Lecture: Approximation Algorithms

Jannik Matuschke

TUM

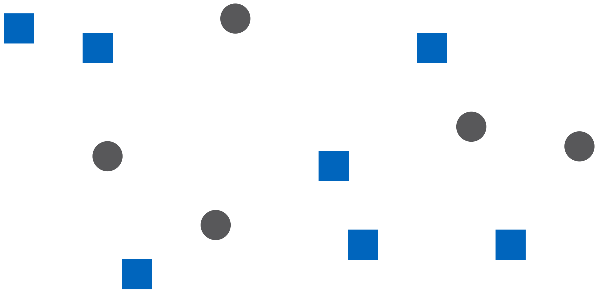
December 5, 2018

Iterated Rounding for
Steiner Tree

Steiner Tree

Input: graph $G = (V, E)$, terminals $R \subseteq V$,
distances $d : E \to \mathbb{R}_+$

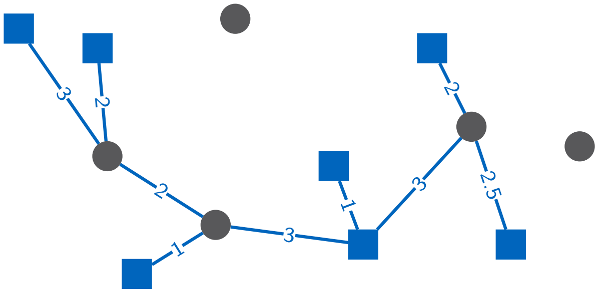
Task: find a tree T spanning R minimizing $\sum_{e \in T} d(e)$



Steiner Tree

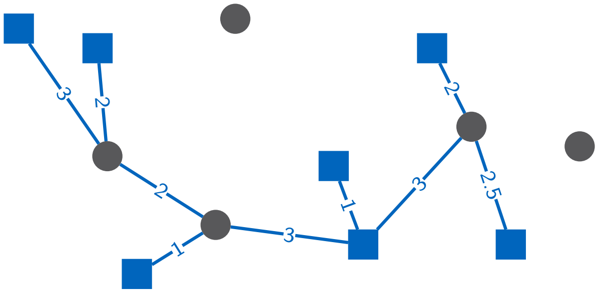
Input: graph $G = (V, E)$, terminals $R \subseteq V$, distances $d : E \to \mathbb{R}_+$

Task: find a tree T spanning R minimizing $\sum_{e \in T} d(e)$



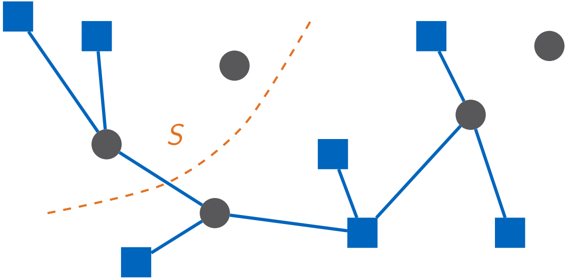
Input: graph $G = (V, E)$, terminals $R \subseteq V$,
distances* $d : E \to \mathbb{R}_+$

Task: find a tree T spanning R minimizing $\sum_{e \in T} d(e)$



*w.l.o.g.: G is complete and d is metric

Undirected cut relaxation



variables:

$x(e) = 1 \;\Leftrightarrow\; e \in T$

$$Z^*_{\mathrm{UC}} := \min \sum_{e \in E} d(e)x(e)$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x(e) \geq 1 \quad \forall\, S \subseteq V, R \cap S \neq \emptyset, R \setminus S \neq \emptyset$$

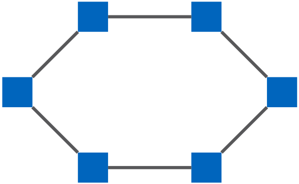
$$x(e) \geq 0 \quad \forall\, e \in E$$

Integrality gap

$$
\begin{aligned}
Z^*_{\mathrm{UC}} := \min \ & \sum_{e \in E} d(e)x(e) \\
\text{s.t.} \quad & \sum_{e \in \delta(S)} x(e) \geq 1 \quad && \forall\, S \subseteq V, R \cap S \neq \emptyset, R \setminus S \neq \emptyset \\
& x(e) \geq 0 && \forall\, e \in E
\end{aligned}
$$

How large can $\mathrm{OPT}/Z^*_{\mathrm{UC}}$ be?

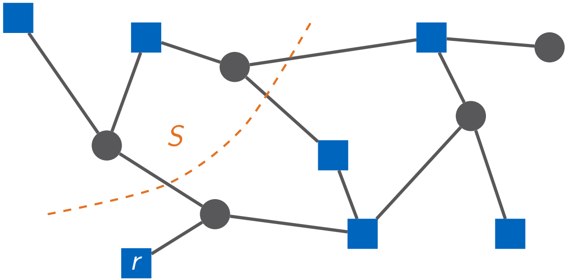
- not larger than 2 (primal-dual for Steiner Forest)
- can get arbitrarily close to 2, even when $R = V$



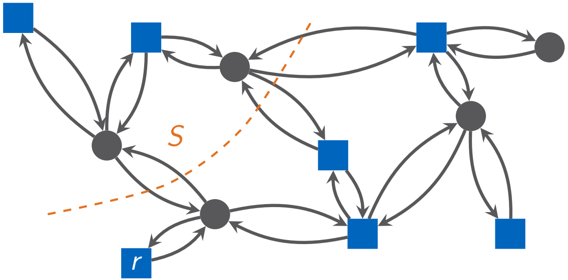
$\mathrm{OPT} = n - 1$

$Z^*_{\mathrm{UC}} = n/2$

Bidirected cut relaxation



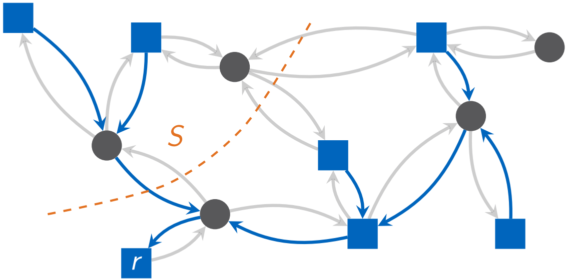
Bidirected cut relaxation



directed graph
$D = (V, A)$

$\delta^+(S) := \{(v, w) \in A : v \in S, w \in V \setminus S\}$

Bidirected cut relaxation



directed graph
$D = (V, A)$

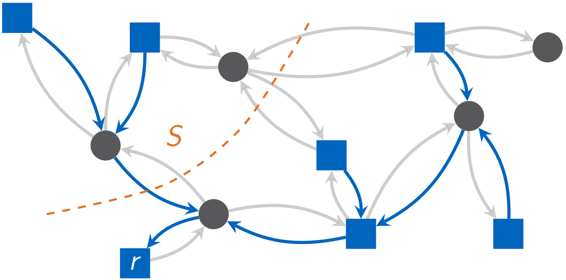
$\delta^+(S) := \{(v, w) \in A : v \in S, w \in V \setminus S\}$

$$Z^*_{\mathrm{BC}} := \min \sum_{a \in A} d(a) x(a)$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(S)} x(a) \geq 1 \quad \forall\, S \subseteq V \setminus \{r\},\ R \cap S \neq \emptyset$$

$$x(a) \geq 0 \quad \forall\, a \in A$$

Bidirected cut relaxation



directed graph
$D = (V, A)$

$\delta^+(S) := \{(v, w) \in A : v \in S, w \in V \setminus S\}$

$$Z^*_{\text{BC}} := \min \sum_{a \in A} d(a)x(a)$$

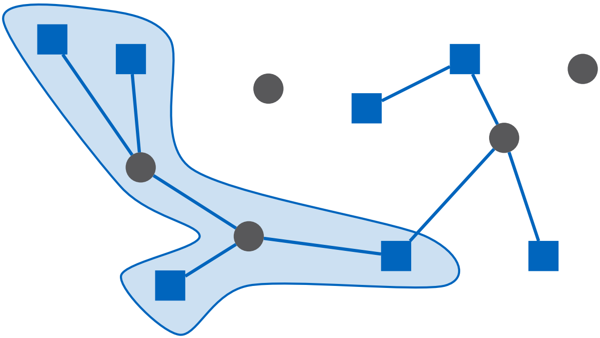
$$\text{s.t.} \quad \sum_{a \in \delta^+(S)} x(a) \geq 1 \quad \forall\, S \subseteq V \setminus \{r\},\ R \cap S \neq \emptyset$$

$$x(a) \geq 0 \quad \forall\, a \in A$$

If $R = V$, then $Z^*_{BC} = \text{OPT}$.

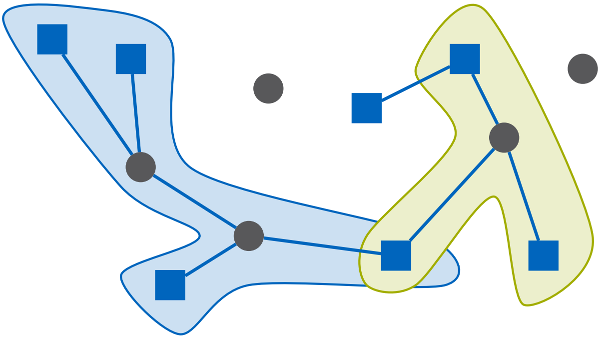
Full components

A full component is a tree in which all non-leaves are Steiner nodes and all leaves are terminals.



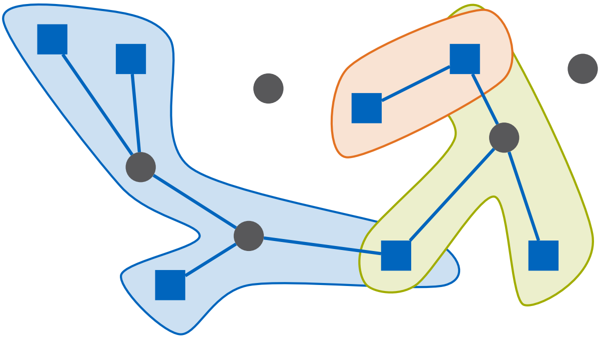
Full components

A full component is a tree in which all non-leaves are Steiner nodes and all leaves are terminals.



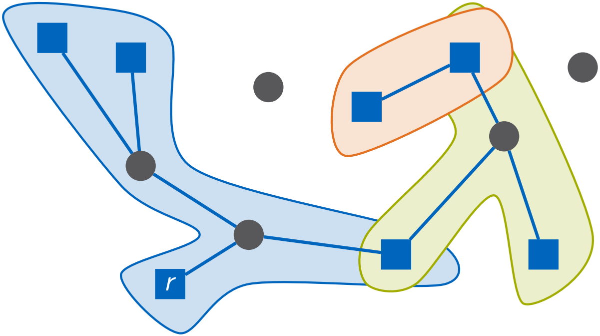
Full components

A full component is a tree in which all non-leaves are Steiner nodes and all leaves are terminals.



Full components

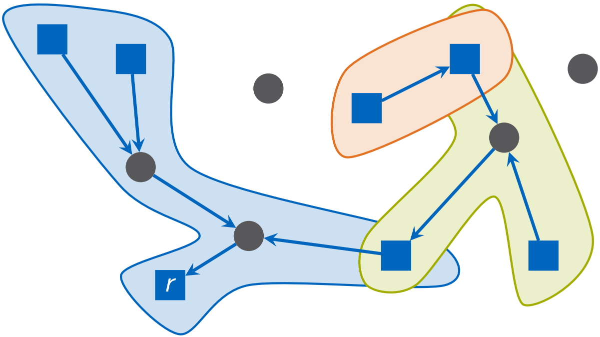
A full component is a tree in which all non-leaves are Steiner nodes and all leaves are terminals.



Fix root $r \in R$.

Full components

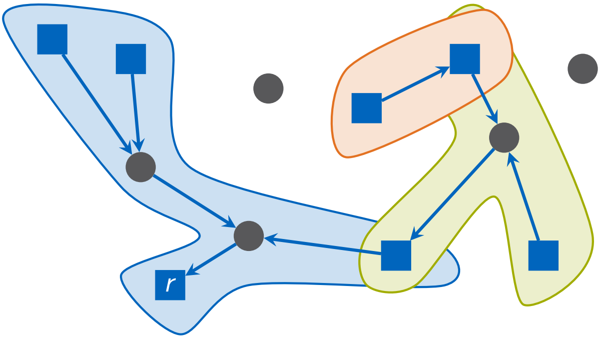
A directed full component is an in-tree in which all non-leaves are Steiner nodes and all leaves are terminals.



Fix root $r \in R$. Direct all edges towards r.

Full components

A directed full component is an in-tree in which all non-leaves are Steiner nodes and all leaves are terminals.

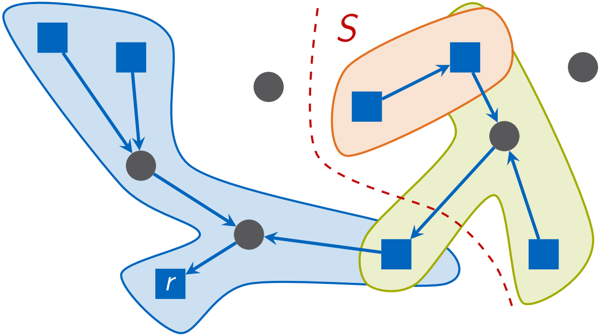


Fix root $r \in R$. Direct all edges towards r.

directed full component C: tree (V_C, E_C) with root r_C

$d_C := \sum_{e \in E_C} d_e$

Directed component relaxation

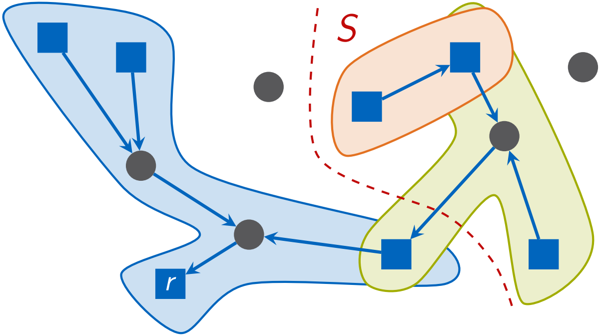


$$\mathcal{C} := \{C \,:\, C \text{ is dir. full comp. of } G\}$$

$$\Delta(S) := \{C \in \mathcal{C} \,:\, r_C \notin S,\, R_C \cap S \neq \emptyset\}$$

$$\begin{aligned} \min \quad & \sum_{C \in \mathcal{C}} d_C x_C \\ \text{s.t.} \quad & \sum_{C \in \Delta(S)} x_C \geq 1 && \forall\, S \subseteq V \setminus \{r\}, S \cap R \neq \emptyset \\ & x_C \in \{0,1\} && \forall\, C \in \mathcal{C} \end{aligned}$$

Directed component relaxation



$\mathcal{C} := \{C \,:\, C \text{ is dir. full comp. of } G\}$

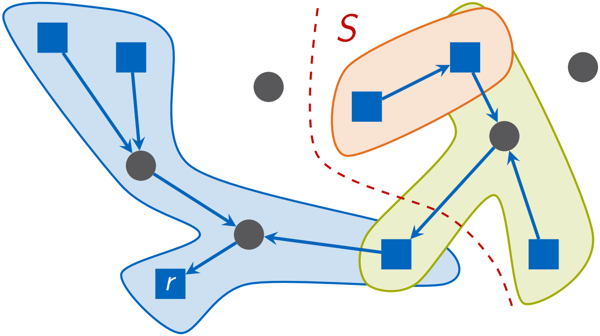
$\Delta(S) := \{C \in \mathcal{C} \,:\, r_C \notin S,\ R_C \cap S \neq \emptyset\}$

$$\min \sum_{C \in \mathcal{C}} d_C x_C$$

$$\text{s.t.} \quad \sum_{C \in \Delta(S)} x_C \geq 1 \qquad \forall\, S \subseteq V \setminus \{r\}, S \cap R \neq \emptyset$$

$$x_C \in \{0, 1\} \qquad \forall\, C \in \mathcal{C}$$

Directed component relaxation



$$\mathcal{C} := \{C : C \text{ is dir. full comp. of } G\}$$

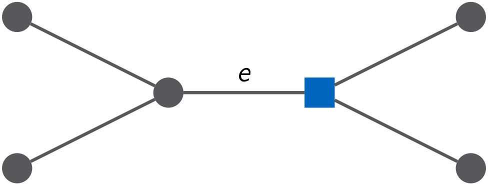
$$\Delta(S) := \{C \in \mathcal{C} : r_C \notin S,\ R_C \cap S \neq \emptyset\}$$

$$Z^*_{\mathrm{DC}} := \min \sum_{C \in \mathcal{C}} d_C x_C$$

$$\text{s.t.} \quad \sum_{C \in \Delta(S)} x_C \geq 1 \qquad \forall\, S \subseteq V \setminus \{r\}, S \cap R \neq \emptyset$$

$$x_C \geq 0 \qquad \forall\, C \in \mathcal{C}$$

Contracting edges

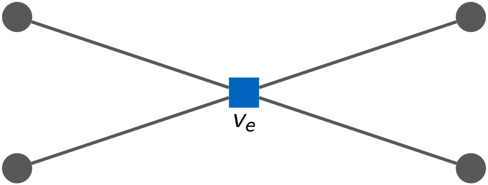


contract edge $e = \{v, w\}$:

- merge v and w into a single node v_e
- $d_{uv_e} = \min\{d_{uv}, d_{uw}\} \quad \forall\, u \in V$
- if v or w was a terminal, v_e is a terminal

Notation Let G/F denote the graph resulting from contracting all edges in F (order does not matter).

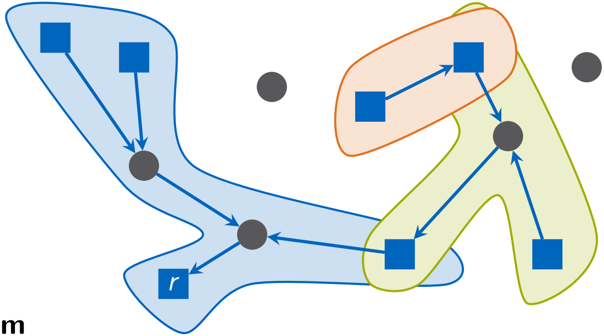
Contracting edges



contract edge $e = \{v, w\}$:

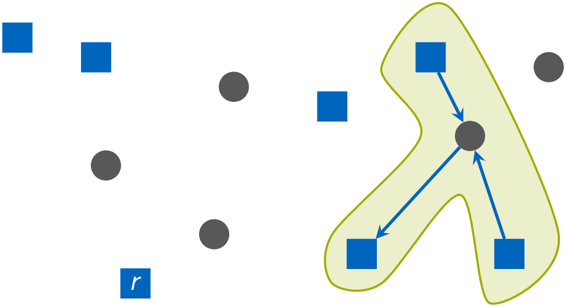
- merge v and w into a single node v_e
- $d_{uv_e} = \min\{d_{uv}, d_{uw}\} \quad \forall\, u \in V$
- if v or w was a terminal, v_e is a terminal

Notation Let G/F denote the graph resulting from contracting all edges in F (order does not matter).



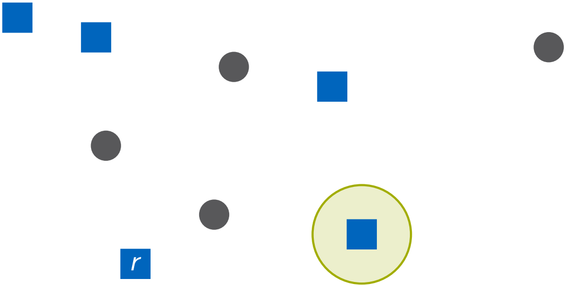
Algorithm

1. $F := \emptyset$
2. for $i := 1$ to ℓ do
   - Compute optimal solution x to the LP for G/F.
   - Select $C \in \mathcal{C}$ at random with probabilities $x_C / \sum_{C' \in \mathcal{C}} x_{C'}$.
   - $F := F \cup E_C$.
3. Let T' be a minimum spanning tree on the terminals in G/F.
4. Return $T' \cup F$.



Algorithm

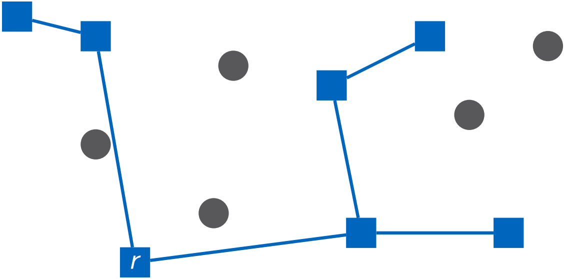
1. $F := \emptyset$
2. for $i := 1$ to ℓ do
    - Compute optimal solution x to the LP for G/F.
    - Select $C \in \mathcal{C}$ at random with probabilities $x_C / \sum_{C' \in \mathcal{C}} x_{C'}$.
    - $F := F \cup E_C$.
3. Let T' be a minimum spanning tree on the terminals in G/F.
4. Return $T' \cup F$.



Algorithm

1. $F := \emptyset$
2. for $i := 1$ to ℓ do
   - Compute optimal solution x to the LP for G/F.
   - Select $C \in \mathcal{C}$ at random with probabilities $x_C / \sum_{C' \in \mathcal{C}} x_{C'}$.
   - $F := F \cup E_C$.
3. Let T' be a minimum spanning tree on the terminals in G/F.
4. Return $T' \cup F$.

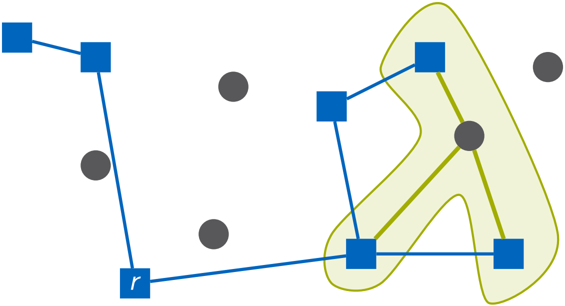
Terminal spanning trees



$$\text{mst}(F) := \text{min cost of a spanning tree on the terminals in } G/F$$

$$\text{drop}_F(C) := \text{mst}(F) - \text{mst}(F \cup E_C)$$

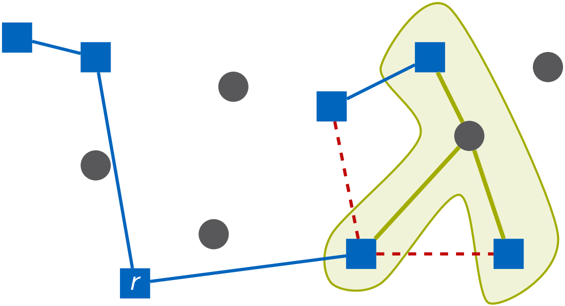
Terminal spanning trees



$$\text{mst}(F) := \text{min cost of a spanning tree on the terminals in } G/F$$

$$\text{drop}_F(C) := \text{mst}(F) - \text{mst}(F \cup E_C)$$

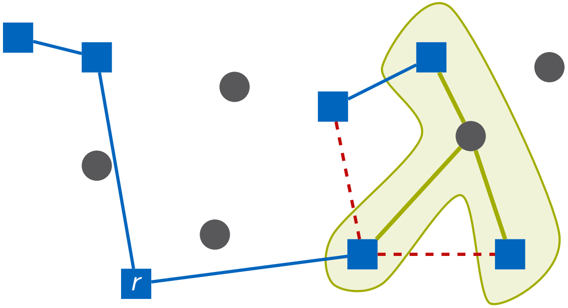
Terminal spanning trees



$$\text{mst}(F) := \text{min cost of a spanning tree on the terminals in } G/F$$

$$\text{drop}_F(C) := \text{mst}(F) - \text{mst}(F \cup E_C)$$

Terminal spanning trees



$$\mathrm{mst}(F) := \text{min cost of a spanning tree on the terminals in } G/F$$

$$\mathrm{drop}_F(C) := \mathrm{mst}(F) - \mathrm{mst}(F \cup E_C)$$

Drop Lemma

$$\mathrm{mst}(F) \leq \sum_{C \in \mathcal{C}} \mathrm{drop}_F(C) x_C$$

Analysis

Algorithm

1. $F := \emptyset$
2. for $i := 1$ to ℓ do
   - Compute optimal solution x to the LP for G/F.
   - Select $C \in \mathcal{C}$ at random with probabilities $x_C / \sum_{C' \in \mathcal{C}} x_{C'}$.
   - $F := F \cup E_C$.
3. Let T' be a minimum spanning tree on the terminals in G/F.
4. Return $T' \cup F$.

Assumptions

- LP relaxation can be solved efficiently.

  $\rightarrow$ restrict to small full components
- In every iteration $\sum_{C \in \mathcal{C}} x_C = \Sigma$.

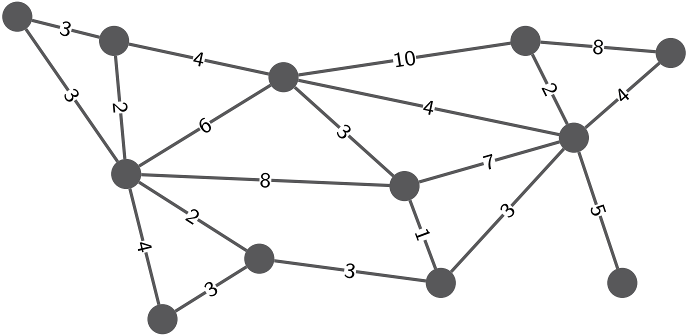
  $\rightarrow$ introduce a dummy component

Rounding an SDP by choosing a random hyperplane: The Maximum Cut Problem

Max Cut

Input: graph $G = (V, E)$, weights $w : E \to \mathbb{R}_+$

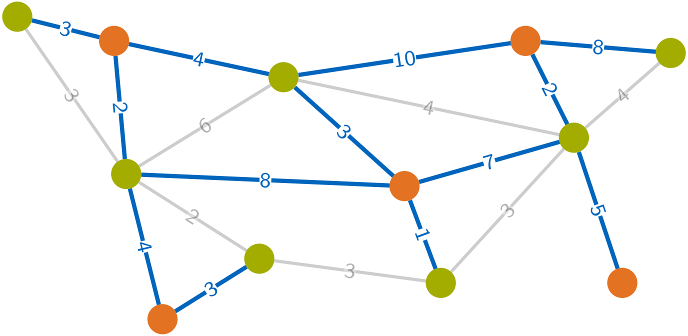
Task: find $S \subseteq V$ maximizing $\sum_{e \in \delta(S)} w(e)$



Max Cut

Input: graph $G = (V, E)$, weights $w : E \to \mathbb{R}_+$

Task: find $S \subseteq V$ maximizing $\sum_{e \in \delta(S)} w(e)$



SDP for MAX CUT

Quadratic program:

$$\max \quad \frac{1}{2} \cdot \sum_{\{i,j\} \in E} w_{ij}(1 - x_i x_j) \qquad\qquad \text{w.l.o.g.: } V = [n]$$

$$\text{s.t.} \quad x_i \in \{-1, +1\} \qquad\qquad \forall\, i \in [n]$$

SDP for MAX CUT

Quadratic program:

$$\max \quad \frac{1}{2} \cdot \sum_{\{i,j\} \in E} w_{ij}(1 - x_i x_j) \qquad\qquad \text{w.l.o.g.: } V = [n]$$

$$\text{s.t.} \quad x_i \in \{-1, +1\} \qquad\qquad \forall\, i \in [n]$$

Relaxation:

$$Z^* \;:=\; \max \quad \frac{1}{2} \cdot \sum_{\{i,j\} \in E} w_{ij}(1 - {v_i}^T v_j)$$

$$\text{s.t.} \qquad {v_i}^T v_i \;=\; 1 \qquad\qquad \forall\, i \in [n]$$

$$v_i \;\in \mathbb{R}^n \qquad\qquad \forall\, i \in [n]$$

Selecting a random hyperplane

$$
\begin{aligned}
Z^* \;:=\; \max \quad & \frac{1}{2} \cdot \sum_{\{i,j\} \in E} w_{ij}(1 - v_i^T v_j) & \\
\text{s.t.} \quad & v_i^T v_i \;=\; 1 & \forall\, i \in [n] \\
& v_i \;\in \mathbb{R}^n & \forall\, i \in [n]
\end{aligned}
$$

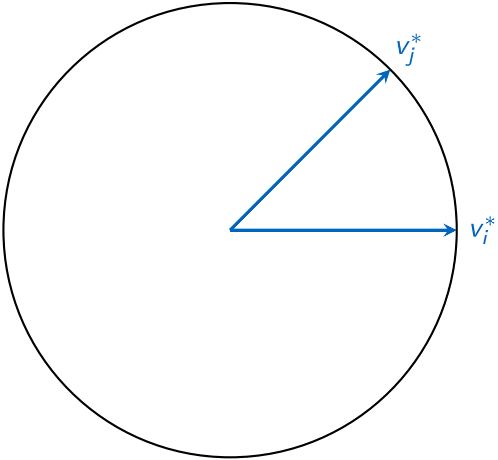
Algorithm

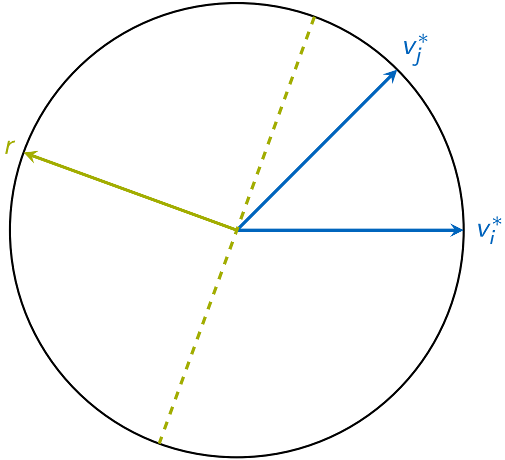
1. Compute optimal solution v^* to SDP.
2. Choose $r \in \mathbb{R}^n$ with $r^T r = 1$ uniformly at random.
3. Return $S := \{i \in [n] : r^T v_i^* \geq 0\}$.

Theorem 13.6

The algorithm is a randomized 0.878-approximation for MAX CUT.

Analysis

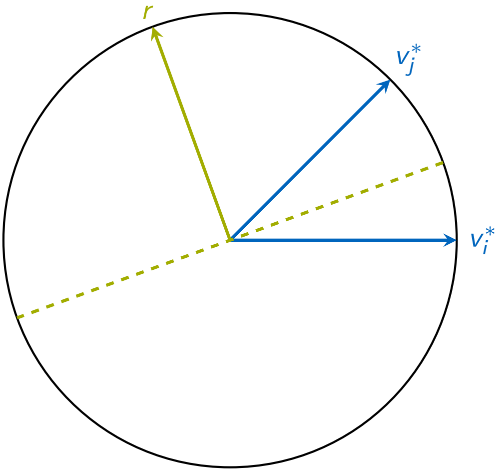




$i \notin S, j \notin S$

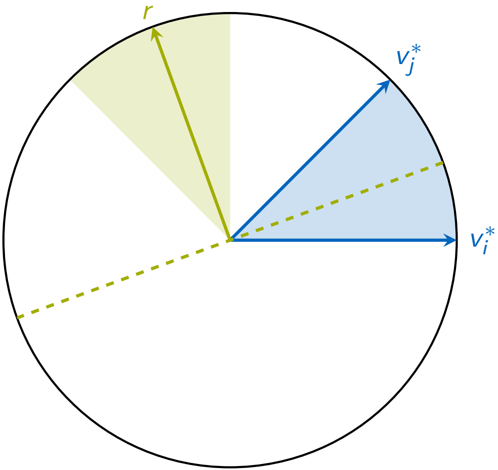
$\{i, j\} \notin \delta(S)$

Analysis



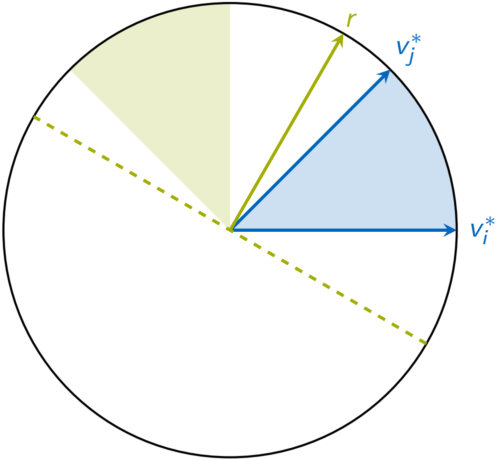
$i \notin S, j \notin S$

$\{i, j\} \in \delta(S)$



$i \notin S, j \notin S$

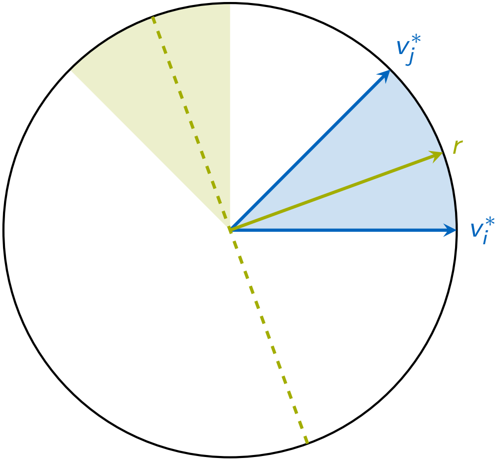
$\{i,j\} \in \delta(S)$



$i \in S, j \in S$

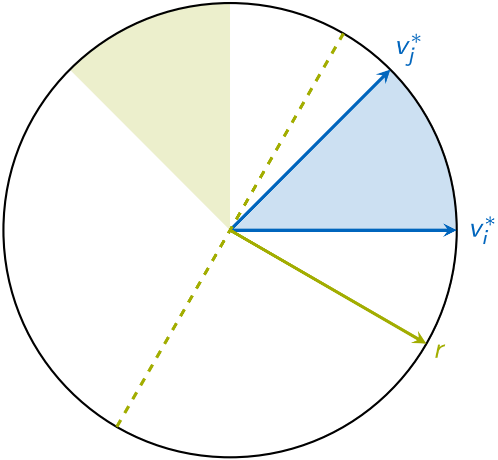
$\{i, j\} \notin \delta(S)$

Analysis



$i \in S, j \in S$

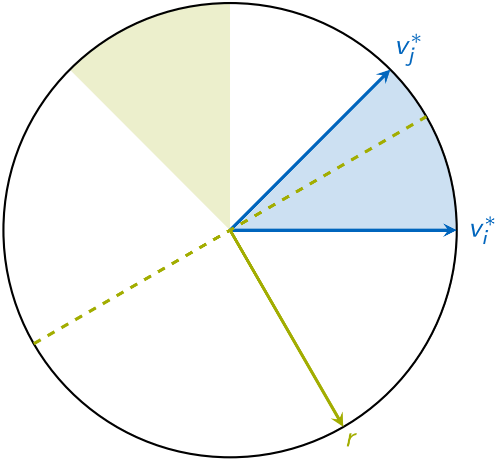
$\{i, j\} \notin \delta(S)$



$i \in S, j \in S$

$\{i, j\} \notin \delta(S)$

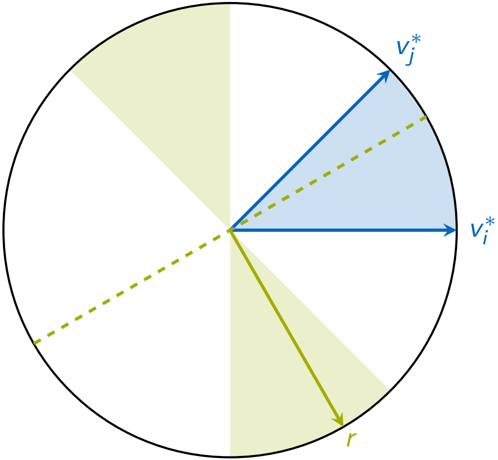
Analysis



$i \notin S, j \in S$

$\{i, j\} \in \delta(S)$

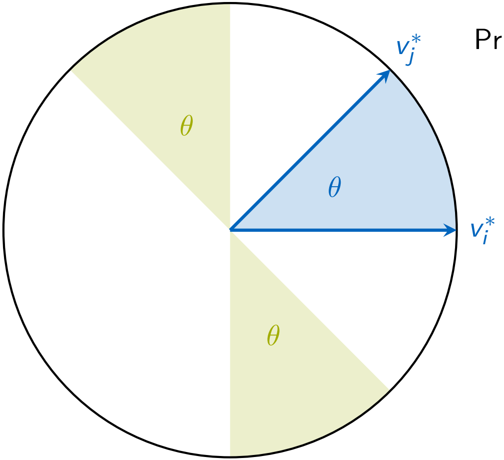
Analysis



$i \notin S, j \in S$

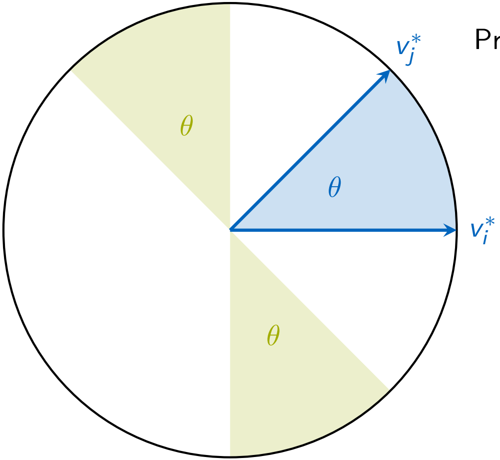
$\{i, j\} \in \delta(S)$

Analysis



$\Pr[\{i,j\} \in \delta(S)] = \frac{2\theta}{2\pi}$

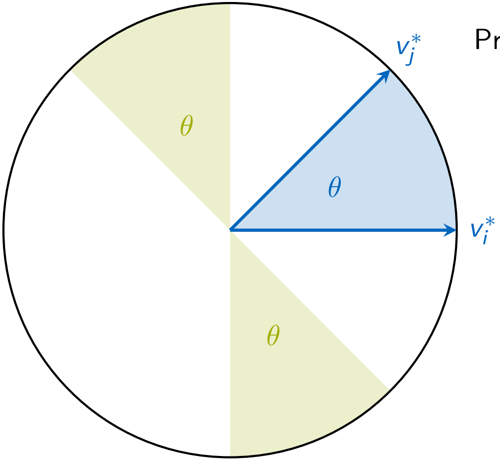
Analysis



$$\Pr[\{i,j\} \in \delta(S)] = \frac{2\theta}{2\pi}$$

$$\theta = \arccos({v_i^*}^T v_j^*)$$

Analysis



$\Pr[\{i,j\} \in \delta(S)] = \frac{2\theta}{2\pi}$

$\theta = \arccos({v_i^*}^T v_j^*)$

Lemma 13.7

$$\frac{1}{\pi}\arccos(x) \;\geq\; 0.878 \cdot \frac{1}{2}(1-x) \qquad \forall\, x \in [-1,1]$$

Next Wednesday: FAQ Session