MIT 6.854/18.415: Advanced Algorithms Spring 2016

Lecture : LP Relaxations for Combinatorial Relaxation

Lecturers: Chris Musco and Ali Vakilian (for Ankur Moitra)

Scribes: Tom Kolokotrones, Mayuri Sridhar, and Vincent Tjeng

In this lecture, we consider methods for developing approximation algorithms for NP-hard combinatorial optimization problems using linear programs (LPs). We will consider two approaches: LP Rounding and the Primal-Dual approach.

1 Last Time (Submodular Functions)

Last week, we talked about submodular functions and showed a way to relax them to convex functions via the Lovasz Theorem. The idea was that, if we could optimize the Lovasz Extension, which is a continuous convex optimization problem, then we could optimize submodular functions. Today, we're going to be doing something very similar to that with one exception: the problems we'll try to approximate aren't easy to solve. In general, we'll focus on NP-hard combinatorial optimization problems, which we'll approximate using similar relaxations.

2 Integer Optimization Problems and Vertex Cover

We'll start with a specific example, which many of you have probably seen and is one of Karp's original 21 NP-Hard problems: VERTEX-COVER.

VERTEX-COVER: Given an undirected graph $G = (V, E)$, find a subset $C \subseteq V$ of minimum size, $|C|$, such that if $(u, v) \in E$, at least one of u or v is in C.

We can write this in the form of an optimization problem by associating each $C \subseteq V$ with a vector x of length $|V|$, so that x has an entry for each vertex of the graph and

$$x_v = \begin{cases} 1 \text{ if } v \in C \\ 0 \text{ otherwise} \end{cases}$$

To ensure that we cover each edge of the graph, for each $(u, v) \in E$ we include the constraint $x_u + x_v \geq 1$, which ensures that x_u or $x_v = 1$, so that $u \in C$ or $v \in C$. Since $|C| = \sum_{v \in V} x_v$ and our goal is to minimize $|C|$, VERTEX-COVER can be formulated as:

Integer Program:

$$\begin{aligned} \min_x \quad & \sum_{v \in V} x_v \\ \text{s.t.} \quad & x_u + x_v \geq 1 \quad \forall (u, v) \in E \\ & x_v \in \{0, 1\} \quad \forall v \in V \end{aligned}$$

Unfortunately, this is not easy to solve. While it looks like a Linear Program, the constraint that x is either 0 or 1 is a discrete constraint, specifically an integer constraint. Problems of this form are called Integer Linear Programs (ILP), since they look like standard Linear Programs, but with additional integer constraints, which generally make them NP-Hard. In fact, NP-Hard optimization problems can typically be written as ILPs, which is nice because we can then relax the problem into something we can actually solve.

In this case, instead of requiring $x_v \in \{0, 1\}$, we can instead simply require $0 \leq x_v \leq 1$, which converts the ILP into a standard LP.

Relaxed Linear Program:

$$\begin{aligned} \min_x \quad & \sum_{v \in V} x_v \\ \text{s.t.} \quad & x_u + x_v \geq 1 \quad \forall (u, v) \in E \\ & 0 \leq x_v \leq 1 \quad \forall v \in V \end{aligned}$$

This is good, and clearly any solution to the ILP will also be a solution to the relaxed LP since $x_v = 0$ or 1 satisfies $0 \leq x_v \leq 1$ (so the optimal value of the relaxed LP will be no worse than the optimal value of the original ILP), but a solution to the LP will not necessarily be a solution to the ILP because the LP can have fractional solutions. A simple example is the fully connected graph on 3 vertices, $G = K_3$ (the triangle). The optimal solution to the LP is given by setting each $x_v = \frac{1}{2}$, which gives a cost of $\frac{3}{2}$, while any solution to the ILP must set at least two $x_v = 1$ in order to satisfy the covering requirement, so that the minimum cost is 2. So clearly we can't, in general, blindly use the LP to solve the integer program, but often we can use the LP solution to help find a discrete solution, using LP Rounding, which is the first basic technique we'll talk about today.

3 LP Rounding

LP Rounding is a general technique that has three main steps:

1. Solve the relaxed LP to obtain the optimal x^*, which may not be integral, and, therefore, may not be a solution to the original ILP.
2. Round the entries of x^* to form $\tilde{x}$, which will be integral and so a feasible solution of the ILP.
3. Bound the gap between the cost of $\tilde{x}$ and the cost of the optimal solution to the ILP.

3.1 Deterministic Rounding

In the case of VERTEX-COVER, there is an extremely simple deterministic rounding algorithm that performs well in finding an approximate solution: rounding to the nearest integer. Starting with x^*, the optimal solution to the relaxed LP, we define $\tilde{x}$ by:

$$\tilde{x}_v = \begin{cases} 1 \text{ if } x_v^* \geq 1/2 \\ 0 \text{ otherwise} \end{cases}$$

We first check that $\tilde{x}$ satisfies VERTEX-COVER. By construction, $\tilde{x}$ is integral. Since x^* satisfies the constraints of the relaxed LP, if $(u, v) \in E$, $x_u^* + x_v^* \geq 1$, so x_u^* or $x_v^* \geq \frac{1}{2}$, so $\tilde{x}_u$ or $\tilde{x}_v = 1$, and, therefore, $\tilde{x}_u + \tilde{x}_v \geq 1$, so $\tilde{x}$ satisifies VERTEX-COVER.

The approximation factor, α, must satisfy: $\sum_v \tilde{x}_v \leq \alpha \cdot OPT_{\text{integral}}$, where OPT_{integral} is the cost of the optimal solution to the ILP and $OPT_{\text{fractional}}$ is the cost of the optimal solution to the LP. Recalling that $OPT_{\text{fractional}} \leq OPT_{\text{integral}}$, since any solution to the original ILP is also a solution to the relaxed LP, and noting that $\tilde{x}_v \leq 2x_v^*$ from the definition of $\tilde{x}$, we have

$$OPT_{\text{integral}} \leq \sum_v \tilde{x}_v \leq 2\sum_v x_v^* = 2 \cdot OPT_{\text{fractional}} \leq 2 \cdot OPT_{\text{integral}}$$

so $\alpha \leq 2$ and our procedure gives a 2-approximation to VERTEX-COVER.

Could we find a smaller α such that $OPT_{\text{integral}} \leq \alpha \sum_v x_v^* = \alpha \cdot OPT_{\text{fractional}}$ using this general approach? The answer is no. Consider, $G = K_3$, the fully conected graph on 3 vertices. Recall that, for this example, $OPT_{fractional} = 3/2$ and $OPT_{\text{integral}} = 2$, so $\alpha \geq 4/3$ and we cannot make the approximation tighter than $4/3$. We can find worse examples than this though: consider the complete graph on n vertices, $G = K_n$. Then the optimal fractional solution still assigns $x_v = 1/2$ to each vertex for a total cost of $\frac{n}{2}$. However, any integer solution must assign 1 to all but one of the vertices, which gives a minimum cost of $n - 1$, so $\alpha \geq \frac{n-1}{n/2} = 2\left(1 - \frac{1}{n}\right)$. Therefore, by taking n arbitrarily large, we can make the lower bound on α arbitrarily close to 2, so that, since $\alpha \leq 2$, $\alpha = 2$. In general, we can bound the cost of $OPT_{\text{integral}}/OPT_{\text{fractional}}$ by taking the supremum over all possible inputs, which is called the integrality gap.

$$IG = \sup \frac{OPT_{\text{integral}}}{OPT_{\text{fractional}}}$$

This is an important concept in these types of combinatorial optimization problems because, while it doesn't rule out the possibility that you can get a better approximation by using a different method (or even working with different LP-relaxation of the problem), it says that using this approach (relax and bound), or even other approaches based on the discussed LP, we can't produce an approximation better than the integrality gap; it is a measure of how tight your LP approximation is in the first place.[1] Here we gave a rounding scheme that matches the integrality gap, so we can't do better.

3.2 Set Cover

Though deterministic rounding performed very well in providing an approximation to the VERTEX-COVER problem, this is not always the case. In particular, it does not perform well on a generalization of VERTEX-COVER, known as SET-COVER.

SET-COVER: Given a set, S, which we will take to be $S = \{1, 2, \ldots, n\}$, and m subsets $S_1, S_2, \ldots, S_m \subseteq S$, such that $S = \bigcup_{i=1}^m S_i$, find a minimal collection $C = \{S_{i_1}, S_{i_2}, \ldots, S_{i_k}\}$, such that $\forall_{j \in S} \exists_{l \leq k} (j \in S_{i_l})$, or, equivalently, $S = \bigcup_{l=1}^k S_{i_l}$.

[1] Note that we could also try to directly bound $\alpha \cdot OPT_{\text{integral}}$, instead of using $\alpha \cdot OPT_{\text{fractional}}$ as an intermediate, which could possibly result in a smaller bound on the overall approximation factor, but this is often extremely difficult and is not typically done in practice, so rounding schemes are almost always analyzed like ours.

Proceeding as we did for VERTEX-COVER, we can associate each C with a vector x of length m, so x has one entry for each S_i and

$$x_i = \begin{cases} 1 \text{ if } S_i \in C \\ 0 \text{ otherwise} \end{cases}$$

The constraint that each $j \in S$ must be contained in at least one $S_{i_l} \in C$ can be written as $\sum_{i:j \in S_i} x_i \geq 1$ and $|C| = \sum_{i=1}^{m} x_i$ so

SET-COVER can be formulated as:

Integer Problem:

$$\begin{aligned} \min_x \quad & \sum_{i=1}^{m} x_i \\ \text{s.t.} \quad & \sum_{i:j \in S_i} x_i \geq 1 \quad \forall j \in \{1, 2, \ldots, n\} \\ & x_i \in \{0, 1\} \quad \forall i \in \{1, 2, \ldots, m\} \end{aligned}$$

To obtain the LP relaxation of above formulation, as before, we relax the constraint $x_i \in \{0, 1\}$ to $0 \leq x \leq 1$, giving:

Relaxation:

$$\begin{aligned} \min_x \quad & \sum_{i=1}^{m} x_i \\ \text{s.t.} \quad & \sum_{i:j \in S_i} x_i \geq 1 \quad \forall j \in \{1, 2, \ldots, n\} \\ & 0 \leq x_i \leq 1 \quad \forall i \in \{1, 2, \ldots, m\} \end{aligned}$$

Unfortunately, for this problem, deterministic rounding doesn't work very well. There is a simple example that shows this. Let $S = \{1, 2, 3, 4\}$, $S_1 = \{2, 3, 4\}$, $S_2 = \{1, 3, 4\}$, $S_3 = \{1, 2, 4\}$, $S_4 = \{1, 2, 3\}$. Then the optimal solution to the relaxed LP gives $x_i = \frac{1}{3}$ for each i. If we just round to the nearest integer, as before, $\tilde{x}_i = 0$ for every i, so we would pick no sets, and, therefore, would not satisfy SET-COVER. However, even if we change the rounding threshold from $\frac{1}{2}$ to some other value, since all the x_is have the same value, we either have to include all the sets or none of the sets using deterministic rounding. If we pick none of the sets, we don't satisfy SET-COVER; if we pick all of them we have a bad approximation, since we only need any 2 of the S_is. If we generalize this example to n elements: $S = \{1, 2, \ldots, n\}$, $S_i = S \setminus \{i\}, i \in \{1, 2, \ldots n\}$, either we pick 0 sets, and don't satisfy SET-COVER, or we pick n sets when we only needed 2, which gives an approximation error of $\frac{n}{2}$, so deterministic rounding does not work well for SET-COVER. Since deterministic rounding doesn't work well, we can instead try randomized rounding.

3.3 Randomized Rounding

Attempt 1: Set $\tilde{x}_i = 1$ with probability x_i^*.

The first thing we need to verify before we can even talk about the approximation factor is that $\tilde{x}$ satisfies the constraints of the linear program, specifically $\sum_{j \in S_i} x_i \geq 1$. Define $Y_j = \sum_{j \in S_i} \tilde{x}_i$. We can't bound this directly, but by considering the expectation and using the linearity of expectation we get, $\mathbb{E}[Y_j] = \sum_{j \in S_i} \mathbb{E}[\tilde{x}_i] = \sum_{j \in S_i} x_i^*$. Since x_i^* is our fractional solution, then it will satisfy the linear program, which is great, so our constraints are satisfied in expectation. However, what we actually want is that they are satisfied with high probability (meaning with probability $\geq 1 - n^{-1}$). To get this, we need to modify things in a very simple way.

Attempt 2: Set $\tilde{x}_i = 1$ with probability αx_i^*.[2]

We will set α later. The reason for multiplying by α is that by using a higher probability, we can ensure that our constraints are satisfied not just in expectation, but with high probability. In order to show this, we'll use Chernoff's Inequality. Y_j is the sum of independent Bernoulli random variables, so the probability that the constraint is not satisfied ($Y_j < 1$) is

$$\mathbb{P}(Y_j < 1) = \mathbb{P}\left(Y_j < \frac{1}{\mathbb{E}[Y_j]}\mathbb{E}[Y_j]\right) = \mathbb{P}\left(Y_j < \left(1 - \left(1 - \frac{1}{\mathbb{E}[Y_j]}\right)\right)\mathbb{E}[Y_j]\right)$$
$$\leq e^{-\left(1 - \frac{1}{E[Y_j]}\right)^2 \frac{E[Y_j]}{2}} = e^{\frac{1}{2}\left(2 - \mathbb{E}[Y_j] - \frac{1}{\mathbb{E}[Y_j]}\right)} \leq e^{1 - \frac{1}{2}\mathbb{E}[Y_j]}$$

which will be $\leq n^{-2}$ as long as $\mathbb{E}[Y_j] \geq 4\log n + 2$. Then, taking the union bound over the n constraints shows that they hold simultaneously with probability $\geq 1 - n^{-1}$, so we can choose an $\alpha \in O(\log n)$ so that randomized rounding gives a feasible solution with high probability.

The last step is to determine the approximation factor. In expectation,

$$OPT_{\text{integral}} \leq \mathbb{E}[Y_j] = \sum_{i=1}^{m} \mathbb{E}[\tilde{x}_i] = \sum_{i=1}^{m} \alpha x_i^* = \alpha \sum_{i=1}^{m} x_i^* = \alpha \cdot OPT_{\text{fractional}} \leq \alpha \cdot OPT_{\text{integral}}$$

so, since $\alpha \in O(\log n)$, randomized rounding gives an $O(\log n)$ approximation, in expectation, for SET-COVER. It's not hard to show that this also holds with high probability by using a Chernoff bound argument, similar to the one above. Next we'll talk about another approach to approximating combinatorial optimization problems that does not use any sort of rounding.

4 Primal-Dual Approach

The Primal-Dual approach is a powerful, general method for generating approximation algorithms for combinatorial problems, by taking advantage of the relationship between an LP and its dual. It was originally proposed by Dantzig, Ford, and Fulkerson in 1956 [1] as a means of solving linear programming problems, but it only works in limited cases and cannot be used, in general, for solving LPs. However, it was later discovered that it could be used for approximation algorithms for many combinatorial optimization problems. We'll use it to (approximately) solve the FACILITY LOCATION problem.

4.1 The (Metric Uncapacitated) Facility Location Problem

In the FACILITY LOCATION problem we wish to assign a collection of clients: D, to a set of facilities: F. We have two cost functions: $d : (D \cup F) \times (D \cup F) \to \mathbb{R}_+$ and $f : F \to \mathbb{R}_+$, where d is a metric

[2]Technically we need to use $\min(\alpha x_i^*, 1)$ to ensure we don't have probabilities above 1, but we will ignore this technical detail in our analysis.

distance function on $(D \cup F)$. The cost of assigning client j to facility i is given by $d_{ij} = d(i,j)$, which is the distance between the two. The cost of opening facility i is $f_i = f(i)$. By default, each facility is closed until the required fee, f_i, is paid to open it. Each facility can serve an unlimited number of clients. The goal is to open some subset of the facilities, and to assign the clients to them, while minimizing the total cost.

We can express the objective function formally as:

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

Define x_{ij} by $x_{ij} = 1$ if client j is connected to facility i, 0 otherwise, and y_i by $y_i = 1$ if facility i is open, 0 otherwise. Then, we can rewrite the cost as:

$$\min_{x,y} \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} d_{ij} x_{ij}$$

We want to ensure that each client is connected to at least one facility, which we can encode using the constraint: $\sum_{i \in F} x_{ij} \geq 1 \; \forall j \in D$. We also only want to assign clients to open facilities, which gives the constraint: $x_{ij} \leq y_i \; \forall i \in F, j \in D$. We can then formulate the problem as:

$$\begin{aligned}
\min_{x,y} \quad & \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} d_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{i \in F} x_{ij} \geq 1 & \forall j \in D \\
& x_{ij} \leq y_i & \forall i \in F, j \in D \\
& x_{ij}, y_i \in \{0,1\} & \forall i \in F, j \in D
\end{aligned}$$

After the standard LP-relaxation, in which we relax the integer constraints to $x_{ij}, y_i \geq 0$, we get:

Primal (P)

$$\begin{aligned}
\min_{x,y} \quad & \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} d_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{i \in F} x_{ij} \geq 1 & \forall j \in D \\
& x_{ij} \leq y_i & \forall i \in F, j \in D \\
& x_{ij}, y_i \geq 0 & \forall i \in F, j \in D
\end{aligned}$$

Dual (D)

$$\begin{aligned}
\max_{\alpha,\beta} \quad & \sum_{j \in D} \alpha_j & \\
\text{s.t.} \quad & \sum_{j \in D} \beta_{ij} \leq f_i & \forall i \in F \\
& \alpha_j - \beta_{ij} \leq d_{ij} & \forall i \in F, j \in D \\
& \alpha_j, \beta_{ij} \geq 0 & \forall i \in F, j \in D
\end{aligned}$$

The dual variables have a nice interpretation: α_j is the amount that client j is willing to pay to be connected to some facility and β_{ij} is the amount that client j is willing to contribute to opening facility i. If the total contribution of all clients to facility i, $\sum_{j \in D} \beta_{ij} \geq f_i$, then we open facility i, but the constraint $\sum_{j \in D} \beta_{ij} \leq f_i$ means that the clients won't overpay.

4.2 Primal-Dual Approximation for the Metric Uncapacitated Facility Location Problem (Jain and Vazirani 2001) [2]

The key insight about the Primal-Dual approach is that, if you find a feasible solution for the dual program, then, by weak duality, it is a lower bound for the optimal solution of the primal. The method will yield three outputs:

- A feasible, integer primal solution.
- A feasible dual solution.
- A proof that cost(integer primal solution) $\leq \lambda \cdot$ cost(dual solution).

Together, this tells us that the integer solution that we construct is a λ-approximate solution to the problem. The general approach is as follows:

- Begin with a feasible dual solution. This will typically be all zeros for the dual variables (which is easy to verify as feasible), but doesn't have to be.
- Start increasing the dual variables in a controlled manner, while ensuring that the dual solution remains feasible. We can't do this forever since eventually a dual constraint will become tight.
- Once this happens, set the corresponding primal value to some integer. This is how we actually construct the primal integer solution. When a dual constraint becomes tight, we modify the primal solution and then return to increasing dual variables until some other variable binds, and repeat the process.
- When we can no longer increase any dual variable, we stop and examine our constructed primal solution. The algorithm will show that it is not too different in cost from the dual solution we've constructed.

For the Metric Uncapacitated Facility Location problem, we proceed as follows:
Phase 1:

- Start with a feasible dual solution; in particular, set $\alpha = \beta = 0$. When $\alpha_j = 0$, client j is not connected to any facility, and when $\sum_{j \in D} \beta_{ij} < f_i$, facility i is not open, so we begin with all clients unassigned and all facilities closed.
- Each client increases its α_j at the same rate; in particular, each client will increase its α_j by 1 at each step, but this cannot continue indefinitely without violating dual feasibility.
- At some point, a constraint will become tight, so that $\alpha_j = d_{ij}$ for some i, j. At this point, client j has paid enough to reach facility i and the edge (i, j) becomes tight.
- In order to maintain dual feasibility in the rest of the algorithm, whenever α_j increases, for any tight edges (i, j), we must also increase β_{ij} by the same amount in order to maintain $\alpha_j - \beta_{ij} \leq d_{ij}$, but this is done only for tight edges.
- After repeating these steps enough, eventually a constraint of the second type, $\sum_{j \in D} \beta_{ij} \leq f_i$ will become tight for some i. If this happens, then facility i has received enough contributions to open, and we declare it to be temporarily open. Then, all *unconnected* clients that have a tight edge in common with facility i (e.g. (i, j)) get connected to facility i. For any such clients, β_{ij} must stop increasing in order to maintain dual feasibility, which means that α_j must also stop increasing for any clients that are connected to i. In general, the dual variables associated with connected clients do not increase after connection.

- When we are no longer able to increase any of the dual variables, all the clients will be connected, but the solution may have an excessively high cost due to too many facilities being open, which will necessitate some clean-up steps (hence why facilities are only referred to as temporarily open).

The dual cost, $\sum_{j \in D} \alpha_j$, is a combination of the connection costs and the facility opening costs in the primal. Even though a client will only connect to one facility, he may have contributed to multiple facilities. This is wasteful, so a clean-up phase will be needed to obtain a good result. In particular, if we were to open every temporarily opened facility, the cost of the primal would typically be much higher than that of the dual.

Phase 2: We say that i and i' are conflicting if $\exists j(\beta_{ij} \cdot \beta_{i'j} > 0)$. Define F_t to be the set of all temporarily opened facilities, T to be the graph whose vertex set is F in which there is an edge between i and i', (i, i'), iff i and i' are conflicting, and H to be the induced subgraph of T over the nodes F_t. Then we continue as follows:

- First, determine the set of open facilities. Select a maximal independent subset, $I \subseteq H$ by beginning with $I = \emptyset$ and adding facilities in the order in which they were opened, while ensuring that I remains independent. I is the set of open facilities.
- Second, assign each client to an open facility. If client j was connected to facility i during Phase 1 and $i \in I$, then assign j to facility i. We refer to this as "directly connected". Otherwise, assign j to one of i's neighbors i' in H such that $i' \in I$. We refer to this as "indirectly connected". Note that, by construction, i and i' will be conflicting. Also, note that some such i' must exist since if $i \notin I$, then it must have at least one neighbor i' in H such that $i' \in I$, since otherwise $\{i\} \cup I$ would be an independent set, contrary to the requirement that I is maximal.
- Finally, construct a solution to the primal integer problem by setting $x_{ij} = 1$ if client j is assigned to facility i, 0 otherwise, and $y_i = 1$ if facility $i \in I$, 0 otherwise.

We now show that this construction provides a feasible solution to the integral primal problem. By the above construction, each x_{ij} and y_i are either 0 or 1, so the integer constraint is satisfied.

Claim 1. *No client j contributes to two different open facilities.*

Proof. This follows from the construction of I as a maximal independent set of H, since otherwise, if some client j contributed to both facilities i and i' then, $\beta_{ij}, \beta_{i'j} > 0$, and, therefore, $\beta_{ij} \cdot \beta_{i'j} > 0$, so i and i' would be conflicting and there would be an edge between them in H, but then I would not be independent in H, contrary to construction. □

Claim 2. *If f_i is open, let S_i be the set of clients directly connected to i. Then*

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

Proof. Any client j with $\beta_{ij} > 0$ is directly connected to i, so $f_i = \sum_{j \in D} \beta_{ij} = \sum_{j \in S_i} \beta_{ij}$. (Note that there may also be directly connected clients with $\beta_{ij} = 0$ if they connected to the facility later.) Additionally, for any client j that is connected to i, the dual constraint $\alpha_j - \beta_{ij} \leq d_{ij}$ is tight, so $f_i = \sum_{j \in S_i} \beta_{ij} = \sum_{j \in S_i} (\alpha_j - d_{ij})$, so $f_i + \sum_{j \in S_i} d_{ij} = \sum_{j \in S_i} \alpha_j$, as claimed. □

Claim 3. *Let client j be indirectly connected to facility i, then $d_{ij} \leq 3\alpha_j$.*

Proof. If j is indirectly connected to i, then, by definition, in Phase 1 it must have been connected to some $i' \notin I$ and there must be an edge (i, i') in H. For i' to be connected to j, (i', j) must be tight and, therefore, $\alpha_j - \beta_{i'j} = d_{i'j}$ so $d_{i'j} \leq \alpha_j$. Since an edge (i, i') is in H iff there exists some $j' \in D$ such that $\beta_{ij'} \cdot \beta_{i'j'} > 0$, both $\beta_{ij'}$ and $\beta_{i'j'} > 0$, so both $d_{ij'}$ and $d_{i'j'} \leq \alpha_{j'}$, because, following the protocol in Phase 1, we can only have $\beta_{ij} > 0$ if the constraint $\alpha_j - \beta_{ij} \leq d_{ij}$ is tight. Let $t_i, t_{i'}$ be the times at which facilities i and i' become temporarily open in Phase 1. Since facility i became temporarily open before i', $\alpha_{j'} = t_{i'} \leq t_i = \alpha_j$. Therefore, since $d(i, j)$ is a metric, the triangle inequality holds so that $d_{ij} \leq d_{ij'} + d_{i'j'} + d_{i'j} \leq \alpha_{j'} + \alpha_{j'} + \alpha_j \leq 3\alpha_j$, so $d_{ij} \leq 3\alpha_j$, as claimed. □

Lemma 4. *The (integral) primal and dual solutions constructed by the algorithm satisfy:*

$$3 \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} d_{ij} x_{ij} \leq 3 \sum_{j \in D} \alpha_j$$

Proof. Let $D^{(d)}$ be the set of directly connected clients and $D^{(i)}$ be the set of indirectly connected clients, so that $D = D^{(d)} \cup D^{(i)}$ and let $S_i^{(d)}$ and $S_i^{(i)}$ be the set of all j directly and indirectly connected to i, respectively. Since, by construction, each client is connected to only a single facility, the $S_i^{(d)}$s and $S_i^{(i)}$s are disjoint and, if j is connected to $i \in I$, then $y_i = 1$, $x_{ij} = 1$ and $x_{i'j} = 0$ for $i' \neq i$. Then, from Claim 2, if j is directly connected to i, we have $f_i y_i + \sum_{j \in S_i^{(d)}} d_{ij} x_{ij} = f_i + \sum_{j \in S_i^{(d)}} d_{ij} = \sum_{j \in S_i^{(d)}} \alpha_j$. Therefore,

$$\sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in D^{(d)}} d_{ij} x_{ij} = \sum_{i \in I} \left(f_i y_i + \sum_{j \in S_i^{(d)}} d_{ij} x_{ij} \right) = \sum_{i \in I} \sum_{j \in S_i^{(d)}} \alpha_j = \sum_{j \in D^{(d)}} \alpha_j$$

so

$$3 \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in D^{(d)}} d_{ij} x_{ij} \leq 3 \sum_{i \in I} f_i y_i + 3 \sum_{i \in I} \sum_{j \in D^{(d)}} d_{ij} x_{ij} = 3 \sum_{j \in D^{(d)}} \alpha_j$$

If j is indirectly connected to i, then, from Claim 3,

$$\sum_{i \in I} \sum_{j \in D^{(i)}} d_{ij} x_{ij} = \sum_{i \in I} \sum_{j \in S_i^{(i)}} d_{ij} x_{ij} = \sum_{i \in I} \sum_{j \in S_i^{(i)}} d_{ij} \leq \sum_{i \in I} \sum_{j \in S_i^{(i)}} 3\alpha_j = 3 \sum_{j \in D^{(i)}} \alpha_j$$

Therefore,

$$\begin{aligned} 3 \sum_{i \in I} f_i y_i + \sum_{i \in I, j \in D} d_{ij} x_{ij} &= 3 \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in D^{(d)}} d_{ij} x_{ij} + \sum_{i \in I} \sum_{j \in D^{(i)}} d_{ij} x_{ij} \\ &\leq 3 \sum_{j \in D^{(d)}} \alpha_j + 3 \sum_{j \in D^{(i)}} \alpha_j \leq 3 \sum_{j \in D} \alpha_j \end{aligned}$$

Since $x_{ij}, y_i = 0$ for $i \notin I$,

$$3\sum_{i\in F} f_i y_i + \sum_{i\in F, j\in D} d_{ij}x_{ij} = 3\sum_{i\in I} f_i y_i + \sum_{i\in I, j\in D} d_{ij}x_{ij} \leq 3\sum_{j\in D}\alpha_j$$

as claimed. □

Together, these results give:

Theorem 5. *The (integral) primal and dual solutions constructed by the algorithm satisfy:*

$$OPT(IP) \leq cost(x,y) \leq 3\sum_{j\in D}\alpha_j \leq 3\cdot OPT(D) \leq 3\cdot OPT(P) \leq 3\cdot OPT(IP)$$

Proof.

$$\begin{aligned} OPT(IP) \leq cost(x,y) &= \sum_{i\in F} f_i y_i \ + \sum_{i\in F, j\in D} d_{ij}x_{ij} \leq 3\sum_{i\in F} f_i y_i \ + \sum_{i\in F, j\in D} d_{ij}x_{ij} \leq 3\sum_{j\in D}\alpha_j \\ &\leq 3\cdot OPT(D) \leq 3\cdot OPT(P) \leq 3\cdot OPT(IP) \end{aligned}$$

□

Corollary 6. *The Primal-Dual algorithm provides a 3-approximation for the Metric Uncapacitated* FACILITY LOCATION *Problem.*

References

[1] G. B. Dantzig, L. R. Ford, and D. R. Fulkerson. A primal-dual algorithm for linear programs. In H. W. Kuhn and A. W. Tucker, editors, Linear Inequalities and Related Systems, pages 171181. Princeton University Press, Princeton, NJ, 1956.

[2] Kamal Jain and Vijay V. Vazirani. Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation. Journal of the ACM, Vol. 48, No. 2, March 2001, pp. 274-296.