Local search for k-median

Section 9.2 in Williamson-Shmoys

Local search

- Local search: starting with an arbitrary solution, compute a locally optimal solution with respect to a fixed set of operations.
- Max-cut in graph $G = (V, E)$: Start with an arbitrary $S \subseteq V$, local search using the following operations.
    1. $M_S(u)$: Update $S \coloneqq S \setminus \{u\}$.
    2. $M_{V \setminus S}(u)$: Update $S \coloneqq S \cup \{u\}$.

- At a locally optimal solution, no operation will improve the cost.

1. For any $u \in S$, $\text{cost}(S) - \text{cost}(S \setminus \{u\}) \geq 0$.
2. For any $u \in V \setminus S$, $\text{cost}(S) - \text{cost}(S \cup \{u\}) \geq 0$.

- Idea: Each operation gives an inequality. Use them to get a bound on the cost of the solution.

- For any $u \in S$, $\text{cost}(S) - \text{cost}(S \setminus \{u\}) = |N(u) \cap (V \setminus S)| - |N(u) \cap S| \geq 0$
- Therefore,

$$2|N(u) \cap (V \setminus S)| \geq |N(u) \cap S| + |N(u) \cap (V \setminus S)| = |N(u)|$$

- Similarly, for any $u \in V \setminus S$, we have $2|N(u) \cap S| \geq |N(u)|$.

- Adding all of them,

$$\underbrace{\sum_{u \in S} |N(u) \cap (V \setminus S)| + \sum_{u \in V \setminus S} |N(u) \cap S|}_{= 2\,\text{cost}(S)} \geq \frac{1}{2}\left(\sum_{u \in S} |N(u)| + \sum_{u \in V \setminus S} |N(u)|\right) = \frac{1}{2} 2m$$

k-median

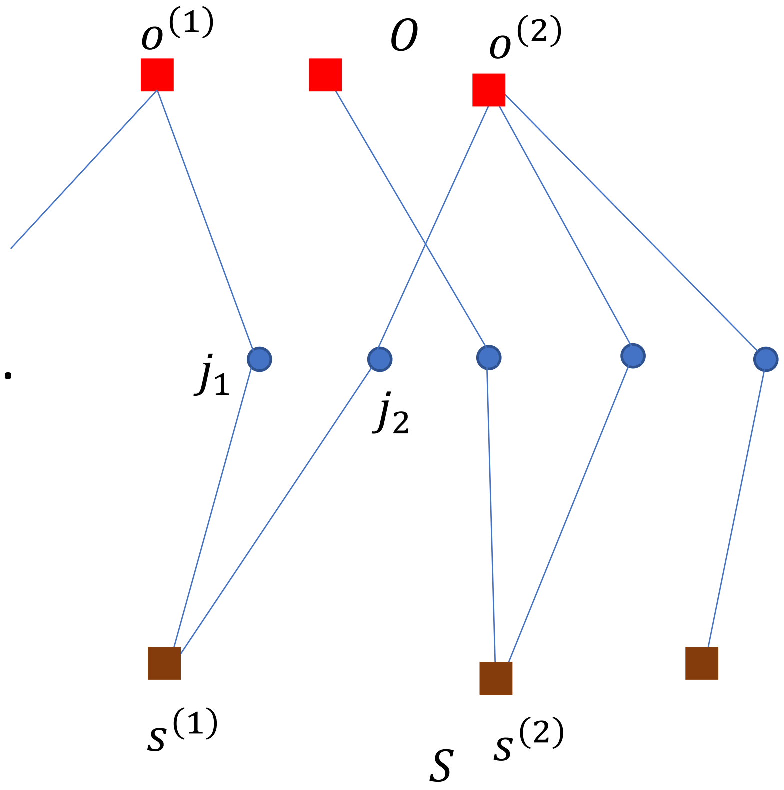
- Goal: open at most k facilities to minimize the service cost.

1. Start with an arbitrary set $S \subseteq \mathcal{F}$ of k open facilities.
2. While there exists a $f_1 \in S$ and $f_2 \in \mathcal{F} \setminus S$, such that $\mathrm{c}\big((S \setminus \{f_1\}) \cup \{f_2\}\big) < \mathrm{c}(S)$,
   1. Swap f_1 and f_2, i.e., set $S \coloneqq (S \setminus \{f_1\}) \cup \{f_2\}$.
3. Output S.

- Let S denote a locally optimal solution and let O denote an optimal solution. For $j \in \mathcal{C}$, let c_j denote the service cost of j in S and let o_j denote the service cost of j in O.

Analysis

- At a locally optimal solution $c\left(S - s^{(1)} + o^{(1)}\right) - c(S) \geq 0$.
- Let $s^{(1)}$ be the nearest facility in S to $o^{(1)}$. Bounding increase in service cost of $S - s^{(1)} + o^{(1)}$?
- Assign clients in $N\left(s^{(1)}\right) \cap N\left(o^{(1)}\right)$ to $o^{(1)}$
  - For j_1, change $= -c_{j_1} + o_{j_1}$

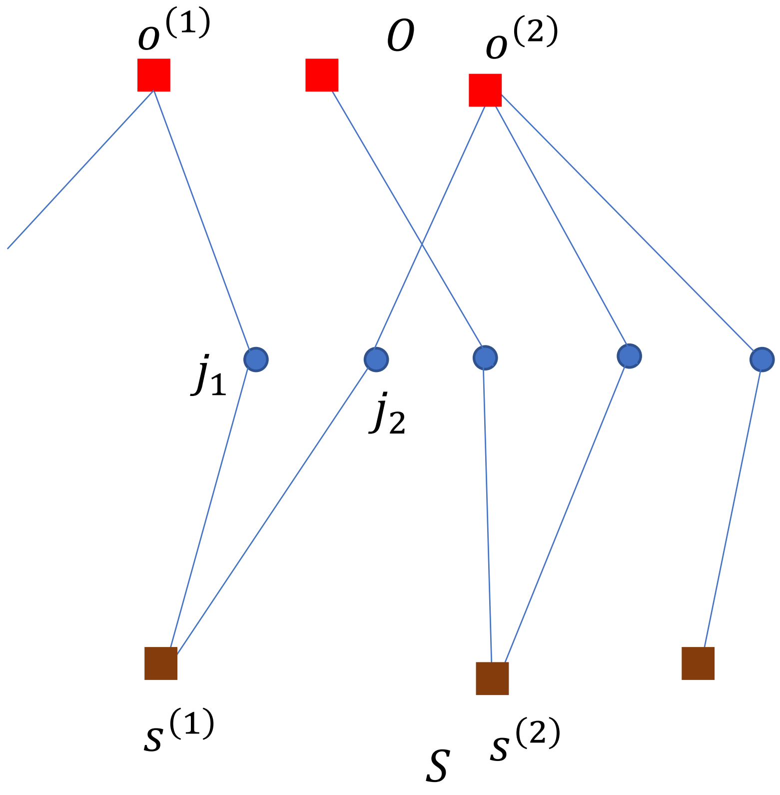


Swap $s^{(1)}$ and $o^{(1)}$

- Assign clients in $N\left(s^{(1)}\right) \setminus N\left(o^{(1)}\right)$ to nearest facility in $S \setminus \left\{s^{(1)}\right\}$
  - For j_2, where $s^{(2)}$ is the nearest facility in S to $o^{(2)}$.

$$\begin{aligned}\text{change} &\leq -c_{j_2} + d\left(j_2, o^{(2)}\right) + d\left(o^{(2)}, s^{(2)}\right) \\ &\leq -c_{j_2} + o_{j_2} + d\left(o^{(2)}, s^{(1)}\right) \\ &\leq -c_{j_2} + o_{j_2} + \left(c_{j_2} + o_{j_2}\right) \\ &= 2o_{j_2}\end{aligned}$$

- Choose swaps carefully!



Choosing the swaps

- For each $i^* \in O$, let $\eta(i^*)$ be the nearest facility in S.
- Let $R_0, R_1, R_{\geq 2} \subseteq S$ be the set of facilities such that number of facilities mapped to them by η is $0, 1, \geq 2$ respectively.
- Construct the set of swaps $\mathcal{S}$ as follows.
  - For each $i \in R_1$, add $\left(i, \eta^{-1}(i)\right)$ to $\mathcal{S}$. Let $O_1 \subseteq O$ be the set of facilities matched in this step.
  - Out of $R_0 \times (O \setminus O_1)$ choose a set of swaps such that each facility in $O \setminus O_1$ appears exactly once and each facility in R_0 appears at most twice, and add all of them to $\mathcal{S}$.

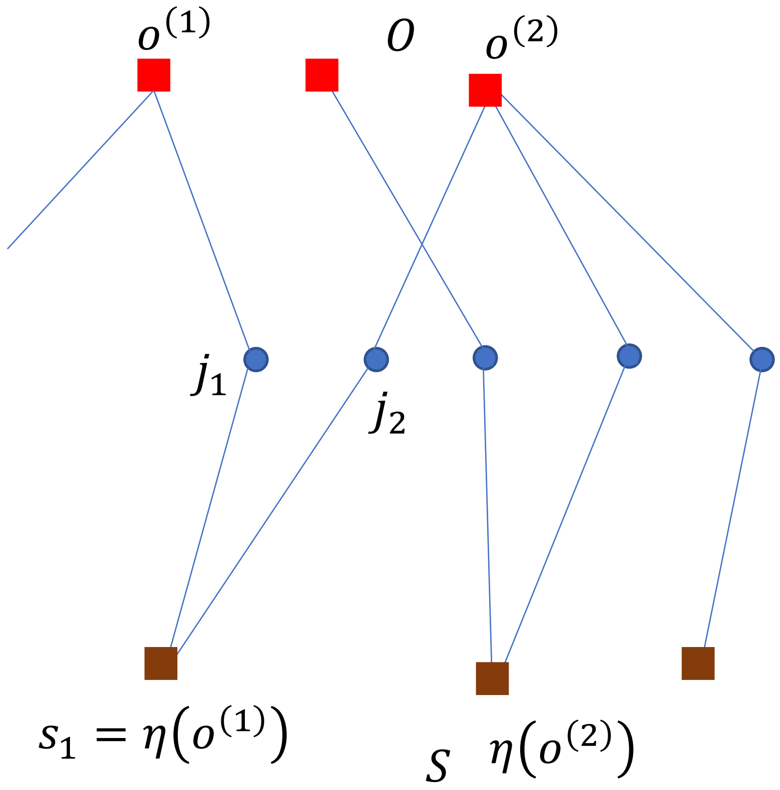
- $|O \setminus O_1| \leq 2|R_0|$ (H.W.).

- Lemma: For each swap $(s, o) \in \mathcal{S}$

$$0 \leq c(S + o - s) - c(S) \leq \sum_{j \in N(o)} (o_j - c_j) + \sum_{j \in N(s)} 2o_j$$

- New assignment
  - For each $j \in N(o)$, assign j to o.
  - For each $j \in N(s) \setminus N(o)$, assign j to $\eta\left(f^O(j)\right)$, where $f^O(j)$ is the facility serving j in O.

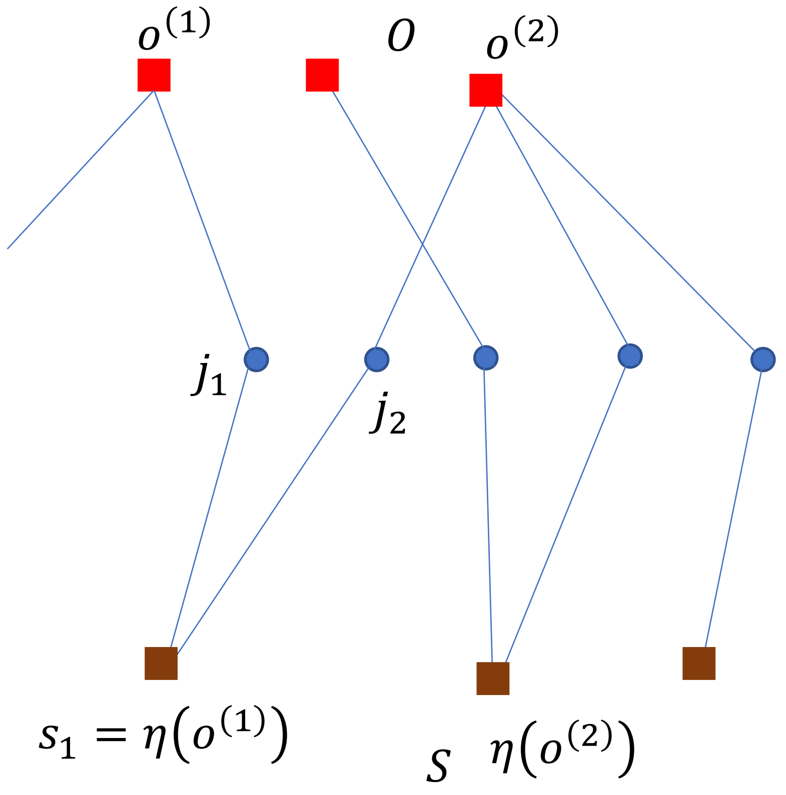
- Observe that in the second case $\eta\left(f^O(j)\right) \neq s$.
  - If $s \in R_1$, then $\eta^{-1}(s) = o$.
  - If $s \in R_0$, then $\eta\left(f^O(j)\right) \neq s$.
  - If $s \in R_{\geq 2}$, then $(s, *) \notin \mathcal{S}$.



- Change in service of cost j
  - If $j \in N(o)$, change $= -c_j + o_j$
  - If $j \in N(s) \setminus N(o)$,

change $\leq -c_j + d\left(j, f^O(j)\right) + d\left(f^O(j), \eta\left(f^O(j)\right)\right)$

$\leq -c_j + o_j + d(f^O(j), s) \leq -c_j + o_j + \left(c_j + o_j\right) = 2o_j$



- Therefore
- $0 \leq c(S + o - s) - c(S) \leq \sum_{j \in N(o)} \left(o_j - c_j\right) + \sum_{j \in N(s)} 2o_j$

- Summing over all swaps in $\mathcal{S}$, we get

$$\begin{aligned}
0 &\leq \sum_{(s,o)\in\mathcal{S}} \left(\sum_{j\in N(o)} (o_j - c_j) + \sum_{j\in N(s)} 2o_j \right) \\
&\leq \sum_{o\in O} \left(\sum_{j\in N(o)} o_j \right) - \sum_{o\in O} \left(\sum_{j\in N(o)} c_j \right) + 2 \sum_{(s,o)\in\mathcal{S}} \left(\sum_{j\in N(s)} o_j \right) \\
&\leq c(O) - c(S) + 2 \cdot 2 \sum_{s\in S} \left(\sum_{j\in N(s)} o_j \right) \\
&= c(O) - c(S) + 4c(O)
\end{aligned}$$

- $c(S) \leq 5c(O)$. Therefore, 5-approximation algorithm.

Running time

- Checking whether a swap exists can be done in polynomial time.

- How many swaps?
- If there is no lower bound on the decrease in cost, there may be too many iterations.
- Perform a swap operation only if the cost decreases by a factor of $(1 - \epsilon)$.
  - Can be shown that this gives a $(5 + \epsilon')$-approximation (details in the book).
  - Running time is polynomial in size of instance and $\frac{1}{\epsilon}$ (H.W.).

Other local search operations

- t-swap: Close $\leq t$ facilities in S and open the same number of facilities.
- Running time: $n^{O(t)}$
- $\left(3 + \frac{2}{t}\right)$-approximation to k-median.

- Current best known approximation factor for k-median is 2.675 [Byrka et al. - 2017] (not based on local search).