# Algorithmen II
# Vorlesung am 28.11.2013

Randomisierte Algorithmen – Max Cut

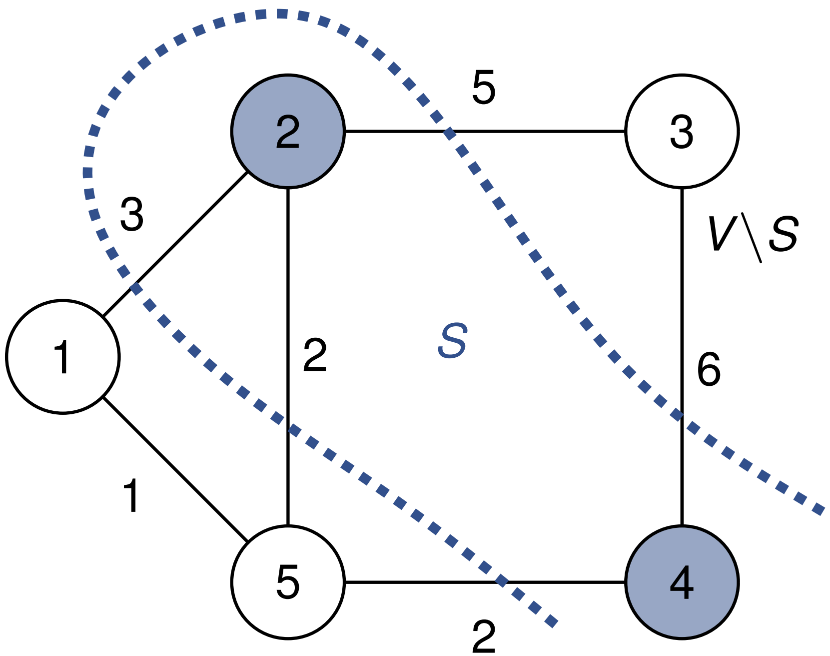
INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

# Problemdefinition

**Problem MAXCUT:** Gegeben ist ein Graph $G = (V, E)$ mit Gewichtsfunktion $c\colon E \longrightarrow \mathbb{N}$. Gesucht ist ein Schnitt $(S, V \setminus S)$ von $G$ mit maximalem Gewicht, d.h.

$$c(S, V \setminus S) := \sum_{\substack{u,v \in E \\ u \in S \text{ und } v \in V \setminus S}} c(\{u, v\}) \quad \text{soll maximal sein.}$$

Problem ist $\mathcal{NP}$-schwer.




$c(S, V \setminus S) = 18$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

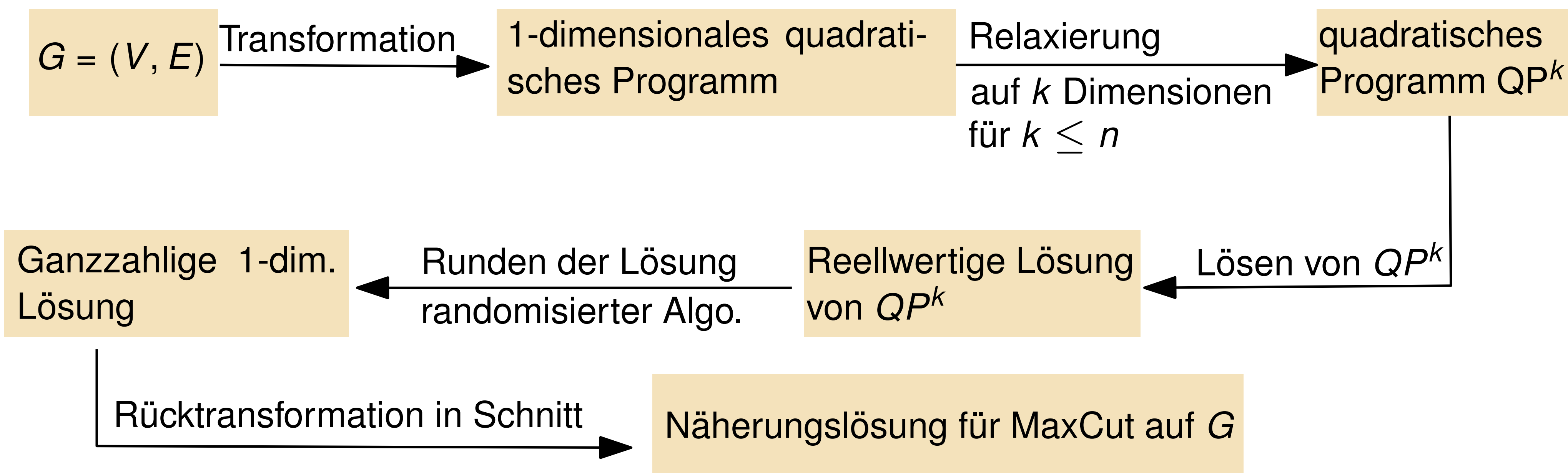
# Problemdefinition

**Problem MaxCut:** Gegeben ist ein Graph $G = (V, E)$ mit Gewichtsfunktion $c\colon E \longrightarrow \mathbb{N}$. Gesucht ist ein Schnitt $(S, V \setminus S)$ von $G$ mit maximalem Gewicht, d.h.

$$c(S, V \setminus S) := \sum_{\substack{u,v \,\in\, E \\ u \in S \text{ und } v \in V \setminus S}} c(\{u, v\}) \qquad \text{soll maximal sein.}$$

Problem ist $\mathcal{NP}$-schwer.

**Idee:** Berechne Lösung, die annähernd so gut ist, wie die optimale Lösung.



# Problemdefinition



**Problem MaxCut:** Gegeben ist ein Graph $G = (V, E)$ mit Gewichtsfunktion $c\colon E \longrightarrow \mathbb{N}$.
Gesucht ist ein Schnitt $(S, V \setminus S)$ von $G$ mit maximalem Gewicht, d.h.

$$c(S, V \setminus S) := \sum_{\substack{u,v \,\in\, E \\ u \in S \text{ und } v \in V \setminus S}} c(\{u, v\}) \qquad \text{soll maximal sein.}$$

Problem ist $\mathcal{NP}$-schwer.

**Idee:** Berechne Lösung, die annähernd so gut ist, wie die optimale Lösung.

- $G = (V, E)$ —Transformation→ 1-dimensionales quadratisches Programm
- 1-dimensionales quadratisches Programm —Relaxierung auf $k$ Dimensionen für $k \leq n$→ quadratisches Programm $QP^k$
- quadratisches Programm $QP^k$ —Lösen von $QP^k$→ Reellwertige Lösung von $QP^k$
- Reellwertige Lösung von $QP^k$ —Runden der Lösung, randomisierter Algo.→ Ganzzahlige 1-dim. Lösung
- Ganzzahlige 1-dim. Lösung —Rücktransformation in Schnitt→ Näherungslösung für MaxCut auf $G$

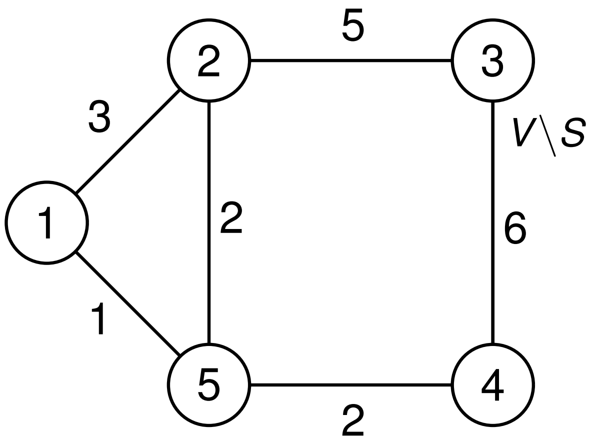
# 1-Dimensionales Quadratisches Programm

Zu $i$ und $j \in V := \{1, \ldots, n\}$ definiere $c_{ij} := \begin{cases} c(\{i,j\}) & \text{falls } \{i,j\} \in E \\ 0 & \text{sonst} \end{cases}$

**QP**($I$):

$$\max \frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter den Nebenbedingungen: $x_i^2 = 1$ für $1 \leq i \leq n$ .




**Idee:**

Jede Variable $x_i$ steht für einen Knoten. Es gilt: $x_i \in \{1, -1\}$

**Es gilt für zwei Knoten i und j:**

| | |
|---|---|
| $x_i = x_j$ | Knoten in der selben Partition. |
| $x_i = -x_j$ | Knoten in unterschiedlichen Partitionen. |

$$c_{ij} \cdot (1 - x_i \cdot x_j) = \begin{cases} 2 & , \text{ wenn } x_i = -x_j \\ 0 & , \text{ wenn } x_i = x_j \end{cases}$$

$\Rightarrow$ $c_{ij}$ geht nur in Zielfunktion ein, wenn $\{i,j\} \in E$ und Knoten befinden sich in unterschiedlichen Partitionen.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

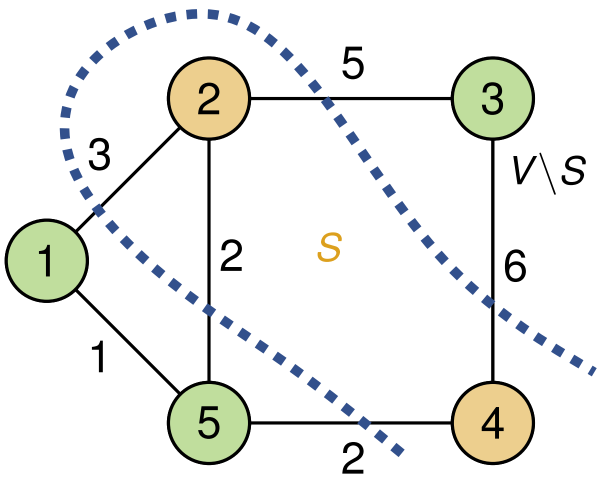
# 1-Dimensionales Quadratisches Programm

Zu $i$ und $j \in V := \{1, \ldots, n\}$ definiere $c_{ij} := \begin{cases} c(\{i,j\}) & \text{falls } \{i,j\} \in E \\ 0 & \text{sonst} \end{cases}$

**QP**($I$):

$$\max \frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter den Nebenbedingungen: $x_i^2 = 1$ für $1 \leq i \leq n$ .




**Gewichtungsmatrix:** $c_{ij}$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | 3 | | | 1 |
| 2 | 3 | | 5 | | 2 |
| 3 | | 5 | 6 | | |
| 4 | | | 6 | | 2 |
| 5 | 1 | 2 | | 2 | |

**Lösung:**

$x_2 = x_4 = 1$

$x_1 = x_5 = x_3 = -1$

# 1-Dimensionales Quadratisches Programm

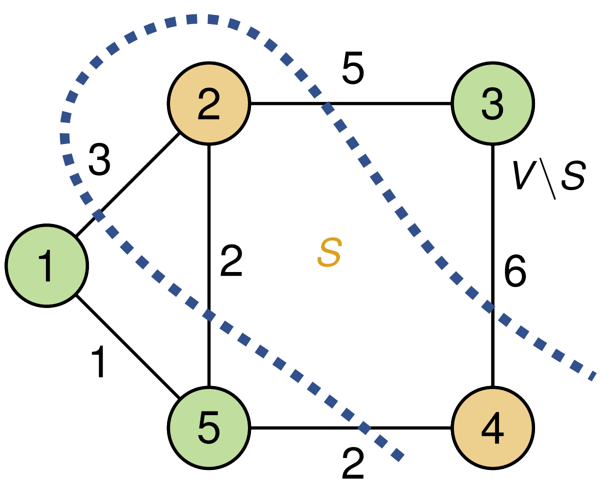


Zu $i$ und $j \in V := \{1, \ldots, n\}$ definiere $c_{ij} := \begin{cases} c(\{i,j\}) & \text{falls } \{i,j\} \in E \\ 0 & \text{sonst} \end{cases}$

**QP**($I$):

$$\max \frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter den Nebenbedingungen: $x_i^2 = 1$ für $1 \leq i \leq n$ .




**Gewichtungsmatrix:** $c_{ij}$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | 3 | | | 1 |
| 2 | 3 | | 5 | | 2 |
| 3 | | 5 | 6 | | |
| 4 | | | 6 | | 2 |
| 5 | 1 | 2 | | 2 | |

**Lösung:**

$x_2 = x_4 = 1$

$x_1 = x_5 = x_3 = -1$

**Problem:**

Lösen von QP(I) ist $\mathcal{NP}$-schwer.

Ansonsten wäre MAXCUT nicht $\mathcal{NP}$-schwer.

# 1-Dimensionales Quadratisches Programm

Karlsruher Institut für Technologie

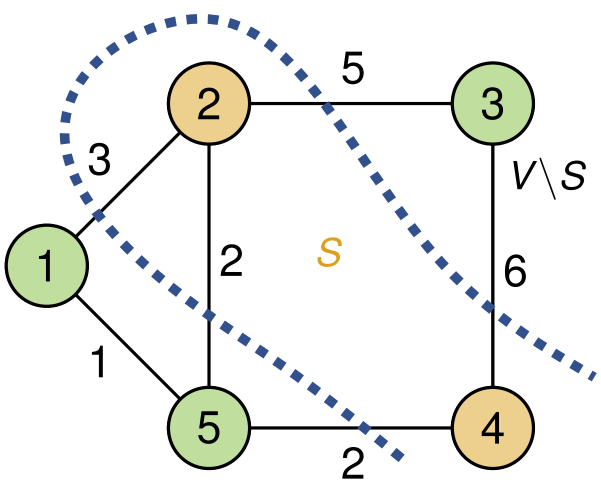
Zu $i$ und $j \in V := \{1, \ldots, n\}$ definiere $c_{ij} := \begin{cases} c(\{i,j\}) & \text{falls } \{i,j\} \in E \\ 0 & \text{sonst} \end{cases}$

**QP**($I$):

$$\max \frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - \underline{x_i \cdot x_j})$$

unter den Nebenbedingungen: $x_i^2 = 1$ für $1 \leq i \leq n$ .

**Beachte:**
$x_i$ und $x_j$ sind 1-dimensionale Vektoren.




**Gewichtungsmatrix:** $c_{ij}$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | 3 | | | 1 |
| 2 | 3 | | 5 | | 2 |
| 3 | | 5 | 6 | | |
| 4 | | | 6 | | 2 |
| 5 | 1 | 2 | | 2 | |

**Lösung:**

$x_2 = x_4 = 1$

$x_1 = x_5 = x_3 = -1$

**Problem:**
Lösen von QP(I) ist $\mathcal{NP}$-schwer.
Ansonsten wäre MaxCut nicht $\mathcal{NP}$-schwer.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Problemdefinition

KIT Karlsruher Institut für Technologie

**Problem MaxCut:** Gegeben ist ein Graph $G = (V, E)$ mit Gewichtsfunktion $c \colon E \longrightarrow \mathbb{N}$.
Gesucht ist ein Schnitt $(S, V \setminus S)$ von $G$ mit maximalem Gewicht, d.h.

$$c(S, V \setminus S) := \sum_{\substack{u,v \,\in\, E \\ u \in S \text{ und } v \in V \setminus S}} c(\{u, v\}) \qquad \text{soll maximal sein.}$$

Problem ist $\mathcal{NP}$-schwer.

**Idee:** Berechne Lösung, die annähernd so gut ist, wie die optimale Lösung.

$G = (V, E)$ Transformation → 1-dimensionales quadratisches Programm — Relaxierung auf $k$ Dimensionen für $k \leq n$ → quadratisches Programm QP$^k$

Ganzzahlige 1-d Lösung

Rücktransf

von $QP^k$

Betrachte den Fall $k = 2$.

**Vorteil:** Prinzip leichter ersichtlich.

**Nachteil:** Nicht bekannt ob $QP^2$ optimal in poly. Zeit gelöst werden kann.

Entsprechendes $QP^n$ kann in poly. Zeit gelöst werden (später).



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Relaxierung von IQP(I)

**QP**$^2(I)$: Sei $x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$ normierter Vektor.

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

unter den Nebenbedingungen $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren

# Relaxierung von IQP(I)

**QP**$^2(I)$: Sei $x^i = (x_1^i, x_2^i) \in \mathbb{R}^2$ normierter Vektor.

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

unter den Nebenbedingungen $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren

Für das Produkt von $x^i$ und $x^j$ gilt: $x^i \cdot x^j = x_1^i \cdot x_1^j + x_2^i \cdot x_2^j = cos(\alpha_{ij})$ mit $0 \leq \alpha_{ij} \leq \pi$

Damit wird $\frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - cos(\alpha_{ij}))$ über Winkel $\alpha_{ij}$ maximiert.

# Relaxierung von IQP(I)

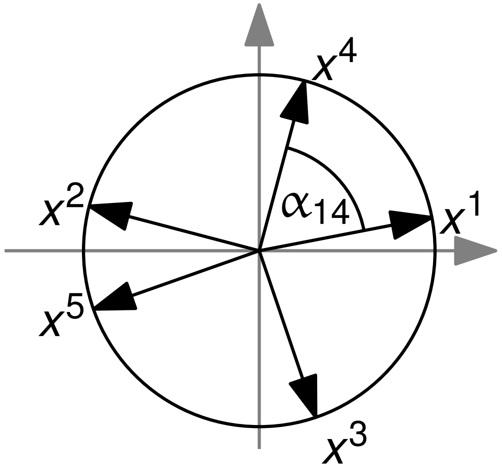
**QP$^2$($I$):** Sei $x^i = (x^i_1, x^i_2) \in \mathbb{R}^2$ normierter Vektor.

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

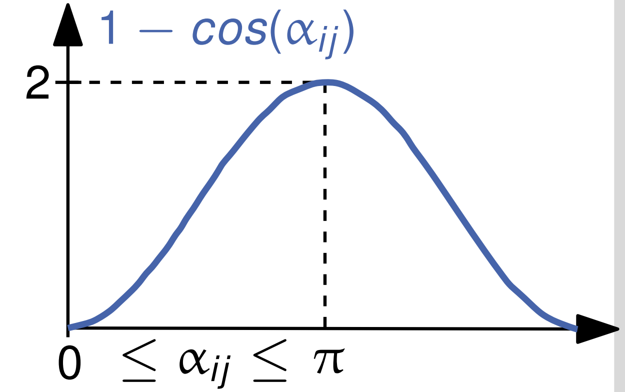
unter den Nebenbedingungen $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren

Für das Produkt von $x^i$ und $x^j$ gilt: $x^i \cdot x^j = x^i_1 \cdot x^j_1 + x^i_2 \cdot x^j_2 = cos(\alpha_{ij})$ mit $0 \leq \alpha_{ij} \leq \pi$

Damit wird $\frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - cos(\alpha_{ij}))$ über Winkel $\alpha_{ij}$ maximiert.




**Beobachtung:**
Je größer der Winkel $\alpha_{ij}$ zwischen $x_i$ und $x_j$ ist, umso mehr trägt $c_{ij}$ zur Summe bei.



# Relaxierung von IQP(I)

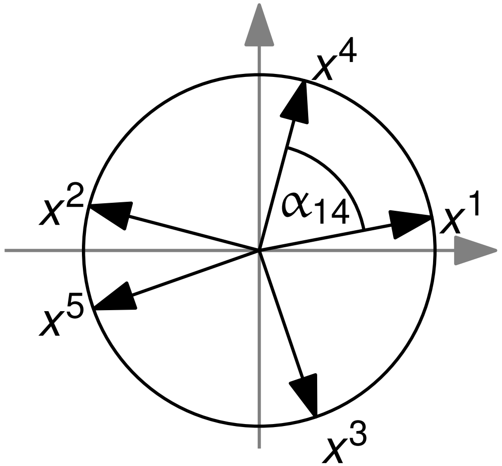
**QP$^2$($I$):** Sei $x^i = (x_1^i, x_2^i) \in \mathbb{R}^2$ normierter Vektor.

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

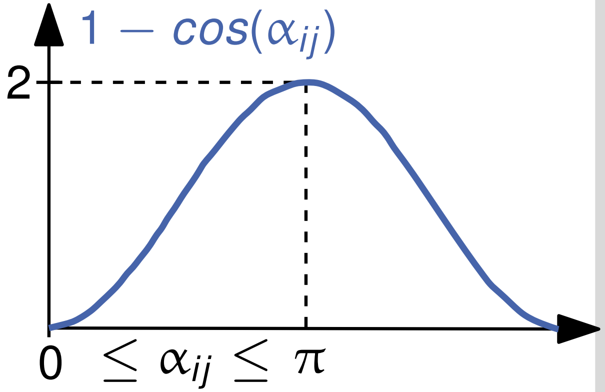
unter den Nebenbedingungen $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren

Für das Produkt von $x^i$ und $x^j$ gilt: $x^i \cdot x^j = x_1^i \cdot x_1^j + x_2^i \cdot x_2^j = cos(\alpha_{ij})$ mit $0 \leq \alpha_{ij} \leq \pi$

Damit wird $\frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - cos(\alpha_{ij}))$ über Winkel $\alpha_{ij}$ maximiert.




**Beobachtung:**
Je größer der Winkel $\alpha_{ij}$ zwischen $x_i$ und $x_j$ ist, umso mehr trägt $c_{ij}$ zur Summe bei.




Lösung $(x_1, \ldots, x_n)$ von $QP(I)$ induziert Lösung $(x^1, \ldots, x^n)$ von QP$^2$($I$) mittels $x^i = (x_i, 0)$.
$\Rightarrow$ **QP$^2$(I) ist Relaxierung von QP(I)**

# Problemdefinition

**Problem MaxCut:** Gegeben ist ein Graph $G = (V, E)$ mit Gewichtsfunktion $c\colon E \longrightarrow \mathbb{N}$.
Gesucht ist ein Schnitt $(S, V \setminus S)$ von $G$ mit maximalem Gewicht, d.h.

$$c(S, V \setminus S) := \sum_{\substack{u,v \,\in\, E \\ u \in S \text{ und } v \in V \setminus S}} c(\{u, v\}) \qquad \text{soll maximal sein.}$$

Problem ist $\mathcal{NP}$-schwer.

**Idee:** Berechne Lösung, die annähernd so gut ist, wie die optimale Lösung.

$G = (V, E)$ — Transformation → 1-dimensionales quadratisches Programm — Relaxierung auf $k$ Dimensionen für $k \leq n$ → quadratisches Programm $QP^k$

Lösen von $QP^k$ → Reellwertige Lösung von $QP^k$

Runden der Lösung, randomisierter Algo. → Ganzzahlige 1-dim. Lösung

Beispielhaft für $k = 2$.

Rücktransformation in Schnitt → Näherungslösung für MaxCut auf $G$

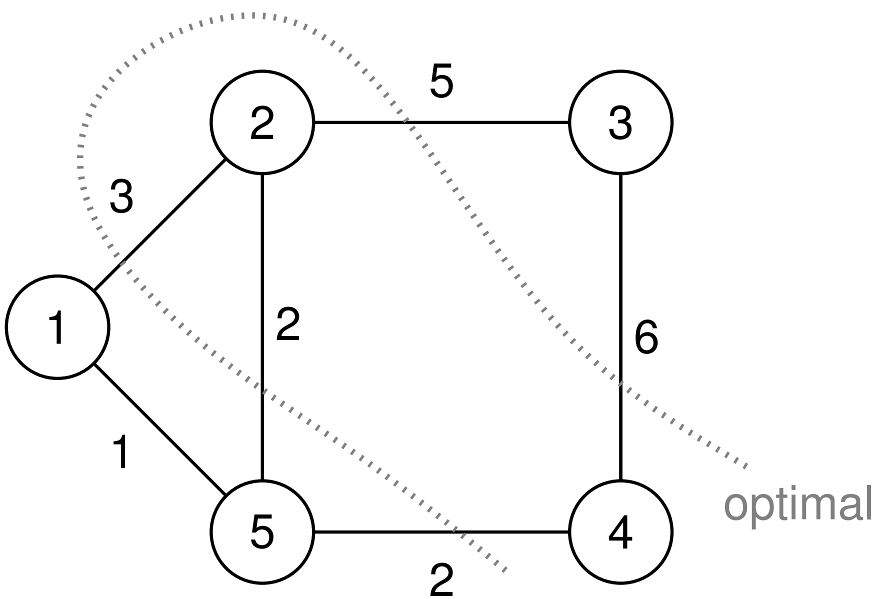
# Random MaxCut

**Eingabe**: Graph $G = (V, E)$ mit einer Gewichtsfunktion $c : E \longrightarrow \mathbb{N}$

**Ausgabe**: Ein Schnitt $(S, V \setminus S)$ in $G$

1. Berechne optimale Lösung $(\tilde{x}^1, \ldots, \tilde{x}^n)$ für QP$^2$(G,c)
2. Wähle zufällig einen zweidimensionalen Vektor $r$ mit Norm 1
3. $S \leftarrow \{i \in V : \underline{\tilde{x}^i \cdot r \geq 0}\}$

$\widetilde{x}_i$ liegt oberhalb der zu $r$ senkrechten Linie $\ell$






Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Random MaxCut

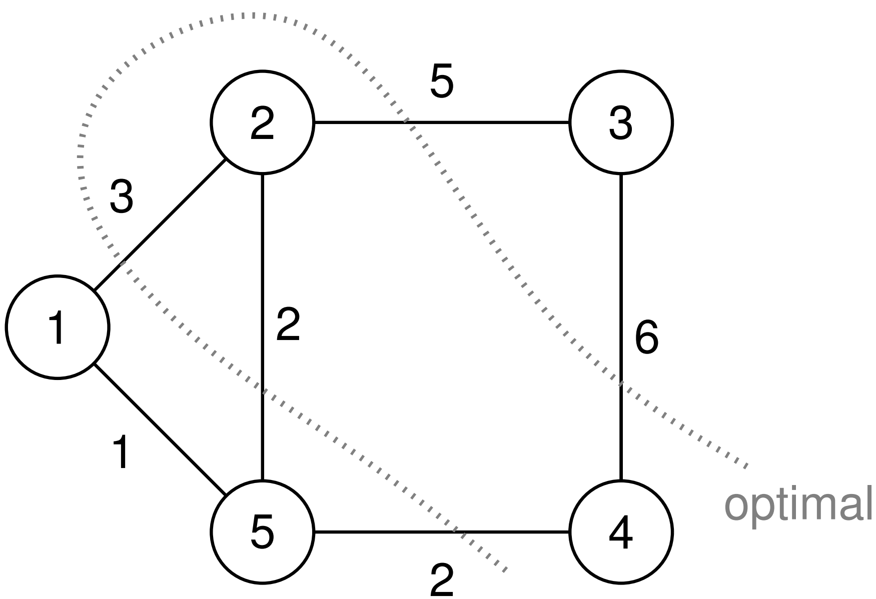


**Eingabe**: Graph $G = (V, E)$ mit einer Gewichtsfunktion $c : E \longrightarrow \mathbb{N}$

**Ausgabe**: Ein Schnitt $(S, V \setminus S)$ in $G$

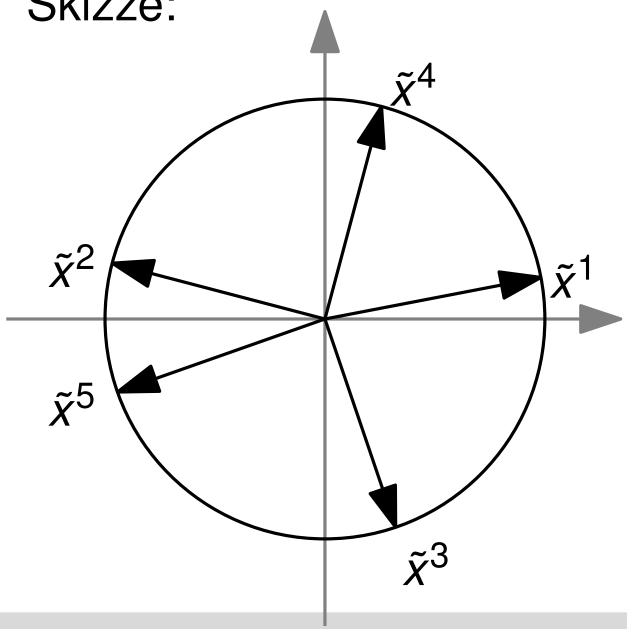
1. Berechne optimale Lösung $(\tilde{x}^1, \dots, \tilde{x}^n)$ für QP$^2$(G,c)
2. Wähle zufällig einen zweidimensionalen Vektor $r$ mit Norm 1
3. $S \leftarrow \{i \in V : \underline{\tilde{x}^i \cdot r \geq 0}\}$

$\rightarrow$ $\tilde{x}_i$ liegt oberhalb der zu $r$ senkrechten Linie $\ell$




**1. Schritt: Berechne Lösung für $QP^2$:**

Skizze:



# Random MaxCut

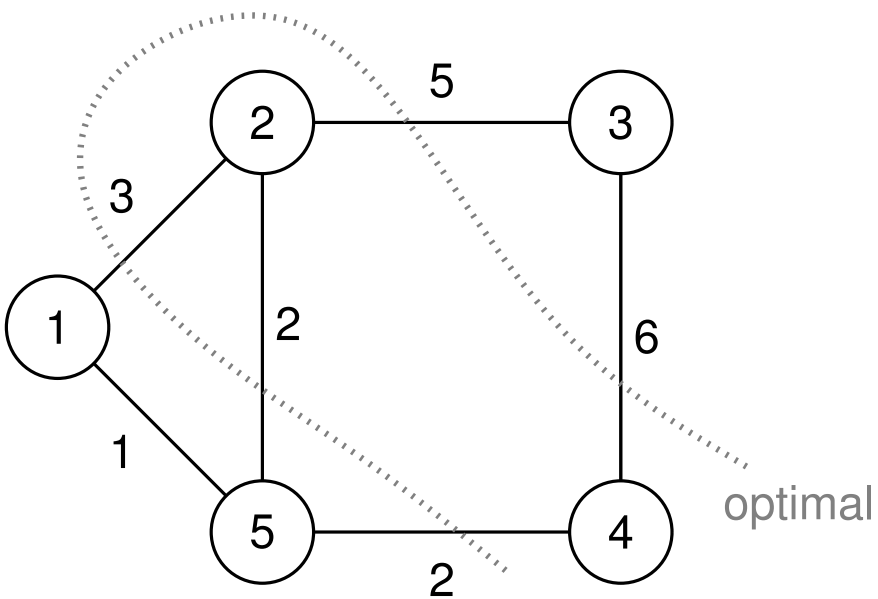
**Eingabe**: Graph $G = (V, E)$ mit einer Gewichtsfunktion $c : E \longrightarrow \mathbb{N}$

**Ausgabe**: Ein Schnitt $(S, V \setminus S)$ in $G$

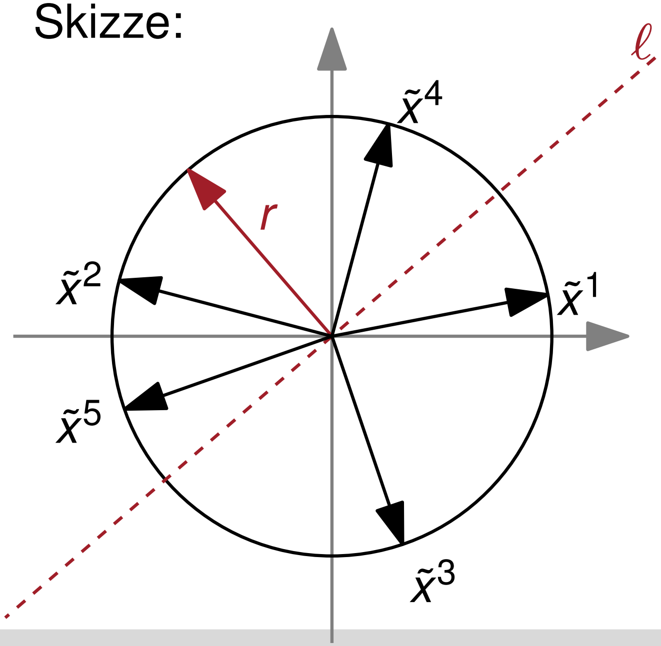
1. Berechne optimale Lösung $(\tilde{x}^1, \ldots, \tilde{x}^n)$ für QP$^2$(G,c)
2. Wähle zufällig einen zweidimensionalen Vektor $r$ mit Norm 1
3. $S \leftarrow \{i \in V : \tilde{x}^i \cdot r \geq 0\}$

$\widetilde{x}_i$ liegt oberhalb der zu $r$ senkrechten Linie $\ell$

**2. Schritt: Rate Vektor $r$**




Skizze:






Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Random MaxCut



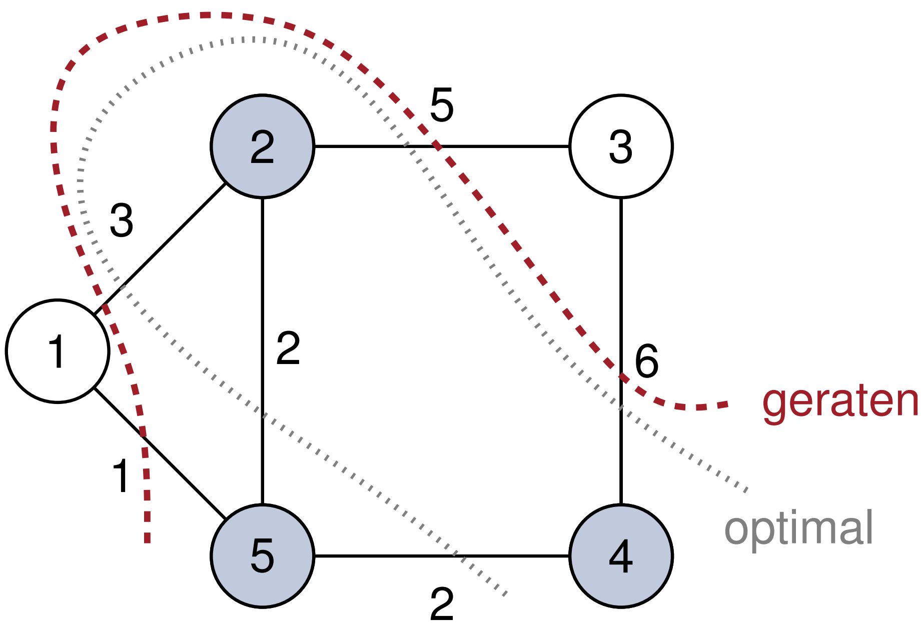
**Eingabe**: Graph $G = (V, E)$ mit einer Gewichtsfunktion $c : E \longrightarrow \mathbb{N}$

**Ausgabe**: Ein Schnitt $(S, V \setminus S)$ in $G$

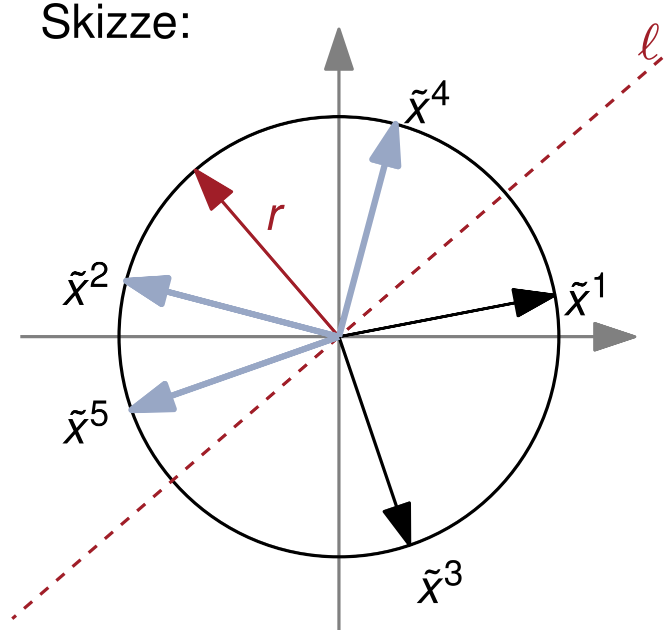
1. Berechne optimale Lösung $(\tilde{x}^1, \ldots, \tilde{x}^n)$ für QP$^2$(G,c)
2. Wähle zufällig einen zweidimensionalen Vektor $r$ mit Norm 1
3. $S \leftarrow \{i \in V : \underbrace{\tilde{x}^i \cdot r \geq 0}\}$

$\longrightarrow$ $\tilde{x}_i$ liegt oberhalb der zu $r$ senkrechten Linie $\ell$

**3. Schritt: Berechne $S$**



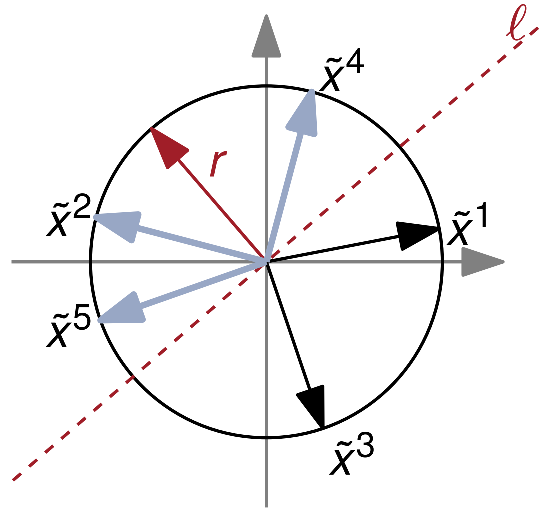

Skizze:



# Random MaxCut

**Satz 8.18:** Sei $I$ eine Instanz für MAXCUT und $C_{\mathrm{RMC}}(I)$ der Wert der Lösung, die RANDOM MAXCUT für $I$ berechnet. Wenn die Vektoren $r$ gleichverteilt angenommen werden, so gilt:

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \arccos(\tilde{x}^i \cdot \tilde{x}^j)$$




**Eingabe**: Graph $G = (V, E)$ mit einer Gewichtsfunktion $c : E \longrightarrow \mathbb{N}$

**Ausgabe**: Ein Schnitt $(S, V \setminus S)$ in $G$

1. Berechne optimale Lösung $(\widetilde{x^1}, \ldots, \widetilde{x^n})$ für QP$^2$(G,c)
2. Wähle zufällig einen zweidimensionalen Vektor $r$ mit Norm 1
3. $S \leftarrow \{i \in V : \widetilde{x^i} \cdot r \geq 0\}$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

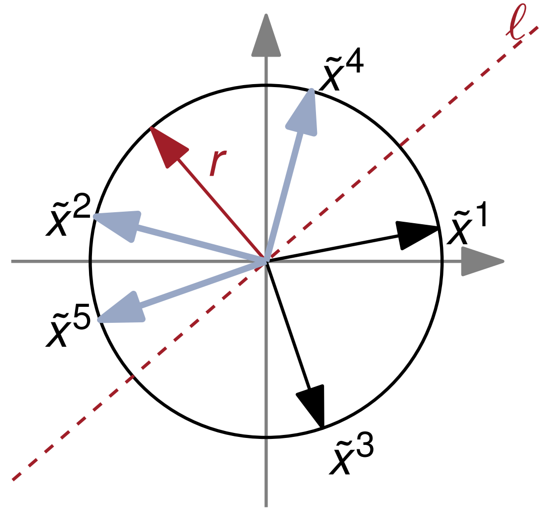
# Random MaxCut

**Satz 8.18:** Sei $I$ eine Instanz für MAXCUT und $C_{\mathrm{RMC}}(I)$ der Wert der Lösung, die RANDOM MAXCUT für $I$ berechnet. Wenn die Vektoren $r$ gleichverteilt angenommen werden, so gilt:

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \arccos(\tilde{x}^i \cdot \tilde{x}^j)$$

**Beweis:**

$$\mathrm{sgn}(x) := \begin{cases} 1 & \text{falls} \quad x \geq 0 \\ -1 & \text{sonst} \end{cases}$$




**Eingabe**: Graph $G = (V, E)$ mit einer Gewichtsfunktion $c : E \longrightarrow \mathbb{N}$

**Ausgabe**: Ein Schnitt $(S, V \setminus S)$ in $G$

1. Berechne optimale Lösung $(\widetilde{x^1}, \ldots, \widetilde{x^n})$ für QP$^2$(G,c)
2. Wähle zufällig einen zweidimensionalen Vektor $r$ mit Norm 1
3. $S \leftarrow \{i \in V : \widetilde{x^i} \cdot r \geq 0\}$

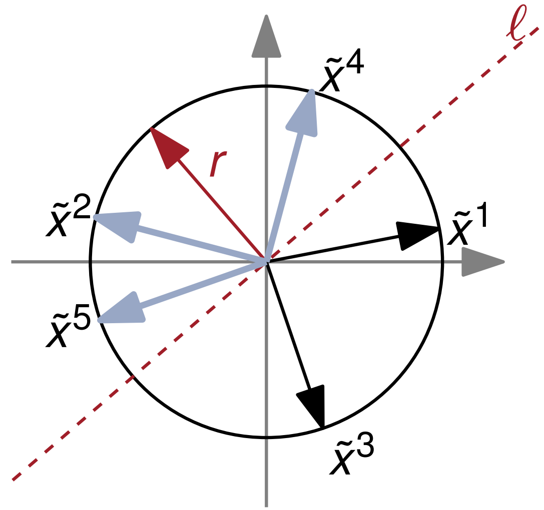
# Random MaxCut

**Satz 8.18:** Sei $I$ eine Instanz für MaxCut und $C_{\mathrm{RMC}}(I)$ der Wert der Lösung, die Random MaxCut für $I$ berechnet. Wenn die Vektoren $r$ gleichverteilt angenommen werden, so gilt:

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \arccos(\tilde{x}^i \cdot \tilde{x}^j)$$

**Beweis:**

$$\operatorname{sgn}(x) := \begin{cases} 1 & \text{falls} \quad x \geq 0 \\ -1 & \text{sonst} \end{cases}$$




**Es gilt:**

$$E(C_{\mathrm{RMC}}(I)) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \Pr[\operatorname{sgn}(\tilde{x}^i \cdot r) \neq \operatorname{sgn}(\tilde{x}^j \cdot r)]$$

wobei $r$ gleichverteilt zufällig gewählt wird.

**Eingabe**: Graph $G = (V, E)$ mit einer Gewichtsfunktion $c : E \longrightarrow \mathbb{N}$
**Ausgabe**: Ein Schnitt $(S, V \setminus S)$ in $G$

1. Berechne optimale Lösung $(\widetilde{x^1}, \ldots, \widetilde{x^n})$ für QP$^2$(G,c)
2. Wähle zufällig einen zweidimensionalen Vektor $r$ mit Norm 1
3. $S \leftarrow \{i \in V : \widetilde{x^i} \cdot r \geq 0\}$

# Random MaxCut

**Satz 8.18:** Sei $I$ eine Instanz für MaxCut und $C_{\mathrm{RMC}}(I)$ der Wert der Lösung, die Random MaxCut für $I$ berechnet. Wenn die Vektoren $r$ gleichverteilt angenommen werden, so gilt:

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \arccos(\tilde{x}^i \cdot \tilde{x}^j)$$

**Beweis:**

$$\mathrm{sgn}(x) := \begin{cases} 1 & \text{falls} \quad x \geq 0 \\ -1 & \text{sonst} \end{cases}$$

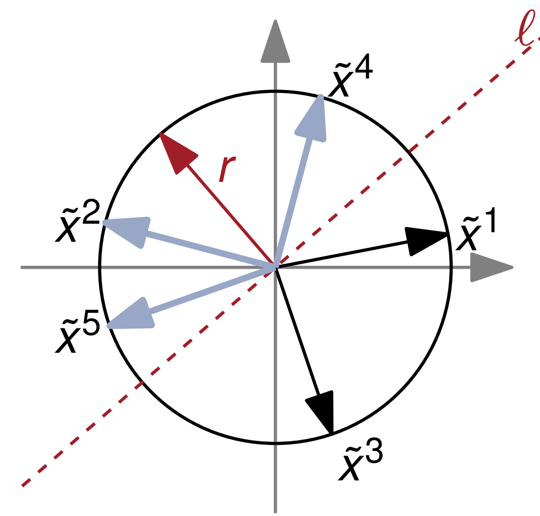
**Es gilt:**

$$E(C_{\mathrm{RMC}}(I)) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \Pr[\mathrm{sgn}(\tilde{x}^i \cdot r) \neq \mathrm{sgn}(\tilde{x}^j \cdot r)]$$

wobei $r$ gleichverteilt zufällig gewählt wird.

**Zeige:**

$$\Pr[\mathrm{sgn}(\tilde{x}^i \cdot r) \neq \mathrm{sgn}(\tilde{x}^j \cdot r)] = \frac{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}{\pi}$$




**Eingabe**: Graph $G = (V, E)$ mit einer Gewichtsfunktion $c : E \longrightarrow \mathbb{N}$

**Ausgabe**: Ein Schnitt $(S, V \setminus S)$ in $G$

1. Berechne optimale Lösung $(\widetilde{x^1}, \ldots, \widetilde{x^n})$ für $\mathrm{QP}^2$(G,c)
2. Wähle zufällig einen zweidimensionalen Vektor $r$ mit Norm 1
3. $S \leftarrow \{i \in V : \widetilde{x^i} \cdot r \geq 0\}$

# Random MaxCut

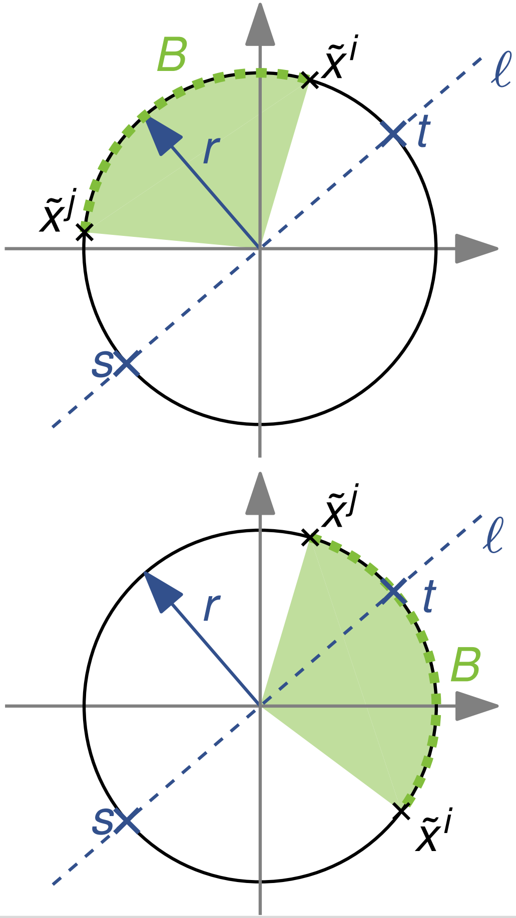
**Satz 8.18:** Sei $I$ eine Instanz für MAXCUT und $C_{\mathrm{RMC}}(I)$ der Wert der Lösung, die RANDOM MAXCUT für $I$ berechnet. Wenn die Vektoren $r$ gleichverteilt angenommen werden, so gilt:

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \arccos(\tilde{x}^i \cdot \tilde{x}^j)$$

**Zeige:**

$$\Pr[\operatorname{sgn}(\tilde{x}^i \cdot r) \neq \operatorname{sgn}(\tilde{x}^j \cdot r)] = \frac{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}{\pi}$$

$\operatorname{sgn}(\tilde{x}^i \cdot r) \neq \operatorname{sgn}(\tilde{x}^j \cdot r) \iff \ell$ trennt $\tilde{x}^i$ und $\tilde{x}^j$




$\ell$=Senkrechte Gerade zu $r$
$s$, $t$= Schnittpunkte von $\ell$ und Einheitskreis.

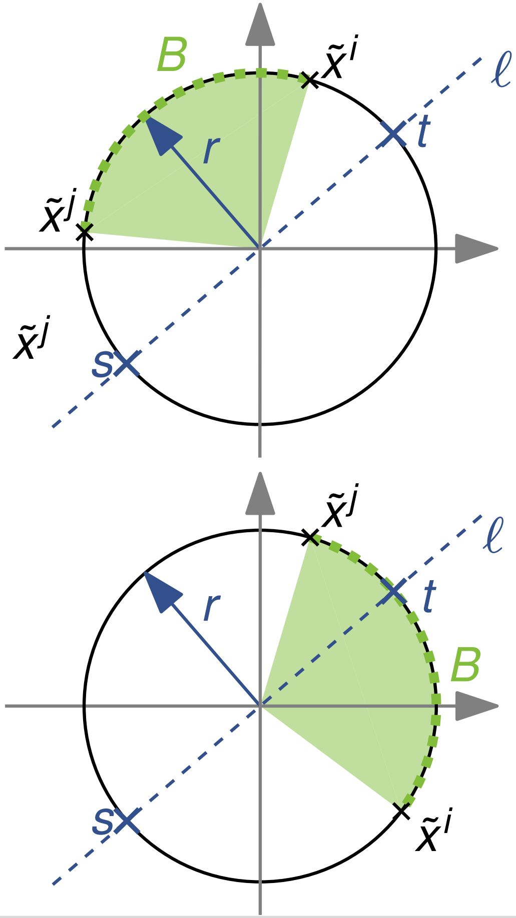
# Random MaxCut

**Satz 8.18:** Sei $I$ eine Instanz für MaxCut und $C_{\text{RMC}}(I)$ der Wert der Lösung, die Random MaxCut für $I$ berechnet. Wenn die Vektoren $r$ gleichverteilt angenommen werden, so gilt:

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \arccos(\tilde{x}^i \cdot \tilde{x}^j)$$

**Zeige:**

$$\Pr[\operatorname{sgn}(\tilde{x}^i \cdot r) \neq \operatorname{sgn}(\tilde{x}^j \cdot r)] = \frac{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}{\pi}$$




$\operatorname{sgn}(\tilde{x}^i \cdot r) \neq \operatorname{sgn}(\tilde{x}^j \cdot r) \Leftrightarrow \ell$ trennt $\tilde{x}^i$ und $\tilde{x}^j$

$\Leftrightarrow$ $s$ oder $t$ liegt auf dem kürzeren Kreisbogen $B$ zwischen $\tilde{x}^i$ und $\tilde{x}^j$

$B$ hat Länge $\arccos(\tilde{x}^i \cdot \tilde{x}^j)$

$\ell$=Senkrechte Gerade zu $r$
$s$, $t$= Schnittpunkte von $\ell$ und Einheitskreis.

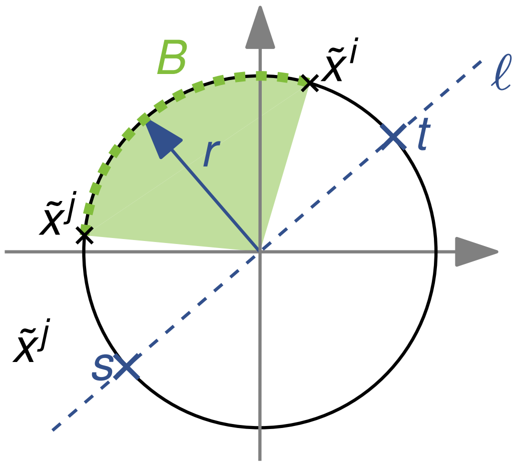
# Random MaxCut

**Satz 8.18:** Sei $I$ eine Instanz für MaxCut und $C_{\text{RMC}}(I)$ der Wert der Lösung, die Random MaxCut für $I$ berechnet. Wenn die Vektoren $r$ gleichverteilt angenommen werden, so gilt:

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \arccos(\tilde{x}^i \cdot \tilde{x}^j)$$

**Zeige:**

$$\Pr[\operatorname{sgn}(\tilde{x}^i \cdot r) \neq \operatorname{sgn}(\tilde{x}^j \cdot r)] = \frac{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}{\pi}$$

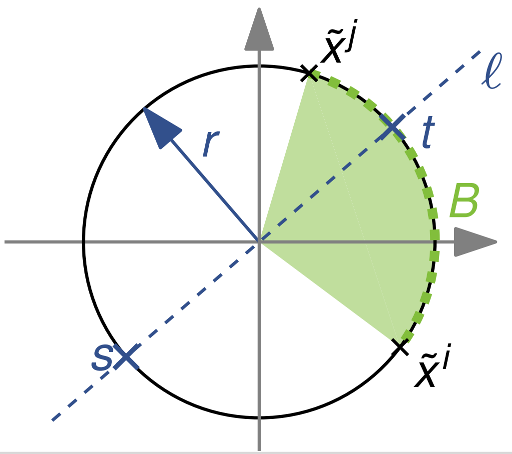



$\operatorname{sgn}(\tilde{x}^i \cdot r) \neq \operatorname{sgn}(\tilde{x}^j \cdot r) \Leftrightarrow \ell$ trennt $\tilde{x}^i$ und $\tilde{x}^j$

$\Leftrightarrow$ $s$ oder $t$ liegt auf dem kürzeren Kreisbogen $B$ zwischen $\tilde{x}^i$ und $\tilde{x}^j$

$B$ hat Länge $\arccos(\tilde{x}^i \cdot \tilde{x}^j)$

$$\Pr[s \text{ oder } t \text{ liegt auf } B] = \frac{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}{2 \cdot \pi} + \frac{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}{2 \cdot \pi} = \frac{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}{\pi}$$




$\ell$=Senkrechte Gerade zu $r$

$s$, $t$= Schnittpunkte von $\ell$ und Einheitskreis.

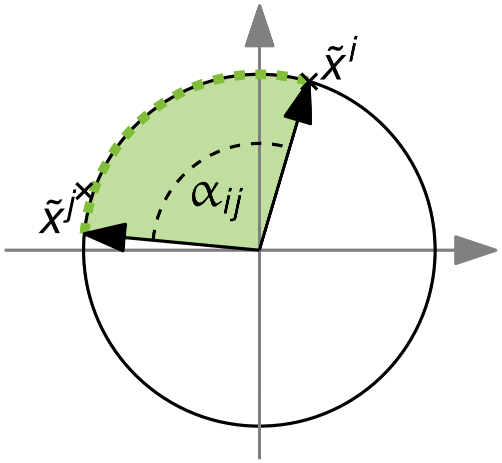


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Random MaxCut – Qualität

**Satz 8.19:** Für eine Instanz $I$ von MaxCut berechnet Random MaxCut eine Lösung mit dem Wert $C_{\text{RMC}}(I)$, für die gilt

$$\frac{E[C_{\text{RMC}}(I)]}{OPT(I)} \geq 0,8785 \ .$$

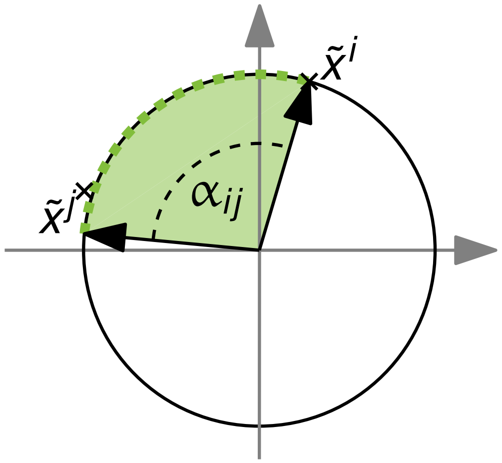


# Random MaxCut – Qualität

**Satz 8.19:** Für eine Instanz $I$ von MaxCut berechnet Random MaxCut eine Lösung mit dem Wert $C_{\mathrm{RMC}}(I)$, für die gilt

$$\frac{E[C_{\mathrm{RMC}}(I)]}{OPT(I)} \geq 0{,}8785 \ .$$

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \underbrace{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}_{\alpha_{ij}} = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \frac{\alpha_{ij}}{\pi}$$

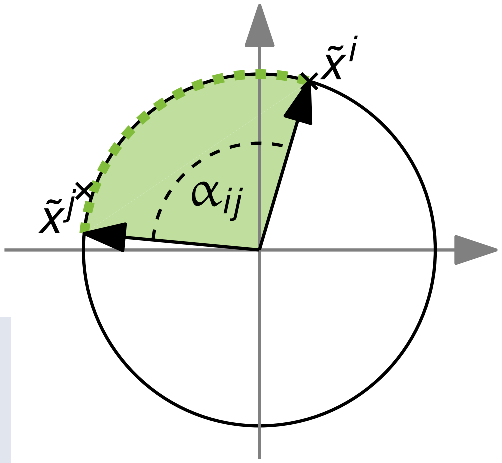


# Random MaxCut – Qualität

**Satz 8.19:** Für eine Instanz $I$ von MaxCut berechnet Random MaxCut eine Lösung mit dem Wert $C_{\text{RMC}}(I)$, für die gilt

$$\frac{E[C_{\text{RMC}}(I)]}{OPT(I)} \geq 0,8785 \ .$$

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \underbrace{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}_{\alpha_{ij}} = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \frac{\alpha_{ij}}{\pi}$$




Sei $\tilde{x}^1, \ldots, \tilde{x}^n$ eine optimale Lösung von $QP^2(I)$ mit dem Wert

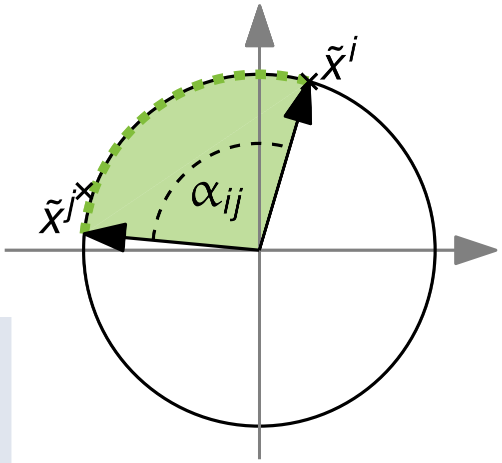
$$C(QP^2(I)) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij}(1 - \tilde{x}^i \cdot \tilde{x}^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \frac{1 - \cos \alpha_{ij}}{2}$$

# Random MaxCut – Qualität

**Satz 8.19:** Für eine Instanz $I$ von MAXCUT berechnet RANDOM MAXCUT eine Lösung mit dem Wert $C_{\text{RMC}}(I)$, für die gilt

$$\frac{E[C_{\text{RMC}}(I)]}{OPT(I)} \geq 0,8785 \ .$$

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \underbrace{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}_{\alpha_{ij}} = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \frac{\alpha_{ij}}{\pi}$$




Sei $\tilde{x}^1, \ldots, \tilde{x}^n$ eine optimale Lösung von $QP^2(I)$ mit dem Wert

$$C(QP^2(I)) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij}(1 - \tilde{x}^i \cdot \tilde{x}^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \frac{1 - \cos \alpha_{ij}}{2}$$

Da $QP^2(I)$ Relaxierung von $IQP(I)$ ist, gilt $OPT(I) \leq C(QP^2(I))$

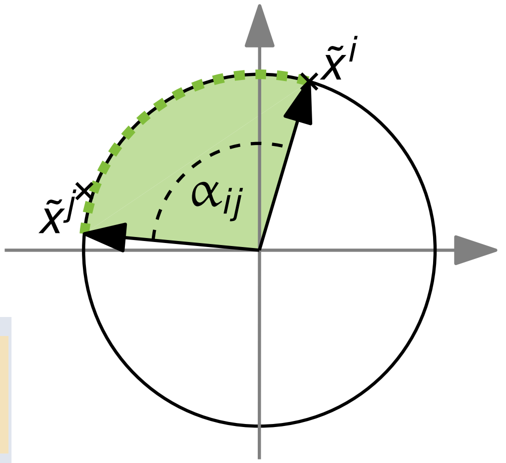
$$\frac{E[C_{RMC}(I)]}{OPT(I)} \geq \frac{E[C_{RMC}(I)]}{C(QP^2(I))}$$

# Random MaxCut – Qualität

**Satz 8.19:** Für eine Instanz $I$ von MaxCut berechnet Random MaxCut eine Lösung mit dem Wert $C_{\mathrm{RMC}}(I)$, für die gilt

$$\frac{E[C_{\mathrm{RMC}}(I)]}{OPT(I)} \geq 0,8785 \ .$$

$$E[C_{RMC}(I)] = \frac{1}{\pi} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \underbrace{\arccos(\tilde{x}^i \cdot \tilde{x}^j)}_{\alpha_{ij}} = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot \frac{\alpha_{ij}}{\pi}$$




Sei $\tilde{x}^1, \ldots, \tilde{x}^n$ eine optimale Lösung von $QP^2(I)$ mit dem Wert

$$C(QP^2(I)) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij}(1 - \tilde{x}^i \cdot \tilde{x}^j) = \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \frac{1 - \cos \alpha_{ij}}{2}$$

Da $QP^2(I)$ Relaxierung von $IQP(I)$ ist, gilt $OPT(I) \leq C(QP^2(I))$

$$\frac{E[C_{RMC}(I)]}{OPT(I)} \geq \frac{E[C_{RMC}(I)]}{C(QP^2(I))}$$

**Zeige:** $\frac{\alpha_{ij}}{\pi} \geq \frac{1 - \cos \alpha ij}{2} \cdot 0.8785$


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Random MaxCut – Qualität

**Satz 8.19:** Für eine Instanz $I$ von MAXCUT berechnet RANDOM MAXCUT eine Lösung mit dem Wert $C_{\text{RMC}}(I)$, für die gilt

$$\frac{E[C_{\text{RMC}}(I)]}{OPT(I)} \geq 0,8785 \quad .$$

$E[C_{RMC}(I)$

Sei $\tilde{x}^1, \ldots$

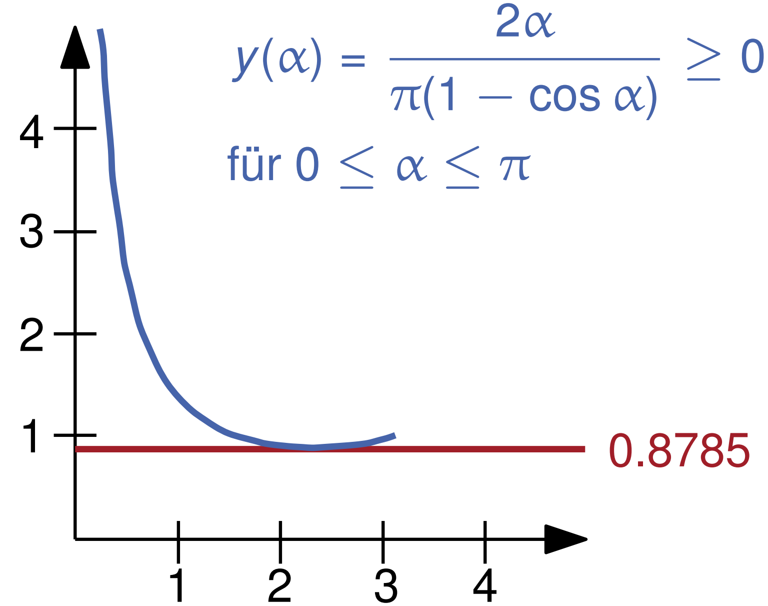
$C(C$

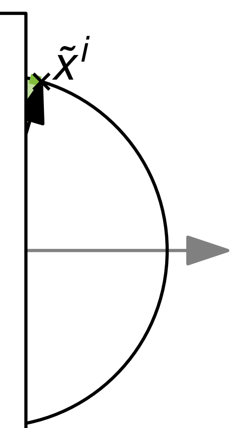
Da $QP^2(I)$

**Man kann zeigen:**

$$y(\alpha) = \frac{2\alpha}{\pi(1-\cos\alpha)} \geq 0.8785$$

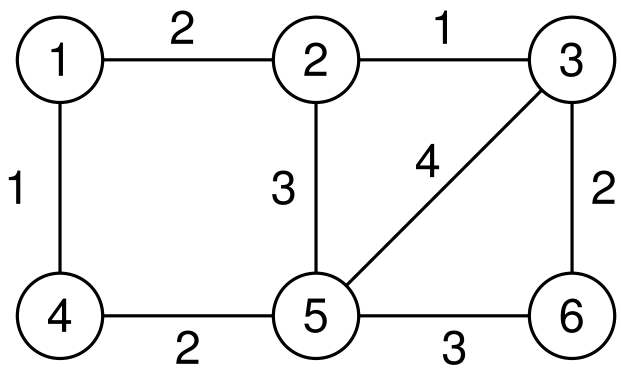
für $0 \leq \alpha \leq \pi$




**Zeige:** $\frac{\alpha_{ij}}{\pi} \geq \frac{1-\cos\alpha ij}{2} \cdot 0.8785$

# Beispiel



# Beispiel

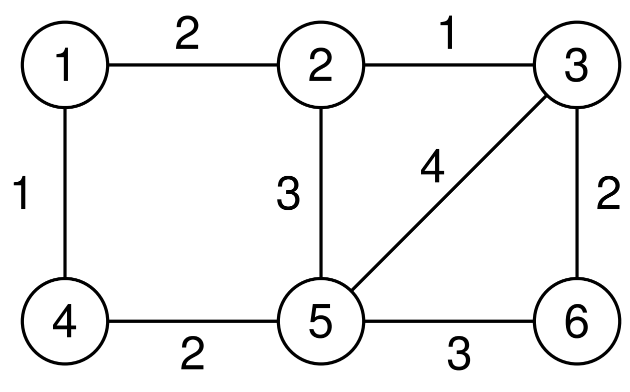
**1. Schritt:** Bestimme QP

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter NB: $x_i^2 = 1$ für $1 \leq i \leq n$ .

**Gewichtungsmatrix:** $c_{ij}$

| | 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 | |






Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Beispiel



**1. Schritt:** Bestimme QP

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter NB: $x_i^2 = 1$ für $1 \leq i \leq n$ .

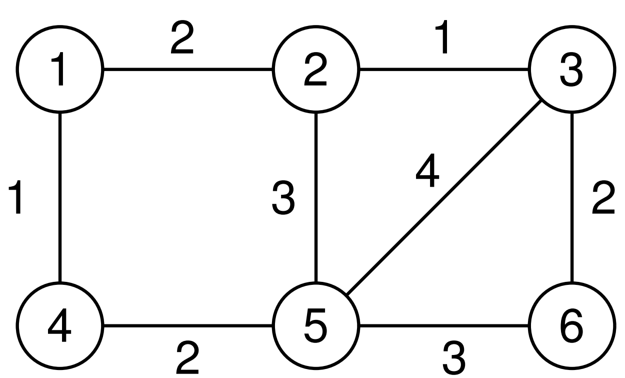
**Gewichtungsmatrix:** $c_{ij}$

| | 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. Schritt:** Relaxiere QP $\rightarrow QP^2$

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x^i \cdot x^j)$$

unter NB: $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren



# Beispiel

Karlsruher Institut für Technologie

**1. Schritt:** Bestimme QP

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter NB: $x_i^2 = 1$ für $1 \leq i \leq n$ .

**Gewichtungsmatrix:** $c_{ij}$

| | 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 | |

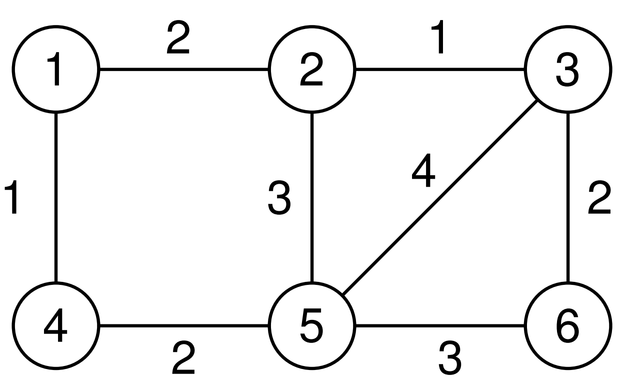
**2. Schritt:** Relaxiere QP $\rightarrow QP^2$

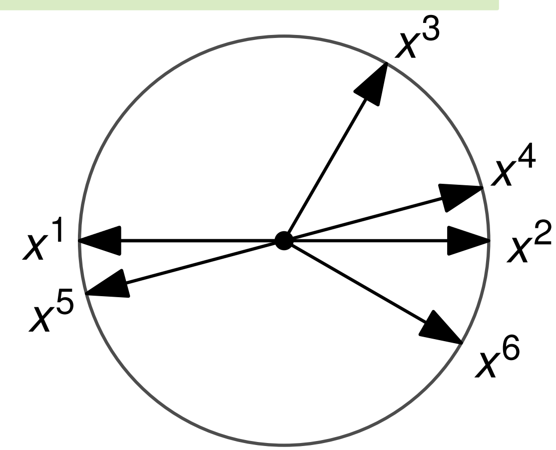
$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x^i \cdot x^j)$$

unter NB: $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren

**3. Schritt:** Löse $QP^2$

| Variable | $x^1$ | $x^2$ | $x^3$ | $x^4$ | $x^5$ | $x^6$ |
|---|---|---|---|---|---|---|
| Winkel | 0 | 180 | 120 | 165 | 345 | 210 |






Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Beispiel

**1. Schritt:** Bestimme QP

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter NB: $x_i^2 = 1$ für $1 \leq i \leq n$ .

**Gewichtungsmatrix:** $c_{ij}$

| | 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 | |

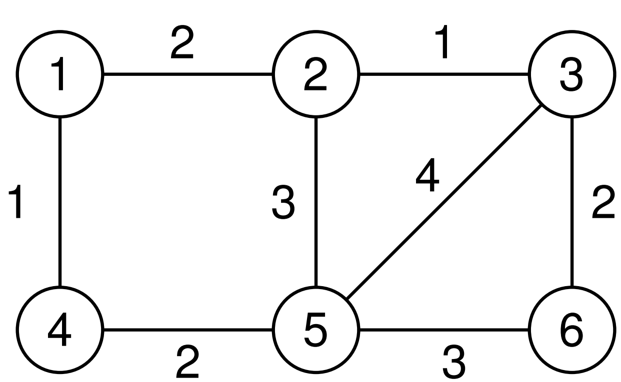
**2. Schritt:** Relaxiere QP $\rightarrow QP^2$

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x^i \cdot x^j)$$

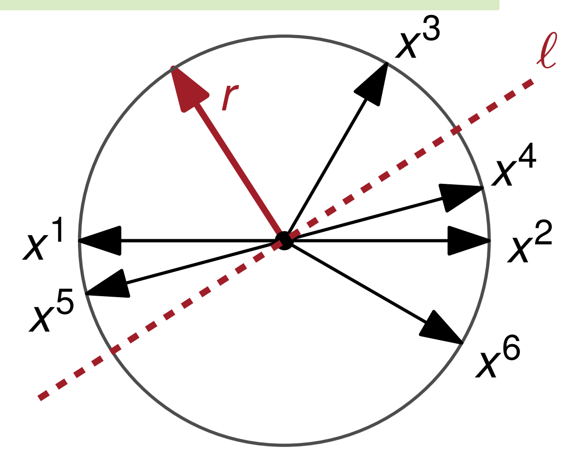
unter NB: $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren

**3. Schritt:** Löse $QP^2$

| Variable | $x^1$ | $x^2$ | $x^3$ | $x^4$ | $x^5$ | $x^6$ |
|---|---|---|---|---|---|---|
| Winkel | 0 | 180 | 120 | 165 | 345 | 210 |




**4. Schritt:** Rate Vektor $r$.



# Beispiel

**1. Schritt:** Bestimme QP

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter NB: $x_i^2 = 1$ für $1 \leq i \leq n$ .

**Gewichtungsmatrix:** $c_{ij}$

| | 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 | |

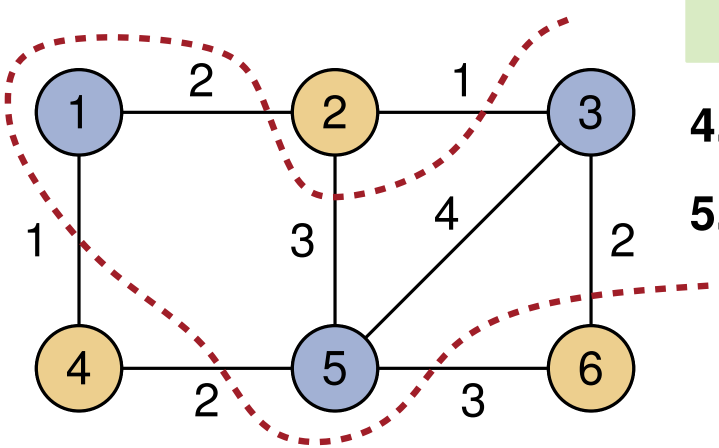
**2. Schritt:** Relaxiere QP $\to QP^2$

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x^i \cdot x^j)$$

unter NB: $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren

**3. Schritt:** Löse $QP^2$

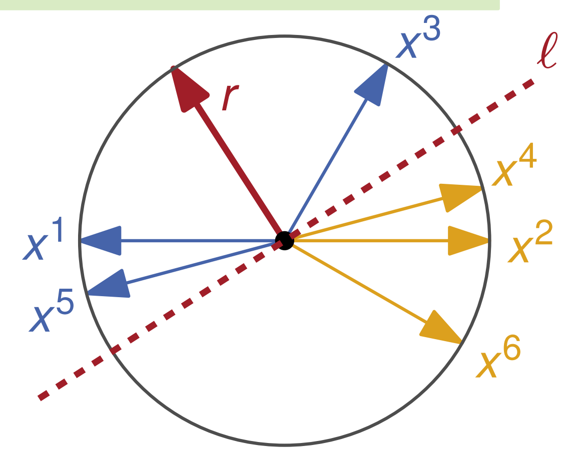
| Variable | $x^1$ | $x^2$ | $x^3$ | $x^4$ | $x^5$ | $x^6$ |
|---|---|---|---|---|---|---|
| Winkel | 0 | 180 | 120 | 165 | 345 | 210 |




**4. Schritt:** Rate Vektor $r$.

**5. Schritt:** Bestimme Schnitt

Gewicht 14






Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

# Beispiel

**1. Schritt:** Bestimme QP

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x_i \cdot x_j)$$

unter NB: $x_i^2 = 1$ für $1 \leq i \leq n$ .

**Gewichtungsmatrix:** $c_{ij}$

| | 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 | |

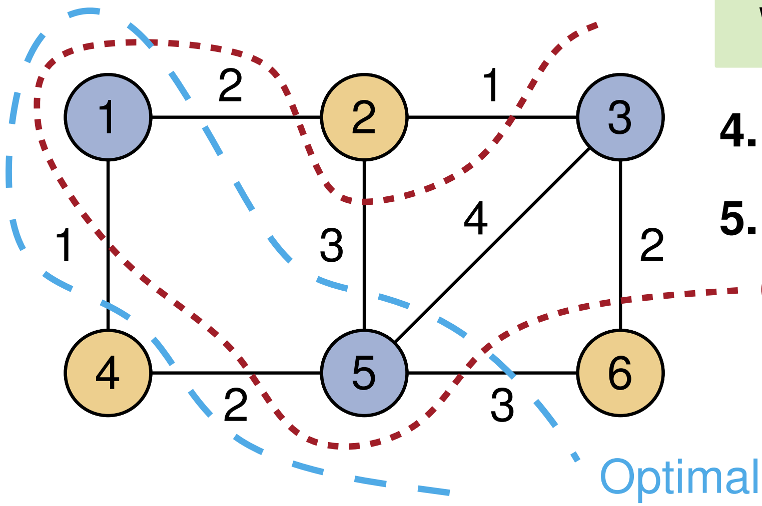
**2. Schritt:** Relaxiere QP $\rightarrow QP^2$

$$\max \frac{1}{2} \cdot \sum_{j=1}^{6} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - x^i \cdot x^j)$$

unter NB: $x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren

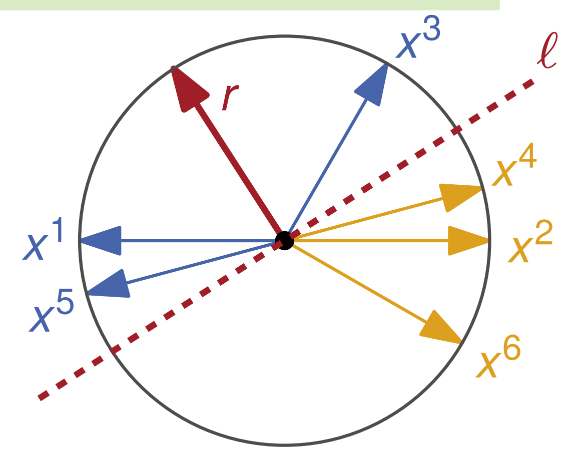
**3. Schritt:** Löse $QP^2$

| Variable | $x^1$ | $x^2$ | $x^3$ | $x^4$ | $x^5$ | $x^6$ |
|---|---|---|---|---|---|---|
| Winkel | 0 | 180 | 120 | 165 | 345 | 210 |




**4. Schritt:** Rate Vektor $r$.

**5. Schritt:** Bestimme Schnitt



# Problemdefinition

Karlsruher Institut für Technologie

**Problem MAXCUT:** Gegeben ist ein Graph $G = (V, E)$ mit Gewichtsfunktion $c \colon E \longrightarrow \mathbb{N}$.
Gesucht ist ein Schnitt $(S, V \setminus S)$ von $G$ mit maximalem Gewicht, d.h.

$$c(S, V \setminus S) := \sum_{\substack{u,v \,\in\, E \\ u \in S \text{ und } v \in V \setminus S}} c(\{u, v\}) \qquad \text{soll maximal sein.}$$

Problem ist $\mathcal{NP}$-schwer.

**Idee:** Berechne … ung.

$G = (V, E)$ Tran… ionen → quadratisches Programm $QP^k$

**Bisher:** k=2

**Problem:**
Nicht bekannt, ob $QP^2$ optimal in poly. Zeit gelöst werden kann.

Entsprechendes $QP^n$ kann in poly. Zeit gelöst werden.

Lösen von $QP^k$ → Reellwertige Lösung von $QP^k$

Runden der Lösung randomisierter Algo. → Ganzzahlige 1-dim. Lösung

Rücktransformation in Schnitt → Näherungslösung für MaxCut auf $G$

# Effiziente Lösung

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

unter den Nebenbedingungen

$x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren.

wird zu

$QP^n$

$$\max \frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - m_{ij})$$

mit $m_{ij} = x^i \cdot x^j$ und $m_{ii} = 1$ und unter den Nebenbedingungen

$x^i, x^j \in \mathbb{R}^n$ sind normierte Vektoren.

# Effiziente Lösung

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

unter den Nebenbedingungen

$x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren.

wird zu

$QP^n$

$$\max \frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - m_{ij})$$

mit $m_{ij} = x^i \cdot x^j$ und $m_{ii} = 1$ und unter den Nebenbedingungen

$x^i, x^j \in \mathbb{R}^n$ sind normierte Vektoren.

**Definition 8.20:** Eine $n \times n$-Matrix $M$ heißt *positiv semidefinit*, falls für jeden Vektor $x \in \mathbb{R}^n$ gilt:

$$x^T \cdot M \cdot x \geq 0$$

**Die Matrix $M := (m_{ij})$ ist positiv semidefinit.**

**Denn:** Symmetrische Matrix $M$ ist genau dann positiv semidefinit, wenn es $m \times n$-Matrix $P$ $(m \leq n)$ gibt, sodass

$$M = P^T \cdot P$$

($P$ ist in polynomieller Zeit berechenbar, falls $M$ positiv semidefinit.)

(Für jede positiv semidefinite $n \times n$-Matrix $M$ mit $m_{ii} = 1$ gilt, dass $n$ normierte Vektoren $x^1, \ldots, x^n \in \mathbb{R}^n$ mit $m_{ij} = x^i \cdot x^j$ in polynomieller Zeit berechnet werden können.)

# Effiziente Lösung

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

unter den Nebenbedingungen

$x^i, x^j \in \mathbb{R}^2$ sind normierte Vektoren.

wird zu

$QP^n$

$$\max \frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{j-1} c_{ij} \cdot (1 - m_{ij})$$

mit $m_{ij} = x^i \cdot x^j$ und $m_{ii} = 1$ und unter den Nebenbedingungen

$x^i, x^j \in \mathbb{R}^n$ sind normierte Vektoren.

**Definition 8.20:** Eine $n \times n$-Matrix $M$ heißt *positiv semidefinit*, falls für jeden Vektor $x \in \mathbb{R}^n$ gilt:

$$x^T \cdot M \cdot x \geq 0$$

**Die Matrix $M := (m_{ij})$ ist positiv semidefinit.**

Damit entspricht $QP^n(I)$ dem Problem SEMI-DEFINIT-CUT(I).

**Satz:** Es gibt Algorithmus $\mathcal{A}_\epsilon$ für jedes $\epsilon$, sodass $\mathcal{A}_\epsilon$ polynomiell in Eingabegröße und $\log(\frac{1}{\epsilon})$ ist und

$$\mathcal{A}_\epsilon(I) \geq OPT_{SD}(I) - \epsilon,$$

wobei $OPT_{SD}(I)$ optimaler Lösungswert von SEMI-DEFINIT-CUT(I).

**(Ohne Beweis)**

**Man kann zeigen:** Für $\epsilon = 10^{-5}$ wird Approximationsgüte von RANDOM MAXCUT erreicht.