{R}^{n \times n}$:

$A \succcurlyeq 0 \Leftrightarrow$ all eigenvalues $\lambda(A) \geq 0 \Leftrightarrow \exists\, B \in \mathbb{R}^{n \times n}, A = B^{\mathrm{T}} B$.

Semidefinite Programing (SDP)

$$C, A_1, \ldots, A_k \in \mathbb{R}^{n \times n}, \quad b_1, b_2, \ldots, b_k \in \mathbb{R} :$$

maximize $\quad \mathrm{tr}(C^T Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} y_{ij}$

subject to $\quad \mathrm{tr}(A_r^T Y) \leq b_r, \qquad \forall 1 \leq r \leq k$

$\quad Y \succeq 0,$

symmetric $\; Y \in \mathbb{R}^{n \times n}.$

Semidefinite Programing (SDP)

$$C, A_1, \ldots, A_k \in \mathbb{R}^{n \times n}, \quad b_1, b_2, \ldots, b_k \in \mathbb{R} :$$

maximize $\quad \mathrm{tr}(C^T Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} y_{ij}$

subject to $\quad \mathrm{tr}(A_r^T Y) \leq b_r, \qquad \forall 1 \leq r \leq k$

$Y \succeq 0,$

symmetric $\; Y \in \mathbb{R}^{n \times n}.$

⇔

$Y = V^T V$

$V \in \mathbb{R}^{n \times n}$

V's column vectors:

$\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$

Semidefinite Programing (SDP)

$$C, A_1, \ldots, A_k \in \mathbb{R}^{n \times n}, \quad b_1, b_2, \ldots, b_k \in \mathbb{R} :$$

maximize $\quad \mathrm{tr}(C^T Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} y_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$

subject to $\quad \mathrm{tr}(A_r^T Y) \leq b_r, \qquad \forall 1 \leq r \leq k$

$Y \succeq 0,$

symmetric $\; Y \in \mathbb{R}^{n \times n}.$

$\Longleftrightarrow$

$Y = V^T V$

$V \in \mathbb{R}^{n \times n}$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(r)} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle \leq b_r$$

V's column vectors: $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$

Semidefinite Programing (SDP)

vector program:

maximize $$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle$$

subject to $$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^{(r)}\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle \le b_r \qquad \forall 1 \le r \le k$$

$$\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$$

Semidefinite Programing (SDP)

vector program: LP for inner products

maximize $$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle$$

linear combinations of inner products

subject to $$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^{(r)}\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle \le b_r \qquad \forall 1 \le r \le k$$

$$\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$$

Semidefinite Programing (SDP)

vector program: LP for inner products

maximize $\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle$

subject to $\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^{(r)}\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle \le b_r \quad \forall 1 \le r \le k$

$\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$

linear combinations of inner products

- SDPs generalize LPs: $v_i = (x_i, 0, \ldots, 0)$
- SDPs are subclass of convex programs.
- "Efficiently solvable" by the ellipsoid method:
  find OPT±ε in time poly(n ,1/ε)

SDP Relaxation

Max-Cut: undirected graph $G(V,E)$, $n = |V|$

$$\begin{aligned} \max \quad & \frac{1}{2}\sum_{uv\in E}(1 - x_u x_v) \\ \text{s.t.} \quad & x_v^2 = 1, \qquad \forall v \in V \\ & x_v \in \mathbb{R}, \qquad \forall v \in V \end{aligned}$$

semidefinite program (SDP) relaxation:

$$\begin{aligned} \max \quad & \frac{1}{2}\sum_{uv\in E}(1 - \langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle) \\ \text{s.t.} \quad & \langle \boldsymbol{x}_v, \boldsymbol{x}_v \rangle = 1, \qquad \forall v \in V \\ & \boldsymbol{x}_v \in \mathbb{R}^n, \qquad \forall v \in V \end{aligned}$$

SDP Relaxation

Max-Cut: undirected graph $G(V,E)$, $n = |V|$

$$\begin{aligned} \text{max} \quad & \frac{1}{2} \sum_{uv \in E} (1 - x_u x_v) \\ \text{s.t.} \quad & x_v^2 = 1, \qquad \forall v \in V \\ & x_v \in \mathbb{R}, \qquad \forall v \in V \end{aligned}$$

semidefinite program (SDP) relaxation:

$$\begin{aligned} \text{max} \quad & \frac{1}{2} \sum_{uv \in E} (1 - \langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle) \\ \text{s.t.} \quad & \|\boldsymbol{x}_v\|_2 = 1, \qquad \forall v \in V \\ & \boldsymbol{x}_v \in \mathbb{R}^n, \qquad \forall v \in V \end{aligned}$$

Rounding

Max-Cut: undirected graph $G(V,E)$, $n = |V|$

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1 - x_u x_v)$$

$$\text{s.t.} \quad x_v^2 = 1, \quad \forall v \in V$$

$$x_v \in \mathbb{R}, \quad \forall v \in V$$

Goemans-Williamson'95:

SDP relaxation:

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1 - \langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle)$$

$$\text{s.t.} \quad \|\boldsymbol{x}_v\|_2 = 1, \quad \forall v \in V$$

$$\boldsymbol{x}_v \in \mathbb{R}^n, \quad \forall v \in V$$

SDP optimal solution: $\boldsymbol{x}_v^*$

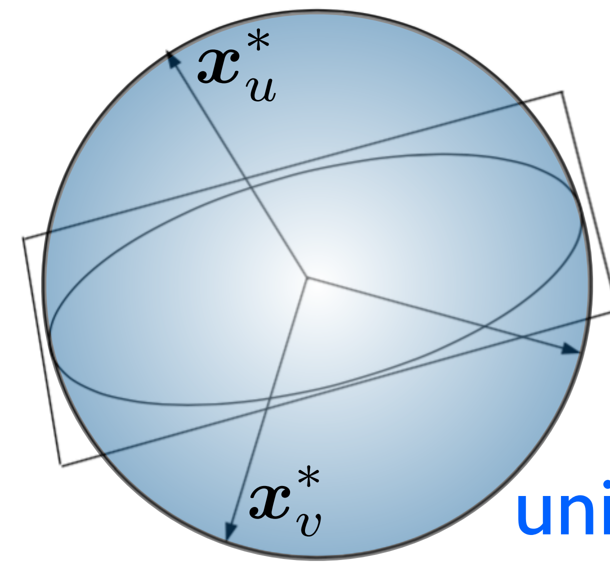
Rounding

Max-Cut: undirected graph $G(V,E)$, $n = |V|$

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1 - x_u x_v)$$

$$\text{s.t.} \quad x_v^2 = 1, \quad \forall v \in V$$

$$x_v \in \mathbb{R}, \quad \forall v \in V$$



Goemans-Williamson'95:

SDP relaxation:

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1 - \langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle)$$

$$\text{s.t.} \quad \|\boldsymbol{x}_v\|_2 = 1, \quad \forall v \in V$$

$$\boldsymbol{x}_v \in \mathbb{R}^n, \quad \forall v \in V$$

SDP optimal solution: $\boldsymbol{x}_v^*$

rounding:

uniform random unit vector

$$\boldsymbol{u} \in \mathbb{R}^n, \quad \|\boldsymbol{u}\|_2 = 1$$

$$\hat{x}_v = \text{sgn}(\langle \boldsymbol{x}_v^*, \boldsymbol{u} \rangle)$$

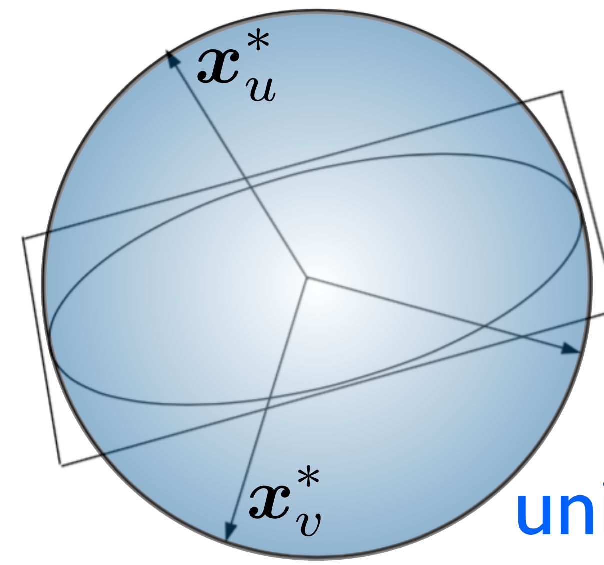
Rounding

Max-Cut: undirected graph $G(V,E)$, $n = |V|$

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1 - x_u x_v)$$

$$\text{s.t.} \quad x_v^2 = 1, \quad \forall v \in V$$

$$x_v \in \mathbb{R}, \quad \forall v \in V$$



Goemans-Williamson'95:

SDP relaxation:

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1 - \langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle)$$

$$\text{s.t.} \quad \|\boldsymbol{x}_v\|_2 = 1, \quad \forall v \in V$$

$$\boldsymbol{x}_v \in \mathbb{R}^n, \quad \forall v \in V$$

SDP optimal solution: $\boldsymbol{x}_v^*$

rounding:

random $\boldsymbol{r} = (r_1, r_2, \ldots, r_n) \in \mathbb{R}^n$

each $r_i \sim N(0,1)$ *i.i.d.*

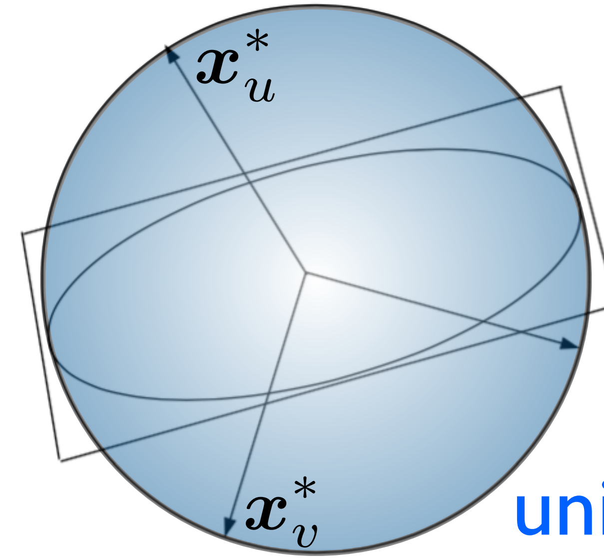
normal distribution

$$\hat{x}_v = \operatorname{sgn}(\langle \boldsymbol{x}_v^*, \boldsymbol{r}\rangle)$$

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1-x_u x_v)$$

$$\text{s.t.} \quad x_v^2 = 1, \qquad \forall v \in V$$

$$x_v \in \mathbb{R}, \qquad \forall v \in V$$



SDP relaxation:

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1-\langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle)$$

$$\text{s.t.} \quad \|\boldsymbol{x}_v\|_2 = 1, \qquad \forall v \in V$$

$$\boldsymbol{x}_v \in \mathbb{R}^n, \qquad \forall v \in V$$

SDP optimal solution: $\boldsymbol{x}_v^*$

rounding:

random $\boldsymbol{r} = (r_1, r_2, \ldots, r_n) \in \mathbb{R}^n$

each $r_i \sim N(0,1)$ *i.i.d.*

normal distribution

$$\hat{x}_v = \operatorname{sgn}(\langle \boldsymbol{x}_v^*, \boldsymbol{r}\rangle)$$

$\boldsymbol{u} = \dfrac{\boldsymbol{r}}{\|\boldsymbol{r}\|_2}$ is uniform random unit vector

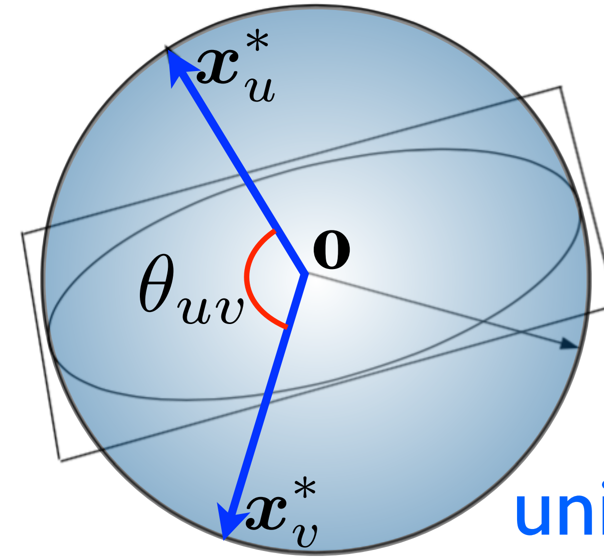
spherically symmetric:

$$\Pr[(r_1,\ldots,r_n)] = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-r_i^2/2} = (2\pi)^{-d/2} e^{-\|\boldsymbol{r}\|^2/2}$$

max $\frac{1}{2}\sum_{uv\in E}(1-x_u x_v)$

s.t. $x_v^2 = 1, \quad \forall v\in V$

$x_v \in \mathbb{R}, \quad \forall v\in V$



SDP relaxation:

max $\frac{1}{2}\sum_{uv\in E}(1-\langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle)$

s.t. $\|\boldsymbol{x}_v\|_2 = 1, \quad \forall v\in V$

$\boldsymbol{x}_v \in \mathbb{R}^n, \quad \forall v\in V$

SDP optimal solution: $\boldsymbol{x}_v^*$

rounding:

random $\boldsymbol{r} = (r_1, r_2, \ldots, r_n) \in \mathbb{R}^n$

each $r_i \sim N(0,1)$ *i.i.d.*

normal distribution

$\hat{x}_v = \mathrm{sgn}(\langle \boldsymbol{x}_v^*, \boldsymbol{r}\rangle)$

$$\mathbf{E}[\mathrm{cut}] = \sum_{uv\in E}\Pr[\mathrm{sgn}(\langle \boldsymbol{x}_u^*\cdot \boldsymbol{r}\rangle)\neq \mathrm{sgn}(\langle \boldsymbol{x}_v^*\cdot \boldsymbol{r}\rangle)] = \sum_{uv\in E}\frac{\theta_{uv}}{\pi}$$

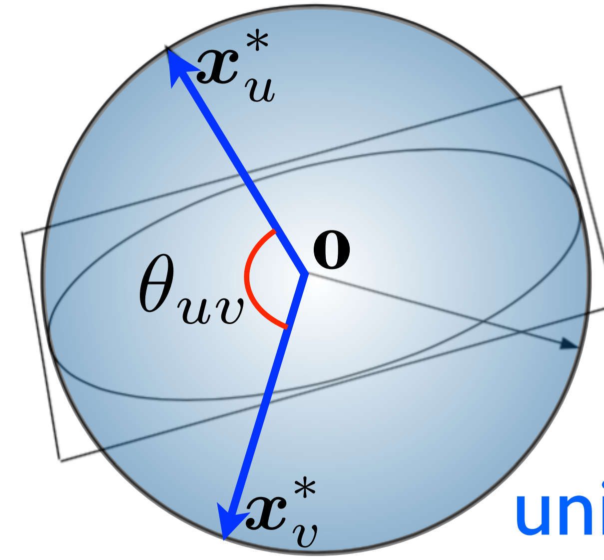
where $\theta_{uv} = \angle \boldsymbol{x}_u^* \boldsymbol{o} \boldsymbol{x}_v^*$

$$\|\boldsymbol{x}_u^*\|\cdot\|\boldsymbol{x}_v^*\|\cdot \cos\theta_{uv} = \langle \boldsymbol{x}_u^*, \boldsymbol{x}_v^*\rangle$$

max $\frac{1}{2}\sum_{uv\in E}(1-x_u x_v)$

s.t. $x_v^2 = 1, \quad \forall v \in V$

$x_v \in \mathbb{R}, \quad \forall v \in V$



SDP relaxation:

max $\frac{1}{2}\sum_{uv\in E}(1-\langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle)$

s.t. $\|\boldsymbol{x}_v\|_2 = 1, \quad \forall v \in V$

$\boldsymbol{x}_v \in \mathbb{R}^n, \quad \forall v \in V$

SDP optimal solution: $\boldsymbol{x}_v^*$

rounding:

random $\boldsymbol{r} = (r_1, r_2, \ldots, r_n) \in \mathbb{R}^n$

each $r_i \sim N(0,1)$ *i.i.d.*

normal distribution

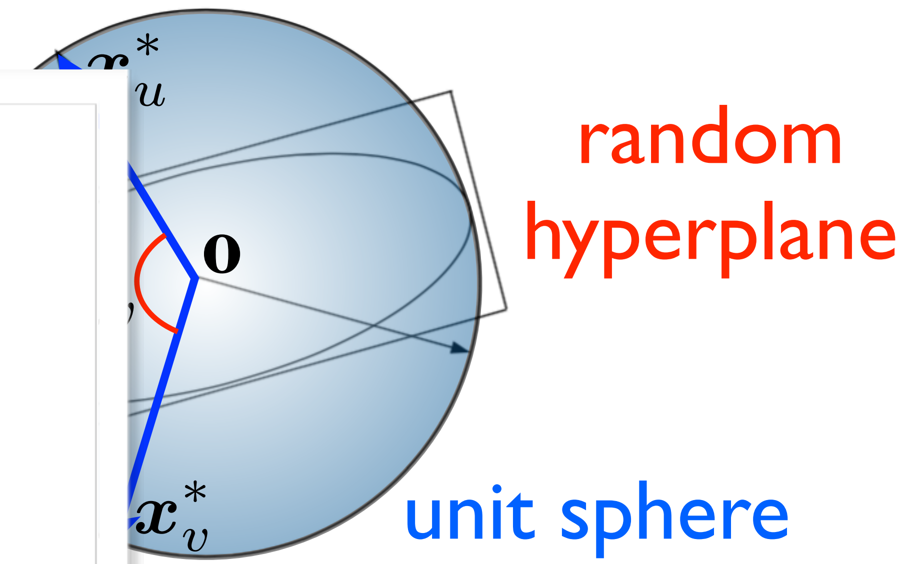
$\hat{x}_v = \mathrm{sgn}(\langle \boldsymbol{x}_v^*, \boldsymbol{r}\rangle)$

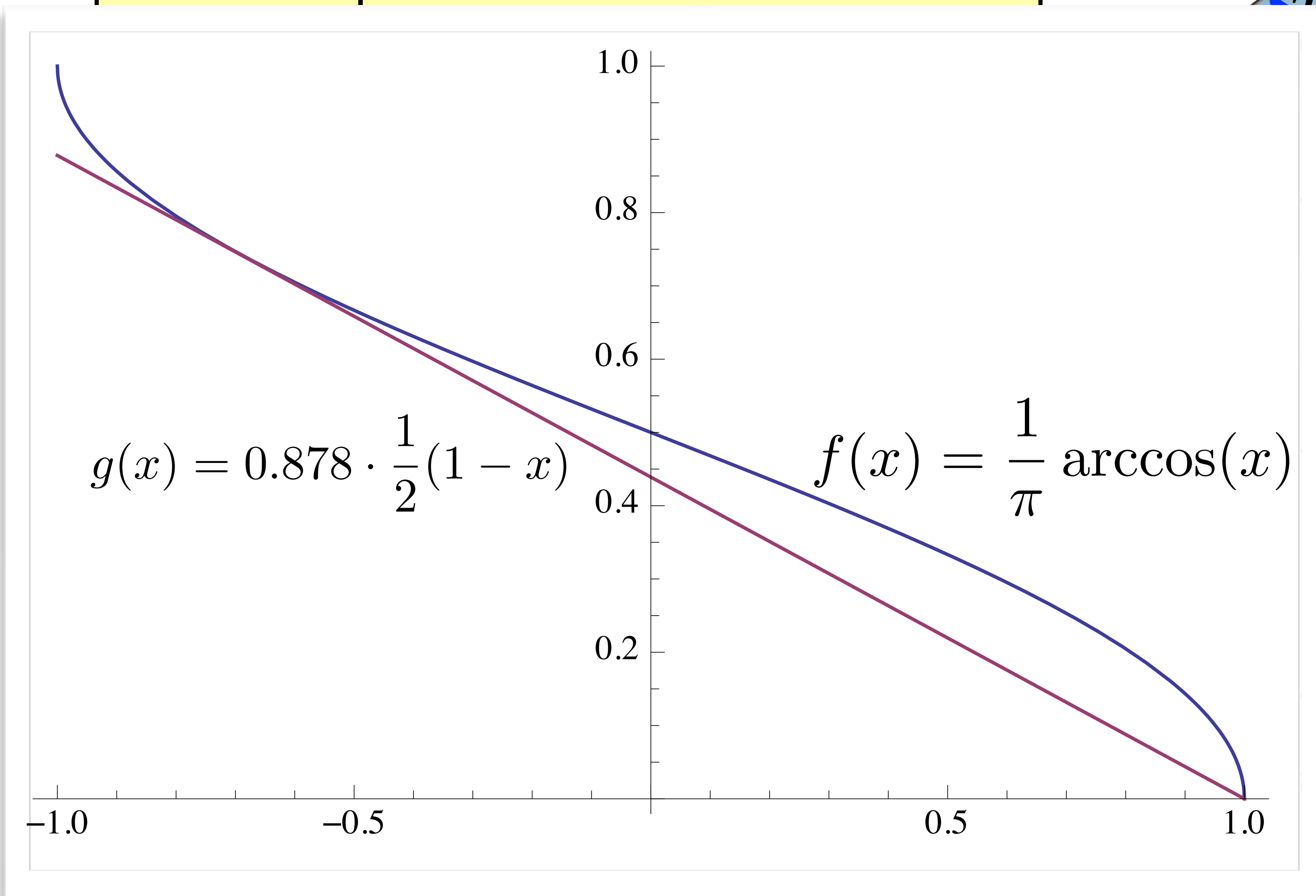
$$\mathbf{E}[\mathrm{cut}] = \sum_{uv\in E} \Pr[\mathrm{sgn}(\langle \boldsymbol{x}_u^* \cdot \boldsymbol{r}\rangle) \neq \mathrm{sgn}(\langle \boldsymbol{x}_v^* \cdot \boldsymbol{r}\rangle)] = \sum_{uv\in E} \frac{\theta_{uv}}{\pi}$$

$$= \sum_{uv\in E} \frac{\arccos\langle \boldsymbol{x}_u^*, \boldsymbol{x}_v^*\rangle}{\pi}$$

where $\theta_{uv} = \angle \boldsymbol{x}_u^* \boldsymbol{o} \boldsymbol{x}_v^*$

$\cos\theta_{uv} = \langle \boldsymbol{x}_u^*, \boldsymbol{x}_v^*\rangle$





$, \ldots, r_n) \in \mathbb{R}^n$

$(0, 1)$ *i.i.d.*

l distribution

SDP optimal solution: $\boldsymbol{x}_v^*$ $\quad \hat{x}_v = \mathrm{sgn}(\langle \boldsymbol{x}_v^*, \boldsymbol{r} \rangle)$

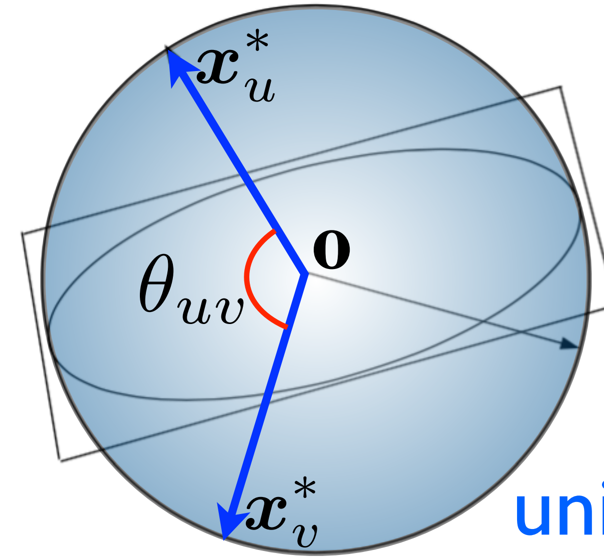
$$\mathbf{E}[\text{cut}] = \sum_{uv \in E} \frac{\arccos\langle \boldsymbol{x}_u^*, \boldsymbol{x}_v^* \rangle}{\pi} \geq \alpha \sum_{uv \in E} \frac{1}{2}(1 - \langle \boldsymbol{x}_u^*, \boldsymbol{x}_v^* \rangle)$$

where $\alpha = \inf_{x \in [-1,1]} \frac{2\arccos(x)}{\pi(1-x)}$ = 0.87856...

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1 - x_u x_v)$$

$$\text{s.t.} \quad x_v^2 = 1, \quad \forall v \in V$$

$$x_v \in \mathbb{R}, \quad \forall v \in V$$



SDP relaxation:

$$\max \quad \frac{1}{2}\sum_{uv\in E}(1 - \langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle)$$

$$\text{s.t.} \quad \|\boldsymbol{x}_v\|_2 = 1, \quad \forall v \in V$$

$$\boldsymbol{x}_v \in \mathbb{R}^n, \quad \forall v \in V$$

SDP optimal solution: $\boldsymbol{x}_v^*$

rounding:

random $\boldsymbol{r} = (r_1, r_2, \ldots, r_n) \in \mathbb{R}^n$

each $r_i \sim N(0,1)$ *i.i.d.*

normal distribution

$$\hat{x}_v = \mathrm{sgn}(\langle \boldsymbol{x}_v^*, \boldsymbol{r}\rangle)$$

$$\mathbf{E}[\text{cut}] = \sum_{uv\in E} \frac{\arccos\langle \boldsymbol{x}_u^*, \boldsymbol{x}_v^*\rangle}{\pi} \ge \alpha \sum_{uv\in E}\frac{1}{2}(1 - \langle \boldsymbol{x}_u^*, \boldsymbol{x}_v^*\rangle) = \alpha\ \mathrm{OPT}_{\mathrm{SDP}}$$

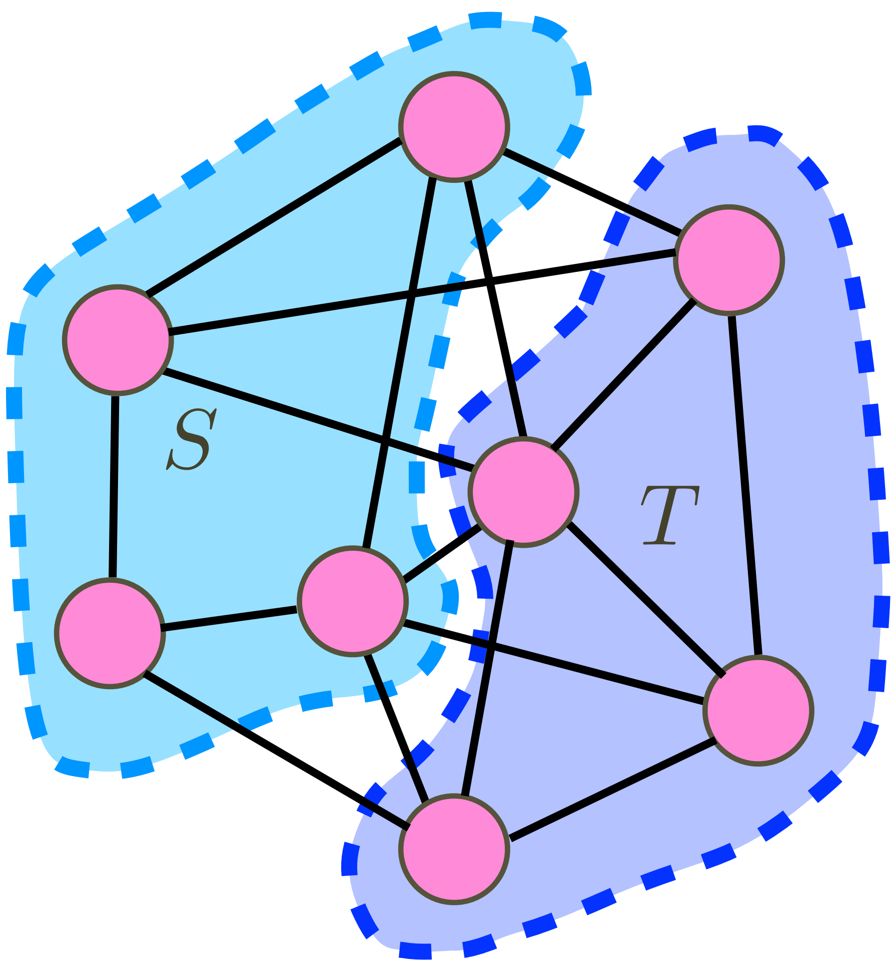
$$\ge \alpha\ \mathrm{OPT}$$

where $\alpha = \inf_{x\in[-1,1]} \frac{2\arccos(x)}{\pi(1-x)} = 0.87856\ldots$

Max-Cut

Instance: An undirected graph $G(V,E)$

Find a *bipartition* of V into S and T that maximize the size of the *cut* $E(S,T) = \{uv \in E \mid u \in S, v \in T\}$.

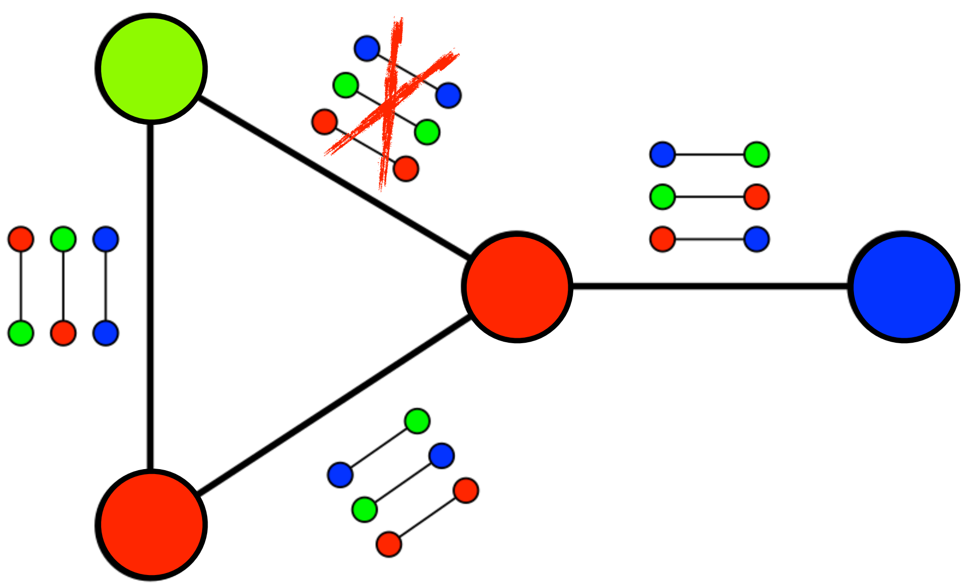


- One of Karp's 21 **NPC** problems.
- Typical Max-CSP.
- Rounding SDP relaxation is $0.878\sim$-approximate.
- Assuming the unique games conjecture: no poly-time algorithm with approx. ratio. $< 0.878\sim$

Unique Games Conjecture

Unique Label Cover (ULC):

Instance: An undirected graph $G(V,E)$; q colors; each edge e associated with a bijection ϕ_e: $[q]\rightarrow[q]$. A coloring $\sigma\in[q]^V$ satisfies the constraint of the edge $e=(u,v) \in E$ if $\phi_e(\sigma_u)= \phi_e(\sigma_v)$.



Unique Games Conjecture:
(Khot 2002)

$\forall\varepsilon$, $\exists q$ such that it is ~~NP~~ ??? -hard to distinguish between ULC instances:

- $>1-\varepsilon$ fraction of edges satisfied by a coloring;
- no more than ε fraction of edges satisfied by any coloring;

Constraint Satisfaction Problem

- variables: $X = \{x_1, x_2, \ldots, x_n\}$
- domain: Ω, usually $\Omega = [q]$ for a finite q
- constraints: $c = (\psi, S)$ where ψ: $\Omega^k \to \{0,1\}$ and scope $S \subseteq X$ is a subset of k variables
- CSP instance: a set of constraints defined on X
- assignment: $\sigma \in \Omega^X$ assigns values to variables
- a constraint $c = (\psi, S)$ is *satisfied* if $\psi(\sigma_S) = 1$
- examples:
  - max-cut: q=2, constraints are $\neq$
  - k-SAT: q=2, constraints are k-clauses
  - matching/cover: q=2, constraints are $\sum\leq 1$ (or $\sum\geq 1$)
  - k-coloring: q=k, constraints are $\neq$
  - graph homomorphism: constraint is adjacency matrix

Constraint Satisfaction Problem

- variables: $X = \{x_1, x_2, \ldots, x_n\}$
- domain: Ω, usually $\Omega = [q]$ for a finite q
- constraints: $c = (\psi, S)$ where $\psi\colon \Omega^k \to \{0,1\}$ and scope $S \subseteq X$ is a subset of k variables
- CSP instance: a set of constraints defined on X
- assignment: $\sigma \in \Omega^X$ assigns values to variables
- a constraint $c = (\psi, S)$ is *satisfied* if $\psi(\sigma_S) = 1$
- examples:
  - unique games: $\Omega = [q]$, each constraint is an arbitrary binary *bijective* predicate:
    $\psi\colon \Omega^2 \to \{0,1\}$ where $\forall a \in \Omega$, $\exists$ unique $b \in \Omega$, $\psi(a,b)=1$

Algorithmic Problems for CSP

Given a CSP instance I:

- satisfiability: decide whether ∃ an assignment satisfying all constraints
  - search: find such a satisfying assignment
- optimization: find an assignment satisfying as many constraints as possible
  - refutation (dual): find a "proof" of "no assignment can satisfy $>m$ constraints" for m as small as possible
- counting: estimate the number of satisfying assignments
  - sampling: random sample a satisfying assignments
  - inference: observing part of a satisfying assignment, guess the value of an unobserved variable

Algorithmic Problems for CSP

CSP	Satisfiability	optimization	counting
2SAT	**P**	**NP**-hard	**#P**-complete
3SAT	**NP**-complete	**NP**-hard	**#P**-complete
matching	perfect matching **P**	max matching **P**	**#P**-complete
cut (2-coloring)	bipartite test **P**	max-cut **NP**-hard	**FP** (poly-time)
3-coloring	**NP**-complete	max-3-cut **NP**-hard	**#P**-complete

A Wishlist for Optimization Algorithms

- Nonlinear, non-convex objectives.
- Powerful enough to tackle hard problems in a systematic way, and meanwhile is still practical.
- Becoming more accurate as we're paying more (but certainly won't beat the inapproximability).
- A generic framework that can be applied obviously to various problems.

sum-of-squares (SoS) SDP, Lasserre hierarchy,
Lovász-Schrijver hierarchy, ...