Advanced Approximation Algorithms **(CMU 15-854B, Spring 2008)**

Lecture 5: Primal-Dual Algorithms and Facility Location

Jan 29, 2008

Lecturer: Anupam Gupta *Scribe: Varun Gupta*

In the last lecture, we saw an LP rounding algorithm for the metric uncapacitated facility location problem that achieves 4-approximation. In this lecture, we will see how the concepts of *linear programming duality* and *complementary slackness* can be used to get better approximation algorithms for UFL.

1 Linear Programming Duality

Figure 1 shows a linear program (P) with m constraints and n variables in canonical form, and the corresponding *dual linear program* (D). The dual program (D) has n constraints, one for each variable in (P), and m variables, one for each constraint in (P).

$$
\begin{array}{lll}
(P) & \min & \boldsymbol{c}^T \boldsymbol{x} \\
 & \text{s.t.} & A\boldsymbol{x} \geq \boldsymbol{b} \\
 & & \boldsymbol{x} \geq 0
\end{array}
\qquad\qquad
\begin{array}{lll}
(D) & \max & \boldsymbol{b}^T \boldsymbol{y} \\
 & \text{s.t.} & A^T\boldsymbol{y} \leq \boldsymbol{c} \\
 & & \boldsymbol{y} \geq 0
\end{array}
$$

$$A \in \mathbb{R}^{m \times n}$$

Figure 1: Primal (P) and dual (D) linear programs in canonical form.

Dual linear programs are very useful to obtain bounds on the optimal value of the primal linear programs using the following two *duality theorems*.

Theorem 1.1. ***Weak duality:*** *Let $\boldsymbol{x}$ be a feasible solution for (P) and $\boldsymbol{y}$ be a feasible solution for (D). Then,*

$$\boldsymbol{c}^T \boldsymbol{x} \geq \boldsymbol{b}^T \boldsymbol{y}.$$

Theorem 1.2. ***Strong duality:*** *For the linear programs in Figure 1, exactly one of the following possibilities occurs:*

1. *Both (P) and (D) are infeasible.*

2. *(P) is infeasible and (D) is unbounded.*

3. *(P) is unbounded and (D) is infeasible.*

4. *Both (P) and (D) are feasible. In this case, both (P) and (D) have an optimal solution. Let $\boldsymbol{x}^*$ be an optimal solution for (P), and $\boldsymbol{y}^*$ be an optimal solution for (D). Then,*

$$\boldsymbol{c}^T\boldsymbol{x}^* = \boldsymbol{b}^T\boldsymbol{y}^*.$$

The weak duality theorem states that the values of the feasible solutions of (D) lower bound the values of the feasible solutions of (P), and hence the optimal value of (P). The strong duality theorem states that this is in fact tight. The weak duality theorem is fairly straightforward to prove:

$$\boldsymbol{c}^T\boldsymbol{x} \geq (A^T\boldsymbol{y})^T\boldsymbol{x} = \boldsymbol{y}^T A\boldsymbol{x} \geq \boldsymbol{y}^T\boldsymbol{b} = \boldsymbol{b}^T\boldsymbol{y},$$

and will often we sufficient for our purposes of obtaining a lower bound on the value of the optimal integral solution.
Note that taking the dual of (D) gives back (P). This is often a useful sanity check for the correctness of (D).

1.1 Complementary slackness

As mentioned earlier, for each primal variable x_j in (P), there is a corresponding dual constraint (call this the jth constraint in (D)):

$$\sum_{i=1}^{m} a_{i,j}y_j \leq c_j$$

Similarly, for each dual variable y_i, there is an associated primal constraint (call this the ith constraint in (P)):

$$\sum_{j=1}^{n} a_{i,j}x_j \geq b_i$$

Complementary slackness is the relationship between the slackness (strict positivity) of a primal variable and the slackness (strict inequality) of the corresponding dual constraint.

Theorem 1.3. ***Complementary slackness:*** *Let $\boldsymbol{x}^*$ be an optimal solution of (P) and $\boldsymbol{y}^*$ be an optimal solution of (D). Then,*

1. *$x_j^* > 0 \implies j$th constraint in (D) is tight.*
2. *jth constraint in (D) is slack $\implies x_j^* = 0$.*
3. *$y_i^* > 0 \implies i$th constraint in (P) is tight.*
4. *ith constraint in (P) is slack $\implies y_i^* = 0$.*

Complementary slackness is a consequence of the strong duality theorem. The book by Chvátal [**?**] provides an economic interpretation of the dual variables as prices in the primal linear program. For the facility location problem, in Section 2 we will see an interpretation of the dual variables as payments made by the clients.

1.2 The ellipsoid method

The ellipsoid method was the first provably polynomial time algorithm for solving linear programs. The ellipsoid method can solve even “large” linear programs (number of constraints may be exponential in the number of variables) in polynomial time, provided there is a *separation oracle* for the problem:

Definition 1.4. Separation oracle: A separation oracle is a polynomial time algorithm, that given a point $\boldsymbol{x}$, answers whether $\boldsymbol{x}$ is a feasible point for the linear program or not, and finds one violated constraint if $\boldsymbol{x}$ is not feasible.

2 Uncapacitated Facility Location using LP Duality

In this section we will see two approximation algorithms for the metric uncapacitated facility location problem based on “solving” the dual program. Recall the metric uncapacitated facility location problem:

Definition 2.1. Metric uncapacitated facility location
Input:

- Set D of clients
- Set F of facilities
- Metric distance function $d : (F \cup D) \times (F \cup D) \to \mathbb{R}_+$ (define $d_{ij} \triangleq d(ij)$)
- Facility costs $f : F \to \mathbb{R}_+$ (define $f_i \triangleq f(i)$)

Output: $S \subseteq F$ that minimizes $\sum_{i \in S} f_i + \sum_{j \in D} \min_{i \in S} d_{ij}$

Lets begin by writing the primal and dual linear programs for the relaxation of the facility location problem (Figure 2).

$$\begin{array}{lll}
(P) & \text{min} & \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} d_{ij} x_{ij} \\
 & & \\
 & \text{s.t.} & x_{ij} \leq y_i \quad \forall i \in F, j \in D \\
 & & \sum_{i \in F} x_{ij} \geq 1 \quad \forall j \in D \\
 & & \\
 & & x_{ij}, y_i \geq 0
\end{array}
\qquad
\begin{array}{lll}
(D) & \text{max} & \sum_{j \in D} \alpha_j \\
 & & \\
 & \text{s.t.} & \sum_{j \in D} \beta_{ij} \leq f_i \quad \forall i \in F \\
 & & \alpha_j - \beta_{ij} \leq d_{ij} \quad \forall i \in F, j \in D \\
 & & \\
 & & \alpha_j, \beta_{ij} \geq 0
\end{array}$$

Figure 2: Primal (P) and dual (D) linear program relaxations for the uncapacitated facility location problem.

Interpretation of the dual variables: Note that the dual program (D) has a variable α_j for every client $j \in D$, and variables β_{ij} for every (facility,client) pair. The total cost of the solution $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is given by $\sum_{j \in D} \alpha_j$. We can think of α_j as the amount of money client j is willing to contribute to the solution, and β_{ij} as client j's contribution towards opening facility i. The first set of constraints in (D),

$$\sum_{j \in D} \beta_{ij} \leq f_i$$

can be interpreted as saying that the total cost of opening facility i is at least the total contribution from every client towards opening facility i. The second set of constraints in (D),

$$\alpha_j \leq d_{ij} + \beta_{ij}$$

can be interpreted as saying that the total payment by client j is at most the sum of the contribution it makes towards opening facility i and the distance it has to travel to connect to i.
We are now ready to present a rounding algorithm based on the solution of the linear programs in Figure 2.

2.1 Another LP rounding algorithm

```
Algorithm LP_Round_Dual(D,F,d,f)

1. Solve $(P)$ and $(D)$. Let $(x, y)$ and $(\alpha, \beta)$ be the primal and dual optimal solutions, respectively.

2. Find $j$ with the smallest $\alpha_j$,

    (a) Let $N_j = \{i : x_{ij} > 0\}$.
    (b) Open the cheapest facility in $N_j$, and assign every $j'$ s.t. $N_j \cap N_{j'} \neq \emptyset$ to the cheapest facility in $N_j$.

3. Repeat step 2 with the unassigned clients until every client is assigned.
```

Figure 3: An LP rounding algorithm for the metric uncapacitated facility location problem.

Let $OPT(P)$ and $OPT(D)$ denote the optimal values of the primal and dual linear programs in Figure 2, respectively.

Claim 2.2. *The total facility opening cost of algorithm LP_Round_Dual is at most* $OPT(P)$.

Proof. Let j be some client picked in step 2 of the algorithm, and let f_{min} be the cost of the cheapest facility in N_j. Then,

$$f_{min} \leq f_{\min} \cdot \sum_{i \in N_j} x_{ij} \leq f_{\min} \sum_{i \in N_j} y_i = \sum_{i \in N_j} f_{\min} y_i \leq \sum_{i \in N_j} f_i y_i$$

If j and j' are two clients picked by the algorithm in step 2, then $N_j \cap N_{j'} = \emptyset$. Therefore the total facility opening cost is at most $\sum_i f_i y_i \leq OPT(P)$. □

Claim 2.3. *The total connection cost of algorithm LP_Round_Dual is at most* $3OPT(D)$.

Proof. Consider a client j picked by step 2, and let i the cheapest facility in N_j. Let i' be any other facility in N_j and j' be some client such that $x_{i'j'} > 0$. We will show,

$$d_{ij'} \leq 3\alpha_{j'}$$

Since the algorithm picks the smallest α from among the unassigned clients,

$$\alpha_j \leq \alpha_{j'}$$

Further, observe that $x_{ij} > 0$, $x_{i'j} > 0$ and $x_{i'j'} > 0$. Therefore, by complimentary slackness, the corresponding dual constraints must be tight. This gives,

$$\begin{aligned} \alpha_j &= d_{ij} + \beta_{ij} \geq d_{ij} \\ \alpha_j &= d_{i'j} + \beta_{i'j} \geq d_{i'j} \\ \alpha_{j'} &= d_{i'j'} + \beta_{i'j'} \geq d_{i'j'} \end{aligned}$$

Finally,

$$\begin{aligned} d_{ij'} &\leq d_{ij} + d_{i'j} + d_{i'j'} \\ &\leq \alpha_j + \alpha_j + \alpha_{j'} \\ &\leq 3\alpha_{j'} \end{aligned}$$

The total connection cost then is at most $\sum_{j \in D} 3\alpha_j \leq 3OPT(D)$. □

Combining Claims 2.2 and 2.3, and noting that $OPT(P) = OPT(D) \leq OPT(\text{Integral})$, we see that algorithm LP_Round_Dual is a 4-approximation algorithm. In the last lecture, we had seen another LP rounding algorithm that achieved 4-approximation, but with an extra filtering step.

A randomized rounding twist: In step 2(b) of the rounding algorithm, we opened the cheapest facility in N_j. Note that for the optimal primal solution, $\sum_i x_{ij} = 1$. Hence for a fixed j, x_{ij} define a probability distribution on the facilities. If we open facility $i \in N_j$ randomly with probability x_{ij}, and assign every j' such that $N_j \cap N_j' \neq \emptyset$ to i, the total expected facility opening cost is upper bounded by $F^* = \sum_{i \in F} f_i y_i$. Let

$$D_{j'} = \sum_{i \in F} x_{ij} d_{ij}.$$

The expected connection cost for j' is bounded by $(\alpha_{j'} + \alpha_j + D_j)$. Now, if in step 2 of the algorithm, we had picked clients in the order of increasing $(\alpha_j + D_j)$, then connection cost of j' is upper bounded by $(2\alpha_{j'} + D_{j'})$. Therefore, the total connection cost is upper bounded by

$2OPT(D) + C^*$, where $C^* = \sum_{j \in D} D_j$ is the connection cost of the primal. The total expected cost of the solution is bounded by $2OPT(D) + OPT(P) = 3OPT(P)$, giving a 3-approximate algorithm.

Another randomized rounding twist: Using another trick, we can completely do away with solving the dual LP. Note that we needed to solve the dual LP to sort the clients in order of α_j (or $(\alpha_j + D_j)$ for the randomized version described above). Let us now pick a client j uniformly at random from the set of unassigned clients, open facility $i \in N_j$ with probability x_{ij}, and assign every j' such that $N_j \cap N_{j'} \neq \emptyset$ to i. We will now prove that this gives an expected 3-approximate algorithm.
Once we have solved the primal LP, we can create a graph with the clients as vertices, and an edge between clients j and j' is there exists a facility i such that both $x_{ij} > 0$ and $x_{ij'} > 0$. Let δ_j be the degree of client j in this graph. While picking clients at random, the probability that client j gets picked is $\frac{1}{\delta_j+1}$ (equivalently, the probability that j appears ahead of any of its neighbors in a random permutation of the clients). If a client j is picked and facility $i \in N_j$ opened, j pays for connecting itself to i ($= D_j$ in expectation), as well as the extra connection cost ($\leq \alpha_j + D_j$) paid by every client j' now assigned to i. If client j is not picked it only pays for connecting to a facility in N_j, and this cost is bounded by α_j. Therefore, the expected payment made by client j is:

$$\begin{aligned} cost_j &\leq \frac{1}{\delta_j + 1}\left(D_j + \delta_j(\alpha_j + D_j)\right) + \frac{\delta_j}{\delta_j + 1}\alpha_j \\ &= D_j + \frac{2\delta_j}{\delta_j + 1}\alpha_j \\ &\leq D_j + 2\alpha_j \end{aligned}$$

The total expected connection cost is therefore bounded by $2OPT(D) + C^*$. The total expected facility opening cost is bounded by F^*. Hence the total expected cost is bounded by $3OPT(P)$.

2.2 A Primal-Dual algorithm for facility location

Definition 2.4. In its most generality, a Primal-Dual algorithm for a minimization problem $\mathcal{P}$ is one that outputs:

1. A feasible integer primal solution for $\mathcal{P}$
2. A feasible solution to the dual of the LP relaxation of $\mathcal{P}$
3. A proof for:

$$cost(\text{primal integer solution}) \leq \alpha \cdot cost(\text{dual solution})$$

Clearly, an algorithm of the type defined in Definition 2.4 will be an α-approximate algorithm for $\mathcal{P}$. Primal-Dual algorithms generally follow the following schema: Begin with a dual feasible

solution (typically all 0s) and start raising the dual variables in a controlled manner until some dual constraint becomes tight. Set the primal variable corresponding to the tight dual constraint to some integral value. While keeping the dual feasible, repeat these steps until a feasible primal solution is obtained.
In this section, we will see a 3-approximate Primal-Dual algorithm for the metric uncapacitated facility location problem due to Jain and Vazirani [**?**].

Algorithm
Phase 1
Start with $\boldsymbol{\alpha}, \boldsymbol{\beta} = 0$, each client is *unconnected* and each facility is *unopened*. Each client raises its dual variable, α_j, at unit rate until $\alpha_j = d_{ij}$ for some facility i. We can interpret this as saying that client j has paid enough to reach facility i. We call edge (i, j) *tight*. From now on, as α_j increases, β_{ij} also increases at unit rate so that the constraint

$$\alpha_j - \beta_{ij} \leq d_{ij}$$

is not violated. β_{ij} can be interpreted as the contribution of j towards opening facility i.
Whenever for some facility i, $\sum_{j \in D} \beta_{ij} = f_i$, facility i is declared *temporarily open*. All unconnected clients having tight edges to i are declared *connected* and i is called the *connecting witness* for these clients. The dual variables, α_j, for the connected clients are not raised anymore, because this would violate some constraint of the dual program. Subsequently, if some client j gets a tight edge to a temporarily opened facility i, client j is also declared connected with connecting witness i. The first phase of the algorithm terminates when all clients are connected.

Remark Note that at the end of Phase 1, the cost of the dual program is $\sum_{j \in D} \alpha_j$. This should approximately pay for the connection cost and facility opening cost for the integer primal solution derived from the dual solution. However, a client may have paid towards temporarily opening several facilities, although it eventually connects to only one. If in the integer solution, we open every temporarily opened facility, the cost of this solution can be very far from the cost of the dual solution. We will see one such example in Section 2.2.1. Phase 2 of the algorithm chooses a subset of the temporarily opened facilities to open in the integer solution.

Phase 2
Let F_t denote the set of temporarily opened facilities. We say that two facilities i and i' in F_t are *conflicting*, if there exists a client j that has offered money to both the facilities ($\beta_{ij}, \beta_{i'j} > 0$). Consider a graph with vertex set F_t, and an edge between two facilities if they are conflicting. Pick a maximal independent set I of facilities from this graph by choosing facilities in the order in which they were temporarily opened if they are not conflicting with a facility already chosen, and declare the set I of facilities *open*.
For a client j, if there exists a facility $i \in I$ such that i was the connecting witness of j, assign client j to facility i, and declare client j *directly connected*. Otherwise, let $i' \notin I$ be the connecting witness of j and let $i \in I$ be a facility which conflicts with i' and was temporarily opened before i'. Assign j to i, and declare client j *indirectly connected*.

Analysis

Claim 2.5. *No client j contributes money to two different* actually *open facilities $i, i' \in I$.*

Proof. True by the construction of the integer solution, since the facilities in I are non-conflicting. □

Claim 2.6. *If a facility i was opened, let S_i denote the set of clients direcly connected to i. Then,*

$$f_i + \sum_{j \in S_i} d_{ij} = \sum_{j \in S_i} \alpha_j.$$

Proof. Since facility i was temporarily opened, the total contribution from the clients directly connected to i must be exactly f_i. Therefore,

$$f_i = \sum_{j \in S_i} \beta_{ij}$$

Note that some β_{ij}'s may be 0 if client j connects to i after it has been temporarily opened. Further, for every client j directly connected to i, the dual constraint

$$\alpha_j - \beta_{ij} \leq d_{ij}$$

is tight. Therefore,

$$f_i = \sum_{j \in S_i} \beta_{ij} = \sum_{j \in S_i} (\alpha_j - d_{ij})$$

or alternately,

$$f_i + \sum_{j \in S_i} d_{ij} = \sum_{j \in S_i} \alpha_j$$

□

Claim 2.7. *Let client j be indirecly connected to i. Then,*

$$d_{ij} \leq 3\alpha_j.$$

Proof. Let i' be the connecting witness of j. Since i and i' are conflicting, there exists a client j' such that $\beta ij', \beta_{i'j'} > 0$. Since j' is contributing money to both i and i', and j is contributing money to i',

$$d_{ij'} \leq \alpha_{j'}$$
$$d_{i'j'} \leq \alpha_{j'}$$
$$d_{i'j} \leq \alpha_j$$

Facility i opens before facility i' by the construction of the set of open facilities. The dual variable α_j stops increasing when its witness facility, i', temporarily opens, and $\alpha_{j'}$ stops increasing when its witness facility, i, temporarily opens. Since α_j and $\alpha_{j'}$ are rising at unit rate , $\alpha_{j'} \leq \alpha_j$. Finally,

$$\begin{aligned} d_{ij} &\leq d_{i'j} + d_{i'j'} + d_{ij'} \\ &\leq \alpha_j + \alpha_{j'} + \alpha_{j'} \\ &\leq 3\alpha_j \end{aligned}$$

□

Combining Claims 2.6 and 2.7, the total cost paid by the integer solution is,

$$cost(\text{integer solution}) \leq 3\sum_j \alpha_j \leq 3OPT(D).$$

Therefore we get a 3-approximate algorithm. Jain and Vazirani [**?**] show that this is the best achievable approximation guarantee by this Primal-Dual algorithm.

Choosing maximal independent sets: In phase 2 of the Primal-Dual algorithm above, we picked the facilities to open in the order they were temporarily opened, as long as they were not conflicting with a facility which was already picked. However, we can choose any maximal independent set. This can be useful if we need the independent set to satisfy some other property as well. The analysis is almost the same as before. Note that Claims 2.5 and 2.6 still hold, so it is sufficient to verify Claim 2.7.

As before, let client j be indirecly connected to i. Let i' be the connecting witness of j. Since i and i' are conflicting, there exists a client j' such that $\beta ij', \beta_{i'j'} > 0$. Let τ_i be the time at which facility i is temporarily opened, and $\tau_{i'}$ be the time when facility i' is temporarily opened. Now,

$$\alpha_{j'} \leq \min\{\tau_i, \tau_{i'}\}$$

since $\alpha_{j'}$ stops increasing whenever a facility it is contributing to temporarily opens. Also, since j is directly connected to i,

$$\alpha_j \geq \tau_i \geq \min\{\tau_i, \tau_{i'}\} \geq \alpha_{j'}$$

The remaining proof is identical to the proof of Claim 2.7.

2.2.1 Need for Phase 2

The following instance of the metric uncapacitated facility location problem illustrates why the second phase of the Primal-Dual algorithm in Section 2.2 is necessary. The instance is illustrated in Figure 4.

- There are $2n$ clients and n potential facilities.

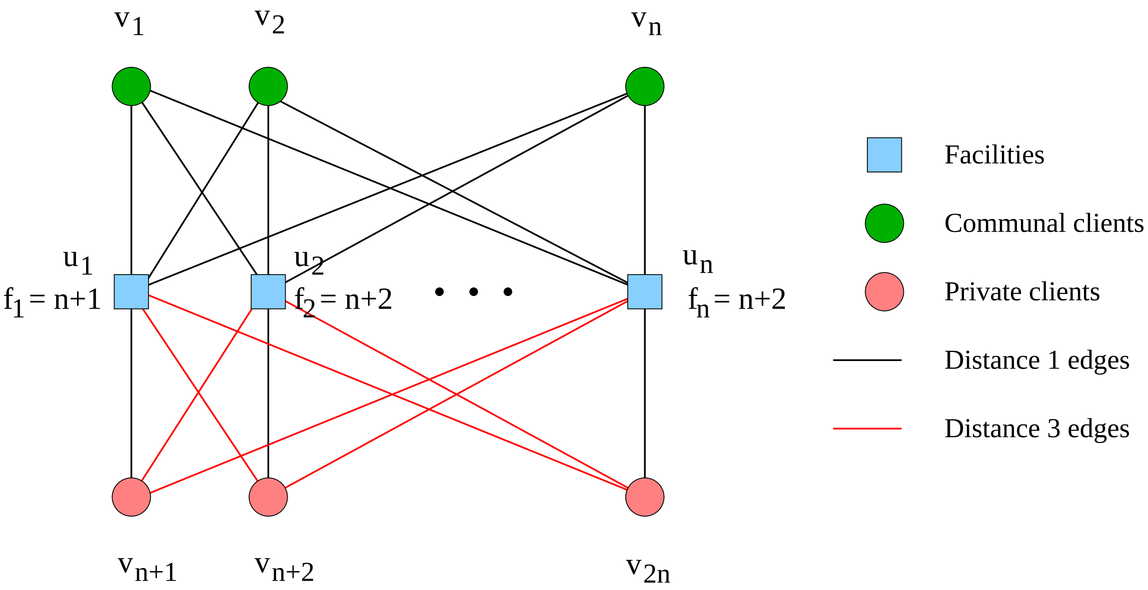

Figure 4: A metric uncapacitated facility location instance illustrating the necessity of the second phase of the Primal-Dual algorithm.

- The first n of the clients (say $v_1, v_2, \cdots, v_n$) are at distance 1 to all the facilities, we call these *communal* clients.
- For the other n clients: client v_{n+i} is at distance 1 to facility u_i and at distance 3 to all other facilities. (We say client v_{n+i} is facility u_i's *private* client.)
- The first facility u_1 costs $n+1$, and all the others cost $n+2$.

If we raise the duals uniformly, then at time $t = 2$, the first facility u_1 will become tentatively opened, and all the communal clients, and private client u_{n+1} become frozen. But the other private clients will continue to raise their duals, and at time $t = 3$, all the remaining facilities will also become tentatively open.
Note that the cost of actually opening all these tentatively open facilities would be $(n+1)+(n-1)(n+2) = \Omega(n^2)$, whereas the total dual value generated is only $O(n)$. Hence the clean-up step is needed.