# Facilities location

- The *uncapacitated facilities location problem* and *clustering problems* have been studied extensively, like the *k-median problem* and the *k-center problem*.
- Typically, in the given $n$-vertex graph, non-negative edge weights obey the triangle inequality.
- In the $k$-center problem we minimize the maximum distance from a *facility*, whereas in the $k$-median problem we minimize the total sum of distances from facilities.
- In both these problems we do not consider and costs for the facilities, unlike in the uncapacitated facilities location problem.

# Facilities location (cont.)

- The *uncapacitated facility location* problem is a combinatorial optimization problem. It has applications in setting up facility distribution centres.
- In the uncapacitated facility location problem, we have a set of *clients* or *demands* $D$ and a set of *facilities* $F$.
- For each client $j \in D$ and facility $i \in F$, there is a cost $c_{ij}$ of assigning client $j$ to facility $i$.
- Furthermore, there is a cost $f_i$ associated with each facility $i \in F$. The aim is to choose a subset $T' \subseteq F$ so as to minimize the total cost of the facilities in $T'$ and the cost of assigning each client $j \in D$ to some facility in $T'$.

# Facilities location (cont.)

- In other words, we wish to find $T' \subseteq F$ and a function $f$ mapping clients to facilties, such that the following cost is minimised,

$$\sum_{i \in T'} f_i + \sum_{j \in D, f(j) \in T'} c_{f(j)j}$$

  where the first part is called *facility cost* and the second part is called *assignment cost* or *service cost*.
- This is an NP-hard problem and therefore we need to design approximation algorithms.

# Integer programming formulation and its linear programing relaxation

- The integer programming formulation for this problem has decision variables $y_i \in \{0, 1\}$ for each facility $f_i \in F$.
- If we decide to open facility $i$, then $y_i = 1$, and $y_i = 0$, otherwise.
- We also introduce decision variables $x_{ij} \in \{0, 1\}$ for all $i \in F$ and all $j \in D$.
- If we assign client $j$ to facility $i$, then $x_{ij} = 1$ while $x_{ij} = 0$, otherwise.
- The objective function becomes

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

# Integer programming formulation and its linear programing relaxation (cont.)

- We need to make sure that each client $j \in D$ is assigned to exactly one facility. This can be done by stating

$$\sum_{i \in F} x_{ij} = 1$$

- We also need to make sure that the client is assigned to a facility that is open. This can be done by ensuring

$$x_{ij} \leq y_i$$

# Integer programming formulation and its linear programing relaxation (cont.)

- Thus, the integer linear programming (ILP) formulation of the facility location problem can be summarized as follows:

$$\begin{aligned} \textit{minimize} \quad & \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij} & \\ \textit{subject to} \quad & \sum_{i \in F} x_{ij} = 1, & \forall j \in D, \\ & x_{ij} \leq y_i, & \forall i \in F, j \in D, \\ & x_{ij} \in \{0, 1\}, & \forall i \in F, j \in D, \\ & y_i \in \{0, 1\}, & i \in F. \end{aligned}$$

# Integer programming formulation and its linear programing relaxation (cont.)

- The linear programming relaxation (LPR) from the ILP can be obtained by replacing the constraint $x_{ij} \in \{0,1\}$ and $y_i \in \{0,1\}$ with $x_{ij} \geq 0$ and $y_i \geq 0$. Thus, the relaxed linear program (LPR) can be summarized as follows:

$$\textit{minimize} \qquad \textstyle\sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij} \tag{10}$$

- 

$$\textit{subject to} \qquad \textstyle\sum_{i \in F} x_{ij} = 1, \qquad \forall j \in D, \tag{11}$$

$$x_{ij} \leq y_i, \qquad \forall i \in F, j \in D, \tag{12}$$

## Integer programming formulation and its linear programing relaxation (cont.)

$$x_{ij} \geq 0, \qquad \forall i \in F, j \in D,$$
$$y_i \geq 0, \qquad i \in F.$$

## Lower bounding using dual linear programs

- The dual maximizing LP, which we will call DLP (corresponding to the minimizing primal LPR), is used to achieve as high lower bounds as possible for the primal ILP objective function.
- Typically, we may start any algorithm for computing a feasible solution for the ILP by initializing all primal ILP and DLP variables to zeros.
- In the course of the algorithm, primal ILP variables can be assigned only integral values whereas DLP variables can be assigned rational values.
- The respective values of the objectives functions for the ILP and the DLP is the approximation ratio achieved in the developing solution.

## Lower bounding using dual linear programs (cont.)

- We now discuss the formulation of a *dual linear program* (DLP) corresponding the relaxed linear program (LPR) as in [3].
- If we ignore costs of facilities by setting $f_i = 0$ for all $i \in F$, the best strategy would be to open all the facilities and assign each client to its nearest facility. We introduce a variable $v_j$ and set it as $v_j = \min_{i \in F} c_{ij}$ to denote the cost of connecting client $j$ to its nearest facility.
- Observe that a lower bound for the primal integer program's objective function cost in an integral solution (of the ILP), is $\sum_{j \in D} v_j$ therefore; we certainly cannot have a better assignment of facilities.

# Lower bounding using dual linear programs (cont.)

- We can improve this lower bound estimate by considering non-zero facility costs as well, as follows.
- Each facility may be viewed as distributing its cost $f_i$, sharing it apportioned amongst the clients it provides service to, that is, $f_i = \sum_{j \in D} w_{ij}$, where each $w_{ij} \geq 0$.
- A client $j$ needs to pay this share only if it uses facility $i$. So, we can now set $v_j = \min_{i \in F}(c_{ij} + w_{ij})$.
- This can be enforced in a linear programming formulation with constraints $v_j \leq c_{ij} + w_{ij}$ (see inequality 15), for each client $j$ (where $i$ ranges over all facilities), with the objective function maximizing $\sum_{j \in D} v_j$, subject to further inequality 14.

# Lower bounding using dual linear programs (cont.)

- Observe that any feasible solution to this dual linear program therefore has objective function value lower bounding the cost of optimal primal objective function value for the (integral) facility location problem ILP.
- We summarize the dual linear program (DLP) for the primal linear program relaxation (LPR) as:

$$maximize \sum_{j \in D} v_j \tag{13}$$

subject to

$$\sum_{j \in D} w_{ij} \leq f_i, \qquad \forall i, \in F \tag{14}$$

# Lower bounding using dual linear programs (cont.)

$$v_j - w_{ij} \le c_{ij}, \qquad \forall i \in F, j \in D \tag{15}$$

$$w_{ij} \ge 0, \qquad \forall i \in F, j \in D \tag{16}$$

$$v_j \ge 0, \qquad \forall j \in D \tag{17}$$

# The design of the algorithm: Phase I

- In Phase I of the algorithm, we first compute (i) a *maximal dual solution*, (ii) a tentative set $T$ of facilities to be opened, and (iii) a temporary facilities mapping for clients, assigning a *connecting witness* facility for each client.
- In the second Phase II, we restrict the facilities allocated to a subset $T'$ of $T$, reworking some assignments of facilites to clients, albeit some additional cost of connectivity, but well within the 3-factor limit (by virtue of triangle inequality).
- A *maximal dual solution* $(v^*, w^*)$ is such that we cannot further enhance the value of any $v_j^*$ and still work out a feasible assignment to variables $w_{ij}^*$.

# The design of the algorithm: Phase I (cont.)

- For such maximal dual LP solutions, consider the definitions (a) of a client $j$ *neighbouring* a facility $i$ when $v_j^* \geq c_{ij}$ (edges $(i, j)$ are called *tight* edges, and $i$ and $j$ are mutually *neighbours* of each other),
  (b) a *saturated* dual constraint Inequality 14 obeying equality when a facility $i$ becomes *tight* or *paid up*, and
  (c) when it is said that a client $j$ *contributes* to a facility $i$, or $w_{ij} > 0$; such edges $(i, j)$ are called *special* edges.
- Furthermore, recall that the neighbours of a facility $i$ are in the set $N(i)$ of clients, and the neighbours of a client $j$ are in the set $N(j)$ of facilities.
- We sketch the algorithm below as in [3].

## The design of the algorithm: Phase I (cont.)

- The algorithmic issues are as follows, providing intuition about its design, correctness and performance bound.
- Suppose the largest $w^*_{ij}$ satisfying the dual inequality 15 with equality, for some $i \in F$ and some $j \in D$, is non-zero.
- If such a $w^*_{ij}$ is non-zero, we have $v^*_j > c_{ij}$. [So, $(i, j)$ is both *special* as well as *tight*, as per the above definitions.]
- Due to the maximality of $w^*_{ij}$, we can set $v^*_j$ to $c_{ij} + w^*_{ij}$, for the smallest such value over all $i \in F$, keeping the solution feasible for the dual LP.
- Such an $i \in F$ is called *saturated*, and is included in the set $T$ of tentatively opened facilities if inequality 14 is satisfied.
- For such a set $T$ we now argue, as in [3], that every client neighbours a facility in $T$.

# Every client neighbours a facility in $T$

- First we note that the dual solution being maximal, it must be that $v_j^* = min_{i \in F}(c_{ij} + w_{ij}^*)$, for some $i \in F$.
- Otherwise, we must have $v_j^* < min_{i \in F}(c_{ij} + w_{ij}^*)$ for all $i \in F$, in which case we can enhance $v_j^*$, contradicting that we have a maximal dual solution.
- So, now suppose client $j$ has no neighbour in $T$, that is, $v_j < c_{ij}$ for all $i \in T$.
- However, the dual solution being maximal, we must have $v_j^*$ as the smallest of $c_{ij} + w_{ij}^*$ for some $i$, and if all such clients $i$ are outside $T$ then it must have been the case that for all such $i \in T \cap F$, $\sum_{k \in D} w_{ij}^* < f_i$.
- So, $i$ was not selected to be in $T$.

## Every client neighbours a facility in $T$ (cont.)

- In this case however, we can enhance $v_j^*$ and $w_{ij}^*$, without violating dual constraint inequalities 14 and 15. This contradicts that we had a maximal dual solution.
- We therefore conclude that all clients will have a neighbour in $T$ once we have computed a dual maximal feasible solution at the termination of Phase I.

# Summarizing Phase I

- In this phase we maintain feasibility in the dual solution and devleop a maximal dual solution $(v^*, w^*)$ as already defined.
- We set $S = D$, the set of clients, and $T = \phi$, the set of temporarily selected facilities.
- We raise $v_j$'s and $w_{ij}$'s uniformly until either (Case 1) some client $j \in D$ *neighbours* some facility $i \in T$, or (Case 2) some facility $i \in F$ becomes *tight*, or *paid for*, or *saturated*.
- Such clients $j \in S$ as in Case 1, that neighbour some facility $i \in T$ $(v_j \geq c_{ij})$, are removed from $S$.
- Such saturated facilities $i$ as in Case 2 $(\sum_{j \in D} w_{ij} = f_i)$ are moved into the set $T$.

# Summarizing Phase I (cont.)

- Whenever a facility $i$ is added to $T$, we remove all clients in the neighbouring set $N(i)$ of facility $i$ from the set $S$.
- When the set $S$ becomes empty and each client neighbours some facility, and Phase I is terminated.
- More precisely, $v_j$ are increased uniformly for all $j \in S$.
- Once $v_j = c_{ij}$ for some $i$, we increase $w_{ij}$ and $v_j$ uniformly so that the complemetary slackness condition (i) 18, that is, $v_j - w_{ij} = c_{ij}$, resulting from dual constraint 15 will continue to hold.
- However, this raising of $w_{ij}$ will not be necessary when the facility $i$ is already *paid up* or *saturated* or *tight*, as per dual inequality 14 and complementary slackness condition (ii) 19.

# Summarizing Phase I (cont.)

- In this case, all client neighbours $j \in N(i)$ are also removed from $S$. Consequently, we also stop raising $w_{i'j}$ for any $i' \in F$, where $i' \neq i$ for clients $j$ removed from $S$.
- We observe that the maximality of the dual solution ensures that all clients have finally got *tight* edges to some facility, thereby acquiring a *connecting witness* as that facility.
- Some client $j$ may have a connecting witness $i$ with $w_{ij} = 0$. Other edges $(i, j)$ will have $w_{ij}$ non-zero, which we have already named as *special edges*.

# Phase II

- Once the whole set $S$ is exhausted and we have computed a maximal feasible DLP solution $(v^*, w^*)$, we assign facilities from a set $T' \subseteq T$ to clients.
- Note one important point that each client has a neighbouring facility in $T$.
- This is due to the maximality of $(v^*, w^*)$ in the Phase **I** process. We refer to the proof of this fact to section 7.6 of [3], as also elaborated in the above discussion.
- Once $T$ is computed in Phase **I**, a subset $T' \subseteq T$ of facilities is opened by selecting one facility at a time to cover a number of clients.

## Phase II (cont.)

- Whenever any such facility $i$ is moved into $T'$, all other facilities $h \in T$ are also removed from $T$ if both $h$ and $i$ are *contributed* to by some common client $j$, that is, if both $w_{ij}$ and $w_{hj}$ are positive.
- Therefore, finally *opening up* only the facilities surviving in $T'$ will ensure contribution from each client to its respective assigned facility, the only facility to which that client contributes (see the genesis of the dual inequality 14).
- Finally, opened facilities from $T'$ are assigned to all clients as follows.
- If a client $j \in D$ neighbours a facility $i \in T'$ then $j$ is assigned to $i$ and has connection cost $c_{ij}$, lower bounding $v_j^*$, that is, $v_j^* \geq c_{ij}$.

# Phase II (cont.)

- Otherwise, we see due to Lemma 7.13 in [3] (and also as we discuss below) that although $j$ does not have a neighbour in $T'$, there is a facility $i \in T'$ such that $v_j^* \geq \frac{c_{ij}}{3}$.

## The analysis: Relaxed complementary slackness

- The 3-factor approximation result of Theorem 7.14 of [3] follows from Lemma 7.13 of [3]; we explain these results in more detail now.
- We know that even if client $j \in D$ neighbours no facility in $T'$, it does neighbour a facility in $T$, as argued above, by virtue of the method used to construct the set $T$.
- It turns out that such a client $j$ neighbours some saturated facility $h \notin T'$ such that some other client contributed to both $h$ and some facility $i \in T'$.
- The client $j$ must have neighboured some $h \in T \setminus T'$ in the algorithm's execution when increasing $v_j$ was stopped; it is known that $j$ does not neighbour any facility in $T'$.

# The analysis: Relaxed complementary slackness (cont.)

- How was $h \in T$ excluded from being in $T'$? Another client $k$ was there that *contributes* to both $h$ and another facility $i \in T'$.
- It is now our goal to show that the cost $c_{ij}$ of assigning $j$ to $i$ is at most $3v_j$.
- In this context, view the client-facility pairs $(j, h)$,$(k, h)$ and $(k, i)$, and the path along these three edges from client $j$ to facility $i$, through facility $h$ and client $k$.
- Clearly the cost of connecting $j$ to $i$ is $c_{ij} \leq c_{hj} + c_{hk} + c_{ik}$, by triangle inequality.
- To show $v_j \geq \frac{c_{ij}}{3}$, it is enough to show the $v_j$ is at least as large as each of $c_{hj}, c_{hk}$, and $c_{ik}$.
- First, we note that $v_j \geq c_{hj}$ as $j$ neighbours $h$.

# The analysis: Relaxed complementary slackness (cont.)

- Second, we also show below that $v_j \geq v_k$. Therefore, $v_k$ being at least as large as both of $c_{ik}$ and $c_{hk}$ (since $k$ contributes to and thus also neighbours both $h$ and $i$), we conclude that $v_j$ at least as large as all the three of $c_{hj}, c_{hk}$, and $c_{ik}$.
- Now we show that $v_j \geq v_k$. We know that $v_j$ stopped increasing when it neighboured a facility in $T$.
- Since $j$ neighbours $h \in T$, we understand that $h$ must have already been in $T$ or must have been included in $T$ when $v_j$ stopped increasing.
- Now since $k$ *contributes* to $h$, and therefore also neighbours $h$, $v_k$ too must have stopped increasing before or when $v_j$ stopped increasing.

## The analysis: Relaxed complementary slackness (cont.)

- Furthermore, since dual variables are increased uniformly in the algorithm, we have $v_j \geq v_k$.
- Now that we have explained how $v_j \geq \frac{c_{ij}}{3}$ (Lemma 7.13 of [3]), we establish the 3-factor bound of Theorem 7.14 of [3] as follows.
- The cost $\sum_{i \in T'} f_i$ of opening facilities has now been shown to be apportioned to clients $j$ that contribute to the respective finally opened facilities $f(j)$ in $T'$ to which $j$ is connected; note that $i \in T'$ means saturating inequality 14 is satisfied as an equality for facility $i$.

# The analysis: Relaxed complementary slackness (cont.)

- So, the total facility opening cost $\sum_{i \in T'} f_i = \sum_{i \in T'} \sum_{j \in A(i)} w_{ij}$, is apportioned to neighbouring clients assigned to facilities $i \in T'$, where $A(i)$ is the set of these clients.
- The connection costs for these clients is $\sum_{i \in T'} \sum_{j \in A(i)} c_{ij}$.
- Summing these two costs for neighbouring clients of facilities in $T'$ gives $\sum_{i \in T'} \sum_{j \in A(i)} (w_{ij} + c_{ij}) = \sum_{i \in T'} \sum_{j \in A(i)} v_j$, because the dual inequalities 15 for all $i \in T'$ attain equality.
- Clients $j \in D$, not neighbouring facilities in $T'$ have connection costs assigned to respective facilities $f(j) \in T'$ such that $c_{f(j)j} \leq 3v_j$, as established already, resulting in a total cost of at most $\sum_{j \in D \setminus \cup_{i \in T'} A(i)} c_{f(j)j} \leq 3 \sum_{j \in D \setminus \cup_{i \in T'} A(i)} v_j$.

# The analysis: Relaxed complementary slackness (cont.)

- So, the total cost including costs apportioned to neighbouring clients of $T'$ and connection costs of clients not neighbouring facilities in $T'$ add up to $3\sum_{j\in D} v_j \leq 3 \times OPT$.
- The 3-factor bound thus follows.
- Note that even if we took three times the cost $\sum_{i\in T'}\sum_{j\in A(i)}(w_{ij}+c_{ij}) = \sum_{i\in T'}\sum_{j\in A(i)} v_j$, of directly connected clients, we still get the same bound of $3\sum_{j\in D} v_j$ for the total cost (Exercise 7.8 in [3]).

## More on relaxed complementary slackness

- We will now view the same algorithm (giving 3-factor approximation) using *relaxed complementary slacknesss conditions* as in [2].
- See inequalities 14 and 15. We continue with the same notations.
- Consider the primal and dual complementary slackness conditions (implications) (i)-(iv) here.

# More on relaxed complementary slackness (cont.)

$$(i) \quad x_{ij} > 0 \to v_j - w_{ij} = c_{ij}, \tag{18}$$

$$(ii) \quad y_i > 0 \to \sum_{j \in D} w_{ij} = f_i, \tag{19}$$

$$(iii) \quad \forall j \in D \;\; y_j > 0 \to \sum_{i \in F} x_{ij} = 1 \tag{20}$$

$$(iv) \quad \forall i \in F \;\; \forall j \in D \;\; w_{ij} > 0 \to y_i = x_{ij} \tag{21}$$

- Suppose the optimal LPR solution is integral. Then, each open facility is tight, that is, its cost is fully paid up as per primal slackness condition (ii), Implication 19.

## More on relaxed complementary slackness (cont.)

- Now consider the (dual) slackness condition (iv) $w_{ij} > 0 \rightarrow y_i = x_{ij}$ (Implication 21); given that a client $j \in D$ is not connected to open facility $i \in F$, that is $y_i = 1 \neq x_{ij}$, it follows that $w_{ij} = 0$, indicating $j$ does not contribute to any facility apart from the one to which it is connected.
- Also, by primal slackness condition (i), Implication18, for any cient $j$ connected to an open facility $i$, we have $v_j = c_{ij} + w_{ij}$.
- So, we interpret the total price $v_j$ paid by client $j$ as $c_{ij}$ as going to the connection from $j$ to $i$, and $w_{ij}$ as the contribution of $j$ to $i$.

## More on relaxed complementary slackness (cont.)

- Now we observe that by relaxing the primal complementary slackness conditions suitably, we may limit the objective function value of the ILP solution to within thrice that of the DLP as follows.
- Assume that $f(j) \in F$ is the facility to which client $j \in D$ is connected. The cost in the ILP is $\sum_{j \in D} c_{f(j)j} + \sum_{i \in T'} f_i$, where $T' \subseteq F$ is the final set of opened facilities.
- So, by altering the primal slackness conditions (i) and (ii) respectively, as
  (I) $\frac{1}{3} c_{f(j)j} \leq v_j - w_{f(j)j} \leq c_{f(j)j}$ for all $j \in D$, and
  (II) $\frac{1}{3} f_i \leq \sum_{j:f(j)=i} w_{ij} \leq f_i$,
  we can ensure factor three approximation because the $w_{ij}$ terms would cancel out on summing the primal objective

## More on relaxed complementary slackness (cont.)

function as seen in the first inequalities in conditions (I) and (II).

- We will however not use these slackness conditions. We will consider two cases of assigning clients to facilities, and call them *direct* and *indirect* assignments, as presented in [2].
- This is done to improve the approximation factor though not in the worst-case, as we present here in Theorem 5.
- However, this analysis is important as we will use the same technique for proving approximation bounds for another important optimization problem, the $k$-median problem in Chapter **??**.

# More on relaxed complementary slackness (cont.)

- For indirect assignment of a client $j$ to a facility $i$, we have $w_{ij} = 0$, whence the condition (I) becomes
  (I')$\frac{1}{3}c_{f(j)j} \leq v_j \leq c_{f(j)j}$.
- Primary complementary slackness condition (i) is preserved for a directly connected facility $j$ such that $x_{ij} > 0$ implies $v_j - w_{f(j)j} = c_{f(j)j}$, and condition (ii) is maintained, rather than condition (II), so that we have
  (II') $\sum_{j \in D} w_{ij} = f_i$,
  where such clients pay for the facilities costs.
- Our algorithm must achieve the raising of dual variables $v_j$, paying for costs of opening facilities as well as connecting clients to facilities maintaining conditions (I') and (II').

# More on relaxed complementary slackness (cont.)

- So, let us view $v_j$ as $v_j^f + v_j^c$, where $v_j^f$ is the facilities part of the cost and $v_j^c$ is the connection cost.
- For an indirectly connected client $j$ therefore, we wish to enforce $v_j^f = 0$ and $v_j^c = v_j$.
- For a directly connected client $j$, we know from the complementary slackness condition (i) that $v_j - w_{f(j)j} = c_{f(j)j}$, where $v_j^f = w_{f(j)j}$ and $v_j^c = c_{ij}$, and by the complementary slackness condition (ii) that $\sum_{j \in D} w_{ij} = f_i$.
- In this context we have two observations.

**Observation**

*Observation 2:*
$\sum_{(i,j):j \in N(i)} v_j^f = f_i$.

# More on relaxed complementary slackness (cont.)

- Clearly, here $j$ *neighbours* $i$ or $j \in N(i)$ and $j$ *contributes* to $i$.
- Note that $v_j^f = w_{ij}$ for the case where $j$ is directly connected to $i$ and zero, otherwise.
- Furthermore, we can deduce

**Observation**

*Observation 3:*
$\sum_{i \in T'} f_i = \sum_{j \in D} v_j^f$.

- Here, $T'$ is the set of finally opened facilities and $D$ is the set of clients.

## More on relaxed complementary slackness (cont.)

- Now we claim the following lemma.

**Lemma**

*Lemma 4:*
$c_{ij} \leq 3v_j^c$ *for* $j \in D$ *assigned indirectly to* $i \in T'$.

- We have already discussed the proof of Lemma 4 in Subsection 6 of Section 14. The theorem follows. This theorem is also established in Subsection 6 of Section 14.

**Theorem**

*Theorem 5:*
$\sum_{j \in D, i \in F} x_{ij} c_{ij} + 3 \sum_{i \in F} f_i y_i \leq 3 \sum_{j \in D} v_j$, *where the variables are from the primal and dual solutions computed by the algorithm.*

# More on relaxed complementary slackness (cont.)

**Proof.**

For a directly connected client $j$, $c_{ij} = v_j^c \leq 3v_j^c$, where $j$ is assigned to $i = f(j)$. Lemma 4 further asserts $\sum_{j\in D, i\in F} c_{ij}x_{ij} \leq 3\sum_{j\in D} v_j^c$, even considering indirectly connected clients. Now adding $3\sum_{i\in T'} f_i = 3\sum_{j\in D} v_j^f$ from Observation 3, concludes the proof of this theorem since $\sum_{j\in D, i\in F} c_{ij}x_{ij} \leq 3\sum_{j\in D} v_j^c$ and $3\sum_{i\in T'} f_i = 3\sum_{j\in D} v_j^f$ imply $\sum_{j\in D, i\in F} x_{ij}c_{ij} + 3\sum_{i\in F} f_i y_i \leq 3\sum_{j\in D}(v_j^c + v_j^f) = 3\sum_{j\in D} v_j$. □

- Here, $OPT \geq \sum_{j\in D} v_j$, thereby implying the 3-factor approximation bound, based on Theorem 5.

## References

Thomas H Cormen et al. *Introduction to algorithms*. MIT press, 2009.

Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.