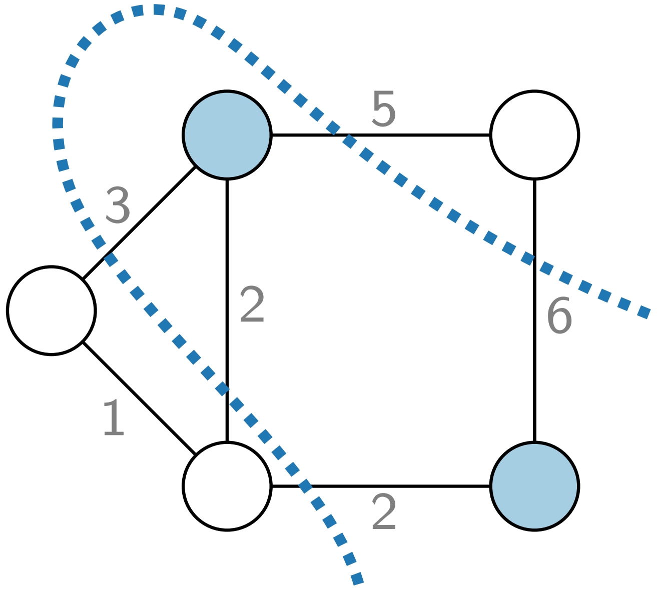
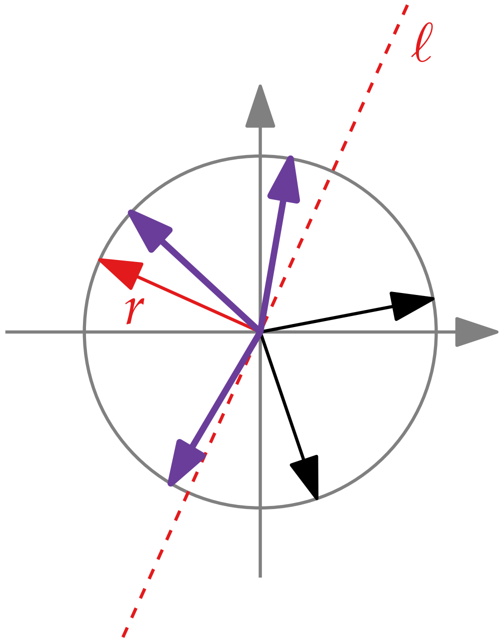
Advanced Algorithms

QP-Relaxation

for MaxCut



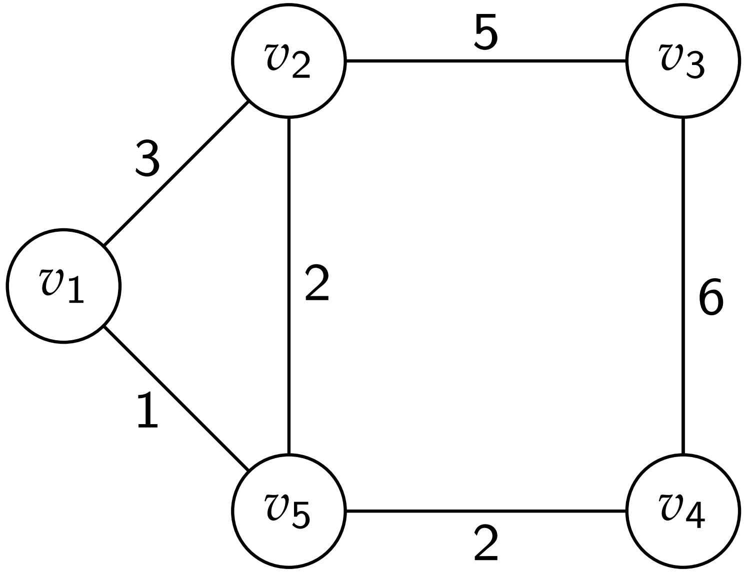


Alexander Wolff

Summer term 2026

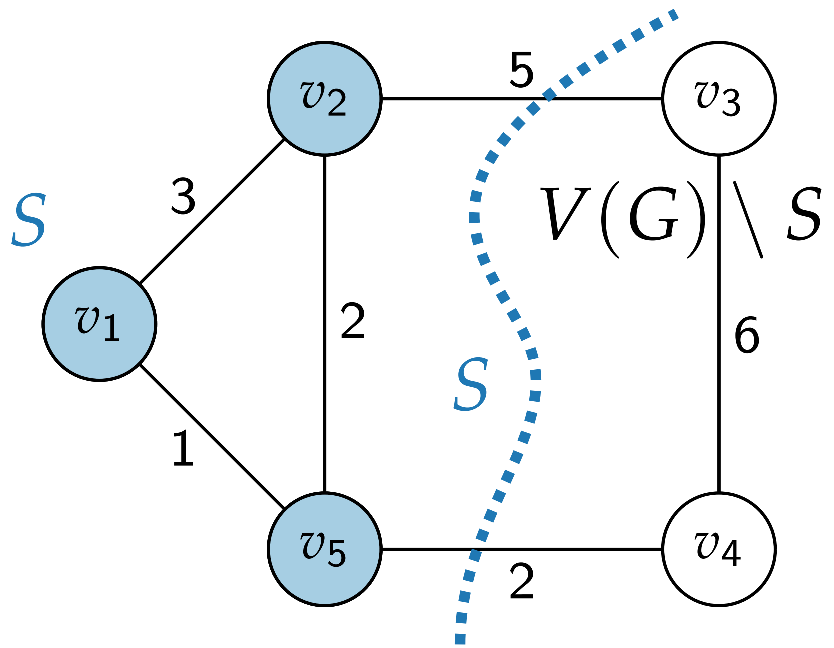
Cut

- Let G be a graph with integral edge weights $w\colon E(G) \to \mathbb{N}$.



Cut

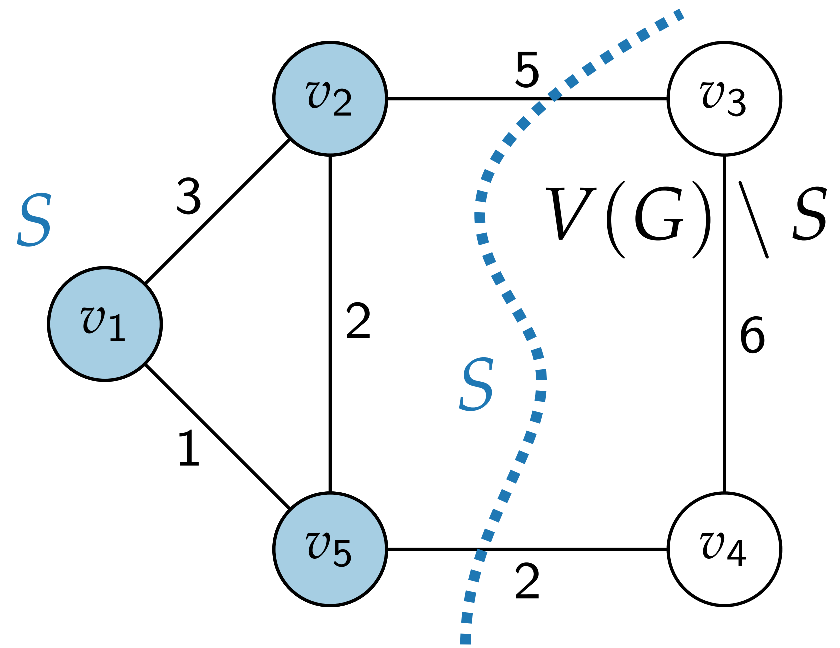
- Let G be a graph with integral edge weights $w\colon E(G) \to \mathbb{N}$.
- A **cut** of G is a partition $(S, V(G) \setminus S)$ of $V(G)$ with $\emptyset \neq S \neq V(G)$.



Cut

- Let G be a graph with integral edge weights $w\colon E(G) \to \mathbb{N}$.
- A **cut** of G is a partition $(S, V(G) \setminus S)$ of $V(G)$ with $\emptyset \neq S \neq V(G)$.
- The **weight** of a cut $(S, V(G) \setminus S)$ is

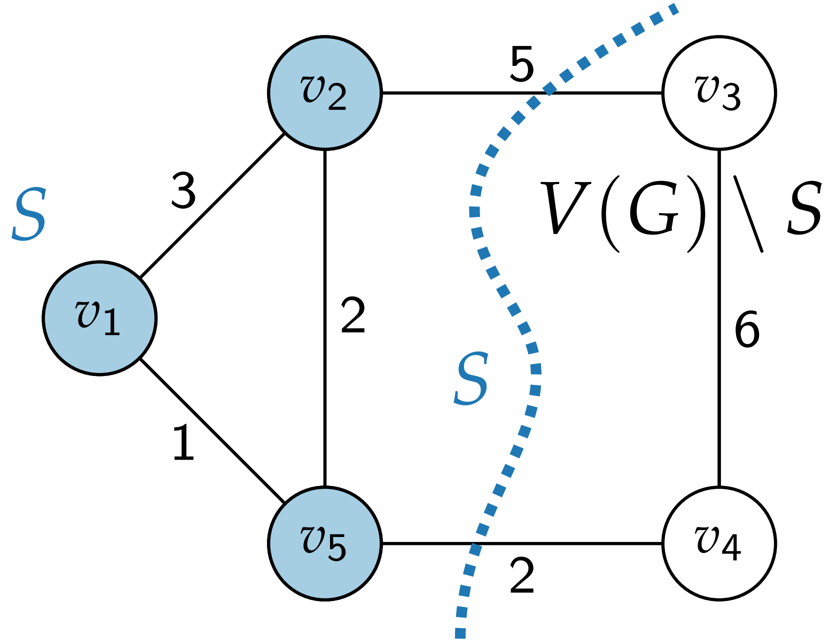
$$w(S) = \sum_{\substack{uv \in E,\\ u \in S, v \notin S}} w(uv)$$



Cut

- Let G be a graph with integral edge weights $w\colon E(G) \to \mathbb{N}$.
- A **cut** of G is a partition $(S, V(G) \setminus S)$ of $V(G)$ with $\emptyset \neq S \neq V(G)$.
- The **weight** of a cut $(S, V(G) \setminus S)$ is

$$w(S) = \sum_{\substack{uv \in E, \\ u \in S, v \notin S}} w(uv)$$

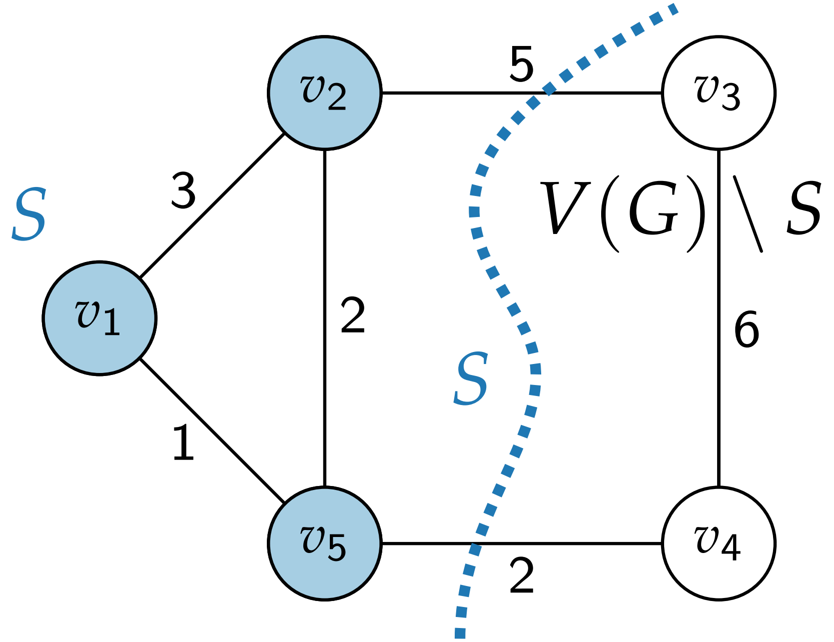


$$w(\{v_1, v_2, v_5\}) = w(\{v_3, v_4\})$$
$$=$$

Cut

- Let G be a graph with integral edge weights $w\colon E(G) \to \mathbb{N}$.
- A **cut** of G is a partition $(S, V(G) \setminus S)$ of $V(G)$ with $\emptyset \neq S \neq V(G)$.
- The **weight** of a cut $(S, V(G) \setminus S)$ is

$$w(S) = \sum_{\substack{uv \in E, \\ u \in S, v \notin S}} w(uv)$$

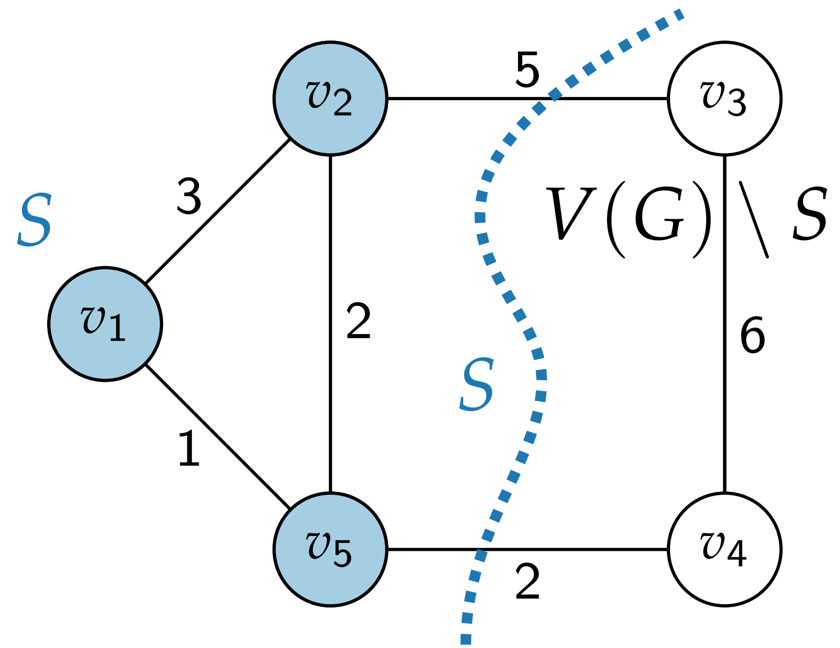


$$w(\{v_1, v_2, v_5\}) = w(\{v_3, v_4\})$$
$$= w(v_2 v_3) + w(v_4 v_5)$$

Cut

- Let G be a graph with integral edge weights $w\colon E(G) \to \mathbb{N}$.
- A **cut** of G is a partition $(S, V(G) \setminus S)$ of $V(G)$ with $\emptyset \neq S \neq V(G)$.
- The **weight** of a cut $(S, V(G) \setminus S)$ is

$$w(S) = \sum_{\substack{uv \in E,\\ u \in S, v \notin S}} w(uv)$$

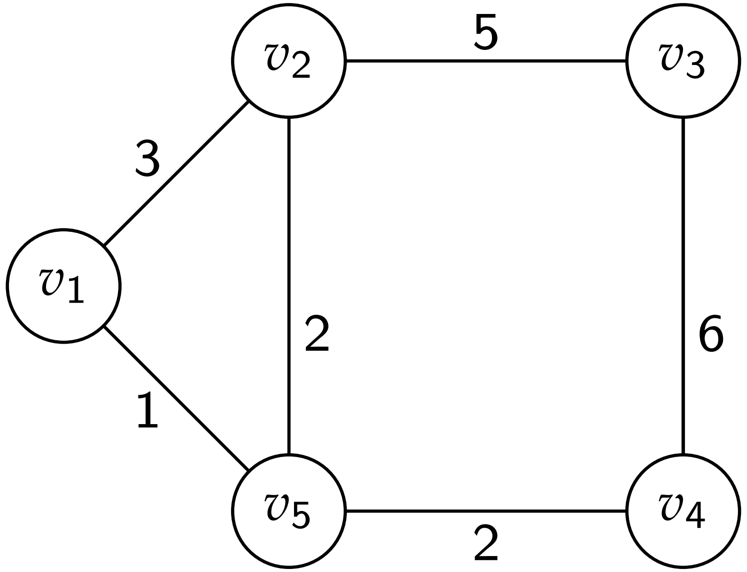


$$w(\{v_1, v_2, v_5\}) = w(\{v_3, v_4\})$$
$$= w(v_2v_3) + w(v_4v_5) = 7$$

The **MinCut** Problem

Input. Graph G, edge weights $w\colon E(G) \to \mathbb{N}$.

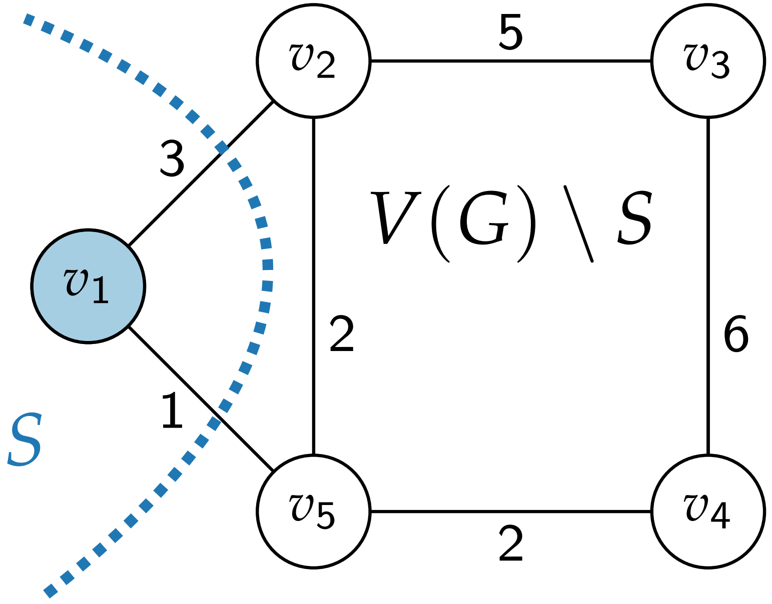
Output. Cut $(S, V(G) \setminus S)$ of G of **minimum** weight.



The **MinCut** Problem

Input. Graph G, edge weights $w\colon E(G) \to \mathbb{N}$.

Output. Cut $(S, V(G) \setminus S)$ of G of **minimum** weight.



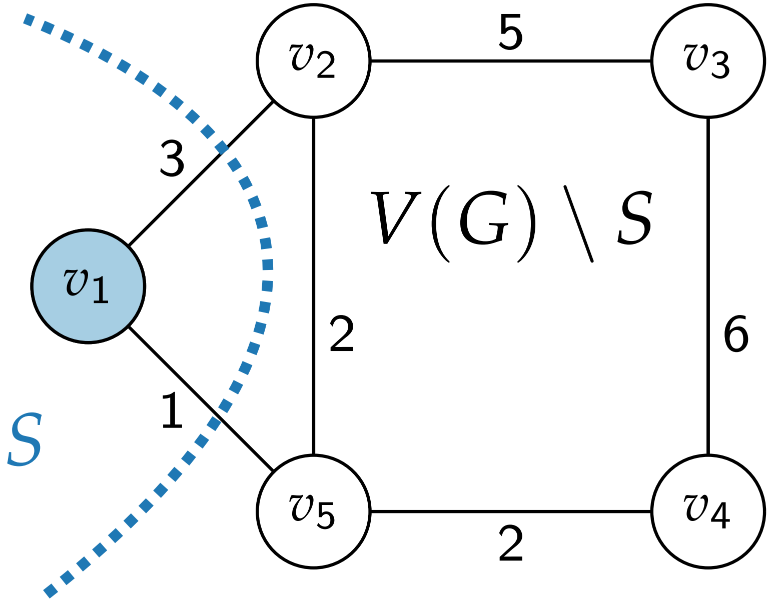
$w(S) = 4$

The **MinCut** Problem

Input. Graph G, edge weights $w \colon E(G) \to \mathbb{N}$.

Output. Cut $(S, V(G) \setminus S)$ of G of **minimum** weight.

- Has applications in flow networks (*max-flow min-cut theorem*), finding a bottleneck in a network, graph partition problems, clustering, . . .



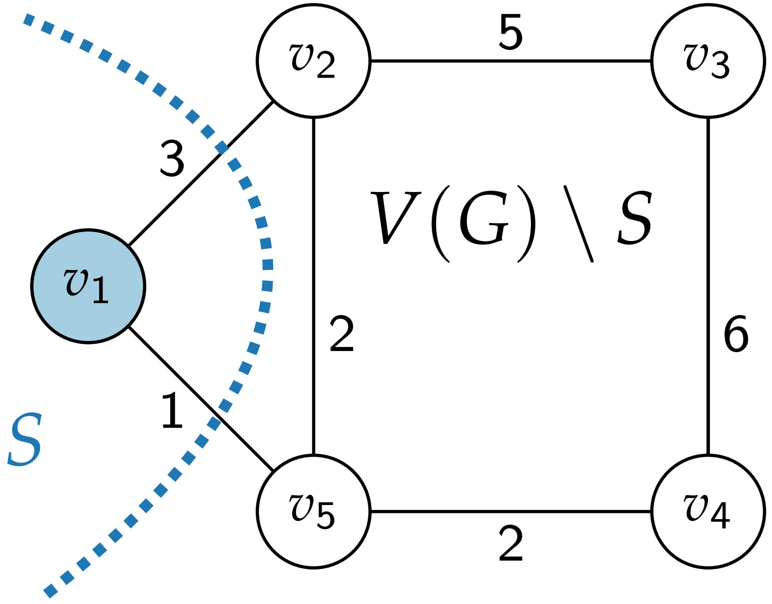
$w(S) = 4$

The **MinCut** Problem

Input. Graph G, edge weights $w \colon E(G) \to \mathbb{N}$.

Output. Cut $(S, V(G) \setminus S)$ of G of **minimum** weight.

- Has applications in flow networks (*max-flow min-cut theorem*), finding a bottleneck in a network, graph partition problems, clustering, . . .
- Can be solved optimally in polynomial time, e.g., by the Stoer–Wagner algorithm.

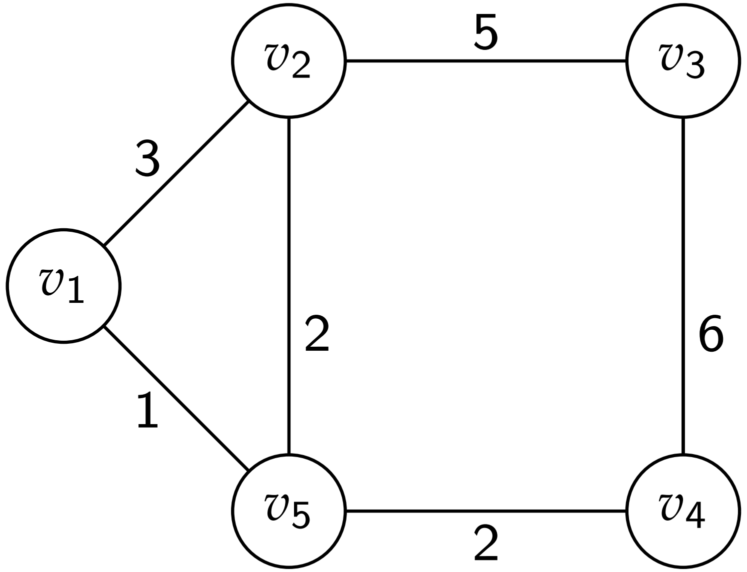


$w(S) = 4$

The **MaxCut** Problem

Input. Graph G, edge weights $w\colon E(G) \to \mathbb{N}$.

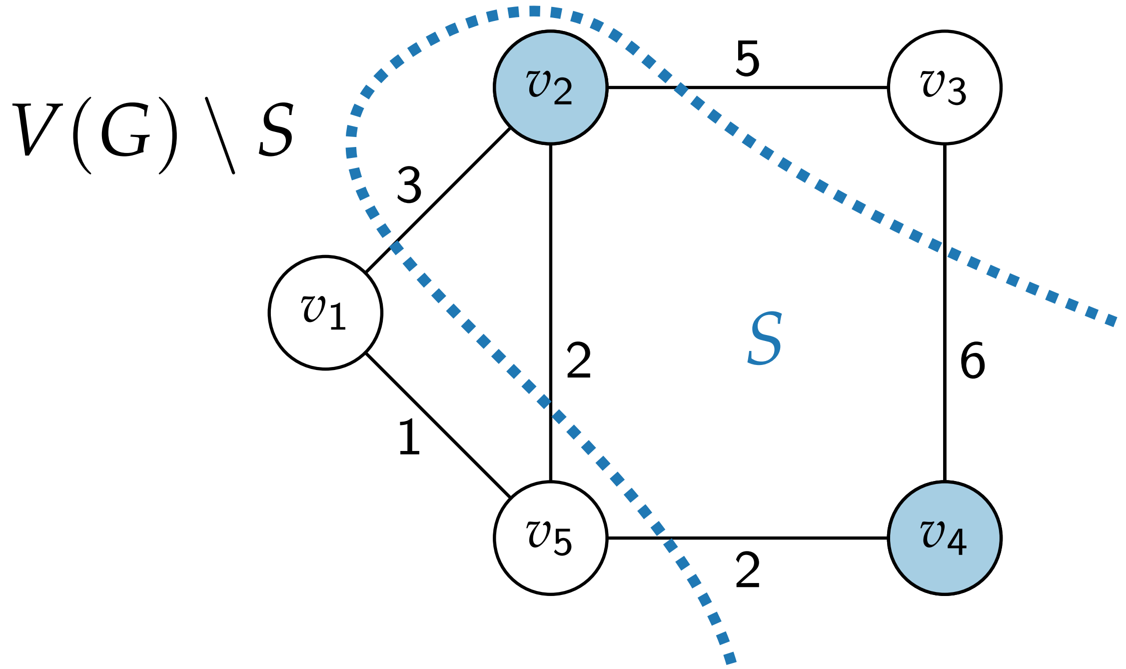
Output. Cut $(S, V(G) \setminus S)$ of G of **maximum** weight.



The **MaxCut** Problem

Input. Graph G, edge weights $w\colon E(G) \to \mathbb{N}$.

Output. Cut $(S, V(G) \setminus S)$ of G of **maximum** weight.



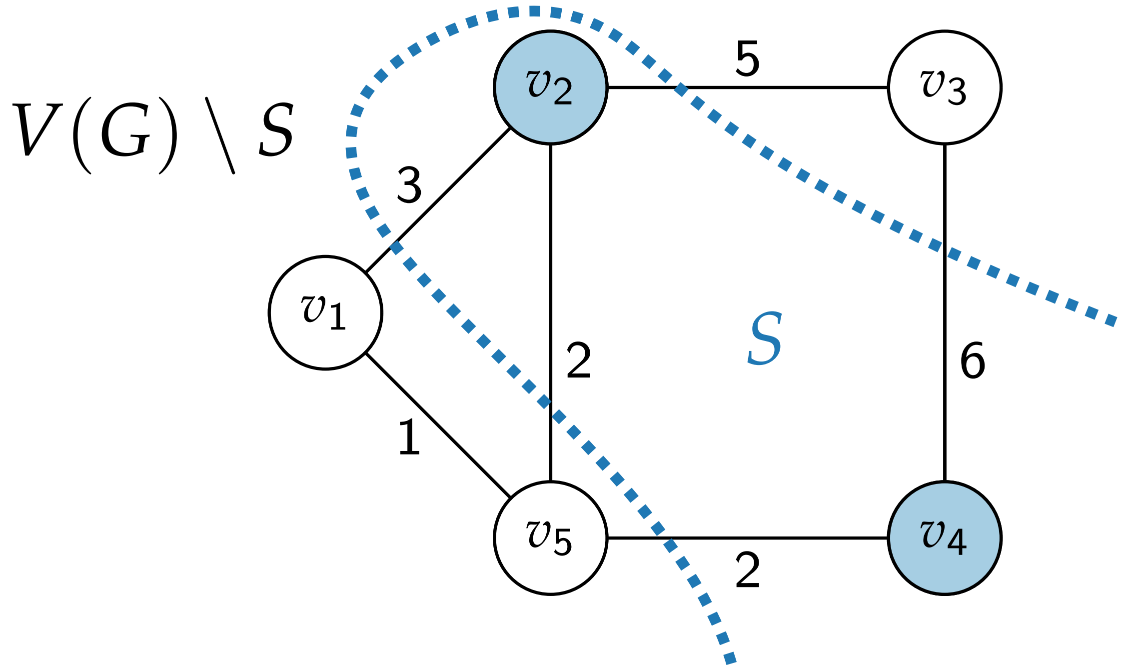
$w(S) = 18$

The **MaxCut** Problem

Input. Graph G, edge weights $w\colon E(G) \to \mathbb{N}$.

Output. Cut $(S, V(G) \setminus S)$ of G of **maximum** weight.

- Has applications in binary classification (vertices are features and weighted edges are distances), statistical physics (equivalent to minimizing the "Hamiltonian" of a spin glass model), and integrated circuit design for computer chips (modeling a specific assignment problem as a graph problem).



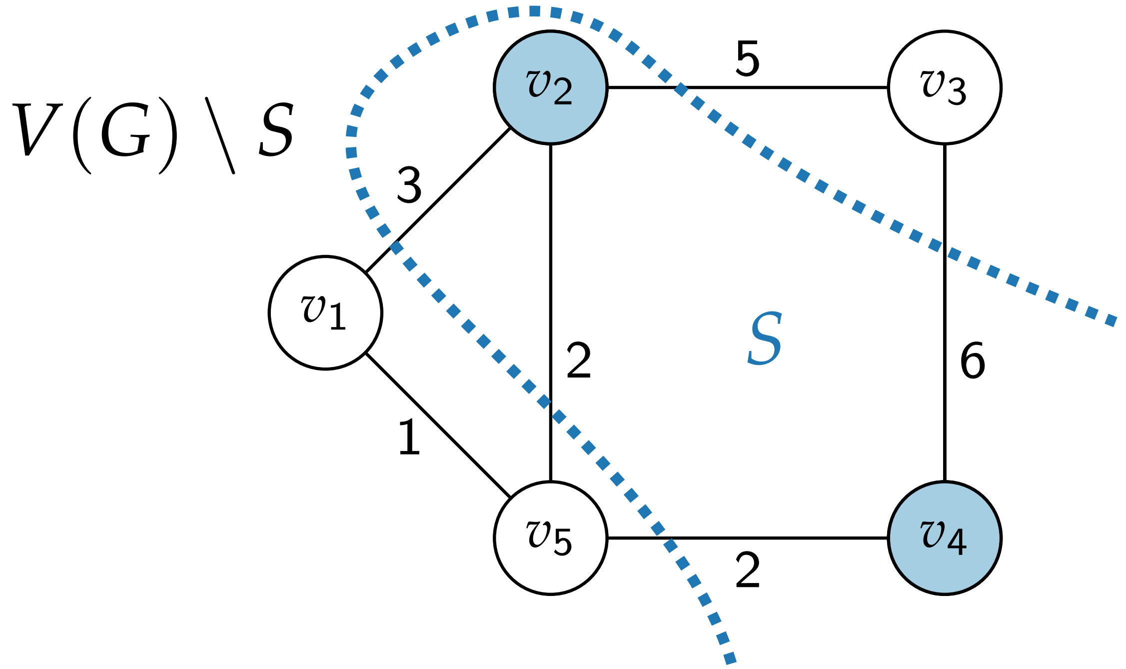
$w(S) = 18$

The **MaxCut** Problem

Input. Graph G, edge weights $w\colon E(G) \to \mathbb{N}$.

Output. Cut $(S, V(G) \setminus S)$ of G of **maximum** weight.

- Has applications in binary classification (vertices are features and weighted edges are distances), statistical physics (equivalent to minimizing the "Hamiltonian" of a spin glass model), and integrated circuit design for computer chips (modeling a specific assignment problem as a graph problem).
- NP-complete to find a cut of maximum weight.



$w(S) = 18$

Randomized Approximation for (Unweighted) MaxCut

$\textsc{CoinFlipMaxCut}(G, w \equiv 1)$

$S \leftarrow \emptyset$
foreach $v \in V(G)$ **do**
  if coin flip shows $\textsc{Heads}$ **then**
    $S \leftarrow S \cup \{v\}$
return $w(S)$ and S

Randomized Approximation for (Unweighted) MaxCut

Theorem 1.
$\textsc{CoinFlipMaxCut}$ is a randomized 0.5-approximation algorithm for MaxCut.

$\textsc{CoinFlipMaxCut}(G, w \equiv 1)$

$S \leftarrow \emptyset$
foreach $v \in V(G)$ **do**
  if coin flip shows $\textsc{Heads}$ **then**
    $S \leftarrow S \cup \{v\}$
return $w(S)$ and S

Randomized Approximation for (Unweighted) MaxCut

Theorem 1.
CoinFlipMaxCut is a randomized 0.5-approximation algorithm for MaxCut.

Proof.

- Runs in $O(|V(G)| + |E(G)|$ time.

```
CoinFlipMaxCut(G, w ≡ 1)
  S ← ∅
  foreach v ∈ V(G) do
      if coin flip shows Heads then
          S ← S ∪ {v}
  return w(S) and S
```

Randomized Approximation for (Unweighted) MaxCut

Theorem 1.
$\textsc{CoinFlipMaxCut}$ is a randomized 0.5-approximation algorithm for MaxCut.

Proof.

- Runs in $O(|V(G)| + |E(G)|$ time.
- Compute expected weight of cut:

$$\mathrm{E}[w(\textsc{CoinFlipMaxCut}(G))] =$$

$$\geq \frac{1}{2}\mathrm{OPT}(G)$$

$\textsc{CoinFlipMaxCut}(G, w \equiv 1)$

```
S ← ∅
foreach v ∈ V(G) do
    if coin flip shows HEADS then
        S ← S ∪ {v}
return w(S) and S
```

Randomized Approximation for (Unweighted) MaxCut

Theorem 1.
$\textsc{CoinFlipMaxCut}$ is a randomized 0.5-approximation algorithm for MaxCut.

$\textsc{CoinFlipMaxCut}(G, w \equiv 1)$

$S \leftarrow \emptyset$
foreach $v \in V(G)$ **do**
  if coin flip shows $\textsc{Heads}$ **then**
    $S \leftarrow S \cup \{v\}$
return $w(S)$ and S

Proof.

- Runs in $O(|V(G)| + |E(G)|$ time.
- Compute expected weight of cut:

$$\mathsf{E}[w(\textsc{CoinFlipMaxCut}(G))] = \mathsf{E}\big[|E(S, V(G) \setminus S)|\big]$$
$$=$$
$$\geq \frac{1}{2}\mathsf{OPT}(G)$$

Randomized Approximation for (Unweighted) MaxCut

Theorem 1.
$\textsc{CoinFlipMaxCut}$ is a randomized 0.5-approximation algorithm for MaxCut.

$\textsc{CoinFlipMaxCut}(G, w \equiv 1)$

$S \leftarrow \emptyset$
foreach $v \in V(G)$ **do**
  if coin flip shows $\textsc{Heads}$ **then**
    $S \leftarrow S \cup \{v\}$
return $w(S)$ and S

Proof.

- Runs in $O(|V(G)| + |E(G)|$ time.
- Compute expected weight of cut:

$$\begin{aligned}\mathsf{E}[w(\textsc{CoinFlipMaxCut}(G))] &= \mathsf{E}\big[|E(S, V(G) \setminus S)|\big] \\ &= \sum_{e \in E(G)} \mathsf{P}[e \in E(S, V(G) \setminus S)] \\ &= \qquad\qquad\qquad \geq \frac{1}{2}\mathsf{OPT}(G)\end{aligned}$$

Randomized Approximation for (Unweighted) MaxCut

Theorem 1.
$\textsc{CoinFlipMaxCut}$ is a randomized 0.5-approximation algorithm for MaxCut.

Proof.

- Runs in $O(|V(G)| + |E(G)|$ time.
- Compute expected weight of cut:

$$\begin{aligned}
\mathsf{E}[w(\textsc{CoinFlipMaxCut}(G))] &= \mathsf{E}\big[|E(S, V(G) \setminus S)|\big] \\
&= \sum_{e \in E(G)} \mathsf{P}[e \in E(S, V(G) \setminus S)] \\
&= \sum_{e \in E(G)} \frac{1}{2} = \frac{1}{2}|E(G)| \geq \frac{1}{2}\mathsf{OPT}(G)
\end{aligned}$$

```
CoinFlipMaxCut(G, w ≡ 1)
  S ← ∅
  foreach v ∈ V(G) do
    if coin flip shows Heads then
      S ← S ∪ {v}
  return w(S) and S
```

Randomized Approximation for (Unweighted) MaxCut

Theorem 1.
$\textsc{CoinFlipMaxCut}$ is a randomized 0.5-approximation algorithm for MaxCut.

$\textsc{CoinFlipMaxCut}(G, w \equiv 1)$

```
S ← ∅
foreach v ∈ V(G) do
    if coin flip shows HEADS then
        S ← S ∪ {v}
return w(S) and S
```

Proof.

- Runs in $O(|V(G)| + |E(G)|$ time.
- Compute expected weight of cut:

$$\begin{aligned}\mathsf{E}[w(\textsc{CoinFlipMaxCut}(G))] &= \mathsf{E}\big[|E(S, V(G) \setminus S)|\big] \\ &= \sum_{e \in E(G)} \mathsf{P}[e \in E(S, V(G) \setminus S)] \\ &= \sum_{e \in E(G)} \frac{1}{2} = \frac{1}{2}|E(G)| \geq \frac{1}{2}\mathsf{OPT}(G)\end{aligned}$$

- Can be "de-randomized". Exercise.

LP-Relaxation

Integer Linear Program

$$
\begin{array}{llll}
\textbf{maximize} & c^\intercal x & & \\
\textbf{subject to} & Ax & \leq & b \\
 & x & \geq & 0 \\
 & x & \in & \mathbb{Z}^n
\end{array}
$$

LP-Relaxation

Integer Linear Program

$$\begin{array}{llcl} \textbf{maximize} & c^\mathsf{T} x & & \\ \textbf{subject to} & Ax & \leq & b \\ & x & \geq & 0 \\ & x & \in & \mathbb{Z}^n \end{array}$$

LP-Relaxation

Linear Program

$$\begin{array}{llcl} \textbf{maximize} & c^\mathsf{T} x & & \\ \textbf{subject to} & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

LP-Relaxation

Integer Linear Program

$$\begin{array}{llll} \textbf{maximize} & c^\top x & & \\ \textbf{subject to} & Ax & \leq & b \\ & x & \geq & 0 \\ & x & \in & \mathbb{Z}^n \end{array}$$

LP-Relaxation

Linear Program

$$\begin{array}{llll} \textbf{maximize} & c^\top x & & \\ \textbf{subject to} & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

Solve in polynomial time

Solution for LP

$x^\star$

LP-Relaxation

Integer Linear Program

$$\begin{array}{llcl} \textbf{maximize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \leq & b \\ & x & \geq & 0 \\ & x & \in & \mathbb{Z}^n \end{array}$$

LP-Relaxation

Linear Program

$$\begin{array}{llcl} \textbf{maximize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

Solve in polynomial time

Solution for LP

$x^\star$

e.g. rounding

Assignment for ILP

$x^\star$

LP-Relaxation

Integer Linear Program

$$\begin{array}{lrcr} \textbf{maximize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \leq & b \\ & x & \geq & 0 \\ & x & \in & \mathbb{Z}^n \end{array}$$

LP-Relaxation

Linear Program

$$\begin{array}{lrcr} \textbf{maximize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

Solve in
polynomial time

Solution for LP

$x^\star$

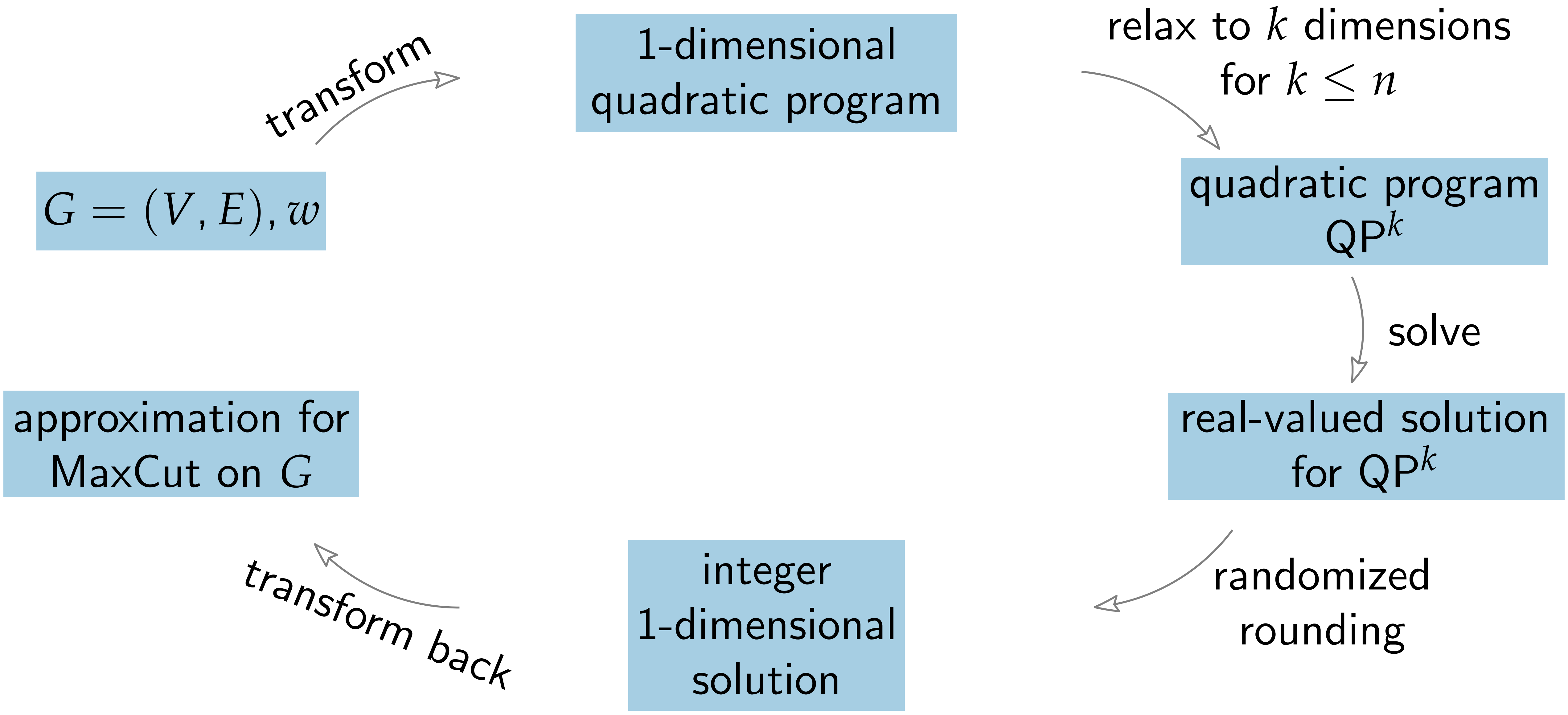
e.g. rounding

Assignment for ILP

$x^\star$

Solution,
approximation,
or bound

Goemans–Williamson Algorithm for MaxCut



Goemans–Williamson Algorithm for MaxCut

$G = (V, E), w$

transform

1-dimensional quadratic program

relax to k dimensions for $k \leq n$

quadratic program QP^k

solve

real-valued solution for QP^k

randomized rounding

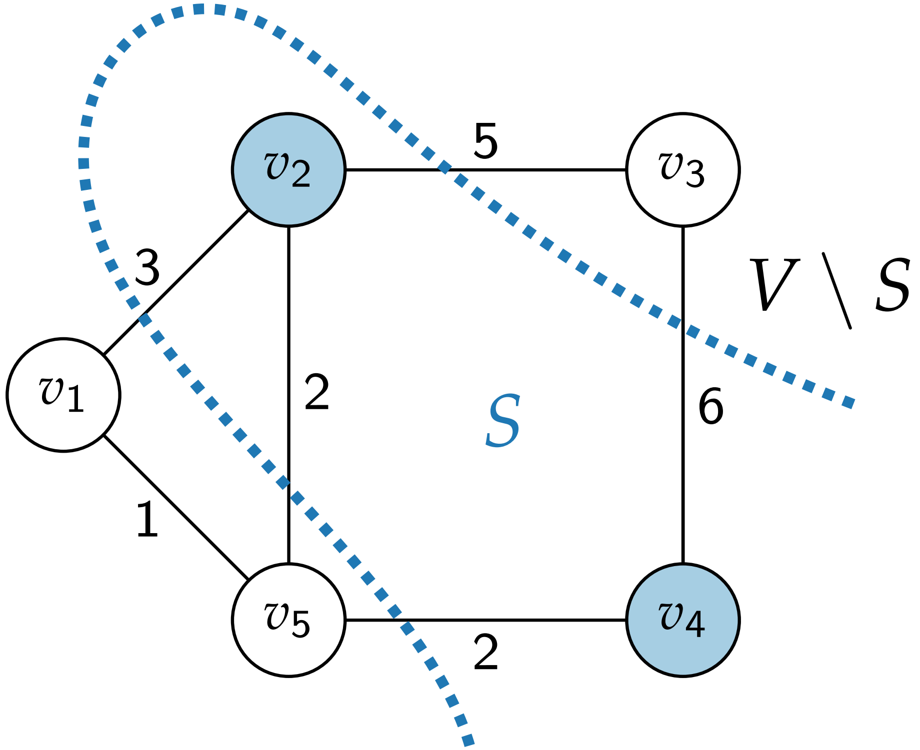
integer 1-dimensional solution

transform back

approximation for MaxCut on G

QP(G, w)

Idea.



QP(G, w)

maximize

subject to

$\mathrm{QP}(G, w)$

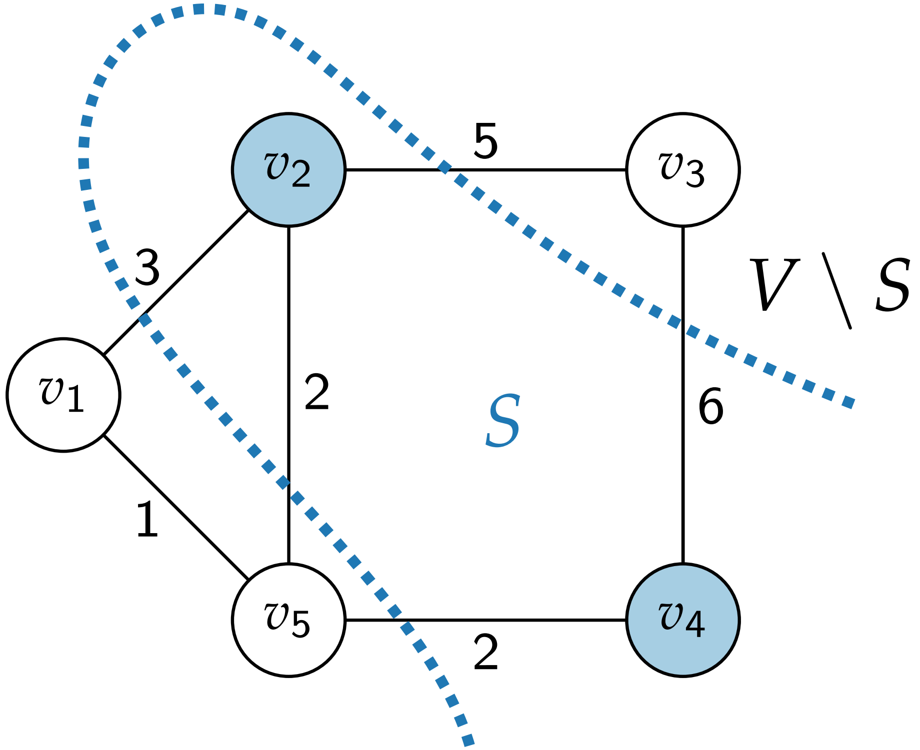
Idea.

- Indicator variable for each vertex v_i:
  $x_i \in \{1, -1\}$

QP(G, w)

maximize

subject to



QP(G, w)

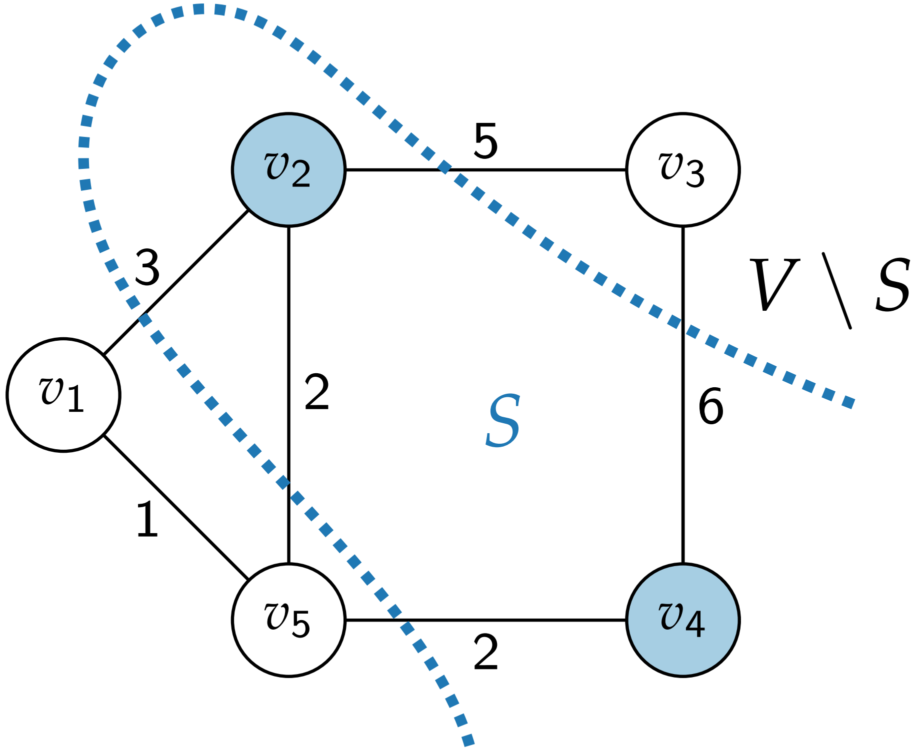
Idea.

- Indicator variable for each vertex v_i: $x_i \in \{1, -1\}$

QP(G, w)

maximize

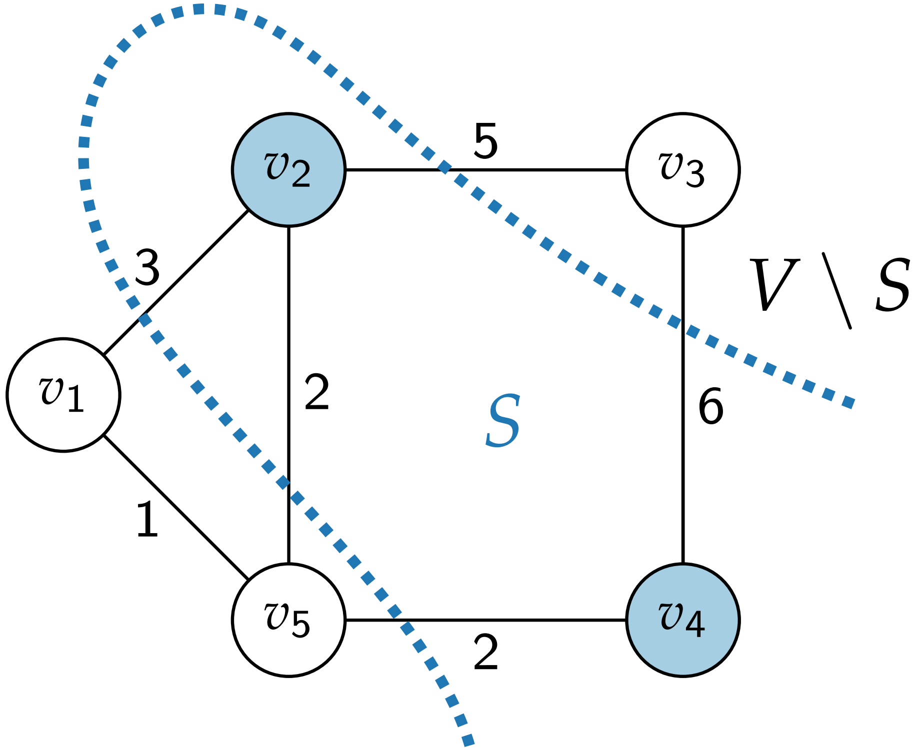
subject to $x_i^2 = 1$



QP(G, w)

Idea.

- Indicator variable for each vertex v_i: $x_i \in \{1, -1\}$
- $x_i \cdot x_j = \begin{cases} 1 & \text{if } i, j \text{ on the same side} \\ -1 & \text{otherwise} \end{cases}$



QP(G, w)

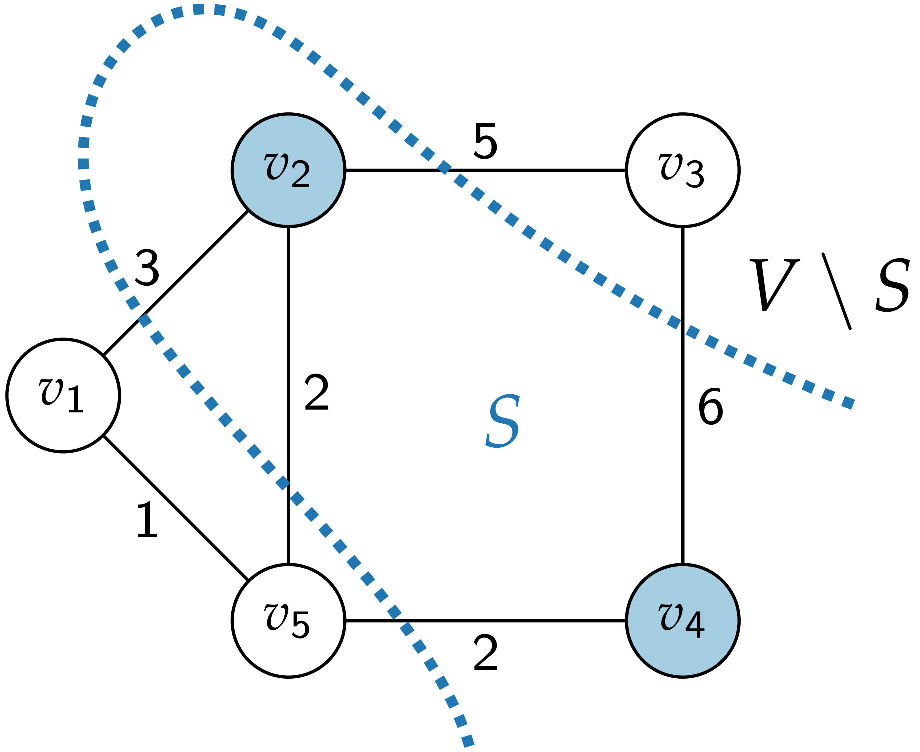
maximize

subject to $x_i^2 = 1$

QP(G, w)

Idea.

- Indicator variable for each vertex v_i: $x_i \in \{1, -1\}$
- $x_i \cdot x_j = \begin{cases} 1 & \text{if } i, j \text{ on the same side} \\ -1 & \text{otherwise} \end{cases}$



QP(G, w)

maximize $(1 - x_i x_j)$

subject to $x_i^2 = 1$

QP(G, w)

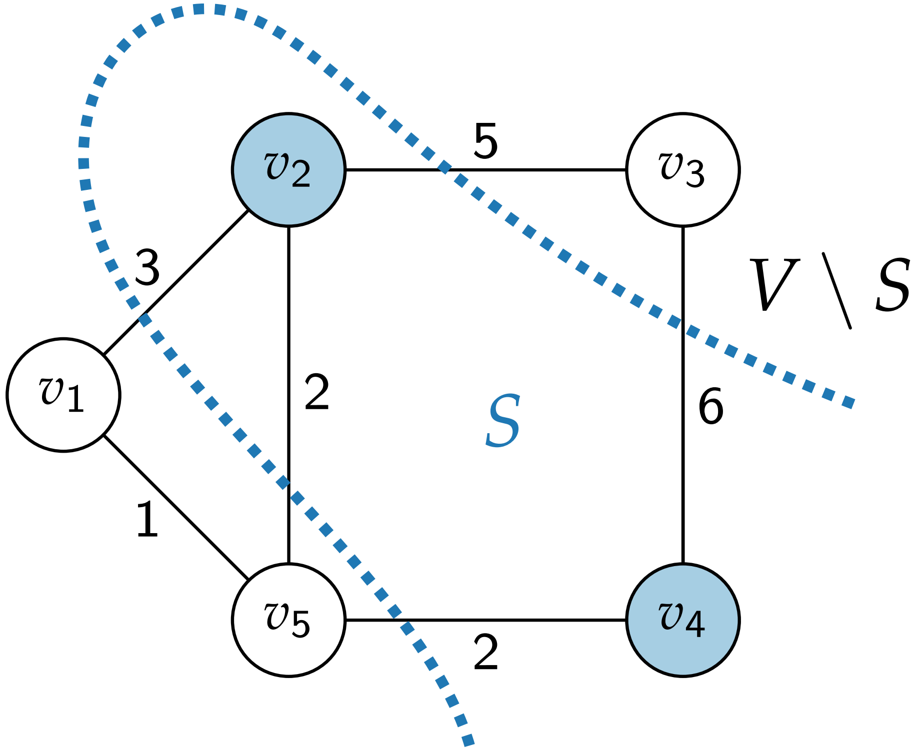
Idea.

- Indicator variable for each vertex v_i: $x_i \in \{1, -1\}$
- $x_i \cdot x_j = \begin{cases} 1 & \text{if } i, j \text{ on the same side} \\ -1 & \text{otherwise} \end{cases}$

QP(G, w)

maximize $\quad w_{ij}(1 - x_i x_j)$

subject to $\quad x_i^2 = 1$



- Weight matrix w_{ij}

	1	2	3	4	5
1		3			1
2	3		5		2
3		5		6	
4			6		2
5	1	2		2	

$\mathrm{QP}(G, w)$

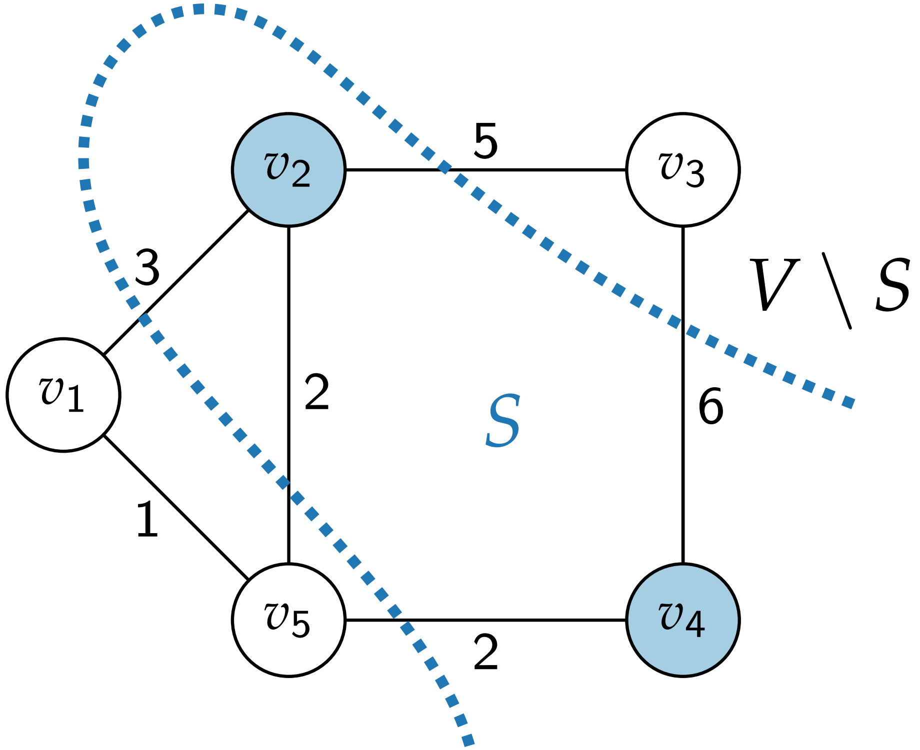
Idea.

- Indicator variable for each vertex v_i: $x_i \in \{1, -1\}$
- $x_i \cdot x_j = \begin{cases} 1 & \text{if } i, j \text{ on the same side} \\ -1 & \text{otherwise} \end{cases}$

QP(G, w)

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j)$

subject to $x_i^2 = 1$



- Weight matrix w_{ij}

	1	2	3	4	5
1		3			1
2	3		5		2
3		5		6	
4			6		2
5	1	2		2	

$\mathrm{QP}(G, w)$

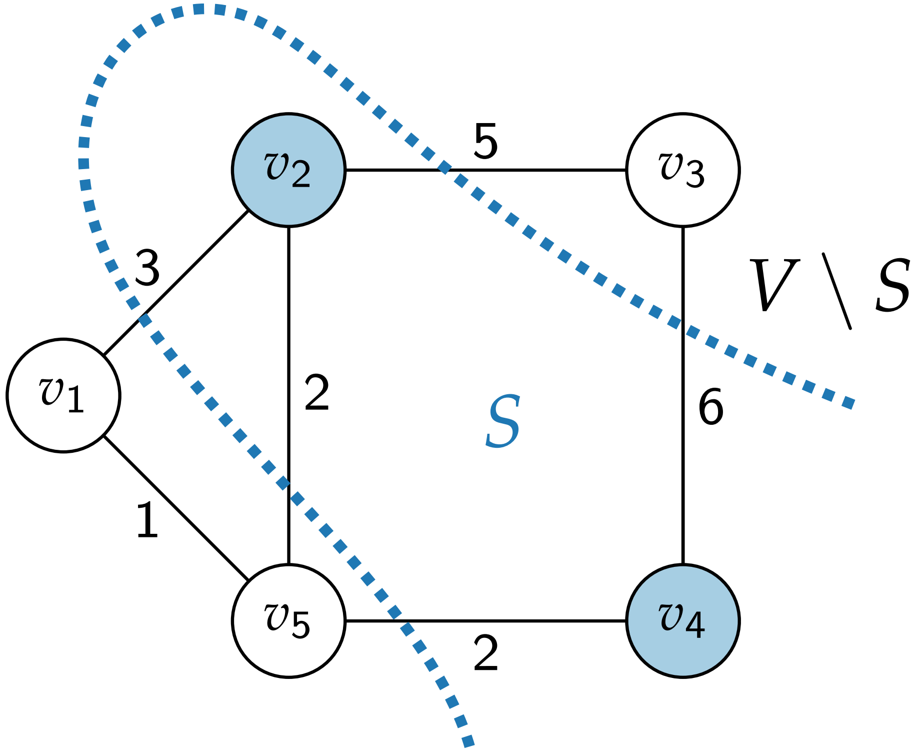
Idea.

- Indicator variable for each vertex v_i:
  $x_i \in \{1, -1\}$
- $x_i \cdot x_j = \begin{cases} 1 & \text{if } i, j \text{ on the same side} \\ -1 & \text{otherwise} \end{cases}$

QP(G, w)

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j)$

subject to $x_i^2 = 1$



- Weight matrix w_{ij}

	1	2	3	4	5
1		3			1
2	3		5		2
3		5		6	
4			6		2
5	1	2		2	

- Solution

  $x_2 = x_4 = 1$

  $x_1 = x_3 = x_5 = -1$

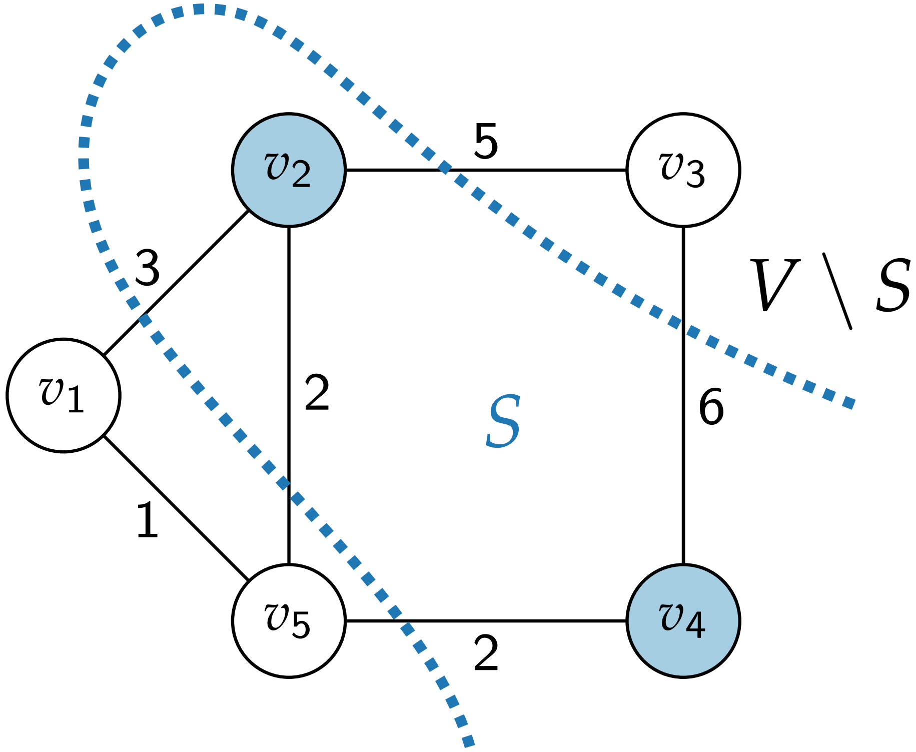
QP(G, w)

Idea.

- Indicator variable for each vertex v_i: $x_i \in \{1, -1\}$
- $x_i \cdot x_j = \begin{cases} 1 & \text{if } i, j \text{ on the same side} \\ -1 & \text{otherwise} \end{cases}$

QP(G, w)

maximize	$\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j)$
subject to	$x_i^2 = 1$



- Weight matrix w_{ij}

	1	2	3	4	5
1		3			1
2	3		5		2
3		5		6	
4			6		2
5	1	2		2	

- Solution

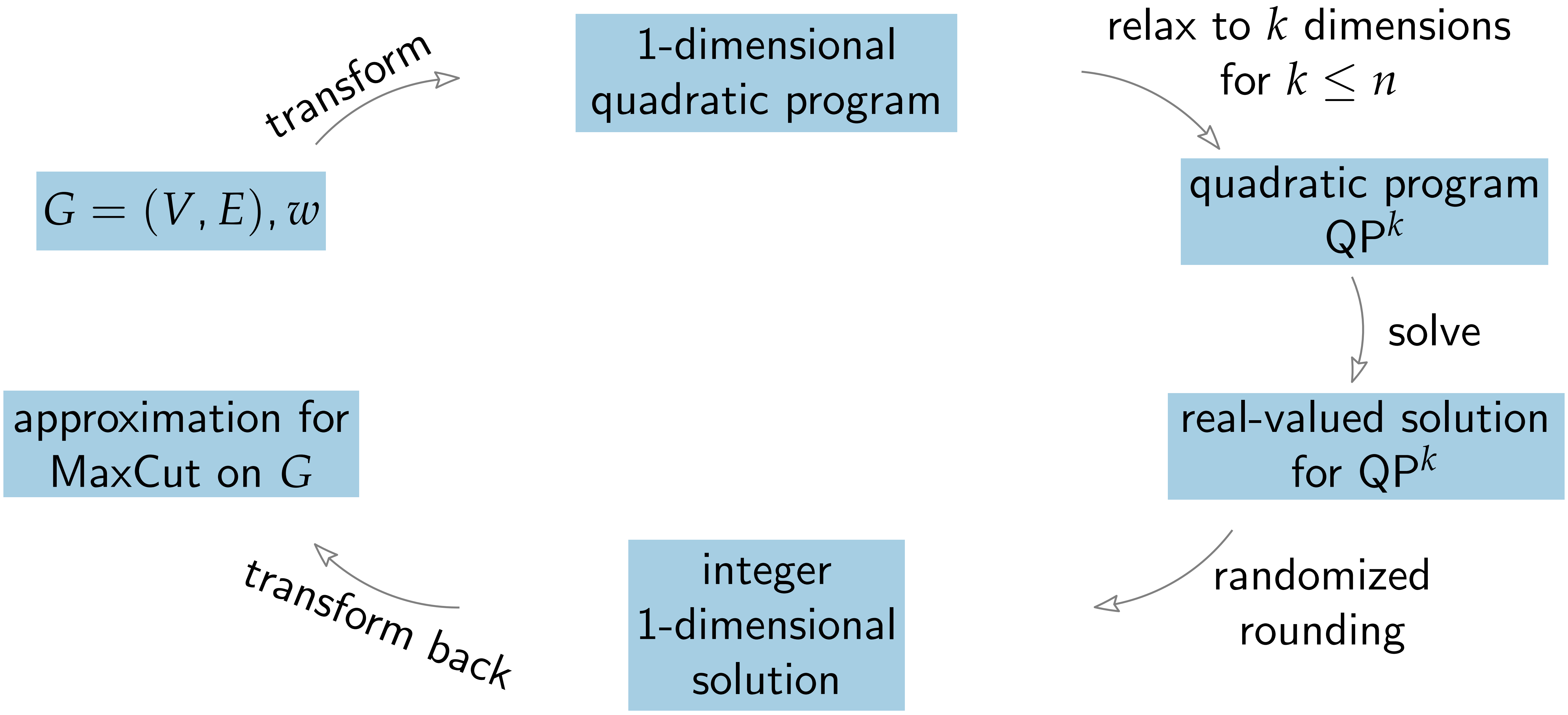
  $x_2 = x_4 = 1$

  $x_1 = x_3 = x_5 = -1$

Note.

- Solving QP(G, w) is NP-hard.
- Otherwise MaxCut would not be NP-hard.

Goemans–Williamson Algorithm for MaxCut



Goemans–Williamson Algorithm for MaxCut

$G = (V, E), w$

transform

1-dimensional quadratic program

relax to k dimensions for $k \leq n$

quadratic program QP^k

solve

real-valued solution for QP^k

randomized rounding

integer 1-dimensional solution

transform back

approximation for MaxCut on G

- Here explained for $k = 2$,
- but unknown if QP^2 can be solved optimally in polynomial time.
- QP^n can be solved in poly. time.

Relaxation of QP(G, w)

QP$^2(G, w)$

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to

$$x^i \cdot x^i = 1$$
$$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$$

Relaxation of QP(G, w)

QP$^2(G, w)$

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to
$$x^i \cdot x^i = 1$$
$$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$$

- The variables are 2-dimensional vectors.

Relaxation of QP(G, w)

QP$^2(G, w)$

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to $x^i \cdot x^i = 1$

$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$

- " $\cdot$ " is scalar (or dot) product.
- The variables are 2-dimensional vectors.

Relaxation of QP(G, w)

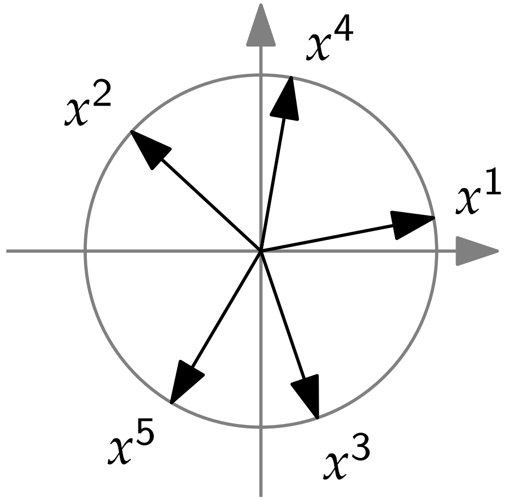
QP^2(G, w)

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to $x^i \cdot x^i = 1$

$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$

- " $\cdot$ " is scalar (or dot) product.
- x^i lies on the unit circle.
- The variables are 2-dimensional vectors.



Relaxation of QP(G, w)

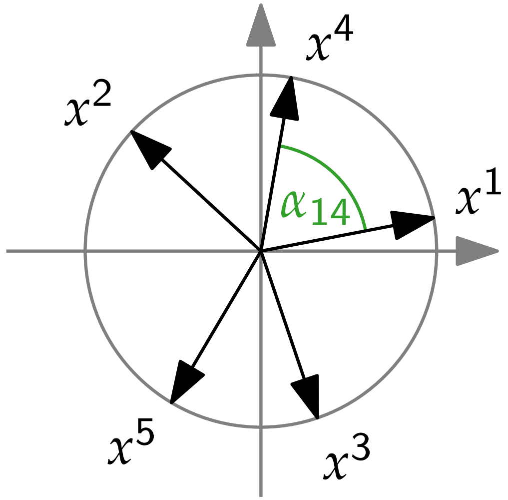
QP$^2(G, w)$

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to $x^i \cdot x^i = 1$

$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$

- "$\cdot$" is scalar (or dot) product.
- x^i lies on the unit circle.
- $x^i \cdot x^j = \|x^i\| \cdot \|x^j\| \cdot \cos(\alpha_{ij})$ $= \cos(\alpha_{ij})$ with $0 \leq \alpha_{ij} \leq \pi$.
- The variables are 2-dimensional vectors.



Relaxation of QP(G, w)

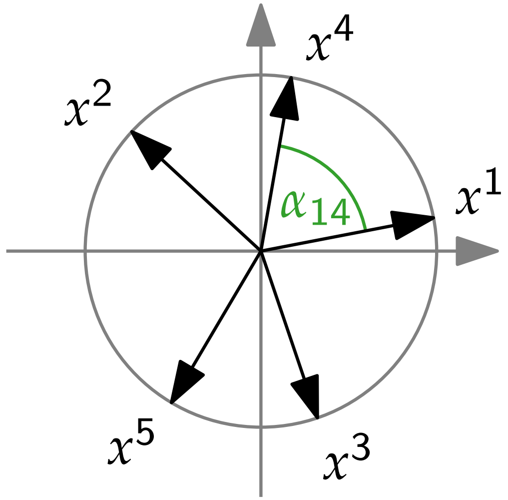
QP$^2(G, w)$

maximize $\quad \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to $\quad x^i \cdot x^i = 1$

$$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$$

- "$\cdot$" is scalar (or dot) product.
- x^i lies on the unit circle.
- $x^i \cdot x^j = \|x^i\| \cdot \|x^j\| \cdot \cos(\alpha_{ij})$ $= \cos(\alpha_{ij})$ with $0 \le \alpha_{ij} \le \pi$.



- The variables are 2-dimensional vectors.
- We maximize angles α_{ij} since larger α_{ij} increases the contribution of w_{ij}.

Relaxation of QP(G, w)

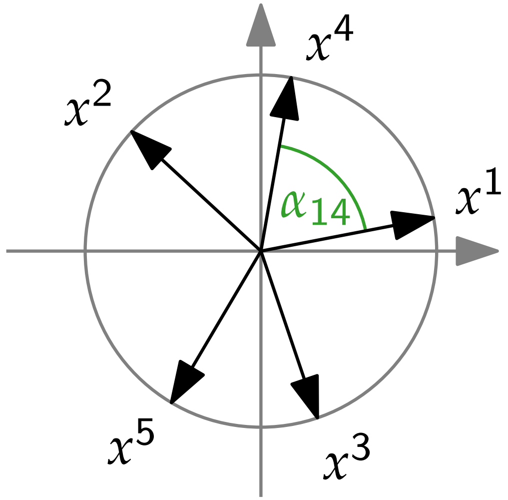
QP$^2(G, w)$

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

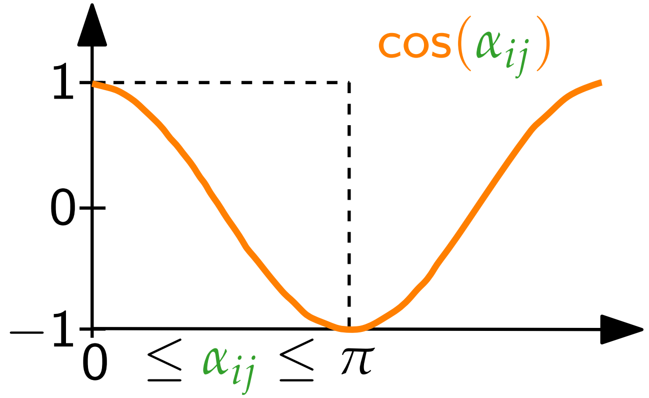
subject to $x^i \cdot x^i = 1$

$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$

- "$\cdot$" is scalar (or dot) product.
- x^i lies on the unit circle.
- $x^i \cdot x^j = \|x^i\| \cdot \|x^j\| \cdot \cos(\alpha_{ij})$ $= \cos(\alpha_{ij})$ with $0 \leq \alpha_{ij} \leq \pi$.



- The variables are 2-dimensional vectors.
- We maximize angles α_{ij} since larger α_{ij} increases the contribution of w_{ij}.



Relaxation of QP(G, w)

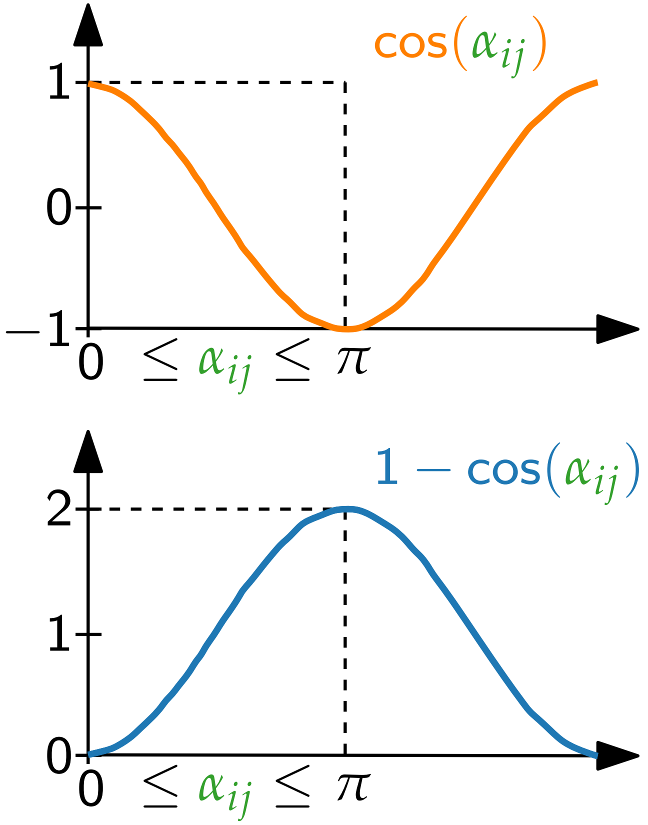
QP^2(G, w)

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

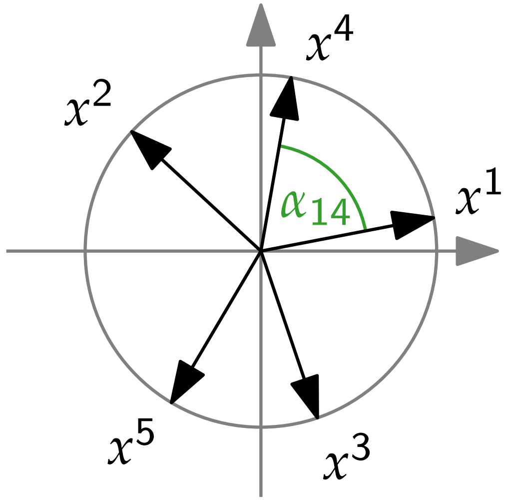
subject to $x^i \cdot x^i = 1$

$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$

- " $\cdot$ " is scalar (or dot) product.
- x^i lies on the unit circle.
- $x^i \cdot x^j = \|x^i\| \cdot \|x^j\| \cdot \cos(\alpha_{ij})$ $= \cos(\alpha_{ij})$ with $0 \le \alpha_{ij} \le \pi$.



- The variables are 2-dimensional vectors.
- We maximize angles α_{ij} since larger α_{ij} increases the contribution of w_{ij}.



Relaxation of QP(G, w)

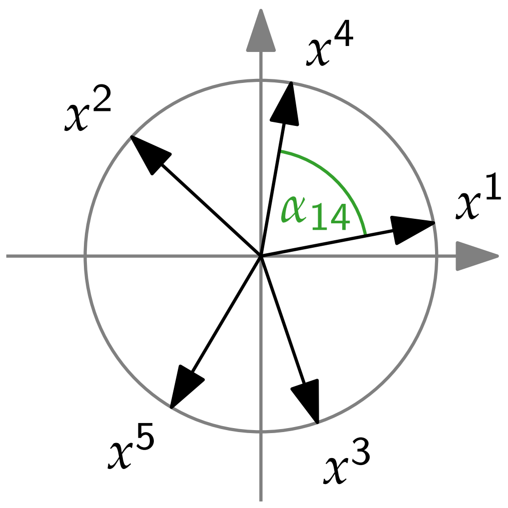
QP$^2(G, w)$

maximize $\quad \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to $\quad x^i \cdot x^i = 1$

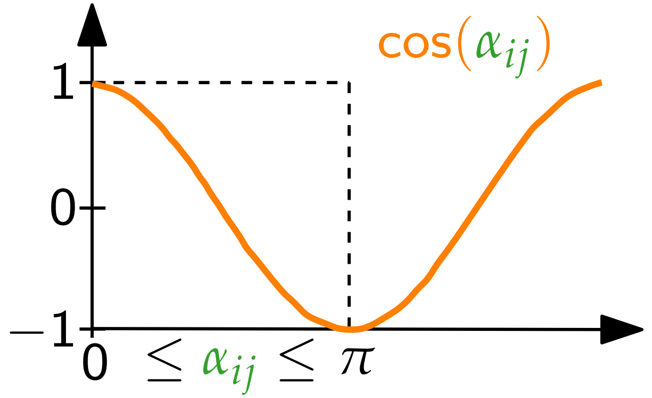
$$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$$

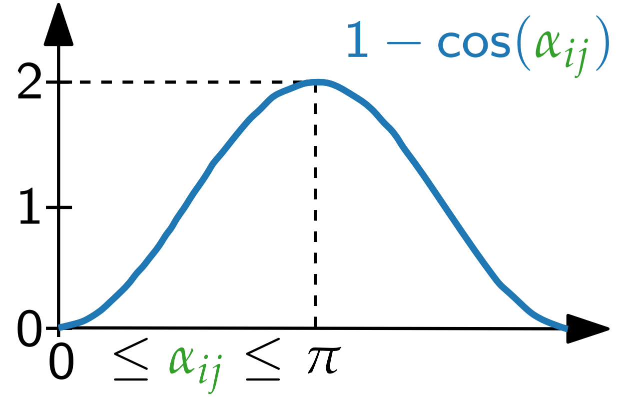
- " $\cdot$ " is scalar (or dot) product.
- x^i lies on the unit circle.
- $x^i \cdot x^j = \|x^i\| \cdot \|x^j\| \cdot \cos(\alpha_{ij})$ $= \cos(\alpha_{ij})$ with $0 \le \alpha_{ij} \le \pi$.



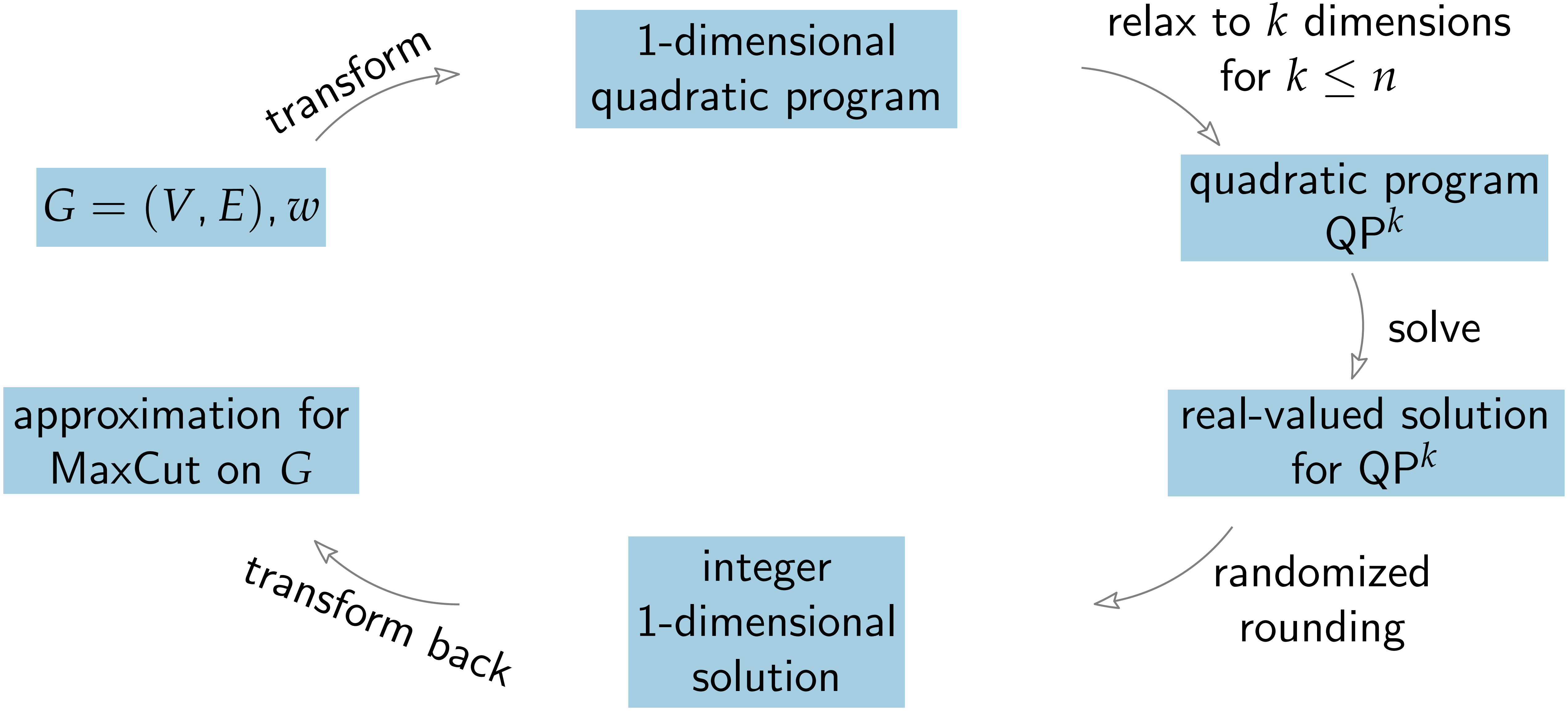
- The variables are 2-dimensional vectors.
- We maximize angles α_{ij} since larger α_{ij} increases the contribution of w_{ij}.
- Hence, our objective is:

$$\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - \cos(\alpha_{ij}))$$

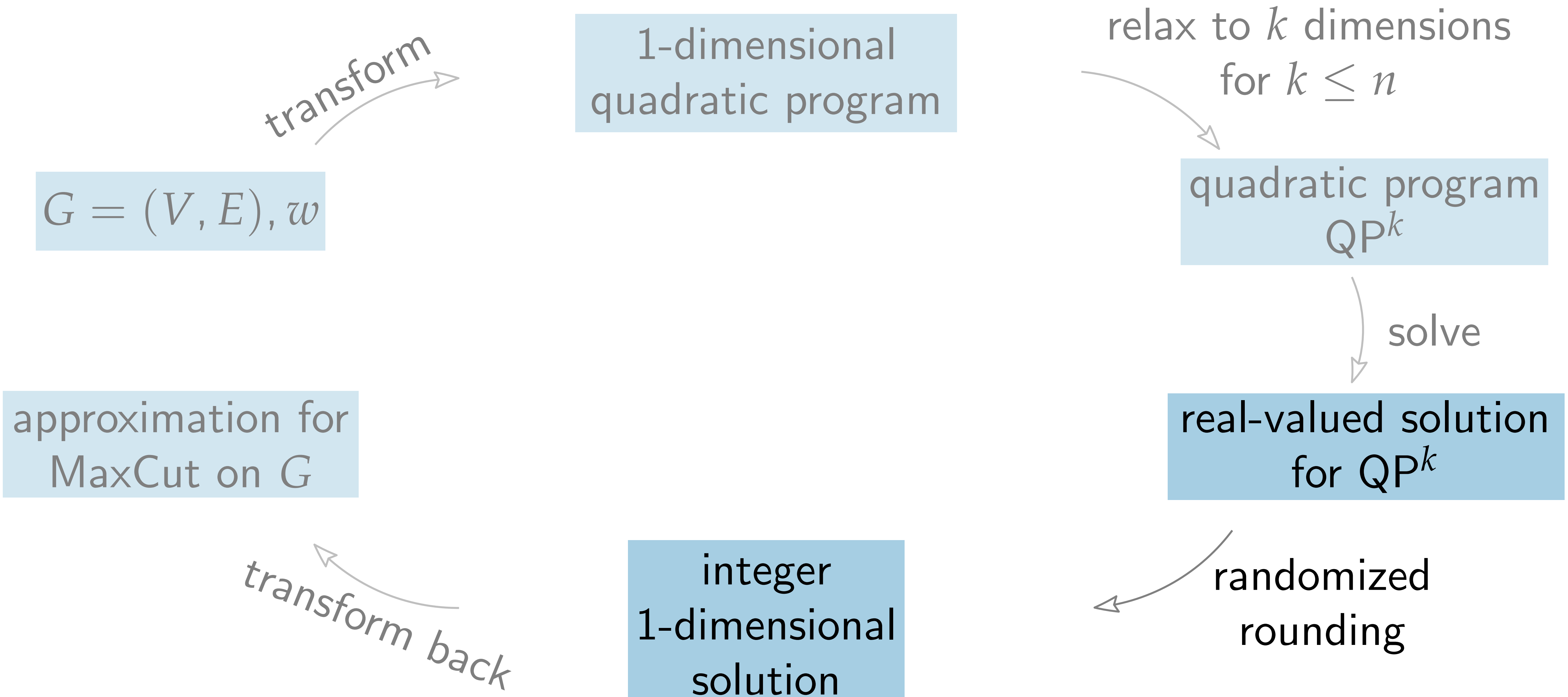




Goemans–Williamson Algorithm for MaxCut



Goemans–Williamson Algorithm for MaxCut



- Here again just for $k = 2$.

Algorithm RandomizedMaxCut

RandomizedMaxCut(G, w)

Compute optimal solution $(\tilde{x}^1, \ldots, \tilde{x}^n)$ for QP$^2(G, w)$

Pick random vector $r \in \mathbb{R}^2$

$S \leftarrow \{v_i \in V(G) : \tilde{x}^i \cdot r \geq 0\}$

return $c(S, V(G) \setminus S)$

Algorithm RandomizedMaxCut

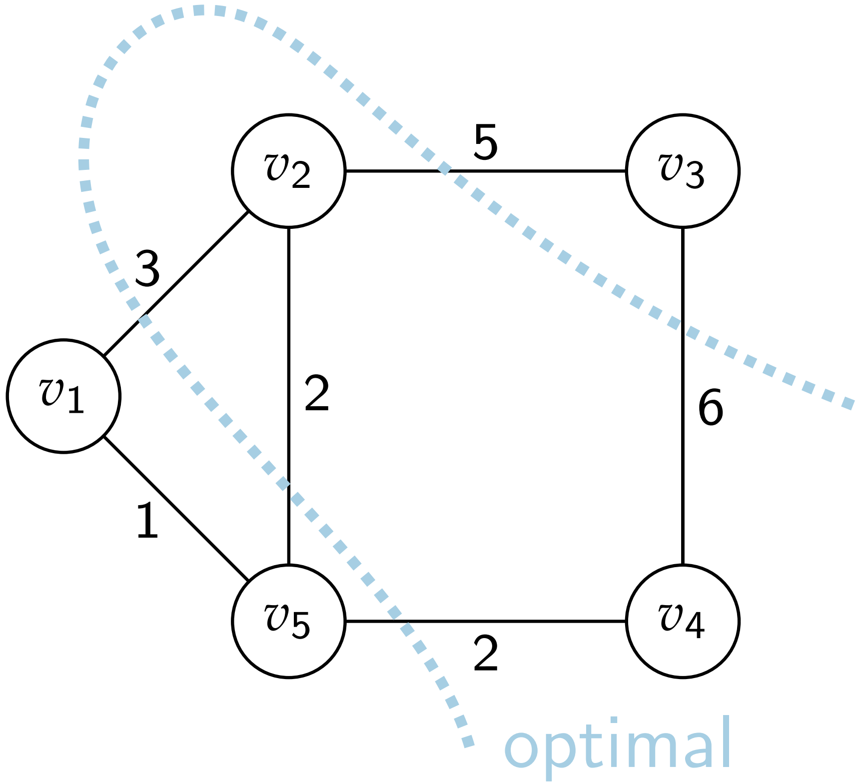
RandomizedMaxCut(G, w)

Compute optimal solution $(\tilde{x}^1, \dots, \tilde{x}^n)$ for QP$^2(G, w)$

Pick random vector $r \in \mathbb{R}^2$

$S \leftarrow \{v_i \in V(G) : \tilde{x}^i \cdot r \geq 0\}$

return $c(S, V(G) \setminus S)$



Algorithm RandomizedMaxCut

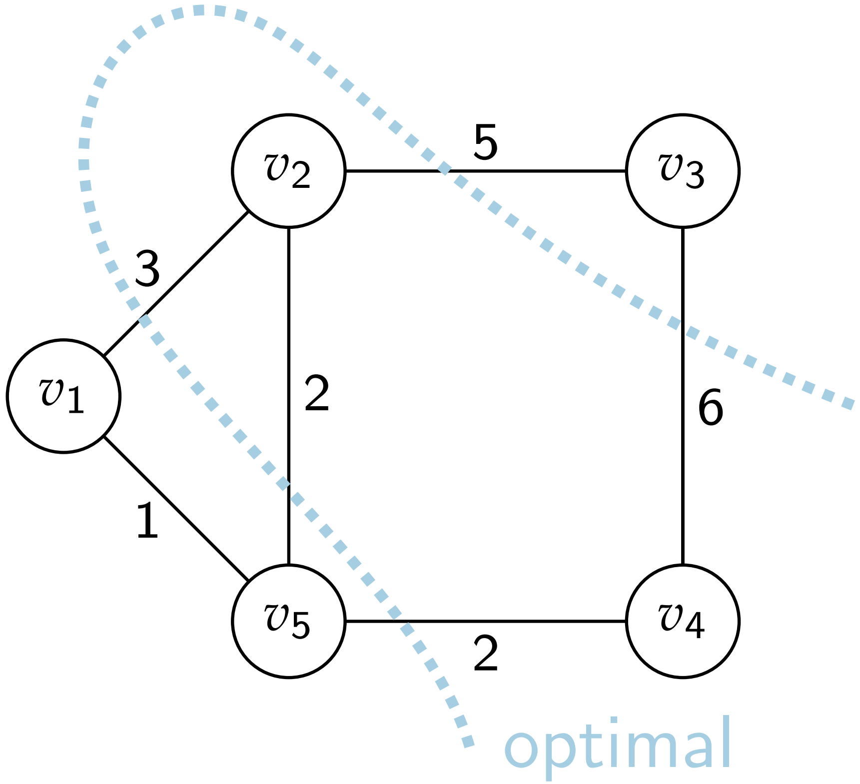
RandomizedMaxCut(G, w)

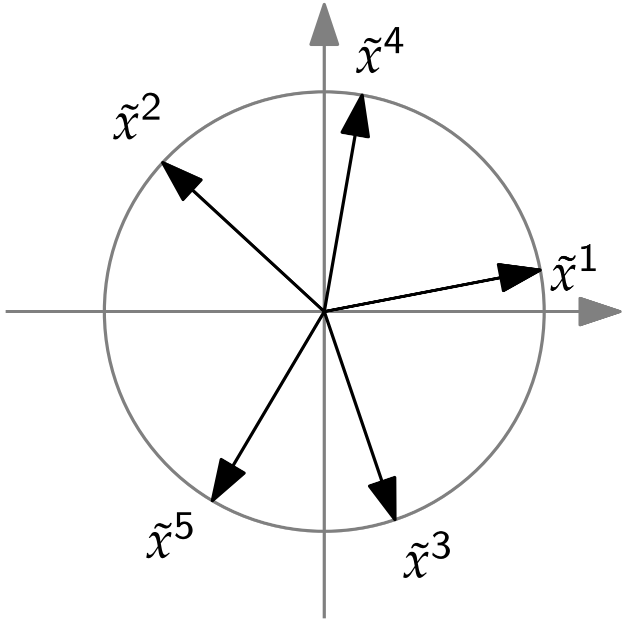
Compute optimal solution $(\tilde{x}^1, \dots, \tilde{x}^n)$ for QP$^2(G, w)$

Pick random vector $r \in \mathbb{R}^2$

$S \leftarrow \{v_i \in V(G) : \tilde{x}^i \cdot r \geq 0\}$

return $c(S, V(G) \setminus S)$





(Sketch)

Algorithm RandomizedMaxCut

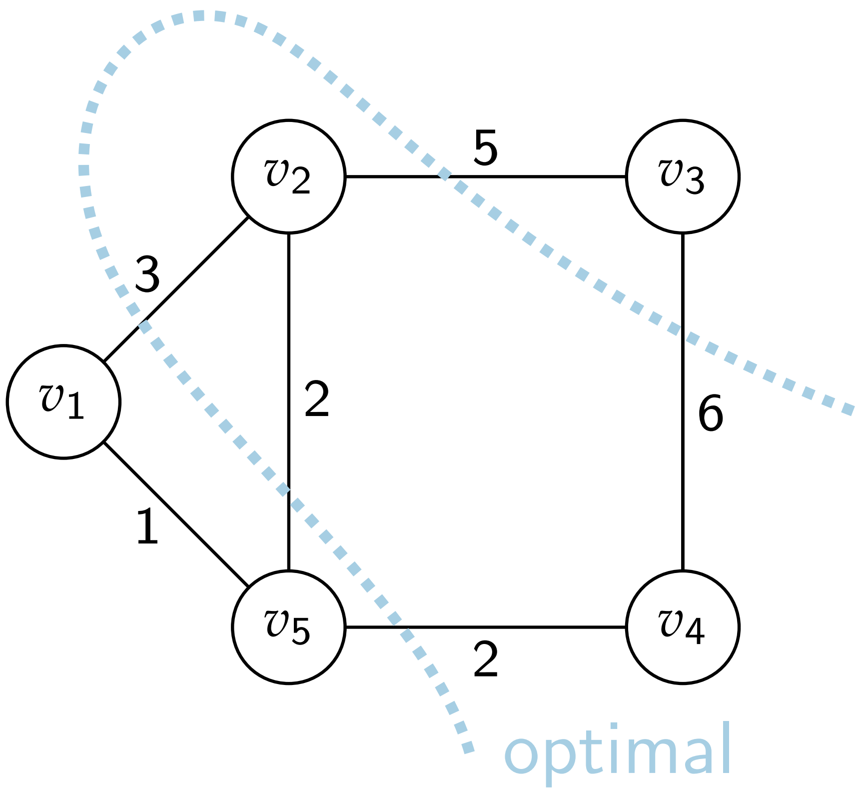
RandomizedMaxCut(G, w)

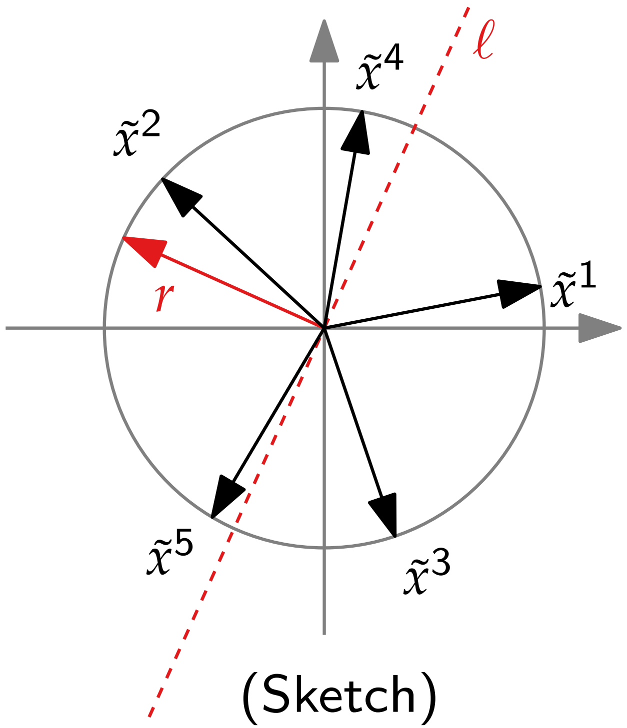
Compute optimal solution $(\tilde{x}^1, \dots, \tilde{x}^n)$ for QP$^2(G, w)$

Pick random vector $r \in \mathbb{R}^2$

$S \leftarrow \{v_i \in V(G) : \tilde{x}^i \cdot r \geq 0\}$

return $c(S, V(G) \setminus S)$





(Sketch)

Algorithm RandomizedMaxCut

RandomizedMaxCut(G, w)

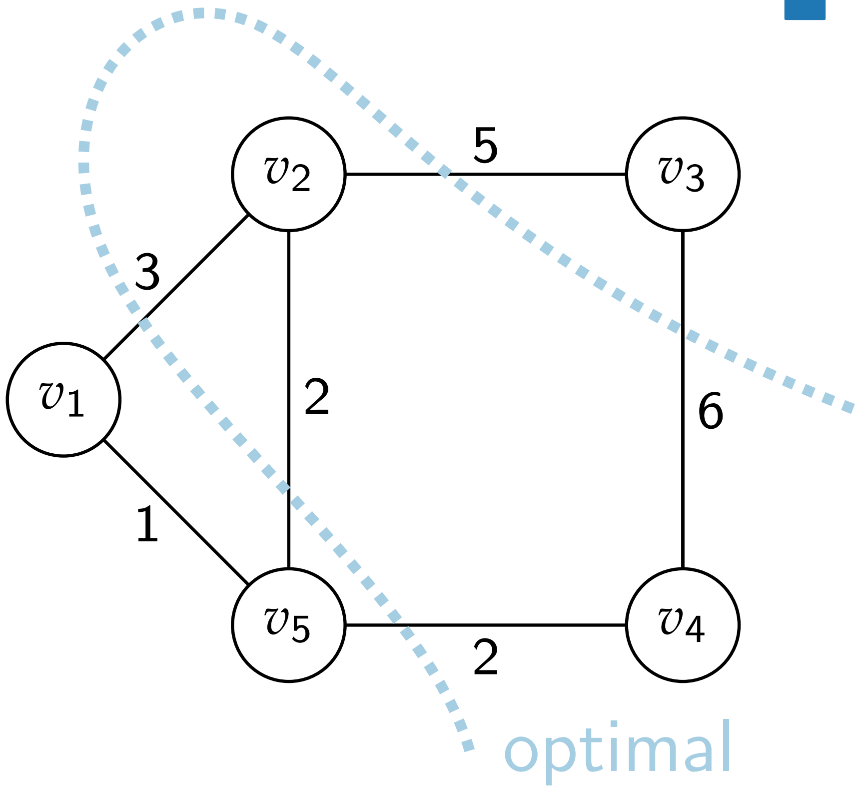
Compute optimal solution $(\tilde{x}^1, \ldots, \tilde{x}^n)$ for QP$^2(G, w)$

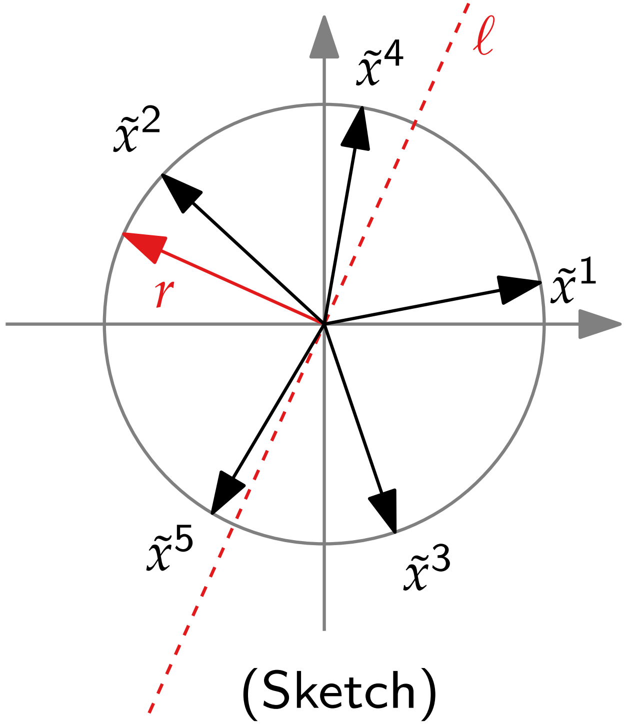
Pick random vector $r \in \mathbb{R}^2$

$S \leftarrow \{v_i \in V(G) : \tilde{x}^i \cdot r \geq 0\}$

return $c(S, V(G) \setminus S)$

- $\tilde{x}^i$ lies above the line ℓ orthogonal to r





(Sketch)

Algorithm RandomizedMaxCut

RandomizedMaxCut(G, w)

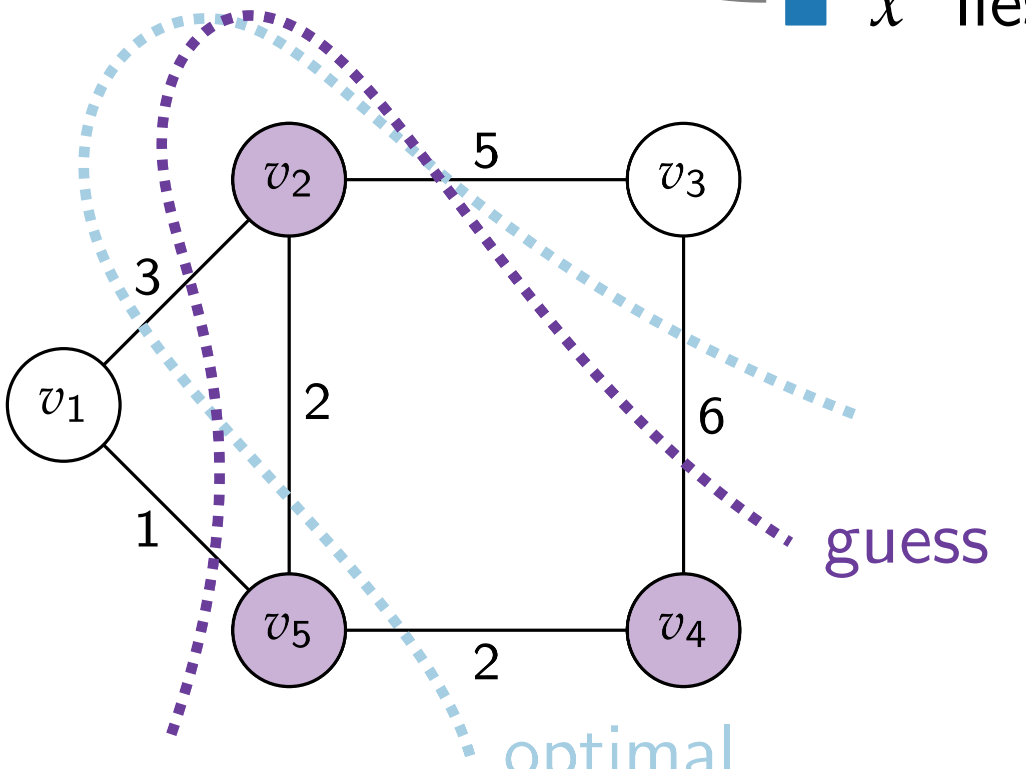
Compute optimal solution $(\tilde{x}^1, \ldots, \tilde{x}^n)$ for QP$^2(G, w)$

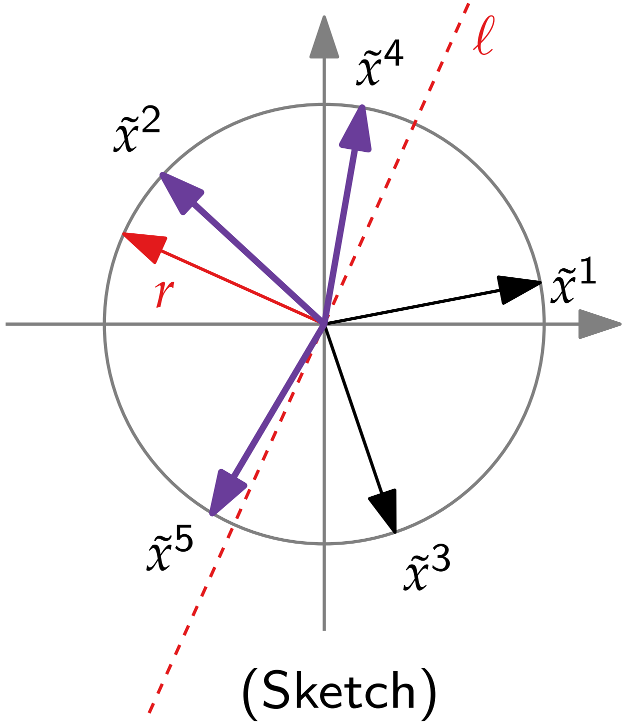
Pick random vector $r \in \mathbb{R}^2$

$S \leftarrow \{v_i \in V(G) : \tilde{x}^i \cdot r \geq 0\}$

return $c(S, V(G) \setminus S)$

- $\tilde{x}^i$ lies above the line ℓ orthogonal to r





(Sketch)

$\textsc{RandomMaxCut}$ – Expected Value

Lemma 2.
Let X be the solution of $\textsc{RandomizedMaxCut}(G, w)$.
If r is picked uniformally at random, then

$$\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} .$$

RandomMaxCut – Expected Value

Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformally at random, then

$$\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} .$$

Proof.

- $E[X] =$

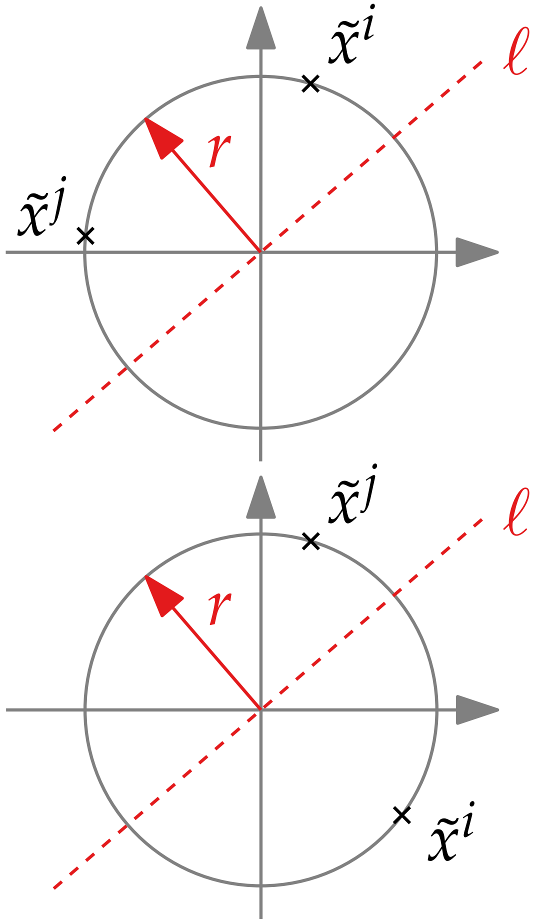
RandomMaxCut – Expected Value

Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformally at random, then

$$\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} \,.$$

Proof.

- $E[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \, \mathsf{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j]$



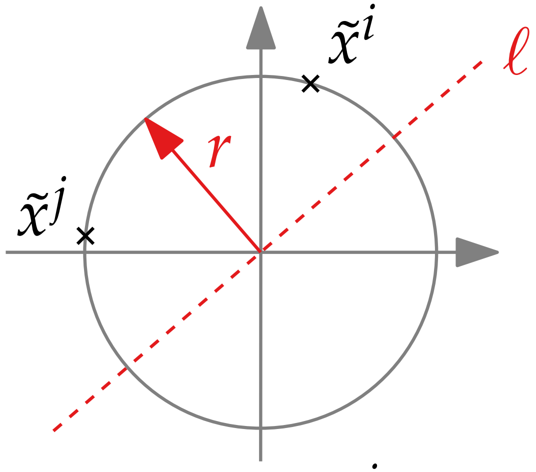
RandomMaxCut – Expected Value

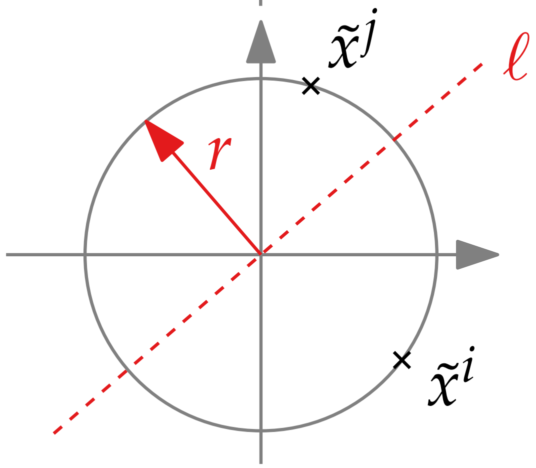
Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformally at random, then

$$\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} .$$

Proof.

- $E[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \,\mathsf{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j]$
- $\mathsf{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j] =$





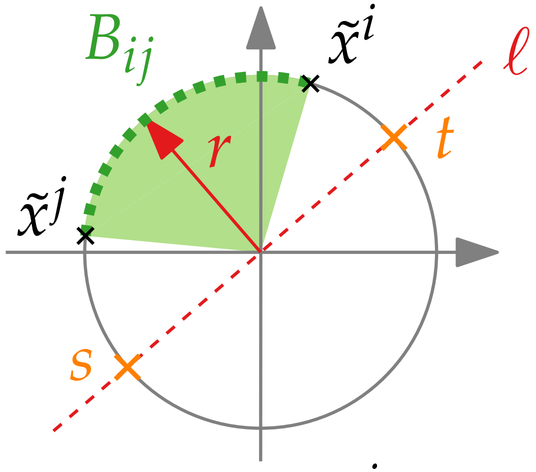
RandomMaxCut – Expected Value

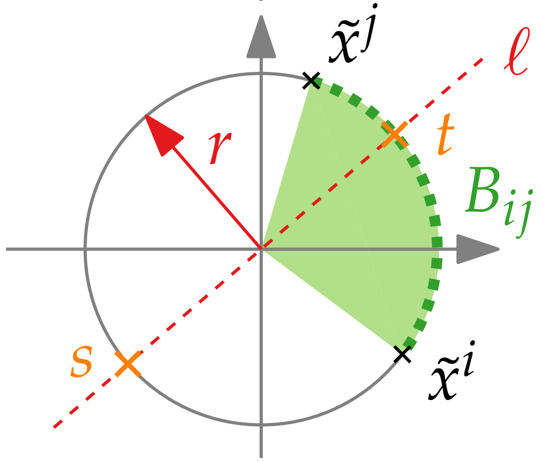
Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformally at random, then

$$\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} \,.$$

Proof.

- $E[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \,\mathsf{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j]$
- $\mathsf{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j] = \mathsf{P}[s \in B_{ij} \text{ or } t \in B_{ij}] =$





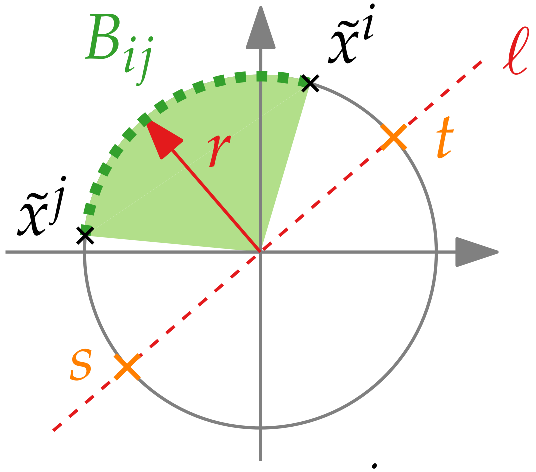
RandomMaxCut – Expected Value

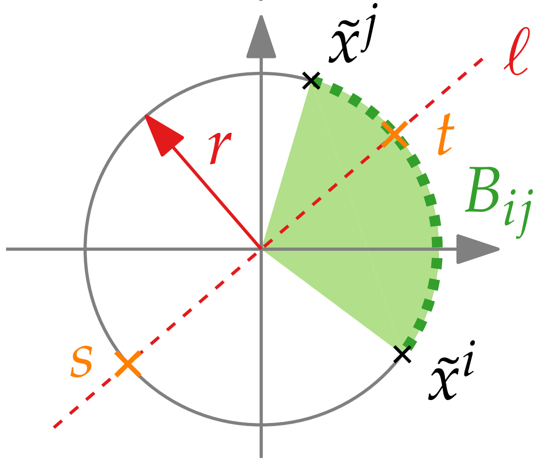
Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformally at random, then

$$\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} .$$

Proof.

- $E[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}\, \mathsf{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j]$
- $\mathsf{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j] = \mathsf{P}[s \in B_{ij} \text{ or } t \in B_{ij}] =$
- B_{ij} has length α_{ij}.

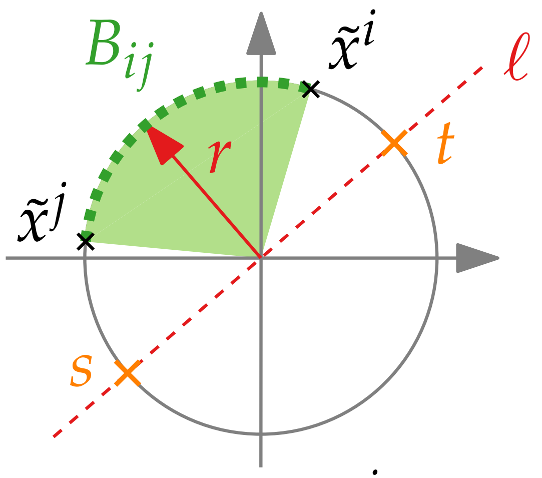


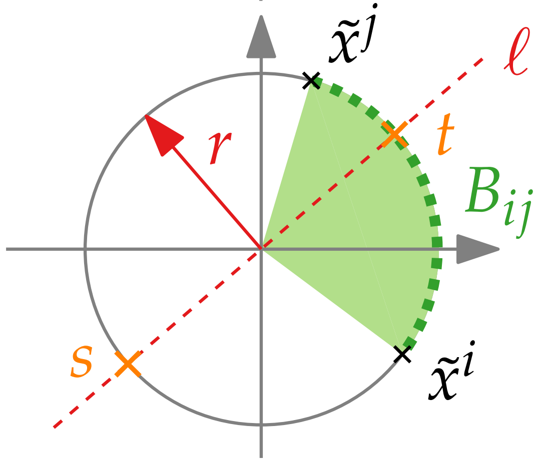


RandomMaxCut – Expected Value

Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformly at random, then

$$\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} \,.$$





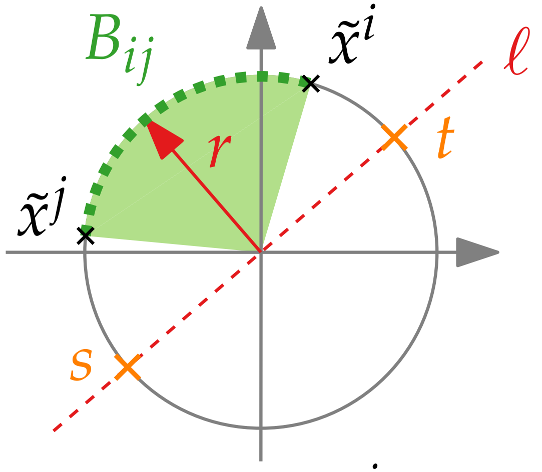
Proof.

- $E[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \, \mathrm{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j]$
- $\mathrm{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j] = \mathrm{P}[s \in B_{ij} \text{ or } t \in B_{ij}] =$
- B_{ij} has length α_{ij}.
- If $\tilde{x}^i$ (or $\tilde{x}^j$) lies $\leq \alpha_{ij}$ before s or t on the perimeter of the unit disk, then s or $t \in B_{ij}$.

RandomMaxCut – Expected Value

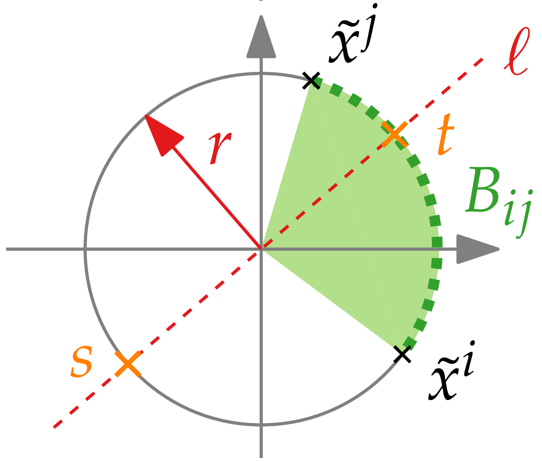
Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformly at random, then

$$\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} .$$



Proof.

- $E[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \, \mathrm{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j]$

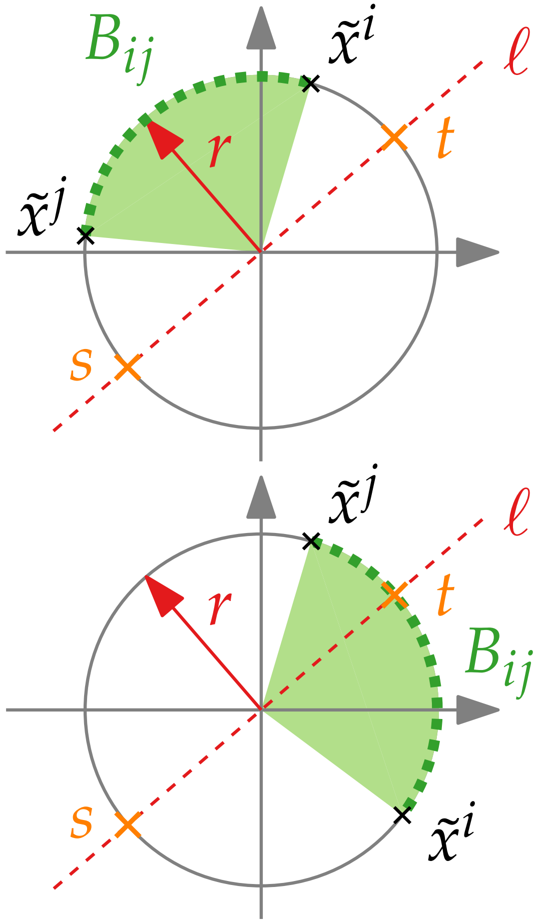


- $\mathrm{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j] = \mathrm{P}[s \in B_{ij} \text{ or } t \in B_{ij}] = \frac{\alpha_{ij}}{2\pi} + \frac{\alpha_{ij}}{2\pi} =$
- B_{ij} has length α_{ij}.
- If $\tilde{x}^i$ (or $\tilde{x}^j$) lies $\leq \alpha_{ij}$ before s or t on the perimeter of the unit disk, then s or $t \in B_{ij}$.

RandomMaxCut – Expected Value

Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformly at random, then

$$\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi} .$$



Proof.

- $E[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \, \mathrm{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j]$
- $\mathrm{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j] = \mathrm{P}[s \in B_{ij} \text{ or } t \in B_{ij}] = \frac{\alpha_{ij}}{2\pi} + \frac{\alpha_{ij}}{2\pi} = \frac{\alpha_{ij}}{\pi}$
- B_{ij} has length α_{ij}.
- If $\tilde{x}^i$ (or $\tilde{x}^j$) lies $\leq \alpha_{ij}$ before s or t on the perimeter of the unit disk, then s or $t \in B_{ij}$.

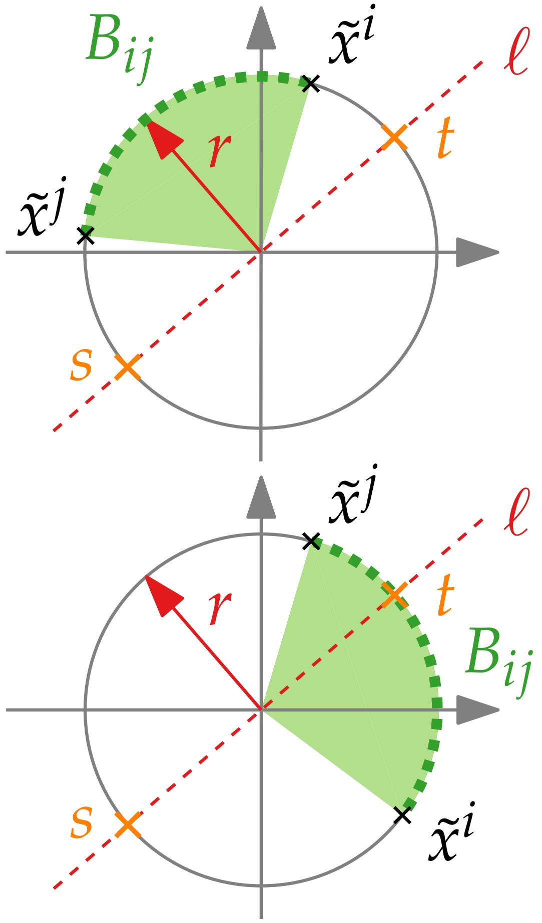
RandomMaxCut – Expected Value

Lemma 2.
Let X be the solution of RandomizedMaxCut(G, w).
If r is picked uniformly at random, then

$$\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}\,.$$

Proof.

- $E[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}\, \mathrm{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- $\mathrm{P}[\ell \text{ separates } \tilde{x}^i, \tilde{x}^j] = \mathrm{P}[s \in B_{ij} \text{ or } t \in B_{ij}] = \frac{\alpha_{ij}}{2\pi} + \frac{\alpha_{ij}}{2\pi} = \frac{\alpha_{ij}}{\pi}$
- B_{ij} has length α_{ij}.
- If $\tilde{x}^i$ (or $\tilde{x}^j$) lies $\leq \alpha_{ij}$ before s or t on the perimeter of the unit disk, then s or $t \in B_{ij}$.



RandomMaxCut – Quality

Theorem 3.
Let X be the solution of $\textsc{RandomizedMaxCut}(G, w)$.
Then
$$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq 0.8785.$$

RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w).
Then
$$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$

RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w).
Then
$$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w). Then
$$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

$$\mathsf{QP}^2(G, w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1 - \cos(\alpha_{ij})}{2}$$

RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w).
Then
$$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

$$\mathrm{QP}^2(G,w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathrm{QP}^2(G,w)$ is relaxation of $\mathrm{QP}(G,w)$:

$$\mathrm{QP}^2(G,w) \geq \mathrm{QP}(G,w) = \mathrm{OPT}(G,w)$$

RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w).
Then
$$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq 0.8785.$$

- $\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq \frac{\mathrm{E}[X]}{\mathrm{QP}^2(G,w)}$

Proof.

- Lemma 2: $\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

$$\mathrm{QP}^2(G, w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathrm{QP}^2(G, w)$ is relaxation of $\mathrm{QP}(G, w)$:

$$\mathrm{QP}^2(G, w) \geq \mathrm{QP}(G, w) = \mathrm{OPT}(G, w)$$

RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w). Then
$$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

$$\mathrm{QP}^2(G, w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathrm{QP}^2(G, w)$ is relaxation of $\mathrm{QP}(G, w)$:
$$\mathrm{QP}^2(G, w) \geq \mathrm{QP}(G, w) = \mathrm{OPT}(G, w)$$

- $$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq \frac{\mathrm{E}[X]}{\mathrm{QP}^2(G,w)} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}}{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}}$$

RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w). Then

$$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

$$\mathsf{QP}^2(G,w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathsf{QP}^2(G,w)$ is relaxation of $\mathsf{QP}(G,w)$:

$$\mathsf{QP}^2(G,w) \geq \mathsf{QP}(G,w) = \mathsf{OPT}(G,w)$$

- $$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq \frac{\mathsf{E}[X]}{\mathsf{QP}^2(G,w)} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}}{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}}$$

- $$\frac{\frac{\alpha_{ij}}{\pi}}{\frac{1-\cos(\alpha_{ij})}{2}}$$

RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w).
Then
$$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq 0.8785.$$

Proof.

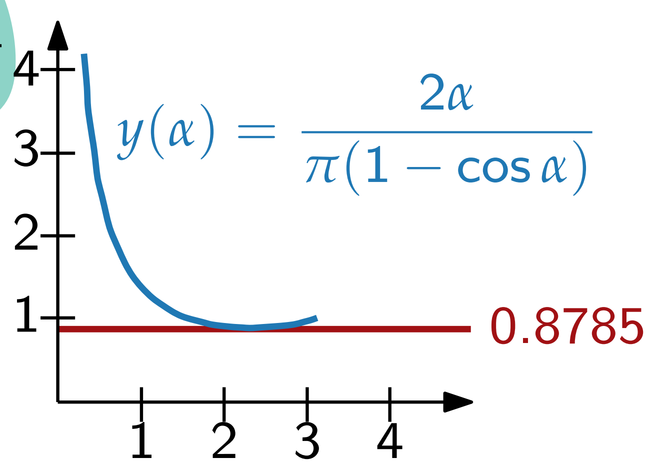
- Lemma 2: $\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

$$\mathrm{QP}^2(G,w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathrm{QP}^2(G,w)$ is relaxation of $\mathrm{QP}(G,w)$:
  $\mathrm{QP}^2(G,w) \geq \mathrm{QP}(G,w) = \mathrm{OPT}(G,w)$

- $$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq \frac{\mathrm{E}[X]}{\mathrm{QP}^2(G,w)} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}}{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}}$$

- $$\frac{\frac{\alpha_{ij}}{\pi}}{\frac{1-\cos(\alpha_{ij})}{2}}$$



RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w).
Then
$$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq 0.8785.$$

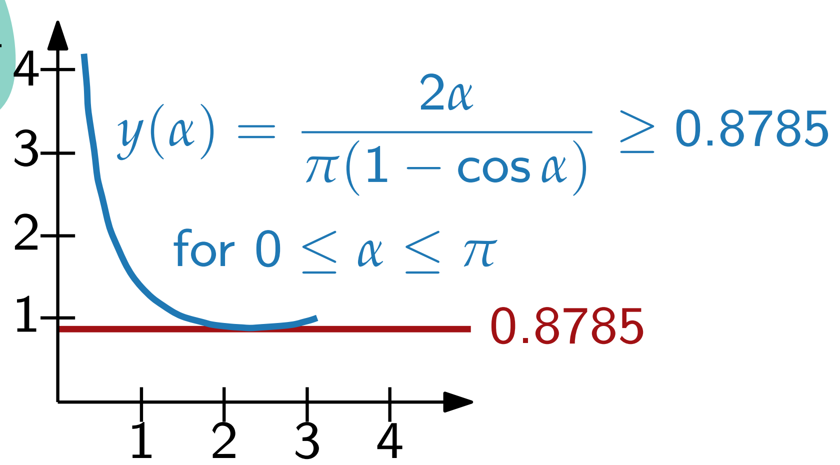
Proof.

- Lemma 2: $\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

$$\mathsf{QP}^2(G, w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathsf{QP}^2(G, w)$ is relaxation of $\mathsf{QP}(G, w)$:
  $$\mathsf{QP}^2(G, w) \geq \mathsf{QP}(G, w) = \mathsf{OPT}(G, w)$$

- $$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq \frac{\mathsf{E}[X]}{\mathsf{QP}^2(G,w)} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}}{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}}$$
- $$\frac{\frac{\alpha_{ij}}{\pi}}{\frac{1-\cos(\alpha_{ij})}{2}}$$



RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w). Then

$$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

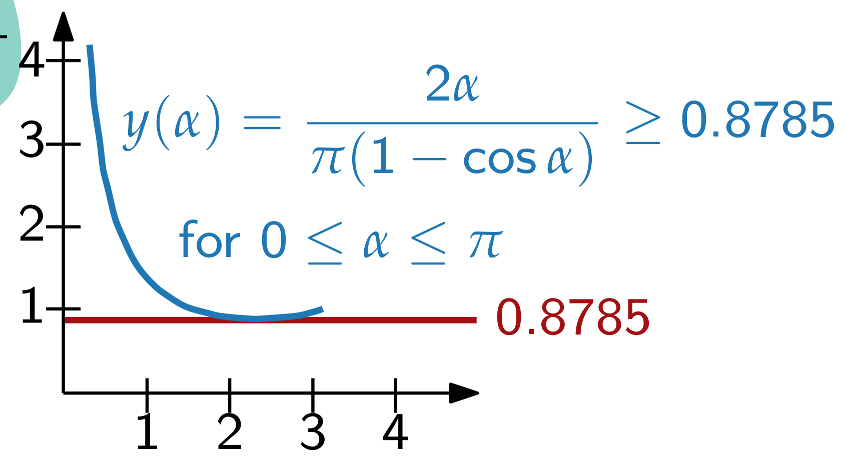
$$\mathrm{QP}^2(G,w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathrm{QP}^2(G,w)$ is relaxation of $\mathrm{QP}(G,w)$:

$$\mathrm{QP}^2(G,w) \geq \mathrm{QP}(G,w) = \mathrm{OPT}(G,w)$$

- $$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq \frac{\mathrm{E}[X]}{\mathrm{QP}^2(G,w)} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}}{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}}$$

- $$\frac{\frac{\alpha_{ij}}{\pi}}{\frac{1-\cos(\alpha_{ij})}{2}} \geq 0.8785$$



RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w). Then
$$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

$$\mathrm{QP}^2(G,w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

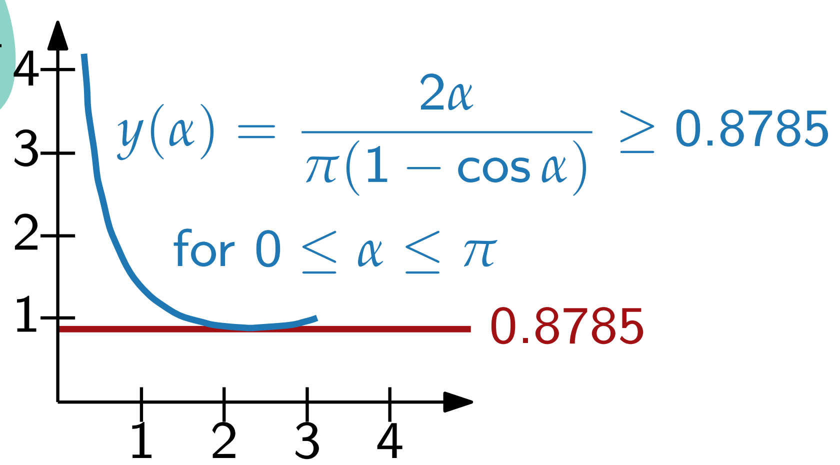
- $\mathrm{QP}^2(G,w)$ is relaxation of $\mathrm{QP}(G,w)$:

$$\mathrm{QP}^2(G,w) \geq \mathrm{QP}(G,w) = \mathrm{OPT}(G,w)$$

- $$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq \frac{\mathrm{E}[X]}{\mathrm{QP}^2(G,w)} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}}{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}}$$

- $$\frac{\frac{\alpha_{ij}}{\pi}}{\frac{1-\cos(\alpha_{ij})}{2}} \geq 0.8785$$

$$\Leftrightarrow \frac{\alpha_{ij}}{\pi} \geq 0.8785 \frac{1-\cos(\alpha_{ij})}{2}$$



RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w).
Then
$$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathsf{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$

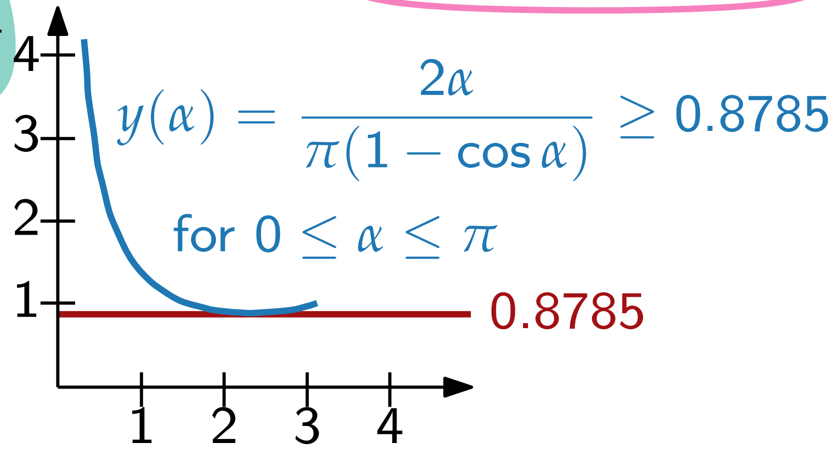
- Optimal solution for QP^2:

$$\mathsf{QP}^2(G,w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathsf{QP}^2(G,w)$ is relaxation of $\mathsf{QP}(G,w)$:
  $\mathsf{QP}^2(G,w) \geq \mathsf{QP}(G,w) = \mathsf{OPT}(G,w)$

- $$\frac{\mathsf{E}[X]}{\mathsf{OPT}(G,w)} \geq \frac{\mathsf{E}[X]}{\mathsf{QP}^2(G,w)} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}}{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}}$$

- $$\frac{\frac{\alpha_{ij}}{\pi}}{\frac{1-\cos(\alpha_{ij})}{2}} \geq 0.8785$$
  $$\Leftrightarrow \frac{\alpha_{ij}}{\pi} \geq 0.8785 \frac{1-\cos(\alpha_{ij})}{2}$$



RandomMaxCut – Quality

Theorem 3.
Let X be the solution of RandomizedMaxCut(G, w). Then

$$\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq 0.8785.$$

Proof.

- Lemma 2: $\mathrm{E}[X] = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}$
- Optimal solution for QP^2:

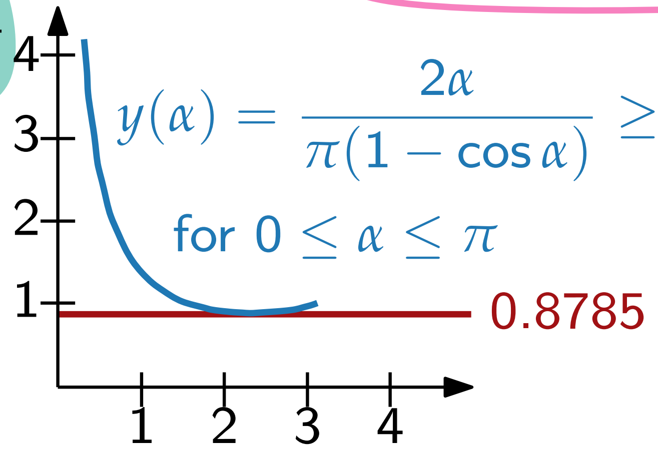
$$\mathrm{QP}^2(G,w) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} (1 - x^i \cdot x^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}$$

- $\mathrm{QP}^2(G,w)$ is relaxation of $\mathrm{QP}(G,w)$:

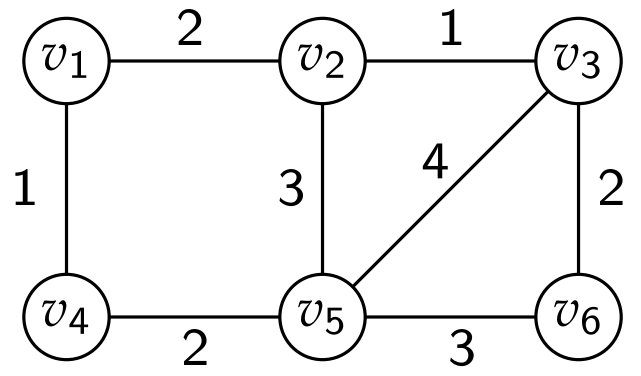
$$\mathrm{QP}^2(G,w) \geq \mathrm{QP}(G,w) = \mathrm{OPT}(G,w)$$

- $\frac{\mathrm{E}[X]}{\mathrm{OPT}(G,w)} \geq \frac{\mathrm{E}[X]}{\mathrm{QP}^2(G,w)} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{\alpha_{ij}}{\pi}}{\sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij} \frac{1-\cos(\alpha_{ij})}{2}} \geq 0.8785$
- $\frac{\frac{\alpha_{ij}}{\pi}}{\frac{1-\cos(\alpha_{ij})}{2}} \geq 0.8785$

$$\Leftrightarrow \frac{\alpha_{ij}}{\pi} \geq 0.8785 \frac{1-\cos(\alpha_{ij})}{2}$$



Example



Example

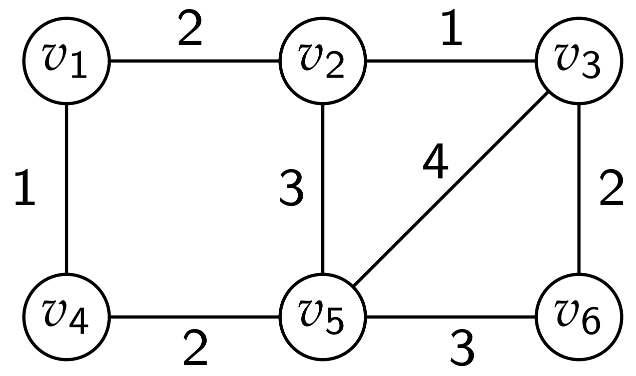
1. Step: Build QP

maximize $\frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j)$

subject to $x_i^2 = 1$

Weight matrix w_{ij}

	1	2	3	4	5	6
1		2		1		
2	2		1		3	
3		1			4	2
4	1				2	
5		3	4	2		3
6			2		3	



Example

1. Step: Build QP

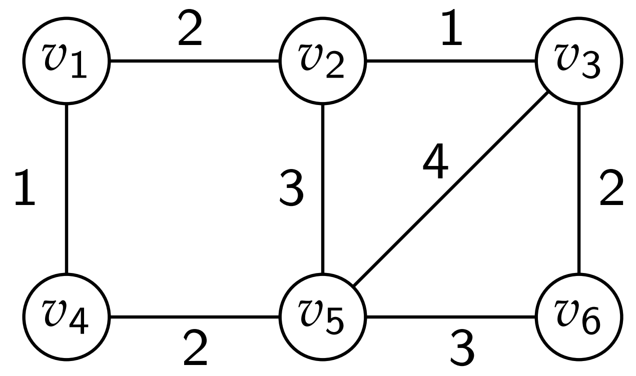
$$\begin{array}{ll} \textbf{maximize} & \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j) \\ \textbf{subject to} & x_i^2 = 1 \end{array}$$

2. Step: Relax QP to QP^2

$$\begin{array}{ll} \textbf{maximize} & \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) \\ \textbf{subject to} & x^i \cdot x^i = 1 \\ & x^i = (x_1^i, x_2^i) \in \mathbb{R}^2 \end{array}$$

Weight matrix w_{ij}

	1	2	3	4	5	6
1		2		1		
2	2		1		3	
3		1			4	2
4	1				2	
5		3	4	2		3
6			2		3	



Example

1. Step: Build QP

$$\begin{aligned} \textbf{maximize} \quad & \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j) \\ \textbf{subject to} \quad & x_i^2 = 1 \end{aligned}$$

Weight matrix w_{ij}

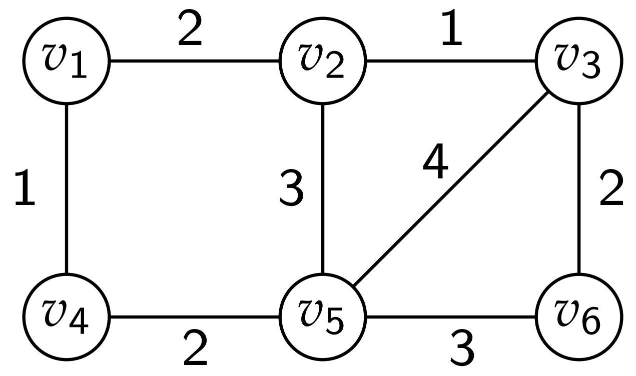
	1	2	3	4	5	6
1		2		1		
2	2		1		3	
3		1			4	2
4	1				2	
5		3	4	2		3
6			2		3	

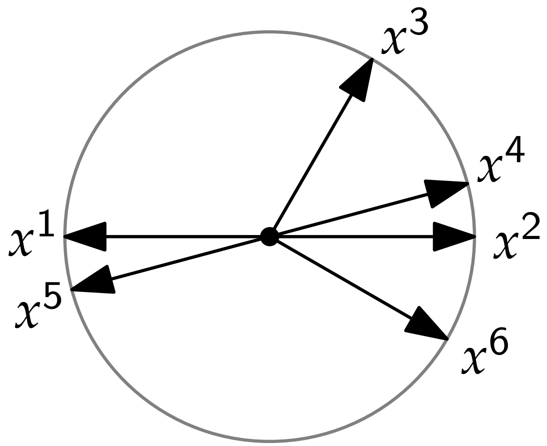
2. Step: Relax QP to QP^2

$$\begin{aligned} \textbf{maximize} \quad & \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) \\ \textbf{subject to} \quad & x^i \cdot x^i = 1 \\ & x^i = (x_1^i, x_2^i) \in \mathbb{R}^2 \end{aligned}$$

3. Step: Solve QP^2

Variable	x^1	x^2	x^3	x^4	x^5	x^6
Angle	0°	180°	120°	165°	345°	210°





Example

1. Step: Build QP

$$\begin{aligned} \textbf{maximize} \quad & \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j) \\ \textbf{subject to} \quad & x_i^2 = 1 \end{aligned}$$

Weight matrix w_{ij}

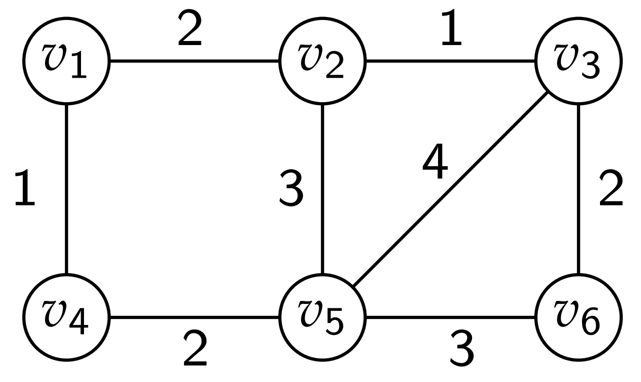
	1	2	3	4	5	6
1		2		1		
2	2		1		3	
3		1			4	2
4	1				2	
5		3	4	2		3
6			2		3	

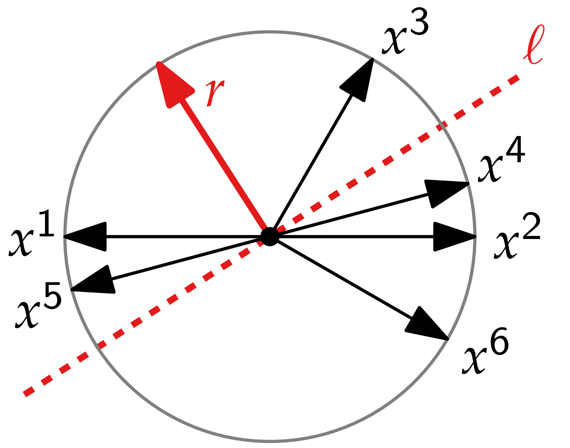
2. Step: Relax QP to QP^2

$$\begin{aligned} \textbf{maximize} \quad & \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) \\ \textbf{subject to} \quad & x^i \cdot x^i = 1 \\ & x^i = (x_1^i, x_2^i) \in \mathbb{R}^2 \end{aligned}$$

3. Step: Solve QP^2

Variable	x^1	x^2	x^3	x^4	x^5	x^6
Angle	0°	180°	120°	165°	345°	210°





4. Step: Guess r

Example

1. Step: Build QP

$$\begin{aligned} \textbf{maximize} \quad & \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j) \\ \textbf{subject to} \quad & x_i^2 = 1 \end{aligned}$$

Weight matrix w_{ij}

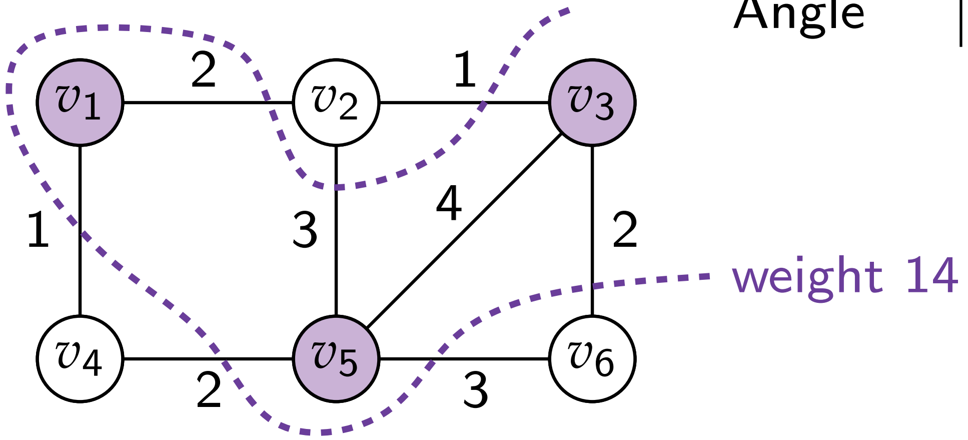
	1	2	3	4	5	6
1		2		1		
2	2		1		3	
3		1			4	2
4	1				2	
5		3	4	2		3
6			2		3	

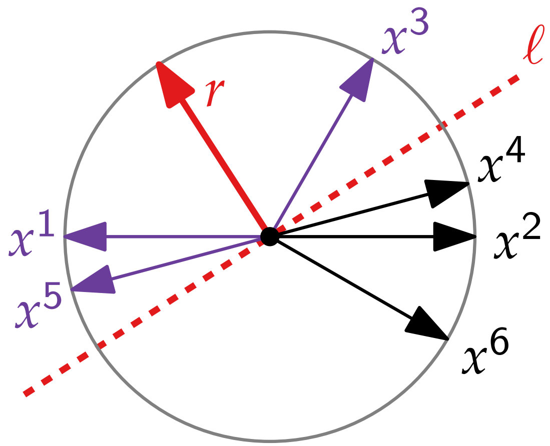
2. Step: Relax QP to QP^2

$$\begin{aligned} \textbf{maximize} \quad & \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) \\ \textbf{subject to} \quad & x^i \cdot x^i = 1 \\ & x^i = (x_1^i, x_2^i) \in \mathbb{R}^2 \end{aligned}$$

3. Step: Solve QP^2

Variable	x^1	x^2	x^3	x^4	x^5	x^6
Angle	0°	180°	120°	165°	345°	210°





4. Step: Guess r

5. Step: Derive S

Example

1. Step: Build QP

$$\text{maximize} \quad \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x_i x_j)$$
$$\text{subject to} \quad x_i^2 = 1$$

Weight matrix w_{ij}

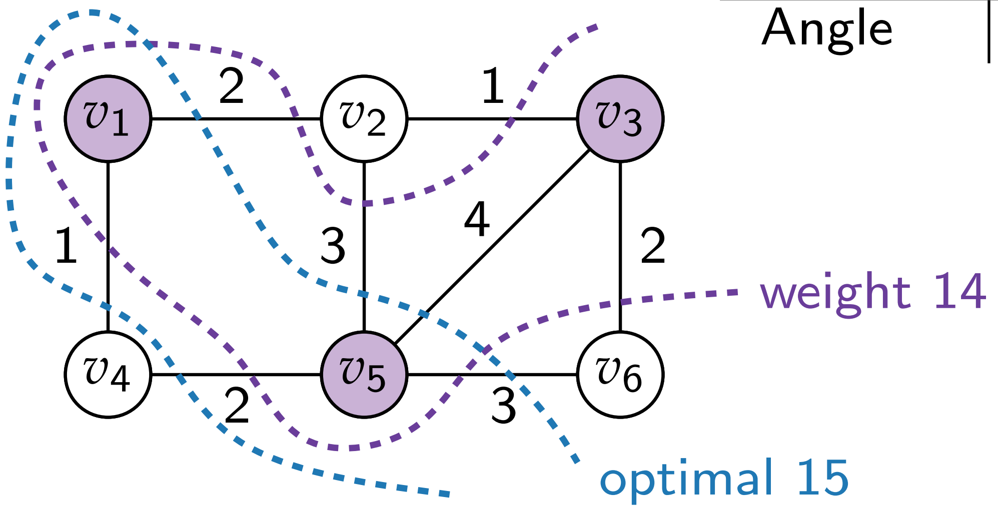
	1	2	3	4	5	6
1		2		1		
2	2		1		3	
3		1			4	2
4	1				2	
5		3	4	2		3
6			2		3	

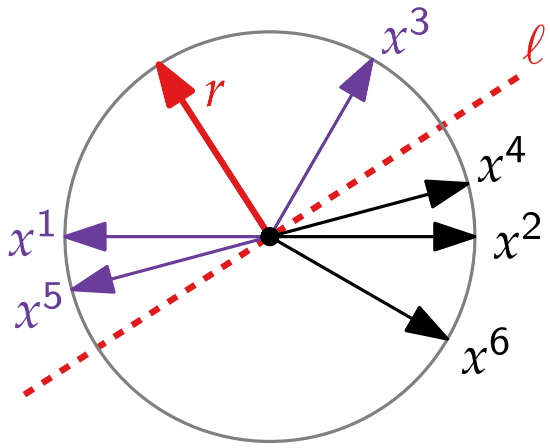
2. Step: Relax QP to QP^2

$$\text{maximize} \quad \frac{1}{2}\sum_{j=1}^{6}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$$
$$\text{subject to} \quad x^i \cdot x^i = 1$$
$$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$$

3. Step: Solve QP^2

Variable	x^1	x^2	x^3	x^4	x^5	x^6
Angle	0°	180°	120°	165°	345°	210°

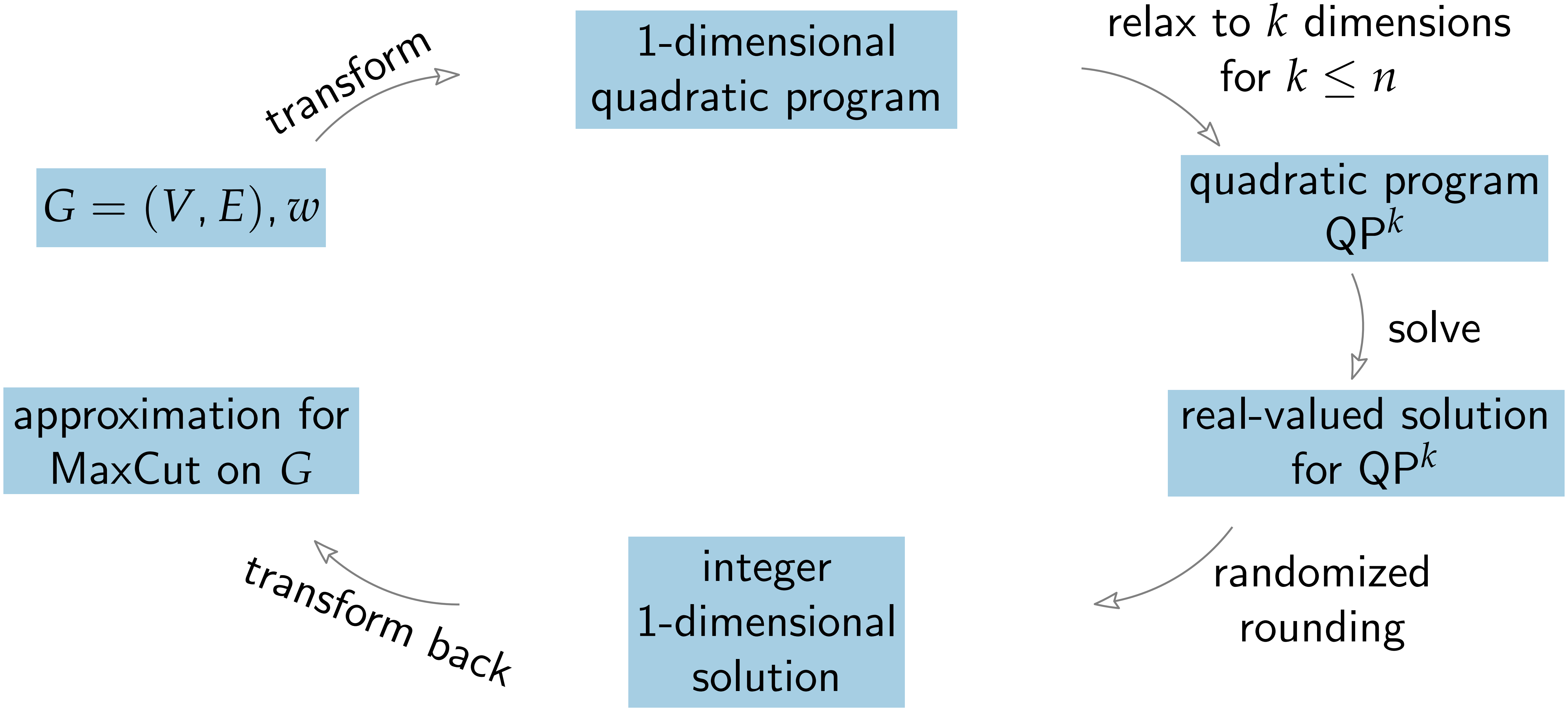




4. Step: Guess r

5. Step: Derive S

Goemans–Williamson Algorithm for MaxCut



Goemans–Williamson Algorithm for MaxCut

$G = (V, E), w$

transform

1-dimensional quadratic program

relax to k dimensions for $k \leq n$

quadratic program QP^k

solve

real-valued solution for QP^k

randomized rounding

integer 1-dimensional solution

transform back

approximation for MaxCut on G

- So far, $k = 2$.
- QP^n can be solved in polynomial time.

$\mathrm{QP}^n(G, w)$

QP$^2(G, w)$

$$\begin{array}{lrl} \textbf{maximize} & \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) & \\ \textbf{subject to} & x^i \cdot x^i & = 1 \\ & x^i = (x^i_1, x^i_2) & \in \mathbb{R}^2 \end{array}$$

QP$^n(G, w)$

$$\begin{array}{lrl} \textbf{maximize} & \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) & \\ \textbf{subject to} & x^i \cdot x^i & = 1 \\ & x^i & \in \mathbb{R}^n \end{array}$$

$\mathrm{QP}^n(G, w)$

QP^2 (G, w)

$$\textbf{maximize} \quad \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$$

$$\textbf{subject to} \quad x^i \cdot x^i = 1$$

$$x^i = (x_1^i, x_2^i) \in \mathbb{R}^2$$

QP^n (G, w)

$$\textbf{maximize} \quad \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$$

$$\textbf{subject to} \quad x^i \cdot x^i = 1$$

$$x^i \in \mathbb{R}^n$$

- A matrix M is called **positive semidefinite** if for any vector $v \in \mathbb{R}^n$:

$$v^\mathsf{T} \cdot M \cdot v \geq 0$$

$\mathrm{QP}^n(G, w)$

$\mathrm{QP}^2(G, w)$

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to
$$x^i \cdot x^i = 1$$
$$x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$$

$\mathrm{QP}^n(G, w)$

maximize $\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j)$

subject to
$$x^i \cdot x^i = 1$$
$$x^i \in \mathbb{R}^n$$

- A matrix M is called **positive semidefinite** if for any vector $v \in \mathbb{R}^n$:
$$v^\mathsf{T} \cdot M \cdot v \geq 0$$
- $M = (m_{ij}) = (x^i \cdot x^j)$ is positive semidefinite.

$\mathrm{QP}^n(G, w)$

$\mathrm{QP}^2(G, w)$

$$\begin{array}{lrl} \textbf{maximize} & \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) & \\ \textbf{subject to} & x^i \cdot x^i & = 1 \\ & x^i = (x_1^i, x_2^i) & \in \mathbb{R}^2 \end{array}$$

$\mathrm{QP}^n(G, w)$

$$\begin{array}{lrl} \textbf{maximize} & \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} w_{ij}(1 - x^i \cdot x^j) & \\ \textbf{subject to} & x^i \cdot x^i & = 1 \\ & x^i & \in \mathbb{R}^n \end{array}$$

- A matrix M is called **positive semidefinite** if for any vector $v \in \mathbb{R}^n$:
  $$v^\intercal \cdot M \cdot v \geq 0$$
- $M = (m_{ij}) = (x^i \cdot x^j)$ is positive semidefinite.
- $\mathrm{QP}^n(G, w)$ becomes the problem $\textsc{SemiDefiniteCut}(G, w)$.
  - Can be approximated in time polynomial in (G, w) and $1/\varepsilon$ with additive guarantee ε.
  - Note that the approximation of $\mathrm{QP}(G, w)$ is an extra step we have seen before. (The approximation of $\mathrm{QP}(G, w)$ with factor 0.8785 works for $\mathrm{QP}^n(G, w)$, too)

Discussion

- If the *Unique Games Conjecture* is true, then the approximation ratio of ≈ 0.8785 achieved by $\textsc{SemiDefiniteCut}$ (and $\textsc{RandomizedMaxCut}$) is best possible.
- Otherwise, no approximation ratio better than $\frac{16}{17} \approx 0.941$ is possible.
  In particular no polynomial-time approximation scheme (PTAS) exists.
- On planar graphs, the MaxCut problem can be solved optimally in polynomial time.

Discussion

- If the *Unique Games Conjecture* is true, then the approximation ratio of ≈ 0.8785 achieved by $\textsc{SemiDefiniteCut}$ (and $\textsc{RandomizedMaxCut}$) is best possible.
- Otherwise, no approximation ratio better than $\frac{16}{17} \approx 0.941$ is possible.
  In particular no polynomial-time approximation scheme (PTAS) exists.
- On planar graphs, the MaxCut problem can be solved optimally in polynomial time.
- Semidefinite programming is a powerful tool to develop approximation algorithms.

Discussion

- If the *Unique Games Conjecture* is true, then the approximation ratio of ≈ 0.8785 achieved by $\textsc{SemiDefiniteCut}$ (and $\textsc{RandomizedMaxCut}$) is best possible.
- Otherwise, no approximation ratio better than $\frac{16}{17} \approx 0.941$ is possible.
  In particular no polynomial-time approximation scheme (PTAS) exists.
- On planar graphs, the MaxCut problem can be solved optimally in polynomial time.
- Semidefinite programming is a powerful tool to develop approximation algorithms.
- Using randomness is another tool to design approximation algorithms.
  $\rightarrow$ See future lectures, in particular the next lecture!

Literature

Original paper:

- [GW '95] "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming"

Source:

- [Vazirani Ch26] "Approximation Algorithms"

Whole book on this topic:

- [Gärtner, Matoušek] "Approximation Algorithms and Semidefinite Progamming"