CS 261 Problem Set 2

Luis A. Perez

Problem 7

Solution:

(a) Let us consider a connected undirected graph $G = (V, E)$ in which each edge e has a weight w_e and the following linear program P with decision variables $\{x_e\}_{e \in E}$:

$$\max \quad \sum_{e \in E} w_e x_e \tag{1}$$

$$\tag{2}$$

subject to

$$\sum_{e \in F} x_e \leq |V| - \kappa(F) \quad \text{for all } F \subseteq E \tag{3}$$

$$\sum_{e \in E} x_e = |V| - 1 \tag{4}$$

$$x_e \geq 0 \quad \text{for all } e \in E \tag{5}$$

We wish to show that the spanning trees of G are in an objective function-preserving one-to-one correspondance with the 0-1 feasible solutions of P.

Lemma 0.1. *All 0-1 feasible solutions correspond to a spanning tree in G.*

Proof. Suppose we have a feasible 0-1 solution to P, $\{x_e\}_{e \in E}$ such that $x_e \in \{0, 1\}$. Then construct the graph $T = (V, E')$ where $E' = \{e \mid x_e = 1\}$. By 4, we know that $|E'| = |V| - 1$. Combining 5 when $F = E'$ with 4, we note the following:

$$\begin{aligned}
|V| - 1 &= \sum_{e \in E} x_e && \text{(By 4)} \\
&= \sum_{e \in E'} x_e + \sum_{e \in (E \setminus E')} x_e && \text{(Seperating into two sets)} \\
&= \sum_{e \in E'} x_e && \text{(Second term is all 0s)} \\
&\leq |V| - \kappa(E') && \text{(By 4 when } F = E'\text{)} \\
\implies \kappa(E') &\leq 1
\end{aligned}$$

This shows that the subgraph $T = (V, E')$ contains $n-1$ edges and only a single-connected component of G. As such, it must be a spanning tree. ☐

Lemma 0.2. *All spanning trees in G correspond to a feasible 0-1 solution of the linear program P.*

Proof. Suppose we have a spanning tree $T = (V, E')$ of G. Then let us define the solution to P as:

$$x_e = \begin{cases} 1 & e \in E' \\ 0 & \text{otherwise} \end{cases}$$

It's immediate that this satisfies 5. We know that $|E'| = |V| - 1$ since T is a spanning tree, as such 4 is also satisfied with the above. Let us now consider some subset $F \subseteq E$. We have:

$$\begin{aligned}
\sum_{e \in F} x_e &= \sum_{e \in F \cap E'} x_e + \sum_{e \in F \cap (E \setminus E')} x_e && \text{(Splitting into edges in } T \text{ and edges not in } T\text{)} \\
&= \sum_{e \in F \cap E'} x_e && \text{(The second sum consists of just 0s)} \\
&= \sum_{e \in E'} x_e - \sum_{e \in E' \setminus F} x_e && \text{(Rewriting set intersection as subtraction)} \\
&\leq (|V| - 1) - (\kappa(F) - 1) && \text{(See below.)} \\
&\leq |V| - \kappa(F)
\end{aligned}$$

We now describe the second to last line in more detail. It's clear that $\sum_{e \in E'} x_e = |V| - 1$, since $|E'|$ is a spanning tree. Next, note that $\sum_{e \in E' \setminus F} x_e$ is simply counting the number of edges in E' (in our spanning tree T) that are **not** in F. We can lowerbound this by $\kappa(F) - 1$, since for each additional disconnected component in (V, F), there must be an edge $e \in E'$ not contained in F that connects that component, contributing 1 to our sum. This implies that $\sum_{e \in E' \setminus F} x_e \geq \kappa(F) - 1$. ☐

Corollary 0.2.1. *With the above two lemmas, we've shown both directions of the one-to-one correspondands between spanning trees in G and 0-1 feasible solutions of P.*

Lemma 0.3. *The 1-1 correspondance above is objective function-preserving.*

Proof. To see this, note that both proposals above essentially use x_e as an indicator variable for whether or not e is included in our spanning tree. As such, the maximization of $\sum_{e \in E} w_e x_e$ in our linear program P maps directly to weight of the corresponding spanning three T. ☐

(b) We use the formula proposed in class to generate the dual. We will have $|E|$ constraints (one for each edge) and one variable for each of the contraints in the primal (yes, we will have a ton of variables – in fact, we'll have $2^{|E|}$ variables) which we label $\{y_F\}_{F \subseteq E}$.

The dual linear program, called D, is given by:

$$\min \sum_{F \subseteq E} (|V| - \kappa(F)) y_F \tag{6}$$

$$\tag{7}$$

subject to

$$\sum_{F \subseteq E : e \in F} y_F \geq w_e \quad \text{for all } e \in E \tag{8}$$

$$y_E \in \mathbb{R} \tag{9}$$

$$y_F \geq 0 \quad \text{for all } F \subset E \tag{10}$$

Note that 6 is simpler than one would initially think, primarily due to the fact that 4 is partially implied by 3, and that we're taking a linear combination of all constraints so there's no need to include a similar constraint multiple times.

The above is the dual D of our original program P. The constraint guarantees that we assign a value to each subset of edges such that the total weight on the edge is sufficient (w_e). The objective is then to minimize the assigned weights for each subset where subsets leading to a more disconnected graph contribute less weight.

(c) The complementatary slackness conditions are as follows for the primal and dual programs, P and D:

- Whenever $x_e \neq 0$, $\{y_F\}$ satisfies the e-th constraint of D with equality.
  For our specific problem, this means that when we select e as an edge to include in our maximum spanning tree of G, the sum of the weights we assign to each subsets F containing e is exactly equal to the edge's weight, w_e.
- Whenever $y_F \neq 0$, $\{x_e\}$ satisfies the F-th constraint of P with equality.
  For our problem , this means that when we assign some non-zero weight to the subset of edges $F \subset E$, then $\sum_{e \in F} x_e = |V| - \kappa(F)$.

(d) We now seek to proof that the modified Kruskal's algorithm proposed is correct (eg, it finds the maximal spanning tree in G).

Lemma 0.4. *Kruskal's algorithm produces a feasible solution to the dual program, D, in (b).*

Proof. We show how to use Kruskal's algorithm to propose feasible values for y_F. Suppose we run Kruskal's on our graph G. Then let $\{e_1, \cdots, e_{n-1}\}$ be the $n-1$

edges selected by Kruskal's algorithm with corresponding weights $\{w_1, w_2, \cdots, w_{n-1}\}$ in the order selected (note this implies that $w_i \geq w_{i+1}$). Define the set $I_{e_i} = \{e \mid \text{e was considered and rejected as a candidate after } e_i \text{ but before } e_{i+1}\}$.

Then define the following sets:

$$\begin{aligned}
F_1 &= \{e_1\} \cup I_{e_1} \\
F_2 &= F_1 \cup \{e_2\} \cup I_{e_2} \\
F_3 &= F_2 \cup \{e_3\} \cup I_{e_3} \\
&\vdots \\
F_{n-2} &= F_{n-3} \cup \{e_{n-2}\} \cup I_{e_{n-2}}
\end{aligned}$$

with corresponding values y_F given by:

$$y_{F_i} = w_i - w_{i+1} \geq 0 \qquad \text{(Since } w_i \geq w_{i+1}\text{, satisfying 10)}$$

Finally, define the set $F_{n-1} = E$ with corresponding variable $y_{F_{n-1}} = w_{n-1} = y_E$.

We claim that the above is a feasible solution to the linear program defined in (b). First note that:

$$y_E = w_{n-1} \in \mathbb{R}$$

which satisfies 9. Above we already showed 5 is satisfied. So now we just need to show 8 is also satisfied. To see this, fix some edge e and then note:

$$\begin{aligned}
\sum_{F \subseteq E : e \in F} y_F &= \sum_{F_k : e \in F_k} y_{F_k} && \text{(Only } y_{F_k} \neq 0\text{)} \\
&= w_{n-1} + \sum_{k=i}^{n-2} (w_k - w_{k+1}) \\
& && \text{(Where } i \text{ corresponds to the first set } F_i \text{ where } e \text{ appears)} \\
&= w_i && \text{(telescoping series simplifies to just } w_i\text{)} \\
&\geq w_e && \text{(By construction since } e \text{ is either } e_i \text{ or was processed after)}
\end{aligned}$$

The above shows that the proposed solution is a feasible solution of (b). □

We now show that the solution is in fact optimal. We do this by showing that the slackness conditions hold for the above proposed solution to the dual. From lecture, we know that if both conditions hold and both the primal and the dual are feasible, then the proposed solution is optimal.

Lemma 0.5. *Whenever $x_e \neq 0$ (eg, Kruskal's selects e as part of the output spanning tree), then $\{y_F\}$ satisfies the e-th constraint of D with equality.*

Proof. We proof this directly. Suppose we have e such that $x_e \neq 0$. This implies that Kruskal's algorithm selected e when processing it. As such, $e = e_i$ for some i in the construction above. We then have:

$$\begin{aligned}
\sum_{F \subseteq E : e \in F} y_F &= \sum_{F_k : e \in F_k} y_{F_k} && \text{(Only } y_{F_k} \neq 0\text{)} \\
&= w_{n-1} + \sum_{k=i}^{n-2} (w_k - w_{k+1}) && \text{(Expanding to all sets)} \\
&= w_i && \text{(Telescoping series simplifies to just } w_i\text{)} \\
&= w_e && \text{(By construction since } e \text{ is } e_i\text{)}
\end{aligned}$$

As such, we equality in the e-th constraint. □

Lemma 0.6. *Whenever $y_F \neq 0$, $\{x_e\}$ satisfies the F-th constraint of P with equality.*

Proof. Let us consider all $y_F \neq 0$ in our proposed solution to the dual from above. Note that we only have $n-1$ such sets, defined above as $F_1, \cdots, F_{n-1}$. Then we have:

$$\begin{aligned}
\sum_{e \in F_i} x_e &= i && \text{(Each new set adds one edge output by Kruskal's)} \\
&= |V| - (|V| - i) && \text{(Adding 0)} \\
&= |V| - \kappa(F_i) && \text{(See below)}
\end{aligned}$$

The last line follows from the fact that $G = (V, F_i)$ is the subgraph containing only the first i edges forming the a spanning tree of G (which means it contains $i+1$ vertices) and edges which led to cycles. As such, $\kappa(F_i) = |V| - i$.

We have therefore shown that the F-th constraint is satisfied with equality, concluding our proof. □

Putting the three lemmas together, using the results from class we now know that the output from Kruskal's is the maximal spanning tree since it the optimal solution to the first linear program.

(e) Let us consider a connected directed graph $G = (V, E)$ with non-negative cost c_e and the following linear program P with decision variables $\{x_e\}_{e \in E}$:

$$\min \quad \sum_{e \in E} c_e x_e \tag{11}$$

$$\tag{12}$$

subject to

$$\begin{aligned}
\sum_{e \in \delta^+(S)} x_e \geq 1 \quad & \text{for all } S \subseteq V \text{ with } s \in S, t \notin S && (13) \\
x_e \geq 0 \quad & \text{for all } e \in E && (14)
\end{aligned}$$

We wish to show that show that every simple s-t path of G corresponds to a 0-1 feasible solution of above linear program, P

Lemma 0.7. *All simple s-t paths of G correspond to a 0-1 feasible solution of P with the same objective function values.*

Proof. Suppose we have a simple s-t path $p \in G$. Then for each $e \in p$, let $x_e = 1$ and $x_e = 0$ otherwise. This satisfies $x_e \geq 0$ for all $e \in E$. Furthermore, since it is an s-t path, for all $S \subseteq V$ where $s \in S$ and $t \notin S$, the path must have at least one edge that crosses out of S. As such, 13 is also satisfied.

Furthermore, the cost of the path is given by $\sum_{e \in E} c_e x_e$, which is the same objective value as the linear program. □

This concludes our proof.

(f) The dual of the above linear program P is given by D below, where we introduce $2^{|V|-2}$ variables y_S.

$$\max \sum_{S \subseteq V : s \in S, t \notin S} y_S \tag{15}$$

$$\tag{16}$$

subject to

$$\sum_{S \subseteq V : s \in S, t \notin S, e \in \delta^+(S)} y_S \leq c_e \quad \text{for all } e \in E \tag{17}$$

$$y_S \geq 0 \quad \text{for all } S \subseteq V, \text{ st. } s \in S, t \notin S \tag{18}$$

We're essentially trying to maximize the weights assigned to subsets S constrained by the fact that for all edges, the weight assigned cannot exceed the cost of the edge crossing the s-t path.

(g)
- Whenever $x_e \neq 0$, $\{y_S\}$ satisfies the e-th constraint of D with equality.

  For our specific problem, this means that when we select e as an edge to include in our simple s-t path, the sum of the weights we assign to each subsets S containing the left end-point of e but not the right-endpoint is exactly equal to the edges cost.
- Whenever $y_S \neq 0$, $\{x_e\}$ satisfies the S-th constraint of P with equality.

  For our problem , this means that when we assign some non-zero weight to the subset of vertices $S \subseteq V$, then $\sum_{e \in \delta^+(S)} x_e = 1$ so there is exactly one edge crossing the cut $S, (V \setminus S)$ in our path.

(h) We let p denote the s-t path returned by Dijkstra's algorithm and prove that the solution to the linear program in (e) corresponding to p is in fact an optimal solution to that linear program, by exhibitinga feasible solution to the dual program in (f) such that the complementary slackness conditions hold.

Lemma 0.8. *The path p is a feasible solution to the primal program in (e).*

Proof. This follows from (e) since p is a simple s-t path and so corresponds to a feasible 0-1 solution that is objective preserving of P. □

Lemma 0.9. *The path p is a feasible solution to the dual program in (f).*

Proof. We show this by converting the path p into feasible values y_S. Suppose we run Dijkstra's on our graph G. Then let $\{v_0 = s, v_1, \cdots, v_{d_s(t)-1}, v_{d_s(t)} = t\}$ where $d_s(t)$ is the number of hops along the s-t path p be the vertices making up our path p as selected by Dijkstra's algorithm in the order they appears in the path. Define the set $I_{v_i} = \{v \in V \mid v_i \text{ was popped from the queue after } v_i \text{ but before } v_{i+1}\}$ (the intermediate nodes considered by Dijkstra's).

Then we can define the following subsets $S_i \subseteq V$ as follows:

$$\begin{aligned}
S_0 &= \{s\} \\
S_1 &= S_0 \cup I_{v_0} \cup \{v_1\} \\
S_2 &= S_1 \cup I_{v_1} \cup \{v_2\} \\
&\vdots \\
S_{d_s(t)-1} &= S_{d_s(t)-2} \cup I_{v_{d_s(t)-2}} \cup \{v_{d_s(t)-1}\}
\end{aligned}$$

We assign each of the above sets the corresponding non-zero values y_S given by:

$$y_{S_i} = c_{v_i, v_{i+1}} \qquad \text{(The cost of the edge along our s-t path that sticks out of } S_i\text{)}$$

We claim that the above is a feasible solution to the linear program defined in (e).

$$y_{S_i} \geq 0 \qquad \text{(Edge costs are non-negative)}$$

which satisfies 18. We now show that we also satisfiy 18. Fix some edge $e \in E$ with endpoints (u, v). Let $S_u \in \{S_i\}$ as defined above be the smallest set containing u. If no such set exists, then we immediately have:

$$\begin{aligned}
\sum_{S \subseteq V : s \in S, t \notin S, e \in \delta^+(S)} y_S &= \sum_{S_i : e \in \delta^+(S_i)} y_{S_i} && \text{(Only non-zero values)} \\
&= 0 \leq c_e && \text{(Since } u \notin S_i \text{ for all } i\text{)}
\end{aligned}$$

Similarly, let $S_v \in \{S_i\}$ be the smallest set containing v. If $S_v \subset S_u$, then the edge $e = (u, v)$ does not stick out of any set S_i (it is fully contained in S_u), and by a similar arguments as above, we must have:

$$\sum_{S \subseteq V : s \in S, t \notin S, e \in \delta^+(S)} y_S = 0 \leq c_e$$

As such, let us suppose that $S_u \subset S_v$. We know that the edge (u, v) exists. As such, we must have:

$$\begin{aligned} \sum_{S \subseteq V : s \in S, t \notin S, e \in \delta^+(S)} y_S &= \sum_{S_u, S_{u+1}, \cdots, S_{v-1}} y_{S_i} && \text{(Sum over only sets until } v \text{ is added)} \\ &= \sum_{v_i = u}^{v-1} c_{v_i, v_{i+1}} \\ & \quad \text{(The length of the path selected by Dijkstra's from } u \text{ to } v\text{)} \\ &\leq c_e \end{aligned}$$

To see why the above inequality holds, consider the path $u \to v$ as constructed by Dijkstra's and the edge (u, v). By definition of Dijktra's, it must be the case that every time we add a node to the path from $u \to v$, we have that the cost along this path is less than the cost along the more direct edge (u, v). Otherwise, we would have popped v out of the queue and connected $u \to v$ directly.

This concludes our proof showcasing a feasible solution to the dual program. □

Lemma 0.10. *The feasible solutions exhibited above satify the complementary slackness conditions.*

Proof. Let us take Dijktra's output path as our solution as outlined above for both the dual and primal LPs. Then suppose $x_e \neq 0$, which means the edge $e = (v_i, v_{i+1})$ is part of the s-t path output by the algorithm (same definition as above). Let us consider the corresponding $\{y_S\}$. We note:

$$\begin{aligned} \sum_{S \subseteq V : s \in S, t \notin S, e \in \delta^+(S)} y_S &= y_{S_i} && \text{(The edge sticks out of only } S_i\text{)} \\ &= c_(v_i, v_{i+1}) && \text{(By definition)} \end{aligned}$$

The reason for the above is that if the edge $e = (v_i, v_{i+1})$ was selected by Dijkstra's, then we must have that $v_i \in S_i$ and $v_{i+1} \in S_{i+1}$. So there is only a single set from which e sticks out.

Now let us suppose that $y_S \neq 0$, which means $S \subseteq V$ is one of the S_i constructed above. Then let us consider:

$$\sum_{e \in \delta^+(S_i)} x_e = 1$$

This follows almost immediately since given a set S_i, the selected edge is (v_i, v_{i+1}). Since the path is a simple s-t path, we cannot have any cycles so we never return to any vertices in S_i, expanding the path only from the endpoints v_{i+1}. □

Putthing the above together, from lecture we know that the path proposed by Dijkstra's is a feasible solution for both the primal and the dual and it also satisfies the slackness conditions. As such, it must be the case that the solution is optimal.

Problem 8

Solution:

(a) Let us take $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \gtrdot$. We assume the version of Farka's Lemma stated in lecture. Then note the following:

Lemma 0.11. *If (i) holds, then (iii) also holds.*

Proof. Assuming (i) holds, we have that $\exists \mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \geq 0$. Then note that this $\mathbf{x}$ trivially satisfies (iii) since it is a less constrained statement. □

Lemma 0.12. *If (i) holds, then (iv) does not hold.*

Proof. For the sake of contradiciton, suppose (i) is true and so is (iv). Then note the following:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} && \text{(By (i))} \\ \implies y^T A x &= y^T b && \text{(Muliply by through } y^T \text{ using } y \text{ from (iv))} \end{aligned}$$

However, we have:

$$\begin{aligned} y^T A x &= (y^T A) x \\ &= 0x && \text{(By (iv))} \\ &= 0 \qquad = y^T b < 0 && \text{(By (iv))} \end{aligned}$$

A contradiction. Therefore we must have that (iv) does not hold. □

Lemma 0.13. *If (ii) holds, then (iii) cannot hold.*

Proof. For the sake of contradiction, suppose (ii) is true and so is (iii). Then note the following:

$$\begin{aligned} \mathbf{y}^T \mathbf{A} &\geq 0 && \text{(By (ii) for some } \mathbf{y}\text{)} \\ \implies \mathbf{y}^T \mathbf{A} \mathbf{x} &\geq 0 && \text{(Mulitply through by } \mathbf{x} \text{ implied by (iii))} \end{aligned}$$

However, we also have:

$$\begin{aligned} \mathbf{y}^T \mathbf{A} \mathbf{x} &\leq \mathbf{y}^T b && \text{(Implied by (iii))} \\ &< 0 && \text{(Implied by (ii))} \end{aligned}$$

However, this is a contradiction. As such, it must be the case that if (ii) holds, then (iii) does not hold. □

Lemma 0.14. *If (ii) holds, then so does (iv).*

Proof. This is immediate by inspection. □

Putting the above results together, we have now deduced the second version of Farka's Lemma.

(b) We follow the provided hint. We let γ^* denote the optimal objective function value of the dual linear program. By weak duality (from lecture), we know that γ^* is an upper bound on the optimal value of P. As such, what we need to proof is that there exists a solution to P that matches this optimal value.

We do this by augmenting P by adding the constraint $c^T x \geq \gamma^*$ (also written as $-c^T x \leq -\gamma^*$, which essentially restricts the solution to be only those which achive the objective value of at least γ^*.

All we need to show is that this new linear program is still feasible, assuming P was feasible to begin with.

Looking at our modified linear program, we have the following:

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} \\ \mathbf{c} \end{bmatrix} \in \mathbb{R}^{(m+1)\times n}$$
$$\mathbf{b}' = \begin{bmatrix} \mathbf{b} \\ -\gamma^* \end{bmatrix} \in \mathbb{R}^{m+1}$$

where the objective remains the same and the constraints are now specified by $\mathbf{A}'\mathbf{x} \leq \mathbf{b}'$

Lemma 0.15. *There does not exists a $y \in \mathbb{R}^{m+1}$ such that $y \geq 0$, $y^T A' = 0$ and $y^T b' < 0$.*

Proof. We proof by contradiction. Suppose such a $y \in \mathbb{R}^{m+1}$ satisfying the above did exist. Let us define $y_m \in \mathbb{R}^m$ as the vector containing the first m elements of y.

Then this implies that the dual program is unbounded. To see why, by assumption, let us take any $y_D \in \mathbb{R}$ as a feasible solution of D. Then note that we can now construct another solution:

$$y'_D = y_D + r y_m$$

with $r > 0$. We claim that this is also a feasible solution, and that this solution is unbounded.

To see the feasibility, note that we have:

$$\begin{aligned}
y_D' &= \underbrace{y_D}_{\geq 0} + \underbrace{cy_m}_{\geq 0 \text{ since } c \geq 0 \text{ and } y \geq 0} \\
&\geq 0 \\
A^T y_D' &= A^T(y_D + ry_m) \\
&= A^T y_D + rA^T y_m \\
&= c + 0 && (A^T y_D = c \text{ and } A^T y_m = y_m^T A = 0) \\
&= c
\end{aligned}$$

As such, the new solution satisfies all of the constraints for the dual D. However, note that the values of this new solution is given by:

$$\begin{aligned}
b^T y_D' &= b^T(y_D + ry_m) \\
&= b^T y_D + rb^T y_m
\end{aligned}$$

Since $b^T y_m < 0$ and we can chose $r > 0$ freely, the value above has no lower bound (we can pick r as large as we like to make it as negative as we like). As such, this implies that the dual program has no optimal solution, which implies the primal program is not feasible (there is no upper bound on it). This is a contradiction on our feasibility assumptions. □

Lemma 0.16. *The modified primal linear program with the additional constraint is feasible.*

Proof. To see why, simply apply Farka's Lemma. By the previous lemma combined with Farka's lemma, we know that there exists $x \in \mathbb{R}^n$ such that $A'x = b'$, with x unrestricted, which is precisely what it means for our modified primal program to be feasible. □

Putting everything together, the above implies that the primal linear program and its dual, if both are feasible, have equal optimal objective function values since the value of the dual is equivalent to an upper bound on the primal (by weak duality), and the the lemma's above, this upper bound from the dual is achievable by the primal.

Problem 9

Solution:

(a) We formulate the multi-commodity flow problem as a linear program as follows. Let $\mathcal{P}$ be the set of all simple s_i-t_i paths in G.

$$\max \quad \sum_{P \in \mathcal{P}} f_P \tag{19}$$

$$\tag{20}$$

subject to

$$\sum_{P \in \mathcal{P} : e \in E} f_P \leq u_e \quad \text{for all } e \in E \tag{21}$$

$$f_P \geq 0 \quad \text{for all } P \in \mathcal{P} \tag{22}$$

Intuitively, this linear program maximizes the total flow (by summing flows f_P along simple s-t paths) subject to the constraint that a flow must be positive and that for any edge, the total flow along that edge cannot exceed its capacity.

Lemma 0.17. *There is a 1-1 correspondance between the optimal solution to the above linear program and the optimal solution to the multi-commodity flow problem.*

Proof. To see this, we proof (1) every feasible solutions corresponds to a multi-commodity flow (2) every multi-commodity flow corresponds to a feasible solution and (3) corresponding feasible solutions and multi-commodity flows have the same objective value.

To see (1), take some feasible solution $\{f_P\}_{P \in \mathcal{P}}$. Then let f_P be the corresponding flow value along the simple s-t path P. This is a valid flow on G. Consider an edge $e \in E$. Then note that by 21, the capacity constraints are immediately satisfied. Furthermore, since the flow consists of simple s-t path flows, the conservation constraints are automatically satisfied as well.

For (2), consider a multi-commodity flow. Then consider the total flow along each s-t path, and assign to the variable f_P the total flow along that specific s-t path P. Then by the fact that flows satisfy capacity constraints, 21 must be satisfied. Furthermore, all flows are positive, so 22 is also satisfied.

Finally, for (3) simply note that the value of a flow corresponding to a feasible solution is precisely the sum of the flows along the simple s-t paths, which is the same as the objective function. □

(b) We now take the dual of the primal linear program above. Following the formula from lecture, we introduce variables f_e for each edge. Then we have:

$$\min \quad \sum_{e \in E} u_e f_e \tag{23}$$

subject to

$$\sum_{e \in P} f_e \geq 1 \quad \text{for all } P \in \mathcal{P} \tag{24}$$

$$f_e \geq 0 \quad \text{for all } e \in E \tag{25}$$

We now proof that every optimal 0-1 solution of this dual is the characteristic vector of the a minimum-value multicut.

Lemma 0.18. *Every optimal 0-1 solution to the dual is a characteristic vector of a minimum multi-cut.*

Proof. To see this, suppose we have some 0-1 solution to the above LP D. Then define the set $F = \{e \mid f_e = 1\} \subseteq E$ and let this be our corresponding multi-cut.

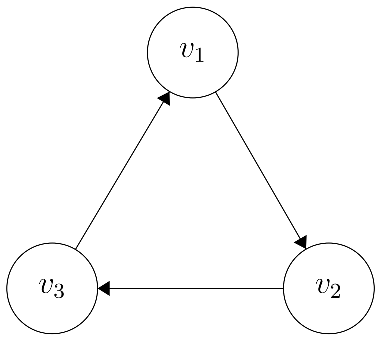
Then note that the value of F is given by $\sum_{e \in F} u_e = \sum_{e \in E} u_e f_e$ which is exactly the same as our dual's objective, so this transformation is objective preserving.

Furthermore, by 24, we know for every possible s_i-t_i path in G, F must contain at least one edge along that path. This implies that there are no s_i-t_i paths in $G = (V, E \setminus F)$, which means F is a valid multi-cut.

As such, it follows that every optimal 0-1 solution to the dual can be mapped into the characteristic vector of a minimum-value multicut. $\square$

(c) We show by example that the optimal solution to this dual linear program can have objective function value strictly smaller than that of every 0-1 feasible solution.

To see this, consider the graph $G = (V, E)$ where $V = \{v_1, v_2, v_3\}$ and $E = \{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}$ where we have $u_e = 1$ for all edges. For concreteness, we draw the graph structure below.



Then let our source sink-pairs be given by $(v_1, v_3), (v_2, v_1), (v_3, v_2)$. This means that we have $\mathcal{P} = \{v_1 \to v_2 \to v_3, v_2 \to v_3 \to v_1, v_3 \to v_1 \to v_2\}$ as our three s_i-t_i paths (one for each commodity).

Lemma 0.19. *The optimal 0-1 solution for our LP from the graph G and commodities as above has value 2.*

Proof. We can see this directly. Note that our constraints are explicitly give as:

$$\begin{aligned} f_{(v_1,v_2)} + f_{(v_2,v_3)} \geq 1 \\ f_{(v_2,v_3)} + f_{(v_3,v_1)} \geq 1 \\ f_{(v_3,v_1)} + f_{(v_1,v_2)} \geq 1 \end{aligned}$$

To satisfy all constraints, we must have that each of the three constraints has one of its variables set to 1. However, note that no variable is shared across all three constraints, so we must set at least 2 distinct variables to 1. Since each constraint shares a variable with one other, setting any two distinct variables to 1 satisfies all the constraints.

As such, the minimum possible value restricting ourselves to integer solutions is 2. □

Lemma 0.20. *There exists a non-integer solution to our LP that achives a lower objective value that the optimal 0-1 solution.*

Proof. We give such a solution. Consider the solution given by $f_e = \frac{1}{2}$ for all $e \in E$. Then note that all of the s-t path constrains are satisfied with equality. As such, this is a feasible solution.

However, the objective value achieved by this solution is $\frac{3}{2} < 2$, a lower objective value than the optimal 0-1 solution. □

From the above, we can see that there isn't a max-flow/min-cut theorem for multicommodity flows. The reason for this is that there exist graphs (such as the above) where the maximum multicommodity flow is non-integral (in the case above, we see that maximum multicommodity flow is $\frac{3}{2}$ by sending $\frac{1}{2}$ flow along each edge), which cannot correspond the the value of the minimum multi-cut, since the multi-cut must be integral (since it's value is the sum of integral capacitoes). For our example, the minimum multi-cut is given by 2.

Problem 10

Solution:

(a) We proof this by induction on the number of recursive steps. We begin by showing that in the two base cases of the algorithm, we return the optimal solution x^*.

Base case # 1

Proof. In this base case, we're given a non-degenerate linear program where $|C_2| = n$. This means that we have exactly n equality constraints as part of the linear program. Furthermore, by assumptions, we know that the linear program has a unique optimal solution, and all feasible solutions satisfy at most n equality constraints. As such, returning the unique point which satisfies every constraint C_2 with equality (since there are n) implies that this is the unique optimal solution for this non-degenerate linear program. □

Base case # 2

Proof. Similar to the above, in this base case we are given a linear program consisting of exactly n constraints, some with equality and some without. Returning the solution to this LP will yield the optimal solution. Note that the constraints B have no impact given the assumption that every feasible solution has coordinages $|x_j| < M$. □

Recursive Step

Proof. We proof by induction on the size of $|C_1|$.

Our inductive hypothesis is that our algorithm returns the optimal solution x^* to all linear programs with constraint set $|C_1| \leq k$. Note that our base cases already proofed this inductive hypothesis for sets $|C_1| \leq n$.

As such, let us suppose our inductive hypothesis holds for all linear programs with $|C_1| \leq k$. We will show that it also holds for all linear program with size $|C_1| = k+1$.

WLOG, we assume that $k + 1 > n$. As such, we entre the recursive step of our algorithm, and recurse with the set $C_1 \setminus \{i\}$. By our inductive hypothesis, the returned point x^* is an optimal solution for the LP defined by $C_1 \setminus \{i\}$ and C_2.

If $a_i^T x \leq b_i$, then note that x^* also satisfies our original linear program (by pure chance). Since the original linear program is strictly more contrained than the one we just solved (recall we removed an inequality constraint), it must be the case that x^* is also an optimal solution for the linear program defined by C_1, C_2 and must be optimal.

In the case where $a_i^T x > b_i$, we've found a solution which is not useful to us in the sense that it violates the contraint we dropped. However, consider for theoretical purpuses the dual program D to our modified linear program. The resulting optimality value x^* implies that in this dual, we must have $y_i \neq 0$. By the complementary slackness conditions (since we are assume our program has a unique optimal solution), it must be the case that the i-th constraint is therefore satisfied by equality. As such, recursing on the modified linear program $C_i \setminus \{i\}$ and $C_2 \cup \{i\}$ is valid, and by our inductive hypothesis will return the optimal solution x^* to the modified LP.

By the complementary slackness condition stated previously, this optimal solution x^* is also the optimal solution to our original linear program. □

(b) We proof that the expected number of recursive calls is given by:

$$T(m,s) = \begin{cases} 1 & \text{if } s = n \text{ or } m + s = n \\ T(m-1,s) + \frac{n-s}{m} \cdot T(m-1,s+1) + 1 \end{cases}$$

Proof. The fact that $T(m,s) = 1$ when $s = n$ or $m + s = n$ is immediately obvious. This corresponds to our two base cases in the algorith. We now focus on the second statement. Note that T is counting the number of expected recursive calls, given that our algorithm is randomized.

As such, let us consider the case where $|C_1| = m$ and $|C_2| = s$ and we have that $s \neq n$ and $m + s \neq n$. In this case, the algorithm always makes at least one recursive call (+1), now with inputs $m - 1$ and n. As such, the expected number of calls is at least $T(m-1,s) + 1$. However, in expectation, it makes one further recursive call with inputs $m - 1, s + 1$ in the case when the selected constraint i is not satified.

Given that we know that the linear program is non-degenerate and that $|C_1| = m$ and $|C_2| = s$, there can be at most $n - s$ constraints in C_1 that could become equality constraints (since every feasible solution has at most n equality constraints, and we've already accumulated $|C_2| = s$ of them).

As such, an upperbound on the expected number of recursive calls is given by:

$$\begin{aligned} T(m,s) \leq & \underbrace{1}_{\text{always at least 1}} \\ & + \underbrace{T(m-1,s)}_{\text{number of calls needed to solve } C_1 \setminus \{i\}, C_2} \\ & + \underbrace{\frac{n-s}{m}}_{\text{max constraints not yet equality}} \underbrace{T(m-1,s+1)}_{\text{number of call to solve } C_1 \setminus \{i\}, C_2 \cup \{i\}} \end{aligned}$$

This is precisely what we wanted to proof. □

(c) We proof that $T(m,0) = O(n! \cdot m)$.

Proof. As per the hint, we substitute $\delta = n - s$.

$$T(m, n-\delta) = \begin{cases} 1 & \text{if } \delta = 0 \text{ or } \delta = m \\ T(m-1, n-\delta) + \frac{\delta}{m} \cdot T(m-1, n-(\delta-1)) + 1 & \text{otherwise} \end{cases}$$

and define $S(m, \delta) = T(m, n-\delta)$ giving us the recurrence:

$$S(m, \delta) = \begin{cases} 1 & \text{if } \delta = 0 \text{ or } \delta = m \\ S(m-1, \delta) + \frac{\delta}{m} \cdot S(m-1, \delta-1) + 1 & \text{otherwise} \end{cases}$$

With this new recurrence, what we want to show is that $T(m, 0) = S(m, n) = O(n! \cdot m)$.

We do this by simulatenous induction on m and δ. We begin with the base case with $m = 0$. We show that for $m = 0$ and for all $0 \le \delta \le n \le m$, our running time is as specified (in particular, it is constant).

$$S(0,0) = 1 = O(n! \cdot m)$$

We now proceed.

Our inductive hypothesis is that $S(m, n) = O(n! \cdot m)$ for all $0 \le m \le i$ and $0 \le n \le j \le i$. We will show that this holds as well when $i' = i + 1$ and when $j' = j + 1$.

For the first case, suppose we now have $i' = i + 1$. For brevity, we only focus on the recursive case (if we hit a base case, the proof is trivial):

$$\begin{aligned} S(i', j') &= S(i+1, j) \\ &= S(i, j) + \frac{j}{m} S(i, j-1) + 1 && \text{(By definition)} \\ &= O(j! \cdot i) + \frac{j}{m} O((j-1)! \cdot i) && \text{(By inductive hypothesis)} \\ &= O(j! \cdot i) + \frac{i}{m} O(j!) && \text{(Associativity)} \\ &\le O(j! \cdot i) + O(j!) && (i \le m) \\ &\le O(j! \cdot (i+1)) \end{aligned}$$

The last line is exactly what we desired. As such, we've shown that the inequality holds for $i' = i + 1$ and $j' = j$.

Let is now consider the other variable, where we have $j' = j + 1$. In this case, we have the below. Let us assume $j < m - 1$ for simplicity.

$$\begin{aligned}
S(i', j') &= S(i, j+1) \\
&= S(i-1, j+1) + \frac{j+1}{m} \cdot S(i-1, j) + 1 && \text{(Definition)} \\
&= O\left((j+1)!(i-1)\right) + \frac{j+1}{m} \cdot S(i-1, j) + 1 \\
& && \text{(By initial inductive hypothesis)} \\
&= O\left((j+1)!(i-1)\right) + \frac{j+1}{m} O(j!(i-1)) + 1 \\
& && \text{(By result proved just above)} \\
&= O\left((j+1)!i - (j+1)! + \frac{i-1}{m}(j+1)!\right) && \text{(Expanding)} \\
&= O\left((j+1)!i\right) && (i - 1 \leq m)
\end{aligned}$$

With the above, we have shown that $S(i, j) = O(j! \cdot i)$. Putting everything together we have:

$$\begin{aligned}
T(m, 0) &= S(m, n) && \text{(By change of variables)} \\
&= O(n! \cdot m)
\end{aligned}$$

Concluding our proof. □

(d) Let us now fix the constant n. From the results in (c), we know that the proposed algorithm will conduct a total of $O(n! \cdot m)$ recursive calls. In each call, we will at most solve a linear program with n total constraints and n variables. As such, the running time of such a program is $O(f(n))$ for some function f.

Putting everything together, the total running time is given by $O(n! \cdot f(n) \cdot m)$, and since n is considered a fixed constant, this implies that the running time of the algorithm is $O(m)$.

Problem 11

Solution:

(a) We provide the following linear program. Our variables are $x_v \forall v \in V$ where we have $0 \leq x_v \leq 1$ (we don't worry about integrality yet). We also have $w_v \forall v \in V$ as the weight of the vertix v.

Then the linear program is as follows:

$$\max \quad \sum_{v \in V} x_v w_v \tag{26}$$

subject to

$$x_u - x_v \geq 0 \quad \text{for all } (u,v) \in E \tag{27}$$
$$0 \leq x_u \leq 1 \quad \text{for all } v \in V \tag{28}$$

Treating x_u as indicator variables, we can define our prefix set $P = \{v : x_v = 1\}$. As such, we see that the constraint $x_u - x_v \geq 0$ simply enforces the fact that for all edges, if $x_u \notin P$, then $x_v \notin P$ (and contrapositively, if $x_v \in P$ then $x_u \in P$). This implies that P has no edges "sticking into it' which is precisely the definition of a prefix of G. As such, all feasible 0-1 solutions of the above linear program are prefixes P of G. Furthermore, the objective function simply computes $\sum_{v \in V} x_v w_v = \sum_{v \in P} w_v$ which is the weight of the prefix P.

(b) We follow the provided proof sketch. Let us star with some feasible solution $\{x_v\}$ to our linear program above, and define for each $t \in [0,1]$ the set $S(t) = \{v : x_v \geq t\}$ and $W(t) = \sum_{v \in S(t)} w_v$.

Lemma 0.21. *For every $t \in [0,1]$, $S(t)$ is a prefix.*

Proof. To show that $S(t)$ is a prefix, we need to show that for all $(u,v) \in E$, $u \notin S(t) \implies v \notin S(t)$. Contrapositively, we need to show that $v \in S(t) \implies u \in S(t)$.

To see this, suppose $v \in S(t)$. Then we must have $x_v \geq t$. However, by 30, we know that $x_u - x_v \geq 0 \implies x_u \geq x_v \geq t$. As such, by definition of $S(t)$, we have that $u \in S(t)$. □

Lemma 0.22. *The objective function value of the linear program on $\{x_v\}$ is $\int_0^1 W(t)dt$*

Proof. We can proof this directly by simplifying the integral. We have:

$$\begin{aligned}\sum_{v\in V} x_v w_v &= \sum_{v \in V}\int_0^{x_v} w_v dt && (x_v = \textstyle\int_0^{x_v} dt)\\ &= \int_0^1 \left(\sum_{v\in S(t)} w_v\right) dt && \text{(Swapping sum and integral and adjusting bounds)}\\ &= \int_0^1 W(t) dt\end{aligned}$$

□

The last lemma above shows that $\sum_{v\in V} x_v w_v$ (a linear combination of our solution with weights w_v) is in-fact equivalent to $\int_0^1 W(t)dt$ which is a (continuous) linear combination of $W(t) = \sum_{v\in S(t)} w_v$ where all coefficients are non-negative and sum to 1 (the coefficients are $t \in [0,1]$). As such x must be a convex combination of integral solutions corresponding to the set $S(t)$.

With that, we conclude that when x is optimal, since this implies x is feasible, there must exists a corresponding integral solution with the same objective values.

(c) We define the linear program in a similar way to above as follows:

$$\max \quad \sum_{v\in V} x_v w_v \tag{29}$$

subject to

$$x_u - x_v = 0 \quad \text{for all } (u,v) \in E \tag{30}$$
$$0 \le x_u \le 1 \quad \text{for all } v \in V \tag{31}$$

Treating x_u as indicator variables, we can define our interval set as $I = \{v : x_v = 1\}$. As such, we see that the constraint $x_u - x_v = 0$ simply enforces the fact that for all edges, if $x_u \notin I$, then $x_v \notin I$. This implies that I has no edges "sticking into it' which is precisely the definition of a prefix of G. Furthermore, if $x_u \in I$ then $x_v \in I$. This implies that I has no edges "sticking out" of it, which is precisely the definition of a suffix of G. As such, I is both a prefix and suffix of G, which implies it is an interval.

Furthermore, the objective function simply computes $\sum_{v\in V} x_v w_v = \sum_{v\in I} w_v$ which is the weight of the interval I.

Problem 12

Solution:

(a) The vertices of F are simply the matrices $A \in F$ such that for all i, j, $a_{ij} = a_{ij}^{\min}$ or $a_{ij} = a_{ij}^{\max}$ since these matrices satisfy the constrains with equalities.

(b) We begin by showing that instead of solving the optimization problem with a constraint for every $A \in F$, it is enough to have a constrain just for the vertices of F.

Let $v(F)$ be the set corresponding to the verticies of F. We will show that every feasible solution $x \in \mathbb{R}$ with the relaxed constraints (constraints just on $v(F)$) is also a solution when considering all the contraints. The opposite direction (all feasible solution of the original LP are feasible in the less contrained LP) is immediate.

Lemma 0.23. *If x is a feasible solution when considering only constrains for $A \in v(F)$, then x is a feasible solution when considering all constraints for $A \in F$.*

Proof. We proof by construction. Suppose we have a feasible solution $x \in \mathbb{R}^n$ that satisfies all constrains for $A \in v(F)$. Then x also satisfies any other constrain $A' \in F$. Since $A' \in F$, we can write it as a convex combination of $A \in v(F)$. That is, we must have:

$$A' = \sum_{V \in v(F)} c_V V$$
$$\sum_{v \in V} c_V = 1$$
$$c_V \geq 0 \quad \text{for all } V \in v(F)$$

This is by definition of F (it is a polytope, as such, any point with F can be written as a convex combination of its vertices).

As such, we now have:

$$\begin{aligned}
A'x &= \left(\sum_{V \in v(F)} c_V V \right) x && \text{(As per above)} \\
&= \sum_{V \in v(F)} (c_V V x) \\
&= \sum_{V \in v(F)} (c_V \underbrace{Vx}_{\geq b}) && \text{(Since } x \text{ is a feasible solution and } V \text{ is a vertex contraint)} \\
&= \sum_{V \in v(F)} \underbrace{(c_V V x)}_{\geq b} && \text{(Since } c_V \geq 0) \\
&\geq b && \text{(Each term is greater than } b \text{, so sum is as well)}
\end{aligned}$$

The above shows that x also satisfies the contraint $A'x \geq b$. We therefore conclude with the fact that it is sufficient to solve the linear program with constrains just on the vertices of the polytope F. □

The resulting linear program is simply:

$$\min \quad c^T x \tag{32}$$

subject to

$$Ax \geq b \quad \text{for all } A \in v(F) \tag{33}$$

The LP has n decision variables (x) and a total of $m \cdot |v(F)|$ constraints. Note that $|v(F)| = 2^{mn}$ (each entry in a matrix can be either $a_{ij}^{\min}$ or $a_{ij}^{\max}$) and there are mn entries.

As such, the size of the linear program is n variables with $m2^{mn}$ constraints, which is not polynomial in the size of the input.

(c) We consider a fixed $x \in \mathbb{R}^n$ and one row of A (the i-th row). Then consider the following LP with $2n$ constraints and n variables a_{ij} for $j = 1, \cdots, n$.

$$\min \quad \sum_{j=1}^{n} a_{ij} x_j \tag{34}$$

subject to

$$a_{ij} \leq a_{ij}^{\max} \text{ for } j = 1, \cdots n \tag{35}$$

$$a_{ij} \geq a_{ij}^{\min} \text{ for } j = 1, \cdots n \tag{36}$$

Then in order to check if the fixed vector x satisfies the condition on the given row of A for all $A \in F$, do the following:

- Construct the above linear program and solve it, achieving optimal objective value γ_i^*.
- If $\gamma_i^* \geq b_i$, then return that x satisfies the condtion on the i-th row for all $A \in F$. Otherwise, return that it does not.

Lemma 0.24. *As long as $F \neq \emptyset$, the LP above is feasible.*

Proof. Since F is non-empty, $\exists A \in F$ which by definition of F must have that the i-th row satisfies all of the constraints in our LP. Therefore, the LP is feasible. □

Lemma 0.25. *The set of values of all feasible solution of the above LP is bounded.*

Proof. To see this, simply note that the variables have a fixed lower and upper bound (by the constraints). Since the objective is simply a linear combination of these variables, it must therefore also be bounded. □

As per the two lemma's above, since $F \neq \emptyset$ (let us assume this), the above LP is feasible and it's optimal solution value is bounded.

Lemma 0.26. *Let γ_i^* be the optimal solution to the above LP. Then if $\gamma_i^* \geq b_i$, we have that $a_i^T x \geq b$ for all $A \in F$ where a_i is the i-th row of A.*

Proof. This is direct. We know that $\sum_{j=1}^n a_{ij}x_j \geq \gamma_i^*$ for all a_{ij} satifying the LP's contraints. However, this is exactly the same as saying that $a_i^T x \geq \gamma_i^*$ for all $A \in F$ where a_i is the i-th row of A (by definition of F). However, we know that $\gamma_i^* \geq b$, and as such, we have $a_i^T x \geq \gamma_i^*$ for all $A \in F$. □

Lemma 0.27. *Similarly, if $\gamma_i < b_i$, we have that $\exists a_i$ such that $a_i^T x < b$.*

Proof. We proof the contrapositive. That is, if for all $A \in F$ we have that $a_i^T x \geq b_i$, then $\gamma_i \geq b_i$. To see this, simply note that the above LP considers only values for the i-th row of $A \in F$ for all A. As such, it must be the case that all $a_i^T x \geq b_i$, and in particular, we must have that the minimum value is greater or equal to b_i, concluding our proof. □

Putting the above together, we have that the LP we've defined correctly checks whether the fixed vector x satisfies the condition on the i-th row of A for all $A \in F$.

We now continue by constructing the dual of this program. The dual is given by the LP D below, where we have $2n$ variables $y_{i1}^-, \cdots, y_{in}^-$ and $y_{i1}^+, \cdots, y_{in}^+$ and n constraints.

$$\max \quad \sum_{j=1}^n (y_{ij}^- a_{ij}^{\min} + y_{ij}^+ a_{ij}^{\max}) \tag{37}$$

subject to

$$y_{ij}^- + y_{ij}^+ = x_j \quad \text{for all } j = 1, \cdots, n \tag{38}$$

$$y_{ij}^- \leq 0 \quad \text{for all } j = 1, \cdots, n \tag{39}$$

$$y_{ij}^+ \geq 0 \quad \text{for all } j = 1, \cdots, n \tag{40}$$

We are essentially decided how to split our x into two vectors y^- and y^+ such that $y^- \cdot a_i^{\min} + y^+ \cdot a_i^{\max}$ is maximized.

(d) From the linear program in (c), we know that the solution x is feasible if and only if there exists $y_{ij}^- \leq 0, y_{ij}^+ \geq 0$ such that:

$$y_{ij}^- + y_{ij}^+ = x_j$$

and

$$\sum_{j=1}^n (y_{ij}^- a_{ij}^{\min} + y_{ij}^+ a_{ij}^{\max}) \geq b_i$$

As such, we can write our original optimization problem more simply as;

$$\min \quad c^T x \tag{41}$$

subject to

$$y_{ij}^- + y_{ij}^+ - x_j = 0 \quad \text{for all } i, j \tag{42}$$

$$\sum_{j=1}^{n} (y_{ij}^- a_{ij}^{\min} + y_{ij}^+ a_{ij}^{\max}) \geq b_i \quad \text{for all } i \tag{43}$$

$$y_{ij}^- \leq 0 \quad \text{for all } i, j \tag{44}$$

$$y_{ij}^+ \geq 0 \quad \text{for all } i, j \tag{45}$$

This linear program solves our original program since it imposes that x must be a feasible solution, and it enforces that each row $a_i^T x \geq b_i$ for all $A \subseteq F$ as in our original problem.

However, we note that this linear program contains only $3mn + m$ constraints and $2mn + n$ variables, which means it is polynomial in size of the input.