The primal-dual method

Sections 7.3 and 7.6 from Williamson-Shmoys.

Shortest $s - t$ path

- Initialize $y = 0, F = \emptyset$.
- While there is no s-t path in (V, F) do
  - Let C be the connected component in (V, F) containing s.
  - Increase y_C until there is an edge $e \in \delta(C)$ such that corresponding dual constraint is tight.
  - Set $F := F \cup \{e\}$.
- Let P be an $s - t$ path in (V, F).
- Output P.

$$\min \sum_{e \in E} c_e x_e$$

Subject to

$$\sum_{e \in \delta(S)} x_e \geq 1 \;\; \forall S \in C_{s,t}$$

$$x_e \geq 0 \; \forall e \in E$$

$$\max \sum_{S \in C_{s,t}} y_S$$

subject to

$$\sum_{S \in C_{s,t} : e \in \delta(S)} y_S \leq c_e \; \forall e \in E$$

$$y_s \geq 0 \;\; \forall S \in C_{s,t}$$

- Lemma: At any point in the algorithm, F forms a tree containing s.
- Proof by induction (H.W.).

- Therefore, algorithm outputs an s-t path, and for each edge e in the path, $c_e = \sum_{S:e\in\delta(s)} y_S$.

$$\sum_{e\in P} c_e = \sum_{e\in P} \sum_{S:e\in\delta(s)} y_S = \sum_{S\in C_{s,t}} |P \cap \delta(S)| y_S$$

- Lemma: For $S \in C_{s,t}$ if $y_S > 0$ then $|P \cap \delta(S)| = 1$.

- Lemma implies that $\sum_{e\in P} c_e = \sum_{S\in C_{s,t}} y_S \leq OPT$ using weak duality.
- Since no s-t path of length $< OPT$, P must have length $= OPT$.

- Lemma: For $S \in C_{s,t}$ if $y_S > 0$ then $|P \cap \delta(S)| = 1.$

Proof:

- Suppose for some $S \in C_{s,t}$ has $y_S > 0$ and $|P \cap \delta(S)| > 1.$
- There must be a sub-path P' of P joining two vertices in S such that only the start and end vertices of P' are in S.
- At the time we increased y_S, F was a tree spanning the vertices in S.
- $F \cup P'$ contains a cycle.
- This is a contradiction.

- Algorithm behaves in the same way as Dijkstra's algorithm.

Primal-dual algorithm for facility location

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

Subject to

$$\sum_{i\in\mathcal{F}} x_{ij} = 1 \ \forall j \in \mathcal{C}$$

$$x_{ij} \le y_i \ \forall i \in \mathcal{F}, j \in \mathcal{C}$$

$$x_{ij} \ge 0 \ \forall i \in \mathcal{F}, j \in \mathcal{C}$$

$$y_i \ge 0 \ \forall i \in \mathcal{F}$$

$$\max \sum_{j\in\mathcal{C}} v_j$$

Subject to

$$v_j - w_{ij} \le d_{ij} \ \forall i \in \mathcal{F}, j \in \mathcal{C}$$

$$\sum_{j\in\mathcal{C}} w_{ij} \le f_i \ \forall i \in \mathcal{F}$$

$$w_{ij} \ge 0 \ \forall i \in \mathcal{F}, j \in \mathcal{C}$$

- For client j, w_{ij} can be viewed as its “contribution” to paying the opening cost of facility i.
- For $i \in \mathcal{F}$, define $N(i) \stackrel{\text{def}}{=} \{j \in \mathcal{C}: v_j \ge d(i,j)\}$.
- For $j \in \mathcal{C}$, define $N(j) \stackrel{\text{def}}{=} \{i \in \mathcal{F}: v_j \ge d(i,j)\}$.

Algorithm

- Initialize $v_j = 0,\ w_{ij} = 0\ \forall i, j, S = \mathcal{C}$ and $T = \emptyset, T' = \emptyset$.
- While $S \neq \emptyset$,
  1. Increase v_j for all $j \in S$ and w_{ij} for all $j \in S, i \in N(j)$ until (1) $v_j = d(i, j)$ for some $j \in S, i \in T$ or (2) $\sum_{j \in \mathcal{C}} w_{ij} = f_i$ for some $i \notin T$.
  2. Case (1): $v_j = d(i, j)$ for some $j \in S, i \in T$.
     - Set $S \coloneqq S \setminus \{j\}$.
  3. Case (2): $\sum_{j \in \mathcal{C}} w_{ij} = f_i$ for some $i \notin T$.
     - Set $S \coloneqq S \setminus N(i)$ and $T \coloneqq T \cup \{i\}$.
- While $T \neq \emptyset$,
  - Pick $i \in T$ and set $T' \coloneqq T' \cup \{i\}$.
  - Set $T \coloneqq T \setminus \{h \in T : \exists j \in \mathcal{C} \text{ such that } w_{ij}, w_{hj} > 0\}$.
- Output T'.

$$\max \sum_{j \in \mathcal{C}} v_j$$

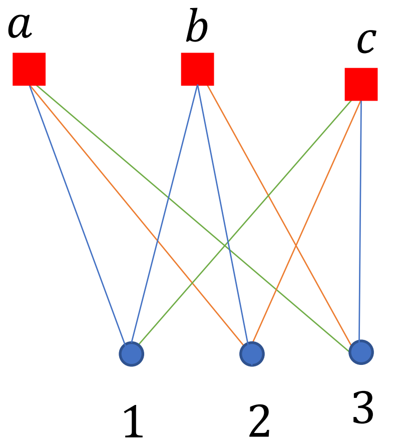
Subject to

$$v_j - w_{ij} \leq d_{ij}\ \forall i \in \mathcal{F}, j \in \mathcal{C}$$

$$\sum_{j \in \mathcal{C}} w_{ij} \leq f_i\ \forall i \in \mathcal{F}$$

$$w_{ij} \geq 0\ \forall i \in \mathcal{F}, j \in \mathcal{C}$$

Algorithm



- Length=1, length=2 and length=3. $f_a = 1, f_b = 3, f_c = 5$.

- Increase v_1, v_2, v_3 till $v_1 = v_2 = v_3 = 1$.
- Increase v_1, v_2, v_3 and $w_{1a}, w_{1b}, w_{2b}, w_{3c}$ till $v_1 = v_2 = v_3 = 2$.
  - $f_a = w_{1a}$. Therefore, $T \coloneqq \{a\}$.
  - $N(a) = \{1,2\}$. Therefore, $S \coloneqq \{3\}$.
- Increase v_3 and w_{3b}, w_{3c} till $v_3 = 3$.
  - $f_b = w_{1b} + w_{2b} + w_{3b}$.
  - Therefore, $T \coloneqq \{a, b\}$ and $S \coloneqq \emptyset$

- $T' \coloneqq \{a\}$.

Bounding cost of T'

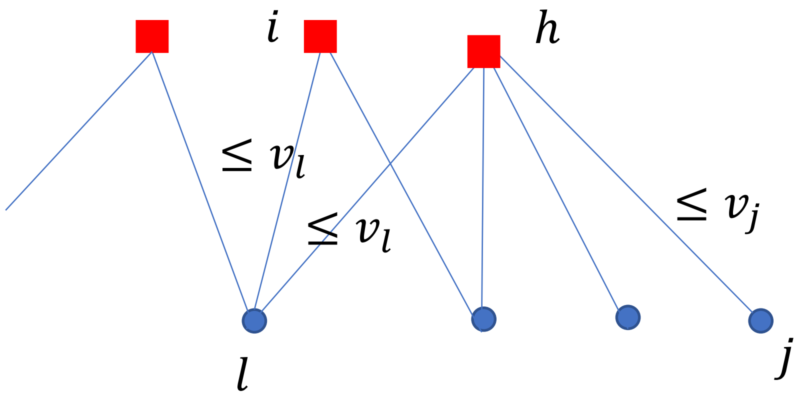
- Lemma: For any facility $i \in T$, $\sum_{j \in \mathcal{C}} w_{ij} = f_i$ (verify).
- Each client $j \in \mathcal{C}$ "pays for" at most one facility in T', i.e. there is at most one $i \in T'$ such that $w_{ij} > 0$.

$$\sum_{i \in T'} f_i = \sum_{i \in T'} \sum_{j \in N(i)} w_{ij} = \sum_{i \in T'} \sum_{j \in N(i)} \left(v_j - d(i,j)\right)$$

- Therefore,

$$\sum_{i \in T'} f_i + \sum_{i \in T'} \sum_{j \in N(i)} d(i,j) = \sum_{i \in T'} \sum_{j \in N(i)} v_j$$

- Lemma: If $j \in \mathcal{C}$ does not have a neighbor in T', then there exists a facility $i \in T'$ such $d(i,j) \leq 3v_j$.



- Let $h \in T \setminus T'$ be a facility because of which we deleted j from S.
- h is not in T' because there is another client $l \in \mathcal{C}$ and a facility $i \in T'$ such that $w_{hl}, w_{il} > 0$.
- We will show $d(h,l), d(l,i) \leq v_j$.
- Consider the point when we stopped increasing v_j. Either (1) h is already in T, or (2) h got added to T now.
- Since $w_{hl} > 0$, either v_l had already stopped increasing or we stop increasing v_l now. In both cases $v_j \geq v_l$.
- Since $w_{hl}, w_{il} > 0$, we have $d(h,l), d(l,i) \leq v_l \leq v_j$.

- Total cost

$$\sum_{i\in T'} f_i + \sum_{i\in T'} \sum_{j\in N(i)} d(i,j) + \sum_{j\in \mathcal{C}\setminus N(T')} d(i,j) = \sum_{i\in T'} \sum_{j\in N(i)} v_j + \sum_{j\in \mathcal{C}\setminus N(T')} 3v_j \leq 3 \sum_{j\in \mathcal{C}} v_j$$

- 3-approximation algorithm.