Lecture: Approximation Algorithms

Jannik Matuschke

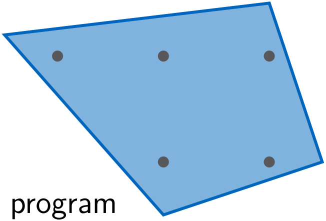
November 12, 2018

LP Rounding

General idea:

1. **formulation**
   formulate problem as an integer program
2. **relaxation**
   drop integrality requirement $\rightarrow$ LP
3. **rounding**
   solve LP and construct solution for original problem with

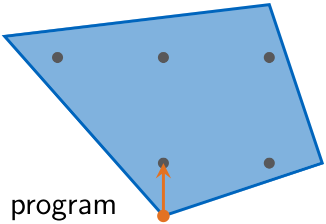
$$\text{ALG} \leq \alpha Z^* \leq \alpha \, \text{OPT}$$



General idea:

1. formulation
   formulate problem as an integer program
2. relaxation
   drop integrality requirement $\rightarrow$ LP
3. rounding
   solve LP and construct solution for original problem with

$$\text{ALG} \leq \alpha Z^* \leq \alpha\,\text{OPT}$$



General idea:

1. **formulation**
   formulate problem as an integer program
2. **relaxation**
   drop integrality requirement $\rightarrow$ LP
3. **rounding**
   solve LP and construct solution for original problem with

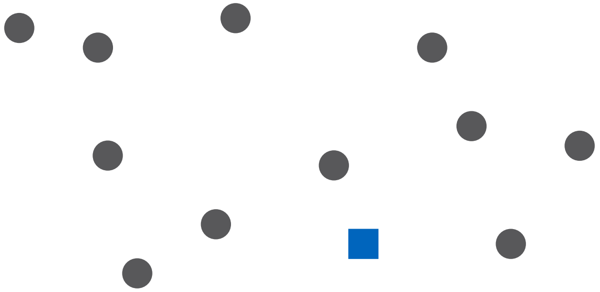
$$\text{ALG} \leq \alpha Z^* \leq \alpha \, \text{OPT}$$

LP Rounding:
Prize-collecting Steiner Tree

Prize-collecting Steiner Tree

Input: graph $G = (V \cup \{r\}, E)$, distances $d : E \to \mathbb{R}_+$,
penalties $\pi : V \to \mathbb{R}_+$

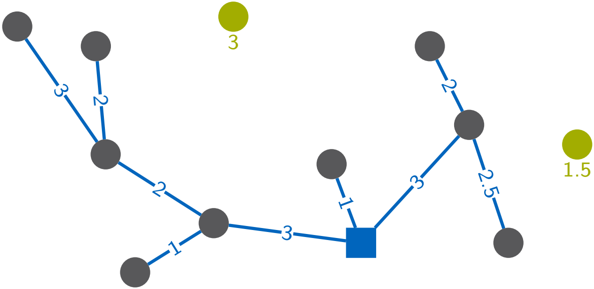
Task: find $U \subseteq V$ and a tree T spanning $U \cup \{r\}$
minimizing $\sum_{e \in T} d(e) + \sum_{v \in V \setminus U} \pi(v)$



Prize-collecting Steiner Tree

Input: graph $G = (V \cup \{r\}, E)$, distances $d : E \to \mathbb{R}_+$,
penalties $\pi : V \to \mathbb{R}_+$

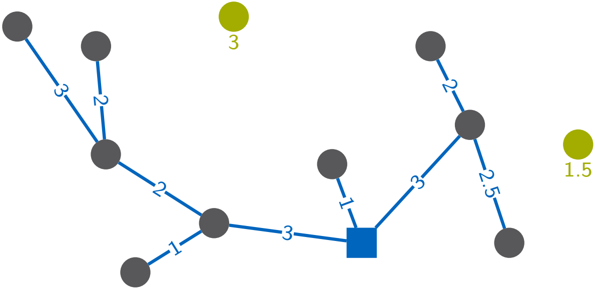
Task: find $U \subseteq V$ and a tree T spanning $U \cup \{r\}$
minimizing $\sum_{e \in T} d(e) + \sum_{v \in V \setminus U} \pi(v)$



Prize-collecting Steiner Tree

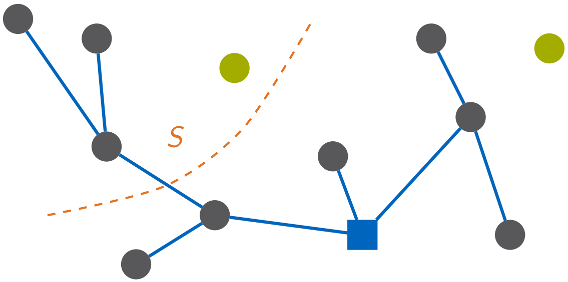
Input: graph $G = (V \cup \{r\}, E)$, distances* $d : E \to \mathbb{R}_+$, penalties $\pi : V \to \mathbb{R}_+$

Task: find $U \subseteq V$ and a tree T spanning $U \cup \{r\}$ minimizing $\sum_{e \in T} d(e) + \sum_{v \in V \setminus U} \pi(v)$



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

LP relaxation



variables:

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

$y(v) = 1 \;\Leftrightarrow\; v \in U$

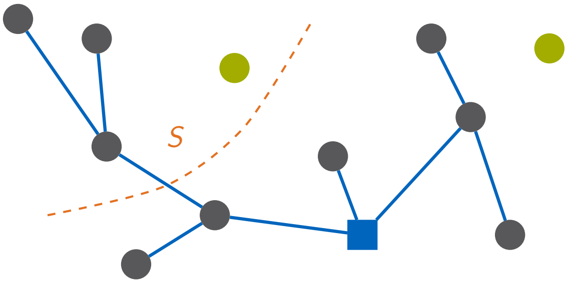
$$\min \sum_{e \in E} d(e)x(e) \;+\; \sum_{v \in V} \pi(v)(1 - y(v))$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x(e) \;\geq\; y(v) \quad \forall\, S \subseteq V,\; \forall\, v \in S$$

$$x(e) \;\in \{0, 1\} \qquad \forall\, e \in E$$

$$y(v) \;\in \{0, 1\} \qquad \forall\, v \in V$$

LP relaxation



variables:

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

$y(v) = 1 \;\Leftrightarrow\; v \in U$

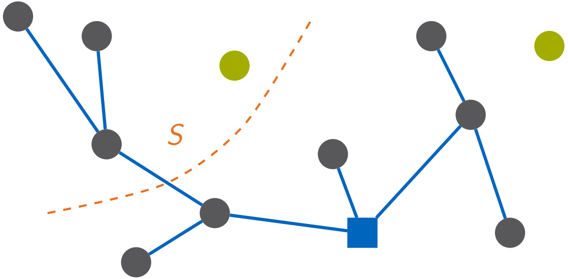
$$\min \sum_{e \in E} d(e)x(e) \;+\; \sum_{v \in V} \pi(v)(1 - y(v))$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x(e) \;\geq\; y(v) \quad \forall\, S \subseteq V,\; \forall\, v \in S$$

$$x(e) \;\in \{0,1\} \qquad \forall\, e \in E$$

$$y(v) \;\in \{0,1\} \qquad \forall\, v \in V$$

LP relaxation



variables:

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

$$y(v) = 1 \;\Leftrightarrow\; v \in U$$

$$Z^* := \min \sum_{e \in E} d(e)x(e) + \sum_{v \in V} \pi(v)(1 - y(v))$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x(e) \geq y(v) \quad \forall\, S \subseteq V,\ \forall\, v \in S$$

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

$$y(v) \geq 0 \quad \forall\, v \in V$$

A 3-approximation algorithm

Algorithm D (Deterministic Rounding)

1. Compute optimal solution (x^*, y^*) to LP.
2. Let $U := \{v \in V : y^*(v) \geq \alpha\}$.
3. Let T be minimum spanning tree on $U \cup \{r\}$.
4. Return T and U.

A 3-approximation algorithm

Algorithm D (Deterministic Rounding)

1. Compute optimal solution (x^*, y^*) to LP.
2. Let $U := \{v \in V \,:\, y^*(v) \geq \alpha\}$.
3. Let T be minimum spanning tree on $U \cup \{r\}$.
4. Return T and U.

Claim 1: $d(T) \;\leq\; \frac{2}{\alpha} \cdot \sum_{e \in E} d(e) x^*(e)$

Claim 2: $\pi(V \setminus U) \;\leq\; \frac{1}{1-\alpha} \cdot \sum_{v \in V} \pi(v)(1 - y^*(e))$

A 3-approximation algorithm

Algorithm D (Deterministic Rounding)

1. Compute optimal solution (x^*, y^*) to LP.
2. Let $U := \{v \in V \;:\; y^*(v) \geq \alpha\}$.
3. Let T be minimum spanning tree on $U \cup \{r\}$.
4. Return T and U.

Claim 1: $d(T) \;\leq\; \frac{2}{\alpha} \cdot \sum_{e \in E} d(e) x^*(e)$

Claim 2: $\pi(V \setminus U) \;\leq\; \frac{1}{1-\alpha} \cdot \sum_{v \in V} \pi(v)(1 - y^*(e))$

Theorem 6.1

Algorithm D is a 3-approximation algorithm for Prize-collecting Steiner Tree (when setting $\alpha = 2/3$).

Improved approximation

Algorithm B (Best of Many)

1. Compute optimal solution (x^*, y^*) to LP.
2. Run Algorithm D for every $\alpha \in \{y^*(v) : v \in V\}$.
3. Return best solution found.

Improved approximation

Algorithm B (Best of Many)

1. Compute optimal solution (x^*, y^*) to LP.
2. Run Algorithm D for every $\alpha \in \{y^*(v) : v \in V\}$.
3. Return best solution found.

Algorithm R (Randomized Rounding)

Choose α uniformly at random from $[\gamma, 1]$ and run Algorithm D.

Observation: Algorithm B is at least as good as Algorithm R.

(randomized analysis)

Improved approximation

Algorithm B (Best of Many)

1. Compute optimal solution (x^*, y^*) to LP.
2. Run Algorithm D for every $\alpha \in \{y^*(v) : v \in V\}$.
3. Return best solution found.

Algorithm R (Randomized Rounding)

Choose α uniformly at random from $[\gamma, 1]$ and run Algorithm D.

Observation: Algorithm B is at least as good as Algorithm R.
(randomized analysis)

Theorem 6.2

Algorithm R is a randomized 2.54-approximation algorithm for Prize-collecting Steiner Tree (when setting $\gamma = \exp(-1/2)$).

Corollary 6.3

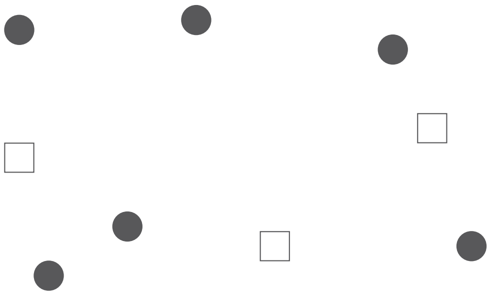
Algorithm B is a 2.54-approximation algorithm for PC-ST.

LP Rounding:
Uncapacitated Facility Location

(Metric) Uncapacitated Facility Location

Input: facilities F, clients C, opening cost f_i for $i \in F$, metric distances d_{ij} for $i \in F$ and $j \in C$

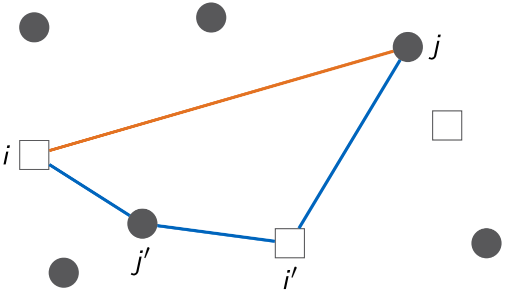
Task: find $S \subseteq F$, minimizing $\sum_{i \in S} f_i + \sum_{j \in C} \min_{i \in S} d_{ij}$



(Metric) Uncapacitated Facility Location

Input: facilities F, clients C, opening cost f_i for $i \in F$,
metric distances d_{ij} for $i \in F$ and $j \in C$

Task: find $S \subseteq F$, minimizing $\sum_{i \in S} f_i + \sum_{j \in C} \min_{i \in S} d_{ij}$

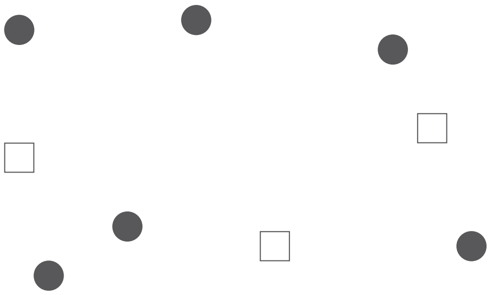


metric: $d_{ij} \leq d_{ij'} + d_{i'j'} + d_{i'j}$

(Metric) Uncapacitated Facility Location

Input: facilities F, clients C, opening cost f_i for $i \in F$, metric distances d_{ij} for $i \in F$ and $j \in C$

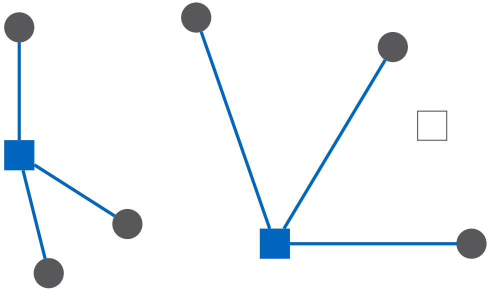
Task: find $S \subseteq F$, minimizing $\sum_{i \in S} f_i + \sum_{j \in C} \min_{i \in S} d_{ij}$



(Metric) Uncapacitated Facility Location

Input: facilities F, clients C, opening cost f_i for $i \in F$, metric distances d_{ij} for $i \in F$ and $j \in C$

Task: find $S \subseteq F$, minimizing $\sum_{i \in S} f_i + \sum_{j \in C} \min_{i \in S} d_{ij}$



LP relaxation

$$\begin{aligned}
\min \quad & \sum_{i \in F} \sum_{j \in C} d_{ij} x_{ij} + \sum_{i \in F} f_i y_i \\
\text{s.t.} \quad & \sum_{i \in F} x_{ij} = 1 && \forall\, j \in C \\
& y_i - x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& x_{ij} \in \{0,1\} && \forall\, i \in F, j \in C \\
& y_i \in \{0,1\} && \forall\, i \in F
\end{aligned}$$

variables:

$x_{ij} = 1 \Leftrightarrow i$ serves j

$y_i = 1 \Leftrightarrow i \in S$

LP relaxation

$$
\begin{aligned}
\min \quad & \sum_{i\in F}\sum_{j\in C} d_{ij}x_{ij} + \sum_{i\in F} f_i y_i \\
\text{s.t.} \quad & \sum_{i\in F} x_{ij} = 1 && \forall\, j \in C \\
& y_i - x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& x_{ij} \in \{0,1\} && \forall\, i \in F, j \in C \\
& y_i \in \{0,1\} && \forall\, i \in F
\end{aligned}
$$

variables:

$x_{ij} = 1 \;\Leftrightarrow\; i$ serves j

$y_i = 1 \;\Leftrightarrow\; i \in S$

LP relaxation

$$
\begin{aligned}
\min \quad & \sum_{i\in F}\sum_{j\in C} d_{ij}x_{ij} + \sum_{i\in F} f_i y_i \\
\text{s.t.} \quad & \sum_{i\in F} x_{ij} = 1 && \forall\, j \in C \\
& y_i - x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& y_i \geq 0 && \forall\, i \in F
\end{aligned}
$$

variables:

$x_{ij} = 1 \Leftrightarrow i$ serves j

$y_i = 1 \Leftrightarrow i \in S$

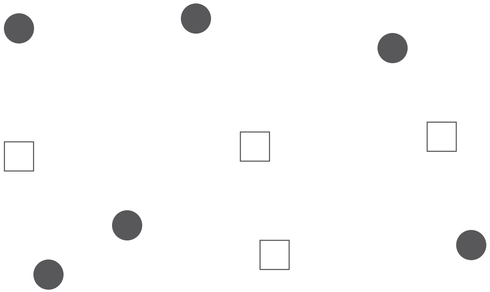
LP relaxation

$$\begin{aligned}
\min \quad & \sum_{i\in F}\sum_{j\in C} d_{ij}x_{ij} + \sum_{i\in F} f_i y_i & \\
\text{s.t.} \quad & \sum_{i\in F} x_{ij} = 1 & \forall\, j \in C \\
& y_i - x_{ij} \geq 0 & \forall\, i \in F, j \in C \\
& x_{ij} \geq 0 & \forall\, i \in F, j \in C \\
& y_i \geq 0 & \forall\, i \in F
\end{aligned}$$

variables:

$x_{ij} = 1 \;\Leftrightarrow\; i$ serves j

$y_i = 1 \;\Leftrightarrow\; i \in S$



LP relaxation

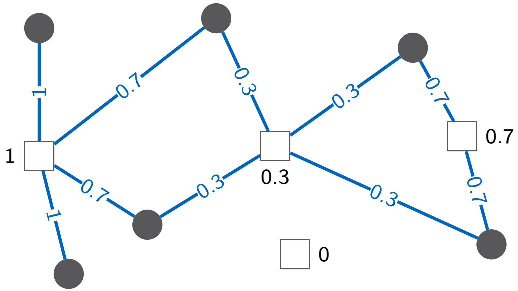
$$\min \sum_{i\in F}\sum_{j\in C} d_{ij}x_{ij} + \sum_{i\in F} f_i y_i$$

$$\begin{aligned} \text{s.t.} \quad \sum_{i\in F} x_{ij} &= 1 && \forall\, j \in C \\ y_i - x_{ij} &\geq 0 && \forall\, i \in F, j \in C \\ x_{ij} &\geq 0 && \forall\, i \in F, j \in C \\ y_i &\geq 0 && \forall\, i \in F \end{aligned}$$

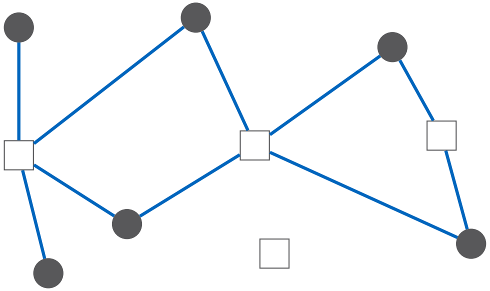
variables:

$x_{ij} = 1 \Leftrightarrow i$ serves j

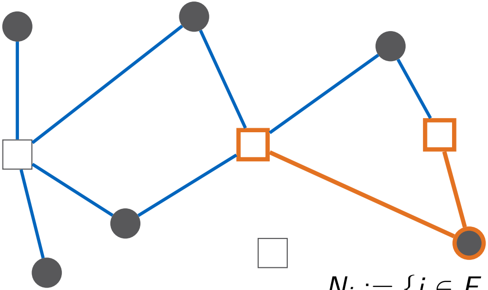
$y_i = 1 \Leftrightarrow i \in S$



Ideas

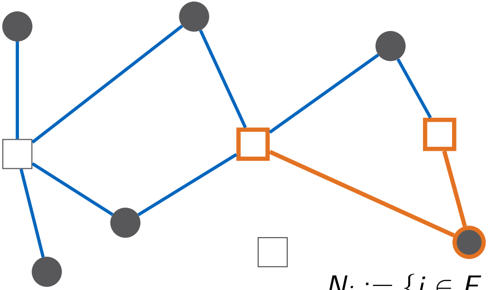


Ideas



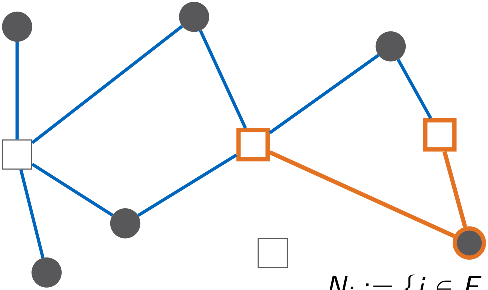
$N_j := \{i \in F \, : \, x^*_{ij} > 0\}$

Ideas



$N_j := \{i \in F \ : \ x^*_{ij} > 0\}$

Idea: For each $j \in C$, open $i \in N_j$ minimizing d_{ij}.

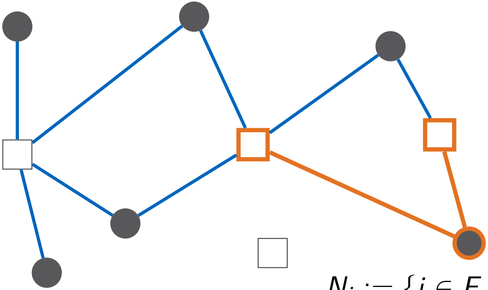


$N_j := \{i \in F \,:\, x^*_{ij} > 0\}$

Idea: For each $j \in C$, open $i \in N_j$ minimizing d_{ij}.

$\rightsquigarrow$ ignores opening costs

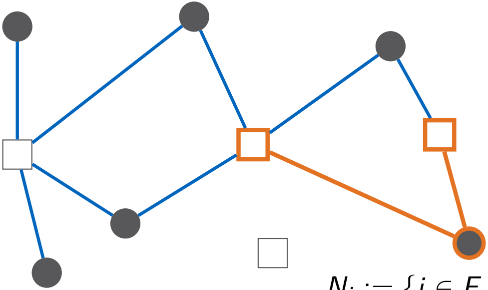
Ideas



$N_j := \{i \in F \ : \ x^*_{ij} > 0\}$

Idea: For each $j \in C$, open $i \in N_j$ minimizing f_i.

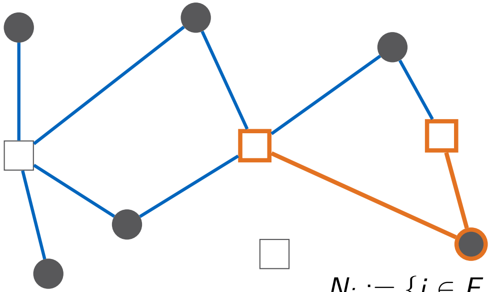
Ideas



$N_j := \{i \in F \;:\; x^*_{ij} > 0\}$

Idea: For each $j \in C$, open $i \in N_j$ minimizing f_i.

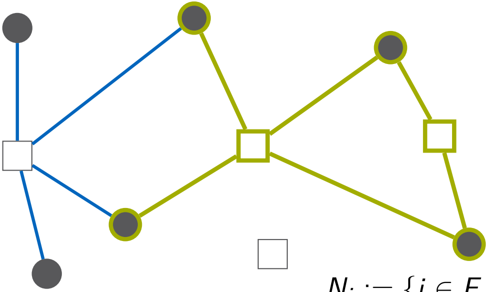
$\rightsquigarrow$ opening costs can stack up



$N_j := \{i \in F \,:\, x^*_{ij} > 0\}$

Idea: Let $X \subseteq C$ with $N_j \cap N_{j'} = \emptyset$ for $j, j' \in X$. For each $j \in X$, open $i \in N_j$ minimizing f_i.

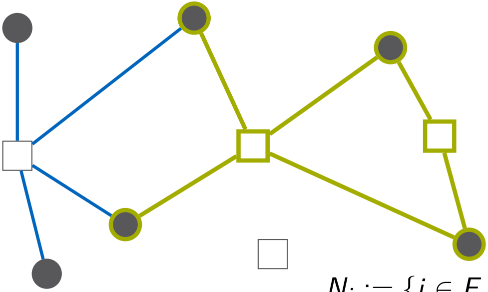
Ideas



$$N_j := \{i \in F \ : \ x^*_{ij} > 0\}$$
$$N^2_j := \{j' \in C \ : \ N_j \cap N_{j'} \neq \emptyset\}$$

Idea: Let $X \subseteq C$ with $N_j \cap N_{j'} = \emptyset$ for $j, j' \in X$. For each $j \in X$, open $i \in N_j$ minimizing f_i.

Ideas

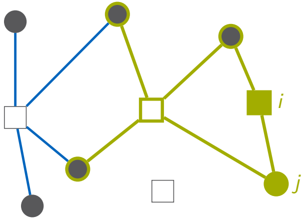


$$N_j := \{i \in F \;:\; x^*_{ij} > 0\}$$
$$N_j^2 := \{j' \in C \;:\; N_j \cap N_{j'} \neq \emptyset\}$$

Idea: Let $X \subseteq C$ with $N_j \cap N_{j'} = \emptyset$ for $j, j' \in X$. For each $j \in X$, open $i \in N_j$ minimizing f_i. $\rightsquigarrow$ Connection costs of $C \setminus X$?

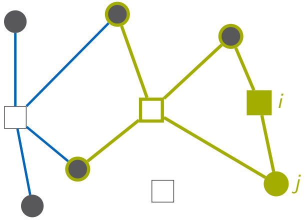
Algorithm 1 (Deterministic Rounding)

1. Compute optimal solutions (x^*, y^*) and (v^*, w^*) to LP relaxation and its dual.
2. Initialize $S := \emptyset$, $C' := C$.
3. while $(C' \neq \emptyset)$
   Choose $j \in C'$ minimizing v_j^*.
   Choose $i \in N_j$ minimizing f_i.
   $C' := C' \setminus N_j^2$ and $S := S \cup \{i\}$.
4. Return S.



Algorithm 1 (Deterministic Rounding)

1. Compute optimal solutions (x^*, y^*) and (v^*, w^*) to LP relaxation and its dual.
2. Initialize $S := \emptyset$, $C' := C$.
3. while $(C' \neq \emptyset)$
   Choose $j \in C'$ minimizing v_j^*.
   Choose $i \in N_j$ minimizing f_i.
   $C' := C' \setminus N_j^2$ and $S := S \cup \{i\}$.
4. Return S.

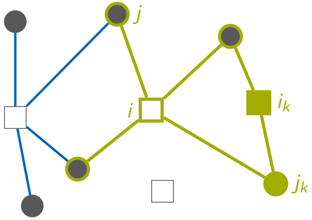


Theorem 6.4

Algorithm 1 is a 4-approximation algorithm for Uncapacitated Facility Location.

Algorithm 1 (Deterministic Rounding)

1. Compute optimal solutions (x^*, y^*) and (v^*, w^*) to LP relaxation and its dual.
2. Initialize $S := \emptyset$, $C' := C$.
3. while $(C' \neq \emptyset)$
   - Choose $j \in C'$ minimizing v_j^*.
   - Choose $i \in N_j$ minimizing f_i.
   - $C' := C' \setminus N_j^2$ and $S := S \cup \{i\}$.
4. Return S.



Theorem 6.4

Algorithm 1 is a 4-approximation algorithm for Uncapacitated Facility Location.

Improved algorithm

Algorithm 2 (Improved Deterministic Rounding)

1. Compute optimal solutions (x^*, y^*) and (v^*, w^*) to LP relaxation and its dual.
2. Initialize $S := \emptyset$, $C' := C$.
3. while $(C' \neq \emptyset)$

   Choose $j \in C'$ minimizing v_j^*.

   Choose $i \in N_j$ minimizing $f_i + \sum_{j' \in N_j^2} d_{ij'}$.

   $C' := C' \setminus N_j^2$ and $S := S \cup \{i\}$.
4. Return S.

Improved algorithm

Algorithm 2 (Improved Deterministic Rounding)

1. Compute optimal solutions (x^*, y^*) and (v^*, w^*) to LP relaxation and its dual.
2. Initialize $S := \emptyset$, $C' := C$.
3. while $(C' \neq \emptyset)$ $\qquad \Delta_j := \sum_{i \in F} d_{ij} x_{ij}$

   Choose $j \in C'$ minimizing $v_j^* + \Delta_j$.

   Choose $i \in N_j$ minimizing $f_i + \sum_{j' \in N_j^2} d_{ij'}$.

   $C' := C' \setminus N_j^2$ and $S := S \cup \{i\}$.
4. Return S.

Improved algorithm

Algorithm 3 (Randomized Rounding)

1. Compute optimal solutions (x^*, y^*) and (v^*, w^*) to LP relaxation and its dual.
2. Initialize $S := \emptyset$, $C' := C$.
3. while $(C' \neq \emptyset)$ $\quad \Delta_j := \sum_{i \in F} d_{ij} x_{ij}$

   Choose $j \in C'$ minimizing $v_j^* + \Delta_j$.

   Choose $i \in N_j$ randomly according to probabilities x_{ij}^*.

   $C' := C' \setminus N_j^2$ and $S := S \cup \{i\}$.
4. Return S.

Algorithm 3 (Randomized Rounding)

1. Compute optimal solutions (x^*, y^*) and (v^*, w^*) to LP relaxation and its dual.
2. Initialize $S := \emptyset$, $C' := C$.
3. while $(C' \neq \emptyset)$ $\qquad \Delta_j := \sum_{i \in F} d_{ij} x_{ij}$

   Choose $j \in C'$ minimizing $v_j^* + \Delta_j$.

   Choose $i \in N_j$ randomly according to probabilities x_{ij}^*.

   $C' := C' \setminus N_j^2$ and $S := S \cup \{i\}$.
4. Return S.

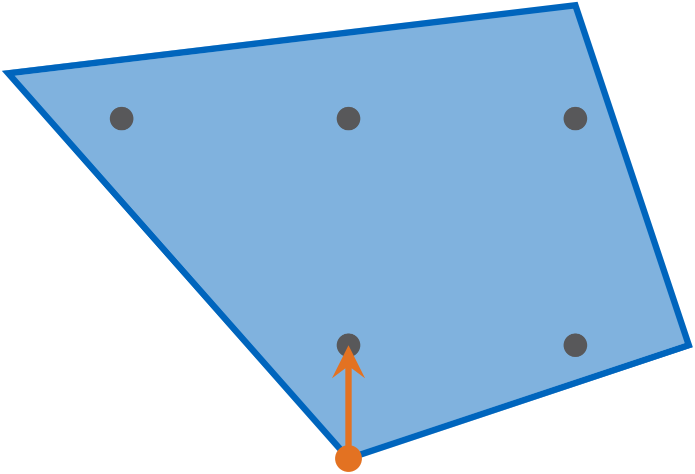
Theorem 6.7

Algorithm 3 is a randomized 3-approximation algorithm for UFL.

Corollary 6.8

Algorithm 2 is a 3-approximation algorithm for UFL.

Today you learnt ...



solve LP and round (randomly)