Topics in Theoretical Computer Science	April 29, 2014

Lecture 10

Lecturer: Ola Svensson *Scribes: Romain Edelmann & Florian Tramèr*

1 Introduction

In the previous lectures, we have introduced linear programming and shown how to devise and analyse approximation algorithms for hard problems based on extreme point solutions. In this lecture, we introduce the important concept of the **dual** of a linear program. We will see that duality can serve as a useful tool to make use of the expressive power of linear programs, without actually having to solve them. We will mostly use duality in order to analyse the performance guarantees of approximation algorithms.

The approximation algorithms based on linear programs we have considered so far usually started from an optimal solution to the LP, which thus must first be found, and then proceeded to round this solution to an integer solution. In this lecture, we will consider more efficient algorithms, using mostly greedy methods, which can also be shown to achieve good approximation guarantees.

2 Linear Programming Duality

2.1 Intuition

Consider the following simple linear program:

$$\begin{aligned} \textbf{Minimize: } & 7x_1 + 3x_2 \\ \textbf{Subject to: } & x_1 + x_2 \geq 2 \\ & 3x_1 + x_2 \geq 4 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Let OPT denote the optimal solution to this LP. Since this is a minimization problem, to find an upper bound on OPT, we can simply look for a feasible solution to the LP. For instance, $x_1 = 1, x_2 = 1$ is a feasible solution with objective value 10.

In order to find a lower bound on OPT, we consider the constraints of the LP which we know to be satisfied for a feasible solution. From the first constraint, we get that:

$$7x_1 + 3x_2 \geq x_1 + x_2 \geq 2.$$

Thus, we can conclude that OPT must be at least 2. Similarly, from the second constraint, we find a bound of OPT ≥ 4. Taking this idea further, we consider linear combinations of the constraints of the LP. For instance, we have that:

$$7x_1 + 3x_2 = (x_1 + x_2) + 2 \cdot (3x_1 + x_2) \geq 2 + 2 \cdot 4 = 10.$$

By considering such linear combinations of the constraints, we can construct the **dual linear program** corresponding to the original **primal** program. For each constraint of the primal program, we associate a dual variable y_i, representing the weight associated to the constraint in the linear combination. Since we are interested in lower bounding the primal OPT, these variables y_i must be constrained such that the linear combination of primal constraints doesn't exceed the primal objective function. Furthermore, we will seek to maximize the objective value of our linear combination to get as good a lower bound on OPT as possible. From all this, we get the following dual linear program:

$$\begin{aligned} \textbf{Maximize: } & 2y_1 + 4y_2 \\ \textbf{Subject to: } & y_1 + 3y_2 \le 7 \\ & y_1 + y_2 \ge 3 \\ & y_1, y_2 \ge 0 \end{aligned}$$

In this case, the optimal solution to the primal and dual LPs coincide. We will later on encounter the strong duality theorem, which states that whenever both the primal and dual LPs are feasible, their optimums coincide.

2.2 General Case

In this section we will show how to derive the dual program of a general linear minimization problem. The analogue for maximization problems follows directly. Consider the following general primal LP with n variables x_i for $i \in [1, n]$ and m constraints:

$$\begin{aligned} \textbf{Minimize: } & \sum_{i=1}^{n} c_i x_i \\ \textbf{Subject to: } & \sum_{i=1}^{n} A_{ji} x_i \ge b_j \quad \forall j = 1, \ldots, m \\ & x \ge 0 \end{aligned}$$

Then, the dual program has m variables y_j for $j \in [1, m]$ and n constraints:

$$\begin{aligned} \textbf{Maximize: } & \sum_{j=1}^{m} b_j y_j \\ \textbf{Subject to: } & \sum_{j=1}^{m} A_{ji} y_j \le c_i \quad \forall i = 1, \ldots, n \\ & y \ge 0 \end{aligned}$$

One can verify that if we take the dual of the dual problem, we get back to the primal problem, as we should expect. We note that finding the dual of a linear program is essentially a mathematical technicality, which could easily be automated for large LP instances.

2.3 Duality Theorems

We now present two of the main results on dual linear programs, known as the weak and strong duality theorems. Again, we will only be concerned with primal minimization problems.

Theorem 1 (Weak Duality) *If x is primal-feasible (meaning that x is a feasible solution to the primal problem) and y is dual-feasible, then*

$$\sum_{i=1}^{n} c_i x_i \ge \sum_{j=1}^{m} b_j y_j.$$

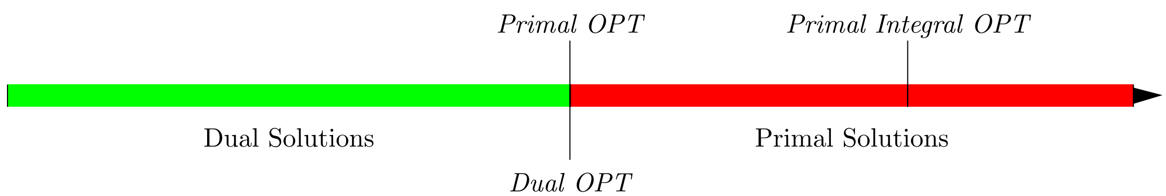
Proof By simple arithmetic, using the fact that both x and y are feasible solutions and thus that all constraints are satisfied, we get:

$$\sum_{j=1}^{m} b_j y_j \leq \sum_{j=1}^{m} \sum_{i=1}^{n} A_{ji} x_i y_j = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} A_{ji} y_j \right) x_i \leq \sum_{i=1}^{n} c_i x_i.$$

■

This theorem is the main result we will use with respect to analysing approximation algorithms. What the theorem tells us is that any dual-feasible solution is a lower bound to any primal-feasible solution. In particular, any dual-feasible solution is a lower bound to the optimal primal solution, which is itself a lower bound to the optimal primal integral solution.

Thus, instead of analysing the approximation ratio of a primal integral solution by bounding it in terms of the optimal primal solution, we may also bound it in terms of **any** dual solution, which might be much simpler. The following figure illustrates how solutions to the primal and dual problems are distributed.



Note that we anticipated the statement of the strong duality theorem, by having the optimal primal and optimal dual solutions coincide.

Theorem 2 (Strong Duality) *If x is an optimal primal solution and y is an optimal dual solution, then*

$$\sum_{i=1}^{n} c_i x_i = \sum_{j=1}^{m} b_j y_j.$$

Furthermore, if the primal is unbounded (respectively infeasible), then the dual is infeasible (respectively unbounded).

For a proof of the theorem, we suggest the classic text on linear programming by Vanderbei [4].

2.4 Applications to Set Cover

Recall the Set Cover problem. Given a universe $\mathcal{U} = \{e_1, \dots, e_m\}$, a family $\mathcal{T}$ of subsets of $\mathcal{U}$ and a cost function $C : \mathcal{T} \to \mathbb{R}$, we want to find a subfamily of $\mathcal{T}$ of sets whose union is $\mathcal{U}$ and whose combined weight is minimized. We also recall the corresponding primal LP:

$$\begin{aligned} \textbf{Minimize: } & \sum_{S \in \mathcal{T}} C(S) \cdot x_S \\ \textbf{Subject to: } & \sum_{\substack{S \in \mathcal{T}: \\ e \in S}} x_S \geq 1 \quad \forall e \in \mathcal{U} \\ & x_S \geq 0 \qquad \forall S \in \mathcal{T} \end{aligned} \tag{1}$$

To derive the dual LP, we introduce a variable y_e for every element $e \in \mathcal{U}$. We get:

$$\begin{aligned} \textbf{Maximize:} \quad & \sum_{e \in \mathcal{U}} y_e \\ \textbf{Subject to:} \quad & \sum_{e \in S} y_e \leq C(S) \quad && \forall S \in \mathcal{T} \\ & y_e \geq 0 && \forall e \in \mathcal{U} \end{aligned} \tag{2}$$

2.4.1 Set Cover via Dual Fitting

We will analyse the following greedy algorithm, making use of the popular idiom 'the most bang for the buck'.

Algorithm 1

$C \leftarrow \emptyset$
while $C \neq \mathcal{U}$ **do**
- Pick the set $S \in \mathcal{T}$ which minimizes $\dfrac{C(S)}{|S \cap (\mathcal{U} \setminus C)|}$
- Add S to C

end while

Consider the algorithm step where set S gets added to C. For each newly added element $e \in S \cap (\mathcal{U} \setminus C)$, we define:

$$\text{price}(e) = \frac{C(S)}{|S \cap (\mathcal{U} \setminus C)|}.$$

This can be seen as having the cost of the set S distributed evenly over all newly added elements from S. From this, the cost of the solution returned by the algorithm is simply

$$\sum_{e \in \mathcal{U}} \text{price}(e).$$

Suppose that setting $y_e = \text{price}(e)$ would yield a feasible solution to the dual LP. This would imply, by the weak duality theorem, that the cost of our returned solution is upper bounded by the optimal primal solution and must thus be optimal. This would imply that we have solved Set Cover exactly, which is only possible if $\mathbf{P} = \mathbf{NP}$. We will thus scale down our dual solution a little in order for it to become feasible. This technique is called 'dual fitting'.

Definition 3 *Let* $H_n = \sum_{i=1}^{n} \frac{1}{i}$ *denote the nth harmonic number.*

Theorem 4 *Let y be defined by* $y_e = \dfrac{\textit{price}(e)}{H_n}$, $\forall e \in \mathcal{U}$. *Then y is a feasible dual solution. This implies that our greedy algorithm is a H_n approximation algorithm for Set Cover.*

Proof To show that y is a feasible dual solution, we will show that it satisfies all the constraints of the dual LP (2). Trivially, all the constraints of the form $y_e \geq 0$ are satisfied. We thus only need to verify the constraints of the form

$$\sum_{e \in S} y_e \leq C(S), \quad S \in \mathcal{T}.$$

Fix some arbitrary $S \in \mathcal{T}$ and let $S = \{e_k, e_{k-1}, \ldots, e_1\}$. We need to show that:

$$\sum_{e \in S} y_e = \sum_{i=1}^{k} \frac{\text{price}(e_i)}{H_n} \leq C(S).$$

Let the elements of S be sorted in the order in which they were covered by the greedy algorithm (e_k is the first element from S which was covered and e_1 was the last).

Claim 5 *price*$(e_i) \leq \frac{C(S)}{i}$, $\forall e_i \in S$.

Proof When e_k was covered by the algorithm, no elements from S were yet covered. If set S was picked at that point, we would have:

$$\text{price}(e_k) = \frac{C(S)}{|S \cap (\mathcal{U} \setminus C)|} = \frac{C(S)}{|S|} = \frac{C(S)}{k}.$$

Since our algorithm picks the set which minimizes the ratio $\frac{C(S)}{|S \cap (\mathcal{U} \setminus C)|}$, the price associated to element e_k is upper bounded by $\frac{C(S)}{k}$.

Now consider the step of the algorithm where element e_i was covered. Since we could have multiple elements of S covered in the same step, let e_j be the first element of S covered in this step (obviously we have $j \geq i$). If set S was picked by the algorithm, we would assign

$$\text{price}(e_i) = \frac{C(S)}{|S \cap (\mathcal{U} \setminus C)|} = \frac{C(S)}{j} \geq \frac{C(S)}{i}.$$

Again, since the algorithm picks the set with minimal ratio, we can upper bound the price assigned to element e_i by $\frac{C(S)}{i}$. ■

From this claim, the proof of the theorem follows directly since

$$\sum_{e \in S} y_e = \sum_{i=1}^{k} \frac{\text{price}(e_i)}{H_n} \leq \frac{C(S)}{H_n} \sum_{i=1}^{k} \frac{1}{i} = C(S) \frac{H_k}{H_n} \leq C(S). \tag{3}$$

Thus, we know that our algorithm outputs a solution to the primal problem, a fraction at most H_n larger than some feasible dual solution. Using the weak duality theorem, we know that our solution is also at most H_n larger than the optimal primal solution. This implies the H_n-approximation ratio for our greedy algorithm. ■

2.4.2 Set Cover via Primal-Dual

We will now present and analyse a different type of algorithm for Set Cover, based on the 'primal-dual' approach.

- We start with an infeasible primal solution x (usually $x = 0$) and a feasible dual solution y (also usually $y = 0$).
- We maintain y to be a feasible solution throughout the execution of the algorithm
- In each step, we attempt to increment y, so as to 'pay' for a change in x.
- When the algorithm terminates, we want a feasible integral solution x, hopefully not much more expensive than the dual solution y.

For Set Cover, our algorithm is as follows:

Algorithm 2

$\mathcal{C} \leftarrow \emptyset$ (sets used)
$\mathcal{V} \leftarrow \emptyset$ (covered elements)

while $\mathcal{V} \neq \mathcal{U}$ **do**
- Pick any uncovered $e \in \mathcal{U} \setminus \mathcal{V}$.
- Increment y_e until some dual constraint becomes tight, formally $\sum_{e \in S} y_e = C(S)$ for some $S \notin \mathcal{C}$.
- Set $x_S = 1$ for all sets S whose corresponding dual constraint is tight and add these sets to $\mathcal{C}$.
- Set $\mathcal{V} = \bigcup_{S \in \mathcal{C}} S$

end while

Definition 6 *Let f be defined as the maximum number of sets $S \in \mathcal{T}$, an element is contained in. Formally,*

$$f = \max_{e \in \mathcal{U}} |\{S \in \mathcal{T} : \ e \in S\}|$$

Theorem 7 *The solution returned by Algorithm 2 has cost at most $f \cdot OPT$, where OPT is the value of the optimal solution to the primal Set Cover LP (1).*

Proof Let $\mathcal{C}$ be the family of sets returned by Algorithm 2. As any $S \in \mathcal{C}$ has a tight corresponding dual constraint, we have that:

$$\text{COST} = \sum_{S \in \mathcal{C}} C(S) = \sum_{S \in \mathcal{C}} \left(\sum_{e \in \mathcal{S}} y_e \right) \leq \sum_{S \in \mathcal{T}} \left(\sum_{e \in \mathcal{S}} y_e \right) = \sum_{e \in \mathcal{U}} y_e \left(\sum_{S: \ e \in S} 1 \right) \leq f \cdot \sum_{e \in \mathcal{U}} y_e \leq f \cdot \text{OPT}, \quad (4)$$

where the last inequality follows from the weak duality theorem. ■

3 Exercises

Exercise 1 Design and analyse an algorithm for Set-Cover with an approximation guarantee of H_k, where k is the largest cardinality of a set $S \in \mathcal{T}$.

Solution 1 From the proof of Theorem 4, we can see that if we define y by $y_e = \frac{\text{price}(e)}{H_k}$, y is a feasible solution to the dual problem. Indeed, for an arbitrary set S, we can rewrite equation (3) as

$$\sum_{e \in S} y_e = \sum_{i=1}^{|S|} \frac{\text{price}(e_i)}{H_k} \leq \frac{C(S)}{H_k} \sum_{i=1}^{|S|} \frac{1}{i} \leq \frac{C(S)}{H_k} \sum_{i=1}^{k} \frac{1}{i} = C(S) \frac{H_k}{H_k} = C(S).$$

Thus, Algorithm 1 is actually a H_k-approximation algorithm for Set Cover.

Exercise 2 Prove that if $\begin{cases} x \text{ is OPT primal} \\ y \text{ is OPT dual} \end{cases}$, then

$$\begin{cases} c_i = \sum_{j=1}^{m} A_{ji} y_j \quad \textbf{or} \quad x_i = 0, \qquad \forall i = 1, \dots, n \qquad \textbf{(a)} \\ \\ b_j = \sum_{i=1}^{n} A_{ji} x_i \quad \textbf{or} \quad y_j = 0, \qquad \forall j = 1, \dots, m \qquad \textbf{(b)} \end{cases}$$

This fact, known as **complementary slackness** states that if a variable is 'used' by an optimal solution, the corresponding dual constraint is necessarily tight.

Furthermore, conditions (a) and (b) are necessary and sufficient conditions for x and y to be optimal solutions to the primal and dual LPs.
(*Hint: Use the proof of the weak duality theorem, along with the result of the strong duality theorem.*)

Solution 2 Let x be the optimal primal solution. From the weak duality theorem proof, we have that:

$$\sum_{j=1}^{m} b_j y_j \leq \sum_{i=1}^{n} \left(\sum_{j=1}^{m} A_{ji} y_j \right) x_i.$$

Now, suppose that condition (a) doesn't hold for some x_i. Without loss of generality, we will assume that:

$$x_1 \neq 0 \quad \text{and} \quad c_1 > \sum_{j=1}^{m} A_{j1} y_j.$$

Then, we get that:

$$\sum_{j=1}^{m} b_j y_j \leq \left(\sum_{j=1}^{m} A_{j1} y_j \right) x_1 + \sum_{i=2}^{n} \left(\sum_{j=1}^{m} A_{ji} y_j \right) x_i < c_1 x_1 + \sum_{i=2}^{n} c_i x_i < \sum_{i=1}^{n} c_i x_i.$$

This result contradicts the statement of the strong duality theorem, implying a contradiction in our assumption that condition (a) doesn't hold for some x_i.

The proof of the analogue result that condition (b) holds for an optimal dual solution y is exactly the same, and not displayed here.

Exercise 3 Suppose you have an **optimal** primal solution x to the Set-Cover LP (1). Consider an algorithm for Set-Cover, which picks all sets S with $x_S > 0$. Show that this is a f-approximation algorithm.

Solution 3 Since x is primal optimal, from exercise 2 we know that all non zero variables x_S have an associated tight dual constraint (complementary slackness). Furthermore, we have seen through the analysis of Algorithm 2, that taking all sets S with an associated tight dual constraint leads to an f-approximation.

Formally, let $\mathcal{C}$ be the family of sets returned by the algorithm, $\mathcal{C} = \{S \in \mathcal{T} : \ x_S > 0\}$. By complementary slackness, we have that:

$$\sum_{e \in S} y_e = C(S), \quad \forall S \in \mathcal{C}.$$

Then, as developed in equation (4), this implies that

$$\text{COST} = \sum_{S \in \mathcal{C}} C(S) = \sum_{S \in \mathcal{C}} \left(\sum_{e \in \mathcal{S}} y_e \right) \leq f \cdot \text{OPT},$$

which concludes the proof.

4 The Metric Uncapacitated Facility Location Problem

4.1 Problem Definition

In this problem we are given a set D of clients[1] and a set F of possible facilities. Each facility $i \in F$ has an opening cost f_i. Also, connecting a client $j \in D$ to a facility $i \in F$ incurs a cost c_{ij}. The goal is to find a subset of facilities $I \subseteq F$ that minimizes the overall cost of openings and the costs of connecting every client to the cheapest facility.

As an additional condition, we have that c_{ij} forms a metric, meaning that for any $i, i' \in F$ and $j, j' \in D$, the following holds:

$$c_{ij} \leq c_{ij'} + c_{i'j'} + c_{i'j}$$

State of the Art It is proven to be **NP**-hard to approximate this problem to a factor 1.463 [1]. The best known polynomial time algorithms can approximate this problem to a factor 1.488 [3].

4.2 Linear Programming Formulations

Primal Let the variable x_{ij} denote that a client $j \in D$ is connected to a facility $i \in F$, and let y_i indicate whether facility $i \in F$ is open or not. Then, the primal linear programming formulation of the problem is as follows:

$$\begin{aligned}
\textbf{Minimize: } & \sum_{i \in F} f_i \cdot y_i + \sum_{i \in F, j \in D} c_{ij} \cdot x_{ij} & \\
\textbf{Subject to: } & \sum_{i \in F} x_{ij} \geq 1 & \forall j \in D \\
& y_i - x_{ij} \geq 0 & \forall i \in F, j \in D \\
& x_{ij}, y_i \in [0, 1] & \forall i \in F, j \in D
\end{aligned} \tag{5}$$

The constraints are respectively there to ensure that:

1. Every client $j \in D$ is connected to at least one facility $i \in F$.
2. If a client is connected to some facility, then the facility is open.
3. Clients can either be connected or not to facilities, and facilities can either be open or closed.

Dual As always, the dual can be obtained mechanically from the primal linear program. Let α_j be a variable for each $j \in D$ and let β_{ij} be a variable for each pair of $i \in F, j \in D$. Then, the dual of the previous linear program is defined as follows:

[1] D stands for "demand".

$$
\begin{aligned}
\textbf{Maximize:} \quad & \sum_{j \in D} \alpha_j \\
\textbf{Subject to:} \quad & \sum_{j \in D} \beta_{ij} \leq f_i && \forall i \in F \\
& \alpha_j - \beta_{ij} \leq c_{ij} && \forall i \in F, j \in D \\
& \alpha_j, \beta_{ij} \geq 0 && \forall i \in F, j \in D
\end{aligned} \tag{6}
$$

4.3 Primal-Dual Algorithm

In this section, we will give a primal-dual algorithm that we will show to be a 3-approximation algorithm. The algorithm was first described in [2]. It runs in two phases.

Phase One In the first phase, we start with $\alpha_j = 0$ and $\beta_{ij} = 0$ for all $i \in F, j \in D$, which is a dual feasible solution. We then start increasing α_j at the same rate for every unassigned client j and react depending on the following events:

- A constraint $\alpha_j \leq c_{ij} + \beta_{ij}$ becomes tight and i is not *temporarily opened*:
  - We add β_{ij} to the set of variables being increased.
- A constraint $\alpha_j \leq c_{ij} + \beta_{ij}$ becomes tight and i is *temporarily opened*:
  - We remove α_j from the set of variables being increased and declare i the *connecting witness* of j.
- A constraint $\sum_{j \in D} \beta_{ij} \leq f_i$ becomes tight:
  - i is added to the set of *temporarily* opened facilities.
  - Each client j with a tight edge to i is assigned to it. i is called a *connecting witness* of all such j's.
  - The associated α's and β's are removed from the set of variables being increased.

When the set of variables to be increased is empty, then Phase one is done.

Note that at the end of this phase, we might have opened too many facilities. For example, take the instance where any number of clients are connected to any number of facilities. Moreover, assume the connection costs and opening costs are all the same. In this case, it is easy to see that *all* facilities will be returned as a result of the first phase, but a single one would have sufficed. Therefore the result of this phase might have arbitrarily greater cost than the optimal. The second phase is there to solve this problem.

Phase Two In the second phase of the algorithm, we decide which of the temporarily opened facilities, denoted by F_t, will indeed be opened. To do so, we construct a graph $G = (V, E)$. The vertex set V of G is the set of temporarily opened facilities $i \in F$ returned from the first phase. We have an edge $(i, i') \in E$ if there exists a client $j \in D$ such that $\beta_{ij}, \beta_{i'j} > 0$. We call an edge (i, j) such that $\beta_{ij} > 0$ a *special* edge. Intuitively, the graph is a conflict graph, each edge indicating that at least one client has contributed to the cost of opening the two given facilities.

To solve the conflict, we construct a *maximal* independent set, meaning that we greedily include independent vertices until it is no longer possible. Note that it is not the same as constructing a maximum independent set. This set of vertices indicates which of the facilities will definitively be opened. We will denote this set by I.

Now that facilities have been opened, we must assign each client to an open facility. Lets denote by $\phi(j)$ the open facility assigned to a client $j \in D$ and let F_j be defined for all $j \in D$ as

$$F_j = \{i \in F_t : \beta_{ij} > 0\}$$

For each client $j \in D$, three cases may arise:

1. $I \cap F_j \neq \emptyset$: In this case, as I is an independent set, we are bound to have that $I \cap F_j = \{i\}$ for some i. We proceed to assign client j to facility i, noted as $\phi(j) = i$. We say that the client is *directly connected*. Note that $\beta_{ij} > 0$.
2. $I \cap F_j = \emptyset$ and the connecting witness of j, say i', is member of I. We set $\phi(j) = i'$ and call this client *directly connected* as well. Note that $\beta_{i'j} = 0$ and $\alpha_j = c_{i'j}$.
3. $I \cap F_j = \emptyset$ and the connecting witness of j, again say i', is *not* member of I. Let $i \in I$ be a neighbour of i' in G. This i must exists since otherwise I wouldn't be maximal. We set $\phi(j) = i$ and call the client *indirectly connected*. Note that $\beta_{ij} = 0$ in this case.

4.4 Analysis of the Algorithm

First, we show how the dual solutions α_j accounts for both the cost of opening facilities and connecting clients to facilities. To show this, lets denote by α_j^f the contribution of j to opening costs and α_j^c the contribution of j to connection costs:

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

If a client j is directly connected to $i = \phi(j)$, we have that $\alpha_j = \beta_{ij} + c_{ij}$. We set $\alpha_j^f = \beta_{ij}$ and $\alpha_j^c = c_{ij}$. In the case when the client j is indirectly connected, we set $\alpha_j^f = 0$ and $\alpha_j^c = \alpha_j$.

Lemma 8 *Let D_i be the set of clients connected to $i \in I$:*

$$D_i = \{j \in D : \phi(j) = i\}$$

For each $i \in I$, the following holds:

$$\sum_{j \in D_i} a_j^f = f_i$$

This means that the opening cost of i is accounted for by the α_j^f of every connected client.

Proof As we had that i was temporarily open at the end of the first phase, it must be the case that:

$$\sum_{j:\beta_{ij}>0} \beta_{ij} = f_i$$

By definition, every client j that contributes to this sum must be *directly connected* to i, since if it was *indirectly connected* β_{ij} would be null.

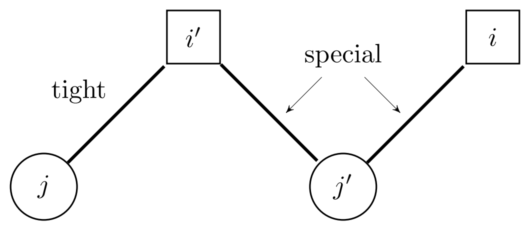
Remember that for *directly connected* clients, we have that $\alpha_j^f = \beta_{ij}$. In the case of *directly connected* clients that have $\beta_{ij} = 0$, which don't contribute to this sum, we have that $\alpha_j^f = \beta_{ij} = 0$.

For *indirectly connected* clients, we have by definition $\alpha_j^f = 0$. ■

Note that, for a client j *directly connected* to $i \in I$, we have that $\alpha_j^c = c_{ij}$, meaning that connection cost of each client is accounted for by its α_j^c. We are now left to analyse the connection cost of *indirectly connected* clients. The following lemma address this point.

Lemma 9 *For a client j* indirectly connected *to $i \in I$, we have that $c_{ij} \leq 3 \cdot \alpha_j^c$, meaning that the connection cost of each of those clients is bounded by three times its α_j^c.*

Proof Let $i' \neq i$ be the connecting witness of client j. As j is *indirectly connected* to i, it must be the case that (i, i') is an edge in the graph G built during the second phase of the algorithm. Since i, i' are neighbours in G, it must be the case that a client j' exists such that $\beta_{ij'}, \beta_{i'j'} > 0$, by definition of G. The following diagram illustrates the situation.



For analysis purposes, lets consider that during the first phase of the algorithms a time variable, called t, is also increased along with all other variables. Let t_1 and t_2 be the values of t respectively when i and i' where declared temporarily opened. As the edge (i, j') is tight, we have that $\alpha_j = c_{i'j} + \beta_{i'j} \geq c_{i'j}$. Since (i, j') and (i', j') are special edges (meaning that $\beta_{ij'}, \beta_{i'j'} > 0$), we have that:

$$\alpha_{j'} \geq c_{ij'} \text{ and } \alpha_{j'} \geq c_{i'j'}$$

By construction, those two edges became special at a time before i or i' were temporarily opened. Note also that after the opening of i or i', $\alpha_{j'}$ would have stopped increasing.

$$\alpha_{j'} \leq \min(t_1, t_2)$$

A final observation we make is that, as i' is the connecting witness of j, we have that:

$$\alpha_j \geq t_2$$

From all those observations, we can derive that:

$$\alpha_j \geq c_{i'j}$$

$$\alpha_j \geq t_2 \geq \min(t_1, t_2) \geq \alpha_{j'} \geq c_{ij'}, c_{i'j'}$$

Using the previous result and the fact that c is a metric, we get that:

$$c_{ij} \leq c_{i'j} + c_{i'j'} + c_{ij'} \leq 3 \cdot \alpha_j$$

As we set $\alpha_j = \alpha_j^c$ for indirectly connected clients, we get that:

$$c_{ij} \leq 3 \cdot \alpha_j = 3 \cdot \alpha_j^c$$

Which concludes the proof. ■

Theorem 10 *The proposed algorithm is a* 3*-approximation algorithm for the metric uncapacitated facility location problem.*

Proof Remember that the cost of a solution is given by:

$$PrimalCost = \sum_{j \in D} c_{\phi(j),j} + \sum_{i \in I} f_i$$

The cost of the dual solution is given by:

$$DualCost = \sum_{j \in D} \alpha_j = \sum_{j \in D} \alpha_j^c + \sum_{j \in D} \alpha_j^f$$

Using the weak duality theorem, we have that $DualCost \leq OPT$ where OPT is the primal optimal. We will proceed to show that $PrimalCost \leq 3 \cdot DualCost \leq 3 \cdot OPT$.

Using the previous lemmas, we can get that:

$$PrimalCost = \sum_{j \in D} c_{\phi(j),j} + \sum_{i \in I} f_i \leq 3 \sum_{j \in D} a_j^c + \sum_{j \in D} a_j^f \leq 3 \sum_{j \in D} a_j = 3 \cdot DualCost \leq 3 \cdot OPT$$

This concludes the proof of the 3-approximation. ■

References

[1] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 649–657. Society for Industrial and Applied Mathematics, 1998.

[2] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 2–13. IEEE, 1999.

[3] S. Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In *Proceedings of the 38th International Conference on Automata, Languages and Programming - Volume Part II*, ICALP'11, pages 77–88, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] R. J. Vanderbei. Linear programming: Foundations and extensions, 1996.