Part 20
Introduction into Primal dual algorithms

Source: *Approximation Algorithms* (Vazirani, Springer Press)

A generic problem

Situation: We want to approximate a problem, which (in many cases) is of the form

$$\begin{aligned} \min \sum_{j=1}^{n} c_j x_j & & \\ \sum_{j=1}^{n} a_{ij} x_j & \geq & b_i \ \forall i = 1, \ldots, m \\ x_j & \in & \{0,1\} \quad \forall j = 1, \ldots, n \end{aligned}$$

Examples so far: SET COVER, STEINER TREE, VERTEX COVER,...

A primal-dual pair

Primal "covering" LP:

$$\min \sum_{j=1}^{n} c_j x_j \qquad (P)$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad \forall i = 1, \ldots, m$$

$$x_j \geq 0 \quad \forall j = 1, \ldots, n$$

Dual "packing" LP:

$$\max \sum_{i=1}^{m} b_i y_i \qquad (D)$$

$$\sum_{i=1}^{m} a_{ij} y_i \leq c_j \quad \forall j = 1, \ldots, n$$

$$y_i \geq 0 \quad \forall i = 1, \ldots, m$$

A generic Approximation algorithm

Generic primal-dual algorithm:

(1) $x := \mathbf{0},\ y = \mathbf{0}$

(2) WHILE x not feasible DO

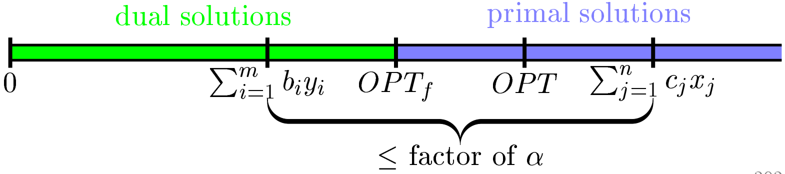
  (3) Increase dual variables in a suitable way until some dual constraint j becomes tight

  (4) Set $x_j := 1$

(5) RETURN x

Generic analysis:

- Show: At the end x is integer and feasible for primal
- Show: At the end y is feasible for dual
- Show: $\sum_{j=1}^{n} c_j x_j \leq \alpha \cdot \sum_{i=1}^{m} b_i y_i$ (α is the apx factor)



Relaxed complementary slackness

Lemma

Let $\alpha, \beta \geq 1$. Let x, y be primal/dual feasible solutions obtained by the algorithm. If

(A) *Relaxed primal compl. slack.:* $x_j > 0 \Rightarrow c_j \leq \alpha \sum_{i=1}^m a_{ij} y_i$

(B) *Relaxed dual compl. slack.:* $y_i > 0 \Rightarrow \sum_{j=1}^n a_{ij} x_j \leq \beta \cdot b_i$

Then $APX \leq \alpha \cdot \beta \cdot OPT_f$.

- Let APX be the cost of the produced solution. Then

$$\begin{aligned} APX &= \sum_{j=1}^n c_j x_j \stackrel{(A)}{\leq} \sum_{j=1}^n x_j \Big(\alpha \sum_{i=1}^m a_{ij} y_i \Big) = \alpha \sum_{i=1}^m y_i \sum_{j=1}^n a_{ij} x_j \\ &\stackrel{(B)}{\leq} \alpha\beta \sum_{i=1}^m y_i b_i \stackrel{y \text{ dual feasible}}{\leq} \alpha\beta \cdot OPT_f \quad \square \end{aligned}$$

Part 21
Steiner Forest

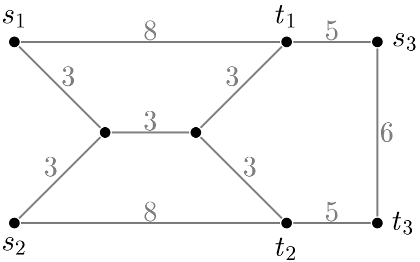
Source: *Approximation Algorithms* (Vazirani, Springer Press)

Steiner Forest

Problem: STEINER FOREST

- Given: Undirected graph $G = (V, E)$, edge cost $c : E \to \mathbb{Q}_+$, terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$
- Find: Minimum cost subgraph F connecting all terminal pairs:

$$OPT = \min_{F \subseteq E} \left\{ \sum_{e \in F} c(e) \mid \forall i = 1, \ldots, k : F \text{ connects } s_i \text{ and } t_i \right\}$$

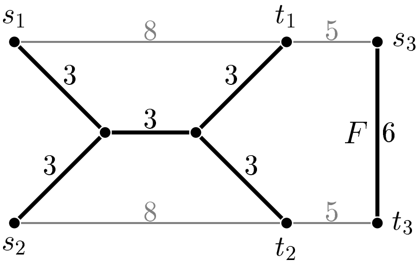


Steiner Forest

Problem: STEINER FOREST

- Given: Undirected graph $G = (V, E)$, edge cost $c : E \to \mathbb{Q}_+$, terminal pairs $(s_1, t_1), \dots, (s_k, t_k)$
- Find: Minimum cost subgraph F connecting all terminal pairs:

$$OPT = \min_{F \subseteq E} \left\{ \sum_{e \in F} c(e) \mid \forall i = 1, \dots, k : F \text{ connects } s_i \text{ and } t_i \right\}$$



The LP relaxation

- For any $S \subseteq V$ define cut requirement

$$f(S) = \begin{cases} 1 & \text{if } \exists i : |S \cap \{s_i, t_i\}| = 1 \\ 0 & \text{otherwise} \end{cases}$$

Primal LP relaxation:

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & & & (P) \\ \sum_{e \in \delta(S)} x_e & \geq & f(S) & \quad \forall S \subseteq V \\ x_e & \geq & 0 & \quad \forall e \in E \end{aligned}$$

Dual LP:

$$\begin{aligned} \max \sum_{S \subseteq V} f(S) y_S & & & (D) \\ \sum_{S : e \in \delta(S)} y_S & \leq & c_e & \quad \forall e \in E \\ y_S & \geq & 0 & \quad \forall S \subseteq V \end{aligned}$$

Preliminaries

- For $F \subseteq E, S \subseteq V$: $\delta_F(S) = \{\{u,v\} \in F \mid u \in S, v \notin S\}$
- A cut $S \subseteq V$ is violated by $F \subseteq E$, if there is a terminal pair (s_i, t_i) with $|\{s_i, t_i\} \cap S| = 1$ but $\delta_F(S) = \emptyset$
- A cut S is active w.r.t. F, if S is violated and minimal (i.e. there is no subset $S' \subset S$ that is also violated).
- An edge e is tight w.r.t. a dual solution $(y_S)_S$ if $\sum_{S:e\in\delta(S)} y_S = c_e$
  (i.e. if the dual constraint of c_e satisfied with equality).

The algorithm

(1) $F := \emptyset$, $y := \mathbf{0}$

(2) WHILE $\exists$ violated cut DO

  (3) Increase simultaneously y_S for all active cuts S, until some edge e gets tight

  (4) Add the tight edge e to F

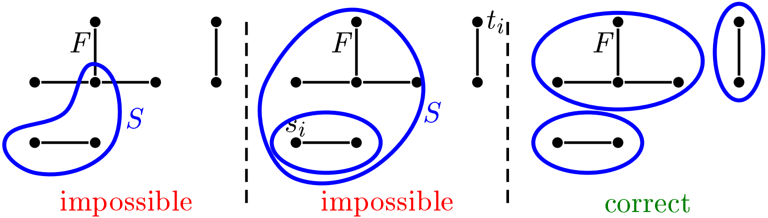
(5) Compute an arbitrary minimal feasible solution $F' \subseteq F$

The active cuts

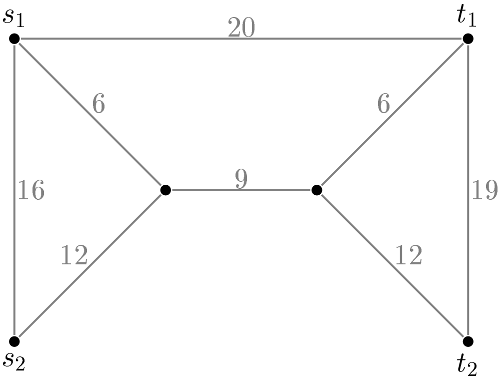
Lemma

The active cuts w.r.t. $F \subseteq E$ are connected components of F.

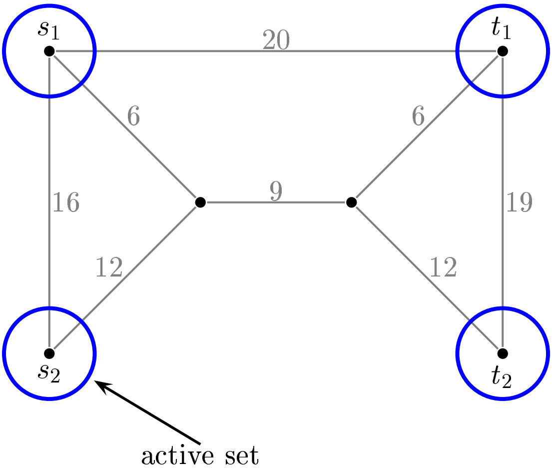
- Consider active cut S (S minimal, $f(S) = 1$, $\delta_F(S) = \emptyset$).
- $\delta_F(S) = \emptyset \Rightarrow$ connected components of F are either fully contained in S or fully outside
- S is violated, hence there is a pair $|\{s_i, t_i\} \cap S| = 1$
- The connected component of F inside S that contains s_i is also violated. Hence, S is a single connected component (or we would have a contradiction). □



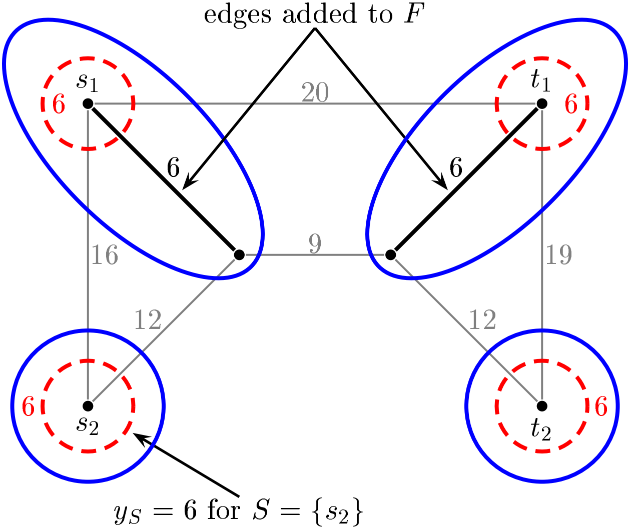
Example



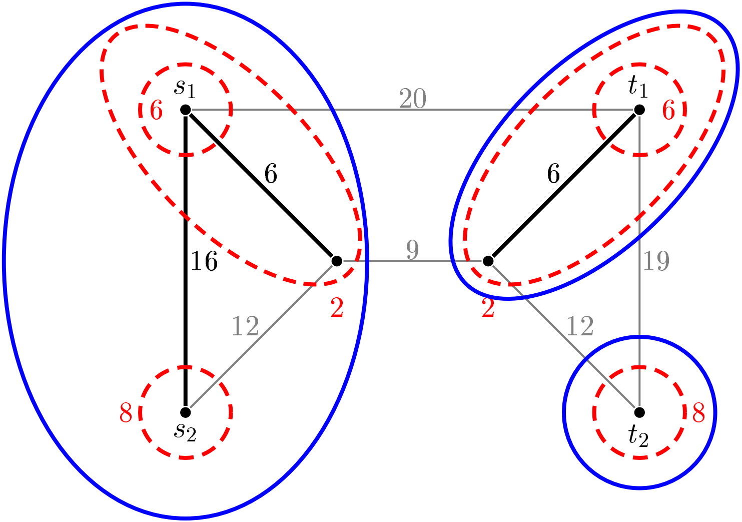
Example



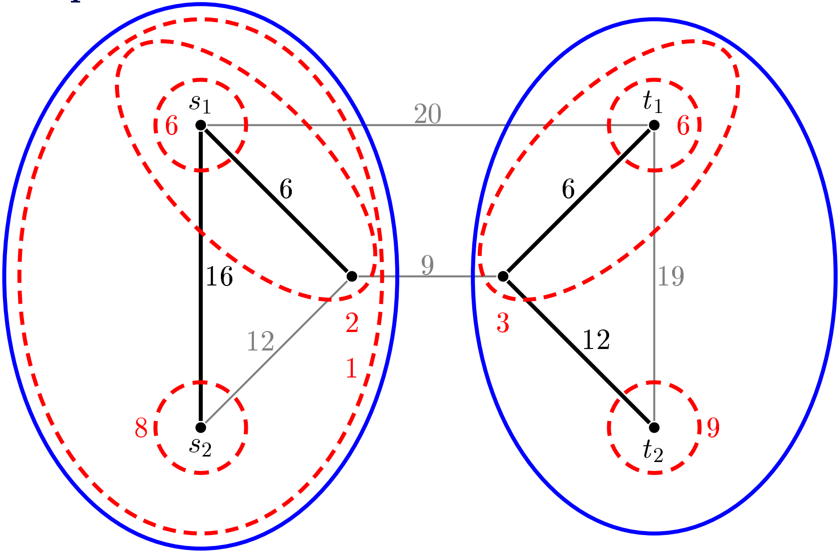
Example



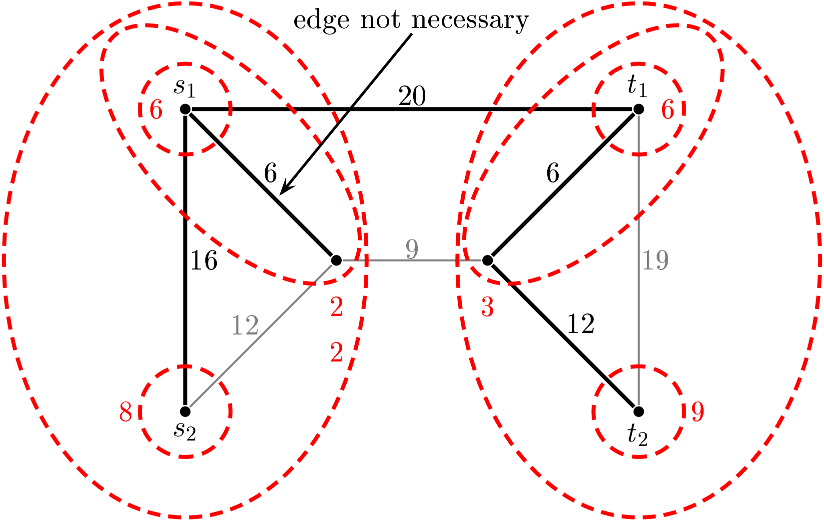
Example



Example

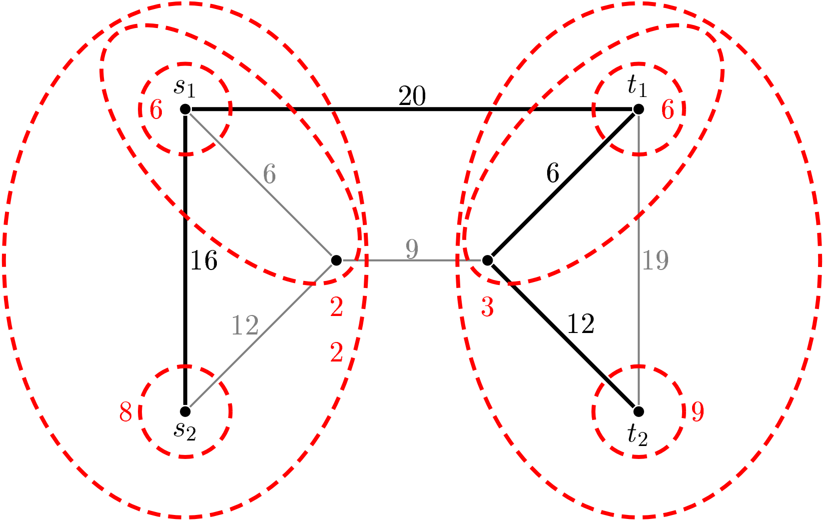


Example



F **at the end of WHILE loop**

Example



Solution F'

Feasibility

Lemma

F' is a feasible solution.

- Let F be the solution at the end of the WHILE loop.
- F is feasible, because there is no violated cut.
- We do not delete necessary edges, hence F' is also feasible. □

Lemma

y is dual feasible, i.e. $\sum_{S:e\in\delta(S)} y_S \leq c_e$ for all $e \in E$.

- Each time that an edge e gets tight (i.e. $\sum_{S:e\in\delta(S)} y_S = c_e$), we add it to F.
- We increase y_S only for violated cuts – not for cuts containing edges of F. □

The main analysis (1)

Lemma

Let y be the dual solution at the end of the algorithm. Then

$$APX = \sum_{e \in F'} c_e \leq 2 \sum_{S \subseteq V} y_S \leq 2 \cdot OPT_f.$$

$$\sum_{e \in F'} c_e \overset{e \text{ tight}}{=} \sum_{e \in F'} \left(\sum_{S : e \in \delta(S)} y_S \right) = \sum_{S \subseteq V} |\delta_{F'}(S)| \cdot y_S \overset{(*)}{\leq} \sum_{S \subseteq V} 2 y_S$$

- Consider any iteration i. Let α be the amount by which the dual variables y_S were increased. We show (*) by proving

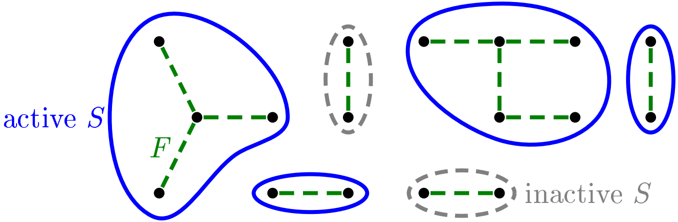
$$\alpha \cdot \sum_{S \text{ active in it.} i} |\delta_{F'}(S)| \leq 2 \cdot \alpha \cdot \#\text{active sets in it.} i$$

The main analysis (2)

- Consider an intermediate iteration i with intermediate F.
- **Remark:** $F' \backslash F$ might contain edges that are added later $F \backslash F'$ might contain edges that are deleted at the end.
- Claim:

$$\sum_{S \text{ active in it.} i} |\delta_{F'}(S)| \leq 2 \cdot \#\text{active sets in iteration } i$$

- Shrink connected components of $F \to H'$ (S becomes node v_S). Nodes v_S steming from active cuts S are active nodes, others are inactive nodes



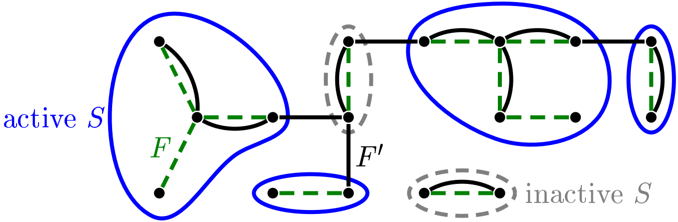
- H' is a forest. Degrees are preserved.

The main analysis (2)

- Consider an intermediate iteration i with intermediate F.
- **Remark:** $F' \backslash F$ might contain edges that are added later $F \backslash F'$ might contain edges that are deleted at the end.
- Claim:

$$\sum_{S \text{ active in it.} i} |\delta_{F'}(S)| \leq 2 \cdot \#\text{active sets in iteration } i$$

- Shrink connected components of $F \to H'$ (S becomes node v_S). Nodes v_S steming from active cuts S are active nodes, others are inactive nodes



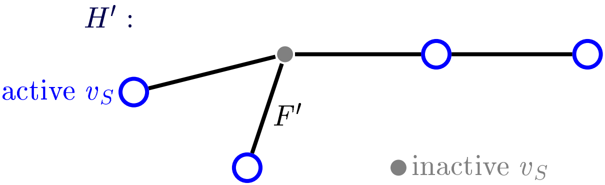
- H' is a forest. Degrees are preserved.

The main analysis (2)

- Consider an intermediate iteration i with intermediate F.
- **Remark:** $F' \backslash F$ might contain edges that are added later $F \backslash F'$ might contain edges that are deleted at the end.
- Claim:

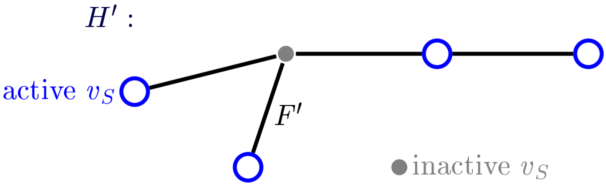
$$\sum_{S \text{ active in it.} i} |\delta_{F'}(S)| \leq 2 \cdot \#\text{active sets in iteration } i$$

- Shrink connected components of $F \to H'$ (S becomes node v_S). Nodes v_S steming from active cuts S are active nodes, others are inactive nodes



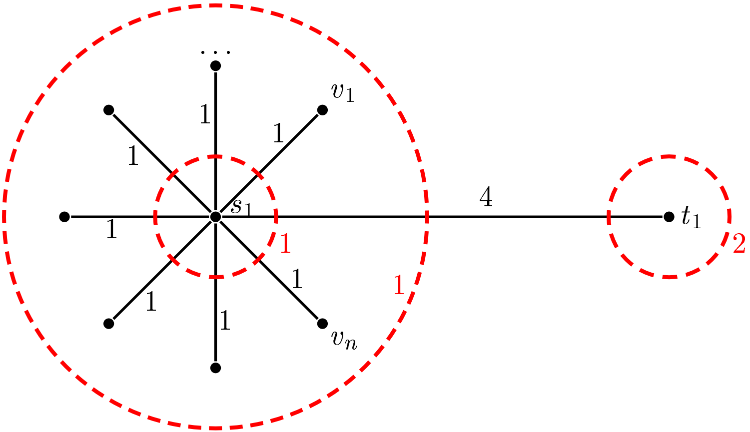
- H' is a forest. Degrees are preserved.

The main analysis (2)



- Consider non-singleton leaf v_S. Edge to v_S was not deleted. Hence $f(S) = 1$. But then S was active (since S is a connected component of F at iteration i).
- Average degree over *all* nodes in a forest is ≤ 2 (since # edges $\leq$ # nodes) and each edge contributes at most 2 to the degrees.
- Inactive nodes are inner nodes of degree ≥ 2, hence average degree of active nodes $\leq$ average degree ≤ 2. □

Deleting redundant edges is crucial



Observation: Without the pruning step at the end of the algorithm, the solution would cost $n + 4$ instead of 4.

Conclusion

Theorem

The primal dual algorithm produces a 2-approximation in time $O(n^2 \log n)$.

Remark: The algorithm works whenever the requirement function $f : 2^V \to \{0,1\}$ is proper, that means

- $f(V) = 0$
- $f(S) = f(V \backslash S)$ (symmetry)
- If $A, B \subseteq V$ are disjoint and $f(A \cup B) = 1$ then $f(A) = 1$ or $f(B) = 1$.

Note: Function f for STEINER FOREST is proper.

State of the art

- There is no $\frac{96}{95}$-approximation algorithm unless $\mathbf{NP} = \mathbf{P}$ (same ratio as for the special case of Steiner tree).
- There is still no better than 2-approximation known.
- The integrality gap of the considered LP is in fact exactly 2.
- There is also no other LP formulation known, which might have a smaller gap.

Part 22
Facility Location

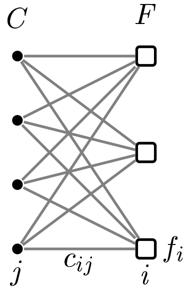
Source: *Approximation Algorithms* (Vazirani, Springer Press)

Facility Location

Problem: FACILITY LOCATION

- Given: Facilities F, cities C, opening cost f_i for every facility i. Metric cost c_{ij} for connecting city j to facility i.
- Find: Set of facilities I and an assignment $\phi : C \to I$ of cities to opened facilities, minimizing the total cost:

$$OPT = \min_{I \subseteq F, \phi: C \to I} \left\{ \sum_{i \in I} f_i + \sum_{j \in C} c_{\phi(j),j} \right\}$$



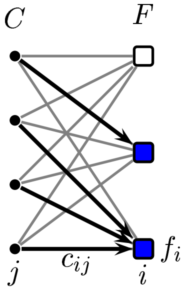
- **Remark:** Without the metric assumption, the problem becomes $\Theta(\log n)$-hard.
- We assume w.l.o.g. $c_{ij}, f_i \in \mathbb{Z}_+$

Facility Location

Problem: Facility Location

- Given: Facilities F, cities C, opening cost f_i for every facility i. Metric cost c_{ij} for connecting city j to facility i.
- Find: Set of facilities I and an assignment $\phi : C \to I$ of cities to opened facilities, minimizing the total cost:

$$OPT = \min_{I \subseteq F, \phi : C \to I} \left\{ \sum_{i \in I} f_i + \sum_{j \in C} c_{\phi(j),j} \right\}$$



- **Remark:** Without the metric assumption, the problem becomes $\Theta(\log n)$-hard.
- We assume w.l.o.g. $c_{ij}, f_i \in \mathbb{Z}_+$

The primal dual pair

Primal LP:

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

$$\begin{array}{rcll} \sum_{i \in F} x_{ij} & \geq & 1 & \forall j \in C \\ x_{ij} & \leq & y_i & \forall i \in F\ \forall j \in C \\ x_{ij} & \geq & 0 & \forall i \in F\ \forall j \in C \\ y_i & \geq & 0 & \forall i \in F \end{array}$$

Dual LP:

$$\max \sum_{j \in C} \alpha_j$$

$$\begin{array}{rcll} \alpha_j & \leq & c_{ij} + \beta_{ij} & \forall i \in F\ \forall j \in C \\ \sum_{j \in C} \beta_{ij} & \leq & f_i & \forall i \in F \\ \alpha_j & \geq & 0 & \forall j \in C \\ \beta_{ij} & \geq & 0 & \forall i \in F\ \forall j \in C \end{array}$$

Intuition:

- α_j is the amount that city j "pays" in total.
- β_{ij} is what city j "pays" to open facility i.

The algorithm - Phase 1:

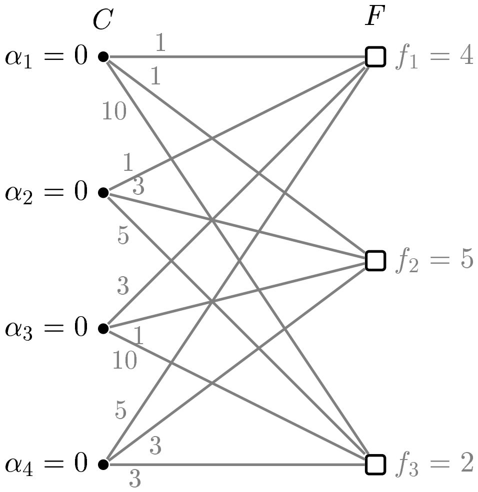
Initially all cities are unconnected
$\alpha := \mathbf{0}, \beta := \mathbf{0}, F_t := \emptyset$
WHILE not all cities are connected DO
    FOR ALL unconnected cities j DO
        Increase α_j (by 1 per time unit)
        For tight edges $\alpha_j = c_{ij} + \beta_{ij}$ increase also β_{ij}
    IF $\sum_j \beta_{ij} = f_i$ (new) THEN
        open facility i temporarily $(F_t := F_t \cup \{i\})$
        FOR ALL cities j where edge (i, j) is tight DO
            connect city to facility i
            facility i is connection witness of j: $w(j) := i$

Phase 2:

Let $H = (F_t, E')$ with $(i, i') \in E'$ if $\exists j \in C : \beta_{ij}, \beta_{i'j} > 0$
Open a maximal independent set $I \subseteq F_t$
FOR ALL $j \in C$ DO
    IF $\exists j \in I : \beta_{ij} > 0$ THEN $\varphi(j) := i$ (j directly conn.)
    ELSE IF $w(j) \in I$ THEN $\varphi(j) := w(j)$ (j directly conn.)
    ELSE $\varphi(j) :=$ a neighbour of $w(j)$ in H (j indir. conn.)

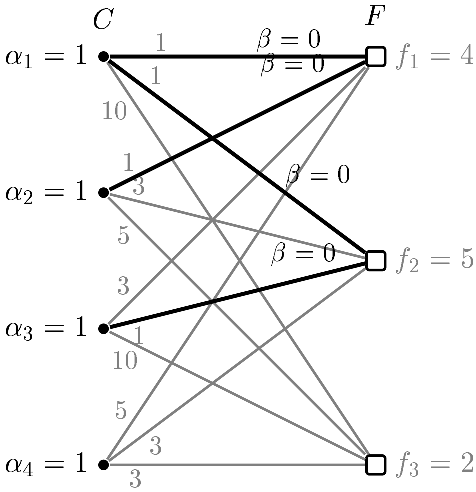
Example:

Phase 1 - Time: 0



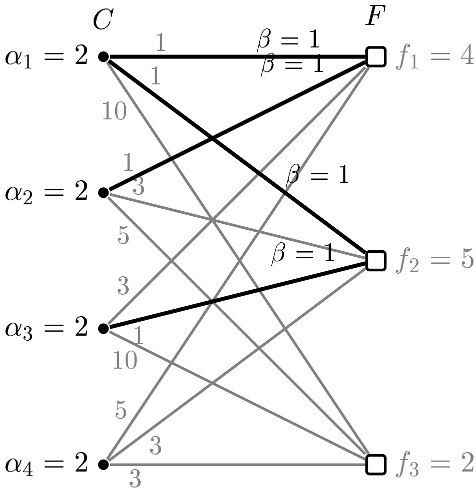
Example:

Phase 1 - Time: 1



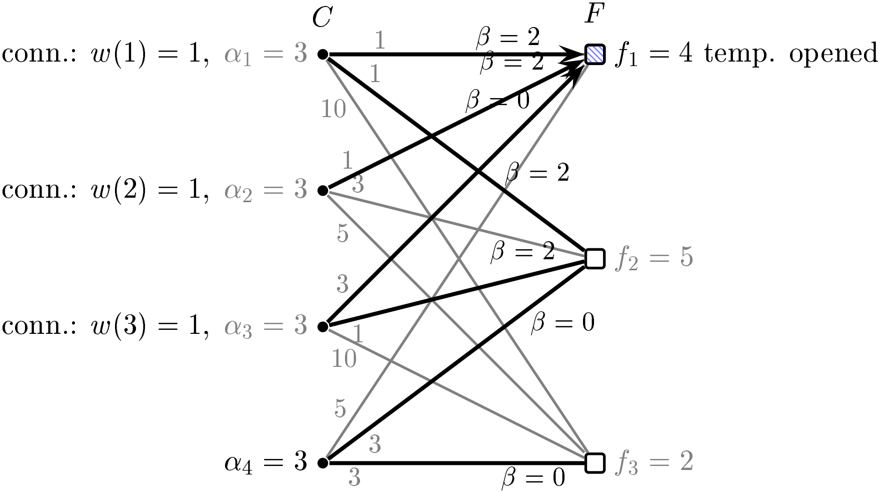
Example:

Phase 1 - Time: 2



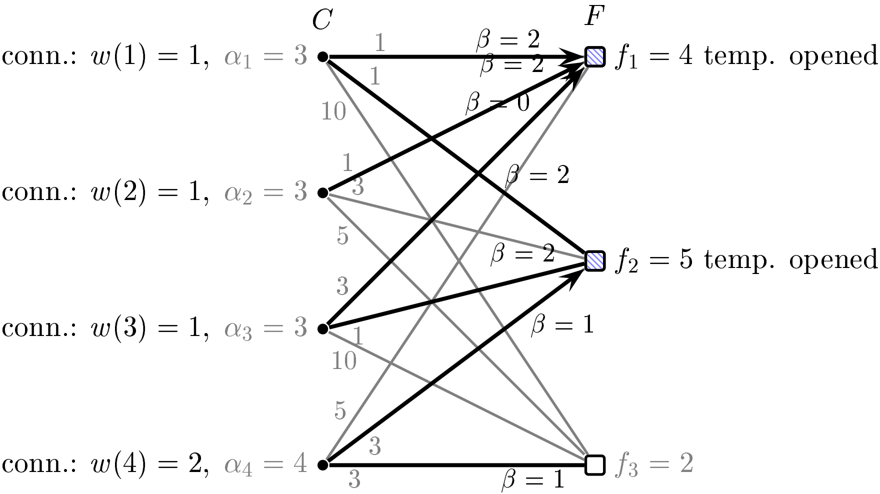
Example:

Phase 1 - Time: 3



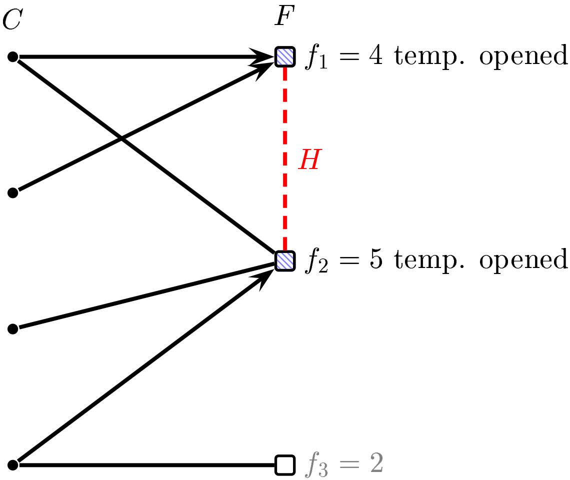
Example:

Phase 1 - Time: 4



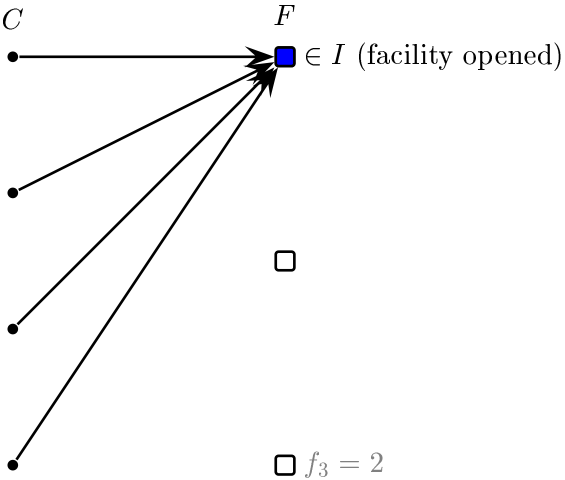
Example:

Phase 2: Graph H



Example:

Phase 2: The solution



Analysis

Theorem

One has $\sum_{j\in C} c_{\varphi(j),j} + \sum_{i\in I} f_i \leq 3\sum_{j\in C}\alpha_j$.

We account the dual "payments"

$$\alpha_j^f := \text{payment for opening} \quad := \begin{cases} \beta_{\varphi(j),j} & \text{if } j \text{ directly connected} \\ 0 & \text{if } j \text{ is indirectly conn.} \end{cases}$$

$$\alpha_j^c := \text{payment for connection} \quad := \begin{cases} c_{\varphi(j),j} & \text{if } j \text{ directly connected} \\ \alpha_j & \text{if } j \text{ is indirectly conn.} \end{cases}$$

Claim: $\alpha_j = \alpha_j^f + \alpha_j^c$.

- For indirectly connected cities: clear
- For directly connected cities: $\alpha_j = c_{\varphi(j),j} + \beta_{\varphi(j),j}$ because edge $(\phi(j), j)$ was tight.

Bounding the opening costs

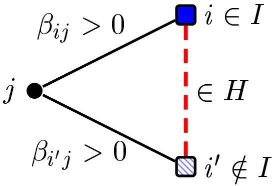
Lemma

The dual prices pay for the opening cost, i.e.

$$\sum_{i\in I} f_i = \sum_{j\in C} \alpha_j^f .$$

- A facility $i \in I$ was temporarily opened because $\sum_j \beta_{ij} = f_i$
- All j with $\beta_{ij} > 0$ must be directly connected to i because: We opened an independent set in H in Phase 2, hence any $i' \in F_t$ with $\beta_{i'j} > 0$ is not in I
- Thus all j with $\beta_{ij} > 0$

$$\sum_{j:\phi(j)=i} \alpha_j^f = \sum_{j:\beta_{ij}>0} \beta_{ij} \overset{i \text{ temp opened}}{=} f_i$$



- The claim follows from

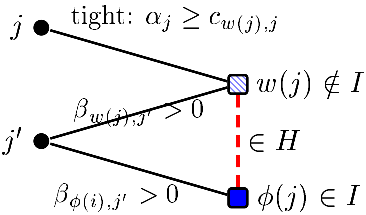
$$\sum_{j\in C} \alpha_j^f = \sum_{i\in I} \sum_{j:\phi(j)=i} \alpha_j^f = \sum_{i\in I} f_i \quad \square$$

Bounding the connection cost

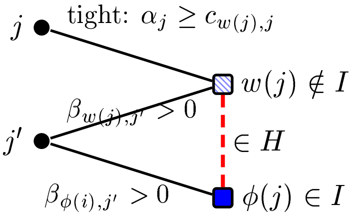
Lemma

For any city $j \in C$ one has $c_{\varphi(j),j} \leq 3\alpha_j^c$.

- If j directly connected, then even $\alpha_j^c = c_{\varphi(j),j}$. Next, suppose j is indirectly connected.
- Then there is an edge $(w(j), \phi(j)) \in H$ (since j was indirectly connected).
- This edge implies that there is a $j' \in C$ with $\beta_{\varphi(j),j'} > 0, \beta_{w(j),j'} > 0$.



Bounding the connection cost (2)



- Event $\beta_{w(j),j} > 0$ only happened if $\alpha_j \geq c_{w(j),j}$. For the same reason: $\alpha_{j'} \geq c_{w(j),j'}$ and $\alpha_{j'} \geq c_{\phi(j),j'}$.
- **Claim $\alpha_j \geq \alpha_{j'}$:** Consider the time t, when $w(j)$ was temporarily opened. Since $w(j)$ is connection witness of j, $\alpha_j \geq t$. At this time t, it was $\beta_{w(j),j'} > 0$ (since if $\beta_{w(j),j'} = 0$ at that time, then $\beta_{w(j),j'} = 0$ forever). At the latest at this time t, also j' was connected and $\alpha_{j'}$ stopped growing. Hence $\alpha_j \geq t \geq \alpha_{j'}$.
- Then

$$c_{\phi(j),j} \overset{\text{metric ineq.}}{\leq} \underbrace{c_{w(j),j}}_{\leq \alpha_j} + \underbrace{c_{w(j),j'}}_{\leq \alpha_{j'} \leq \alpha_j} + \underbrace{c_{\phi(j),j'}}_{\leq \alpha_{j'} \leq \alpha_j} \leq 3\alpha_j = 3\alpha_j^c \qquad \square$$

Conclusion

Theorem

The algorithm produces a 3-approximation in time $O(m \cdot \log(m))$*, where* $m = |C| \cdot |F|$ *is the number of edges.*

State of the art:

Theorem (Byrka '07)

There is a 1.499*-apx for* FACILITY LOCATION.

- The integrality gap for the considered LP lies in $[1.463, 1.499]$.

Theorem

There is no polynomial time 1.463*-apx for* FACILITY LOCATION *unless* $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{O(\log\log n)})$.