Lecture: Approximation Algorithms

Jannik Matuschke

TUM

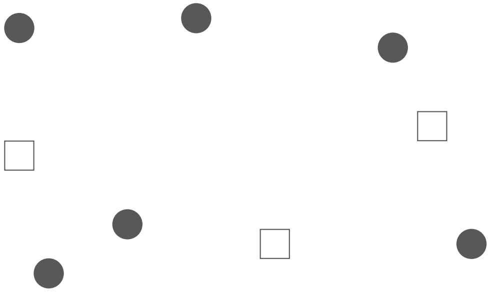
November 26, 2018

The primal-dual method for Uncapacitated Facility Location

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 $j \in C$

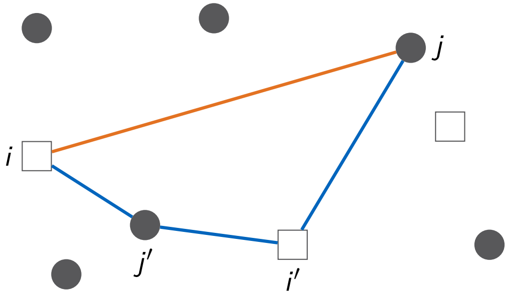
Task: find $S \subseteq F$, minimizing $\sum_{i \in S} f_i + \sum_{j \in C} d(S, j)$ where $d(S, j) := \min_{i \in S} d_{ij}$



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 $\sum_{i \in S} f_i + \sum_{j \in C} d(S, j)$
where $d(S, j) := \min_{i \in S} d_{ij}$

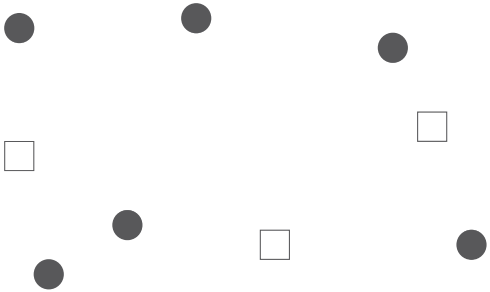


metric: $d_{ij} \leq d_{ij'} + d_{i'j'} + d_{i'j}$

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 $j \in C$

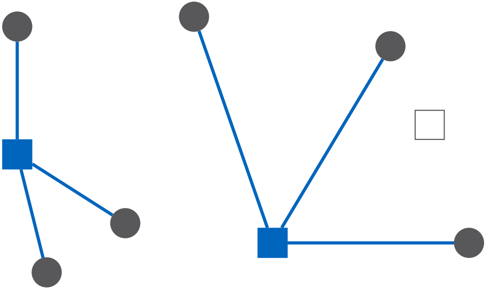
Task: find $S \subseteq F$, minimizing $\sum_{i \in S} f_i + \sum_{j \in C} d(S,j)$ where $d(S,j) := \min_{i \in S} d_{ij}$



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 $\sum_{i \in S} f_i + \sum_{j \in C} d(S, j)$ where $d(S, j) := \min_{i \in S} d_{ij}$



LP relaxation

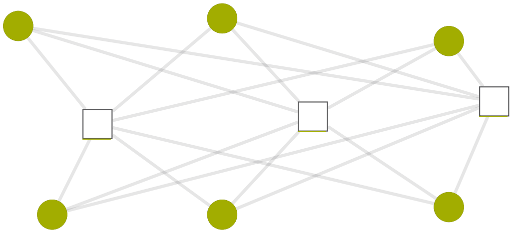
$$
\begin{aligned}
\min \quad & \sum_{i \in F} \sum_{j \in C} d_{ij} x_{ij} \; + \; \sum_{i \in F} f_i y_i \\
\text{s.t.} \quad & \sum_{i \in F} x_{ij} = 1 && \forall\, j \in C \\
& y_i - x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& y_i \geq 0 && \forall\,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

variables:

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

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

Primal-dual algorithm

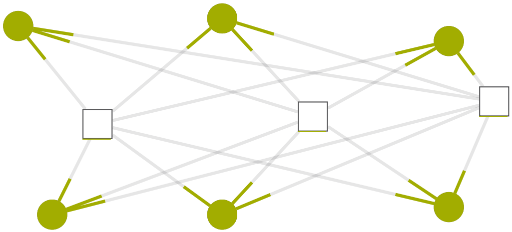


$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$
   Increase v_j for all $j \in C'$ until (a) or (b) happens:
   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.
   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm

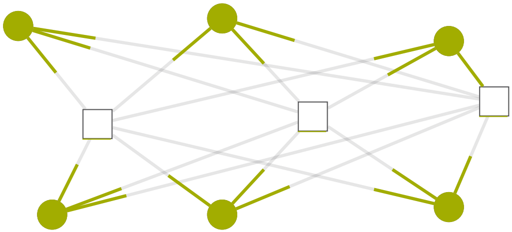


$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$
   Increase v_j for all $j \in C'$ until (a) or (b) happens:
   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.
   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm

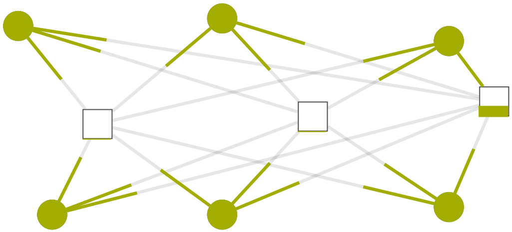


$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$
   Increase v_j for all $j \in C'$ until (a) or (b) happens:
   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.
   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm

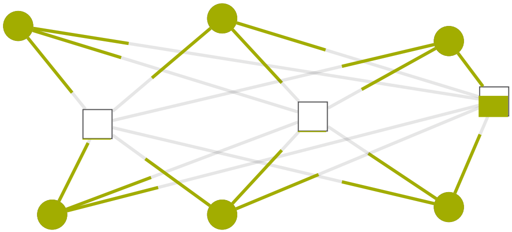


$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$
   Increase v_j for all $j \in C'$ until (a) or (b) happens:
   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.
   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm



$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

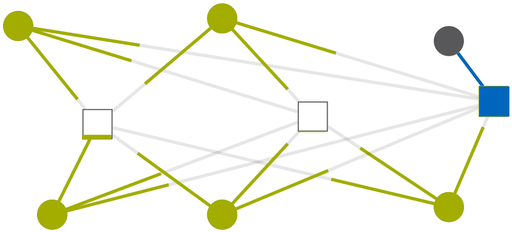
1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$

   Increase v_j for all $j \in C'$ until (a) or (b) happens:

   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.

   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm



$w_{ij} := (v_j - d_{ij})^+$

$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

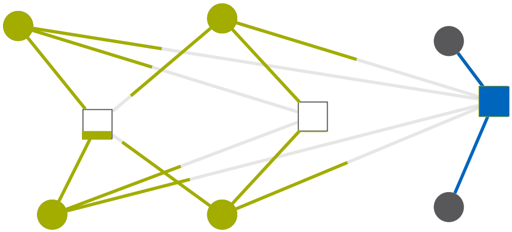
1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$

   Increase v_j for all $j \in C'$ until (a) or (b) happens:

   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.

   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm

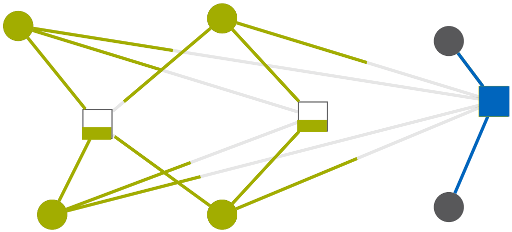


$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$
   Increase v_j for all $j \in C'$ until (a) or (b) happens:
   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.
   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm

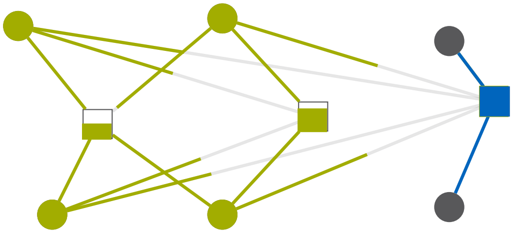


$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$
   Increase v_j for all $j \in C'$ until (a) or (b) happens:
   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.
   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm



$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

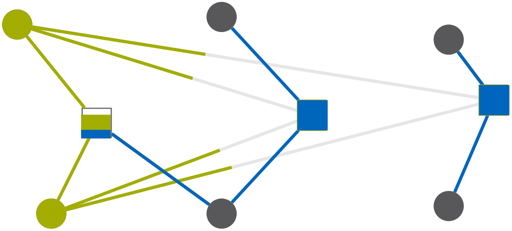
1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$

   Increase v_j for all $j \in C'$ until (a) or (b) happens:

   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.

   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm

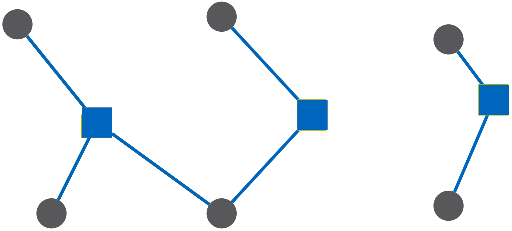


$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$
   Increase v_j for all $j \in C'$ until (a) or (b) happens:
   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.
   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm



$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

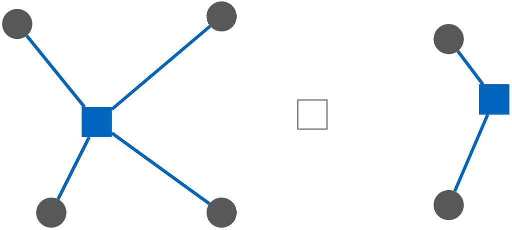
1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$

   Increase v_j for all $j \in C'$ until (a) or (b) happens:

   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.

   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm



$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

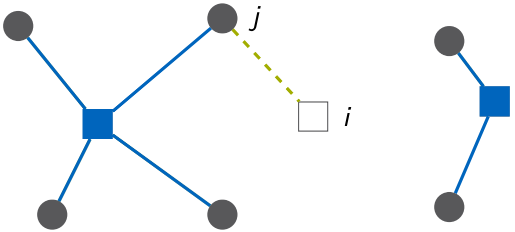
Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$

   Increase v_j for all $j \in C'$ until (a) or (b) happens:

   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.

   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

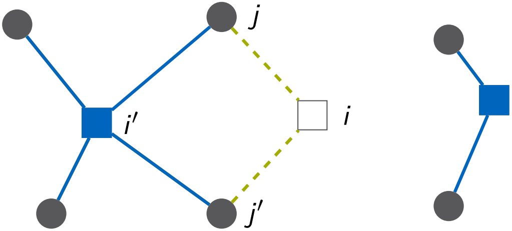


$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$
   Increase v_j for all $j \in C'$ until (a) or (b) happens:
   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.
   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

Primal-dual algorithm



$w_{ij} := (v_j - d_{ij})^+$
$N_i := \{j \in C : w_{ij} > 0\}$

Algorithm

1. $S := \emptyset,\ v := 0,\ C' := C$
2. while $C' \neq \emptyset$

   Increase v_j for all $j \in C'$ until (a) or (b) happens:

   (a) If $\sum_{j \in C} w_{ij} = f_i$ for some $i \in F \setminus S$ then
   $S := S \cup \{i\}$ and $C' := C' \setminus N_i$.

   (b) If $v_j = d_{ij}$ for some $i \in S$ and $j \in C'$ then
   $C' := C' \setminus \{j\}$.

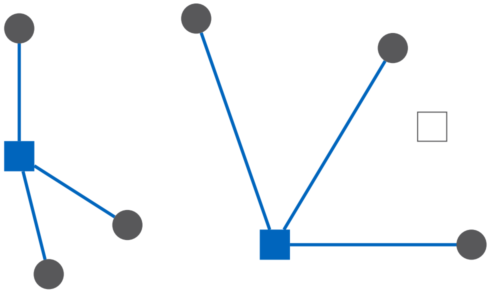
3. Let $\bar{S} \subseteq S$ be $\subseteq$-max with $N_i \cap N_{i'} = \emptyset$ for $i, i' \in \bar{S}$.
   Return $\bar{S}$.

The primal-dual method and Lagrangean relaxation for k-Median

k-MEDIAN

Input: facilities F, clients C, $k \in \mathbb{N}$
metric distances d_{ij} for $i \in F$ and $j \in C$

Task: find $S \subseteq F$, $|S| \leq k$ minimizing $\sum_{j \in C} d(S, j)$



Lagrangean relaxation

$$
\begin{aligned}
[P] \quad Z^* \;=\; \min \;& \sum_{i \in F} \sum_{j \in C} d_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{i \in F} x_{ij} = 1 && \forall\, j \in C \\
& y_i - x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& \sum_{i \in F} y_i \leq k \\
& x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& y_i \geq 0 && \forall\, i \in F
\end{aligned}
$$

Lagrangean relaxation

$$
\begin{aligned}
[P(\lambda)] \quad g(\lambda) \;:=\; \min \quad & \sum_{i \in F} \sum_{j \in C} d_{ij} x_{ij} + \lambda \left(\sum_{i \in F} y_i \;-\; k \right) \\
\text{s.t.} \quad & \sum_{i \in F} x_{ij} = 1 && \forall\, j \in C \\
& y_i - x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& y_i \geq 0 && \forall\, i \in F
\end{aligned}
$$

Lagrangean relaxation

$$
\begin{aligned}
[P(\lambda)] \quad g(\lambda) := \min \; & \sum_{i\in F}\sum_{j\in C} d_{ij}x_{ij} + \sum_{i\in F}\lambda y_i - \lambda k \\
\text{s.t.} \; & \sum_{i\in F} x_{ij} = 1 && \forall\, j \in C \\
& y_i - x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& x_{ij} \geq 0 && \forall\, i \in F, j \in C \\
& y_i \geq 0 && \forall\, i \in F
\end{aligned}
$$

Lagrangean relaxation

$$
\begin{aligned}
[D(\lambda)] \quad g(\lambda) \;=\; \max \quad & \sum_{j\in C} v_j \;-\; \lambda k \\
\text{s.t.} \quad & \sum_{j\in C} w_{ij} \;\leq\; \lambda && \forall\, i \in F \\
& v_j - w_{ij} \;\leq\; d_{ij} && \forall\, i \in F, j \in C \\
& w_{ij} \;\geq\; 0 && \forall\, i \in F, j \in C
\end{aligned}
$$

Lagrangean relaxation

$$
\begin{aligned}
[D(\lambda)] \quad g(\lambda) \;=\; \max \quad & \sum_{j \in C} v_j \;-\; \lambda k \\
\text{s.t.} \quad & \sum_{j \in C} w_{ij} \;\leq\; \lambda & \forall\, i \in F \\
& v_j - w_{ij} \;\leq\; d_{ij} & \forall\, i \in F, j \in C \\
& w_{ij} \;\geq\; 0 & \forall\, i \in F, j \in C
\end{aligned}
$$

Idea: Choose some $\lambda \geq 0$. Run primal-dual algorithm for UFL instance with facility costs $f_i = \lambda$ for all $i \in F$.

Lagrangean relaxation

$$
\begin{aligned}
[D(\lambda)] \quad g(\lambda) \;=\; \max \quad & \sum_{j\in C} v_j \;-\; \lambda k \\
\text{s.t.} \quad & \sum_{j\in C} w_{ij} \;\le\; \lambda && \forall\, i \in F \\
& v_j - w_{ij} \;\le\; d_{ij} && \forall\, i \in F, j \in C \\
& w_{ij} \;\ge\; 0 && \forall\, i \in F, j \in C
\end{aligned}
$$

Idea: Choose some $\lambda \geq 0$. Run primal-dual algorithm for UFL instance with facility costs $f_i = \lambda$ for all $i \in F$.

if $|S| \leq k$: S feasible
if $|S| \geq k$: $f(S) \leq 3\,\text{OPT}$

Can we get both at the same time?

Combining two solutions

Bisection search

We can find in polynomial time $\lambda_1, \lambda_2 \geq 0$ and corresponding $S_1, S_2 \subseteq F$ computed by the primal-dual algorithm such that

- $0 \leq \lambda_2 - \lambda_1 \leq \frac{\varepsilon \, \mathrm{OPT}}{3|F|}$ and
- $|S_1| \geq k \geq |S_2|$.

Combining two solutions

Bisection search

We can find in polynomial time $\lambda_1, \lambda_2 \geq 0$ and corresponding $S_1, S_2 \subseteq F$ computed by the primal-dual algorithm such that

- $0 \leq \lambda_2 - \lambda_1 \leq \frac{\varepsilon \, \mathrm{OPT}}{3|F|}$ and
- $|S_1| \geq k \geq |S_2|$.

Algorithm

1. Find λ_1, S_1 and λ_2, S_2 as above. Define $\alpha := \dfrac{k - |S_2|}{|S_2| - |S_1|}$.
2. If $\alpha \leq 1/2$ then return S_2.
3. If $\alpha > 1/2$ then
   - For $i \in S_2$ let $\phi(i)$ be the nearest facility to i in S_1.
   - Let $\bar{S} := \{\phi(i) : i \in S_2\}$.
   - Select $X \subseteq S_1 \setminus \bar{S}$ with $|X| = k - |\bar{S}|$ uniformly at random.
   - Return $|\bar{S} \cup X|$.

How to compute λ_1, λ_2?

Algorithm (Bisection Search)

1. $\lambda_1 := 0$, $\lambda_2 := \sum_{j \in C} \max_{i \in F} d_{ij}$
   $\Delta := \frac{\varepsilon}{3|F|} \min\{d_{ij} \,:\, i \in F, j \in C, d_{ij} > 0\}$
2. while $(\lambda_2 - \lambda_1 > \Delta)$
   - Compute $S \subseteq F$ with primal-dual for facility cost $\lambda := (\lambda_1 + \lambda_2)/2$.
   - If $|S| \geq k$ then $\lambda_1 := \lambda$ and $S_1 := S$.
   - If $|S| \leq k$ then $\lambda_2 := \lambda$ and $S_2 := S$.

How to compute λ_1, λ_2?

Algorithm (Bisection Search)

1. $\lambda_1 := 0$, $\lambda_2 := \sum_{j \in C} \max_{i \in F} d_{ij}$
   $\Delta := \frac{\varepsilon}{3|F|} \min\{d_{ij} \,:\, i \in F, j \in C, d_{ij} > 0\}$
2. while $(\lambda_2 - \lambda_1 > \Delta)$
   - Compute $S \subseteq F$ with primal-dual for facility cost $\lambda := (\lambda_1 + \lambda_2)/2$.
   - If $|S| \geq k$ then $\lambda_1 := \lambda$ and $S_1 := S$.
   - If $|S| \leq k$ then $\lambda_2 := \lambda$ and $S_2 := S$.

Theorem 10.7

Lagrangean relaxation is a $(6 + \varepsilon)$-approximation algorithm for k-Median (for any fixed $\varepsilon > 0$).

Theorem 10.8

There is no 1.735-approximation for k-Median unless every problem in NP has an $O(n^{O(\log \log n)})$-time algorithm.