Fall 2020
ECE 236A
Christina Fragouli
TAs: Mine Dogan and Kaan Ozkara

Linear Programming

Homework 4
Due: 9 am, Friday Nov. 20th

Problem 1 (`3 points`): Explain why the following proof for strong duality is flawed:

From week duality we have

$$d^* \leq p^*. \tag{1}$$

Since the dual of the dual problem is the primal problem, applying weak duality again gives

$$p^* \leq d^*. \tag{2}$$

From (1), (2), $p^* = d^*$.

Solution :

Let's say we have the following generic primal: min $c^T x$ $s.t.$ $Ax \leq b$, then we have the following dual: max $-b^T z$ $s.t.$ $A^T z + c = 0$. To apply weak duality as studied in the lecture, we need to put the dual problem in a minimization form, then we have $-\min\ b^T z$ $s.t.$ $A^T z + c = 0$. And the dual of the dual is: $-\max\ c^T x$ $s.t.$ $Ax \leq b$. Applying weak duality to dual in minimization form and dual of the dual in maximization form gives us:

$$-p^* \leq -d^*$$

This is the same as inequality (1). Hence, this does not prove strong duality.

Problem 2 (`3 points`):

The following are the max-flow LP and the min-cut ILP formulations as we've seen in class and the provided chapter notes (the variables are as described there):

$$\begin{array}{ll} \text{maximize} & f_{ts} \\ \text{subject to} & f_{ij} \leq c_{ij},\ \forall (i,j) \in \mathcal{E} \\ & \sum\limits_{j:(j,i)\in\mathcal{E}} f_{ji} - \sum\limits_{k:(i,k)\in\mathcal{E}} f_{ik} \leq 0,\ \forall i \in \mathcal{V} \\ & f_{ij} \geq 0,\ (i,j) \in \mathcal{E} \end{array}$$

$$\begin{array}{ll} \text{minimize} & c^T d \\ \text{subject to} & p_s - p_t \geq 1, \\ & d_{ij} - p_i + p_j \geq 0, \\ & d_{ij} \in \{0,1\}, \\ & p_i \in \{0,1\} \end{array}$$

(a) Prove that the constraint matrix of the max-flow LP is TUM.

(b) Prove that the dual of the max-flow LP gives the same solution as the ILP (we will derive the dual LP in class, you do not need to re-derive it).

<u>**Solution**</u>:

(a) We use a similar approach to that you used in the class. The max-flow LP can be written as

$$\begin{array}{ll} \text{maximize} & f_{ts} \\ \text{subject to} & \underbrace{\begin{bmatrix} \mathbf{0}_{|\mathcal{E}|\times 1} \quad I_{|\mathcal{E}|} \\ -I_{|\mathcal{E}|+1} \\ M \end{bmatrix}}_{A} \mathbf{f} \leq \begin{bmatrix} \mathbf{c} \\ \mathbf{0}_{|\mathcal{E}|+1} \\ \mathbf{0}_{|\mathcal{V}|} \end{bmatrix} \end{array}$$

where $\mathbf{f}$ and $\mathbf{c}$ are the vector stacking of respectively the flows and the capacity constraints of all the edges, and M is the edge-vertex adjacency matrix as defined in class. We know that M is TUM (*i.e.*, and square sub-matrix of M has a determinant of 0,+1 or -1). Then we can prove that the matrix A is TUM by proving that every square sub-matrix of dimension k has a determinant of $0, 1$ or -1. We do so by induction over the size of the square sub-matrix that we pick k:

Base Case - $k = 1$. This trivially holds because all elements of the matrix A are $0, 1$ or -1.
Assumption Step. Assume that any square sub-matrix of dimension $k - 1$ is has a determinant of $0, 1$ or -1.
Inductive Step. Consider a sub-matrix of size k.

- If there is one row that has all zeros, then the determinant of the matrix is 0.
- If not, then assume there is one row that is picked from either the first part or the second of the matrix A,
  - Since this row is not all-zero, then there is exactly one 1 in that row. Therefore the determinant of this sub-matrix is equal to the determinant of the square sub-matrix of size $k - 1$ obtained by removing the row and column of that 1. By our assumption, this has a determinant of $0, 1$ or -1.
- If not, then the whole sub-matrix comes from the M part of the matrix A, which we know is TUM.

Thus, the matrix A is TUM.

(b) First we relax the provided ILP into the following LP

$$\begin{array}{ll} \text{minimize} & c^T d \\ \text{subject to} & p_s - p_t \geq 1, \\ & d_{ij} - p_i + p_j \geq 0, \\ & 0 \leq d_{ij} \leq 1, \\ & 0 \leq p_i \leq 1 \end{array} \tag{3}$$

Note that the LP in (3) is similar to the dual of the max-flow LP, except for the two sets of conditions in (3): $d_{ij} \leq 1$ and $p_i \leq 1$. However, in the following we prove that these constraints are redundant. We can rewrite the dual of the max-flow LP as

$$\begin{array}{ll} \text{minimize} & c^T d \\ \text{subject to} & \underbrace{\begin{bmatrix} \mathbf{0}_{1\times|\mathcal{E}|} & M^T \\ I_{|\mathcal{E}|} & \\ & I_{|\mathcal{E}|+|\mathcal{V}|} \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \mathbf{d} \\ \mathbf{p} \end{bmatrix}}_{\tilde{x}} \geq \underbrace{\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}}_{\tilde{b}} \end{array} \tag{4}$$

Now note that similar to our discussion in part 1, the matrix $\tilde{A}$ is TUM. From the definition of TUM matrices, this also implies that any non-singular square submatrix of $\tilde{A}$ is TUM.
Now consider any extreme point of the LP (4). Let J be the set of $|\mathcal{E}|+|\mathcal{V}|$ active constraints at the extreme points. A_J is a non-singular submatrix of $\tilde{A}$ and therefore also TUM. the extreme point $\tilde{x}_J$ is equal to

$$\tilde{x}_J = A_J^{-1}\tilde{b}$$

Since $\tilde{b}$ has a single non-zero element (in the first position) and it is equal to one, then we can write the elements of $\tilde{x}_J$ as $(\tilde{x}_J)_i = (A_J^{-1})_{i1}$. Finally, we note that since A_J is TUM then A_J^{-1} is also TUM and therefore its elements take values $\{-1, 0, 1\}$. However, since $\tilde{x}_J$ is also feasible, we know that $\tilde{x} \geq 0$. Therefore, for any extreme point x_J, we have that $0 \leq x_J \leq 1$, which proves that the missing constraints are redundant.

Problem 3 (`3 points`):

Hospital H is going through a hiring phase. With the staff of surgeons that they already have, the hospital knows that there are some surgeries which they are not equipped to handle. These surgeries are enumerated in the list $\mathcal{S} = \{S_1, S_2, \cdots, S_n\}$. After advertising for surgeon positions, they received a list of surgeon applicants $\mathcal{A} = \{A_1, A_2, \cdots, A_m\}$. Each applicant in their resume indicated which of the surgeries in $\mathcal{S}$ they are able to perform, and the salaries they wish to get if they are hired. The hospital would like to hire a number of surgeons which can handle all the surgeries in $\mathcal{S}$ while paying the minimum amount in salaries. Write an ILP that solves the hospital's problem.

Solution:

The problem can be formulated as a set-cover problem: the nodes are the surgeries, and the sets are the surgeons. An edge between a node and a surgery indicates that this surgeon is able to perform the surgery. Denote the resulting graph as $\mathcal{G} = (\mathcal{S} \cup \mathcal{N}, \mathcal{E})$, where $\mathcal{S}$ is the collection of sets (surgeons) and $\mathcal{N}$ is the collection of nodes (surgeries). Let w_s be the salary required by surgeon number s. We define b_s to be the selector variable of set number s in $\mathcal{S}$. Finally, let $N(n)$ be the collection of neighbor sets for node number n. We can therefore write the following ILP:

$$
\begin{aligned}
\min_{b} \quad & \sum_{s=1}^{|\mathcal{S}|} w_s b_s \\
\text{subject to} \quad & \sum_{s \in N(n)} b_s \geq 1, \quad \forall n \in \{1, \cdots, |\mathcal{N}|\} \\
& b_s \in \{0, 1\}.
\end{aligned}
$$

<u>Problem 4</u> (`3 points`):

Coach P wants to organize some one-on-one training basketball matches between the players of her team. She wants to set up those matches so that a match is beneficial to both the players in the match. Based on her experiences with the players, she now has a list of possible match-ups between her players: each of these match-ups will help the two players develop their game and overcome their playing deficiencies. The list looks like this

Names of players	**Notes on the match**
James and Kevin	Will improve their defensive plays
David and Mike	Will improve their speed plays
Kevin and Shaun	Learn how to play a game with size mismatches
⋮	⋮

In the mind of Coach P, all these match-ups are equally beneficial. However, she notices that it is not the case that each player is a part of a "single" possible match-up (e.g., Kevin in the table above). Moreover, she notices that some players are part of possible match-ups more than the others. Out of this list of possible match-ups, she would like to set up the maximum number of games, to run in parallel.

a) Write an ILP that would help Coach P. Can the constraints of the ILP be expressed using a TUM matrix?

b) Suppose that Coach P wants to devote three separate Sundays to these 1-on-1 matches. Coach P wants to arrange as many matches as possible on the three days, without repeating any matchups. Thus, each player can play in up to three different matches over the three days. Write an ILP that Coach P can use.

<u>Solution</u>:

a) This problem can be modeled as a matching problem. We represent the situation as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the players, and an edge $e \in \mathcal{E}$ exists between two nodes in $\mathcal{V}$ to indicate that these two players can play together. The problem can therefore be cast as follows: find the maximum matching on graph $\mathcal{G}$. For a node v, let $N(v) \subseteq \mathcal{E}$ be the set of edges that are adjacent to v. Then the problem can be expressed as the following ILP

$$
\begin{aligned}
\max_{b} \quad & \sum_{e \in \mathcal{E}} b_e \\
\text{subject to} \quad & \sum_{e \in N(v)} b_e \leq 1, \quad \forall v \in \{1, \cdots, |\mathcal{V}|\} \\
& b_e \in \{0, 1\}.
\end{aligned}
$$

In general, the constraint matrix is not TUM, since the edge-adjacency matrix of a general graph is not TUM in general.

b) This problem can be modeled as an edge-coloring modification of the above, with three different variables for each edge representing each week (one color per week). We have the restriction that each edge can have at most one color. Let w represent the week we are selecting. Then the problem can be expressed as the following ILP:

$$\begin{aligned}
\max_{b} \quad & \sum_{w=1}^{3} \sum_{e \in \mathcal{E}} b_{e,w} \\
\text{subject to} \quad & \sum_{e \in N(v)} b_{e,w} \leq 1, \quad \forall v \in \{1, \cdots, |\mathcal{V}|\}, \forall w = 1, 2, 3 \\
& b_{e,w} \in \{0, 1\}. \\
& \sum_{w=1}^{3} b_{e,w} \leq 1, \forall e \in \mathcal{E}
\end{aligned}$$

Problem 5 (4 points): Bob just started his new job as a bike delivery boy in a famous restaurant. As his first task, he was assigned to deliver meals to a set of houses $\mathcal{H}$. Bob has a very accurate map, so he knows exactly how far it is to go from any given house directly to the other. He wants to find the best route that gets him to do his deliveries with the minimum amount of cycling.

But there is a catch! The bad thing about his job is that he knows that the food at the restaurant he works is really bad. So he wants to make sure that he doesn't go by a house that he has already delivered food to, lest the customers there catch him on his way and complain. Can you help him find such a route using the LP techniques that you learned? *Hint:* transform this problem into a graph problem, and see what properties your graph solution needs to have. It is okay to use an ILP instead of an LP.

Solution:

This can be formulated as the famous Travelling Salesman Problem (TSP). We model the problem as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertices are the houses $\mathcal{V} = \mathcal{H}$ and the edges are the routes from one house to the other (a complete graph). Each edge $(i, j) \in \mathcal{E}$ has a penalty, denoted w_{ij}, equal to the distance traveled between house i and house j. The problem is to find a subset of edges $S \subseteq \mathcal{E}$ that form a single circle across all the vertices with a minimum sum of weight.

There are various methods to solve this problem using ILP. The main idea is to define a binary variable b_e, for any $e \in \mathcal{E}$, as the indicator of whether we select the edge e in the set S. Our objective is to minimize $\sum_{e \in \mathcal{E}} w_e b_e$. One constraint is that each vertex has a degree 2; the other constraint is to exclude the case where multiple disjoint cycles exist. We present three methods to formulate this problem.

Method 1: Let us first define $\delta(U) = \{\{i, j\} \in \mathcal{E} | i \in U, j \notin U\}$ as the subset of edges that connect a subset of vertices U from outside of U. Then the problem can be formulated as:

$$\begin{aligned}
\text{minimize} \quad & \sum_{e\in\mathcal{E}} w_e b_e \\
\text{subject to} \quad & \sum_{e\in\delta(\{v\})} b_e = 2,\ \forall v \in \mathcal{V} \\
& \sum_{e\in\delta(U)} b_e \geq 2,\ \forall U \subset \mathcal{V},\ \text{and } 2 \leq |U| \leq |\mathcal{V}| - 1 \\
& b_e \in \{0, 1\},
\end{aligned}$$

where the first set of constraints describes that any vertex v has degree 2 in the solution S; and the second set of constraints describes that for any subset of vertices U with size $2, 3, \ldots, |\mathcal{V}| - 1$, the number of edges that connect U from outside of U can not be less than 2, in order to exclude the scenario of disjoint cycles.

Method 2: This method uses a directed graph to formulate the problem. An undirected graph can be changed into a directed graph by splitting an edge e connecting u and v into two edges (u, v) and (v, u). Then we define a variable b_{uv} to indicate whether an edge (u, v) is selected in the solution. Let us also denote by $\mathcal{E}$ the edges in the directed graph and by w_{uv} the weight of the edge connecting u and v. Then the problem can be formulated as:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(u,v)\in\mathcal{E}} w_{uv} b_{uv} \\
\text{subject to} \quad & \sum_{v\in\mathcal{V}} b_{uv} = 1,\ \forall u \in \mathcal{V} \\
& \sum_{v\in\mathcal{V}} b_{vu} = 1,\ \forall u \in \mathcal{V} \\
& y_u - y_v + 1 \leq (|\mathcal{V}| - 1)(1 - b_{uv}),\ \forall (u, v) \in \mathcal{E},\ u \neq 1,\ v \neq 1 \\
& 2 \leq y_v \leq |\mathcal{V}|,\ \forall v \in \mathcal{V}, v \neq 1 \\
& b_{uv} \in \{0, 1\},\ \forall (u, v) \in \mathcal{E},
\end{aligned}$$

where the first two sets of constraints describe that both the input degree and output degree for a vertex are 1. The third set of constraints is called the MTZ constraint, which is used to exclude the disjoint cycles. We select an arbitrary vertex as vertex 1, and the MTZ constraint is to guarantee that there is no cycle after removing vertex 1. To see this, we first consider the edges that are not selected, i.e., $b_{uv} = 0$. For these edges, the inequality is satisfied since the maximum difference between any two values in $[2, |\mathcal{V}|]$ is $|\mathcal{V}| - 2$. When an edge (u, v) is selected, we have $b_{uv} = 1$ and the constraint becomes $y_u + 1 \leq y_v$. This constraint can not be satisfied when a cycle exists after removing vertex 1, but can be satisfied for a path (after removing vertex 1) by sequentially setting y_v the values $2, 3, \ldots, |\mathcal{V}|$ along the path.

Method 3: In this method, the main idea is: there is no cycle for any induced subgraph containing $2, 3, \ldots, |\mathcal{V}| - 1$ vertices. We can formulate the problem as follows.

$$\begin{aligned}
\text{minimize} \quad & \sum_{e\in\mathcal{E}} w_e b_e \\
\text{subject to} \quad & \sum_{e\in\delta(\{v\})} b_e = 2,\ \forall v \in \mathcal{V} \\
& \sum_{\substack{e=uv\in\mathcal{E} \\ u\in U, v\in U}} b_e \leq |U| - 1,\ \forall U \subseteq \mathcal{V}, 2 \leq |U| \leq |\mathcal{V}| - 1 \\
& b_e \in \{0, 1\}.
\end{aligned}$$

The first set of constraints describes that each vertex is of degree 2. The second set of constraints aims to exclude the disjoint cycles. To see this, note that when we select any induced subgraph by $U \subset \mathcal{V}$, with $2, 3, \ldots, |\mathcal{V}| - 1$ vertices, the set of edges in the subgraph with $b_e = 1$ should form a tree. We know that a tree with n vertices has $n-1$ edges; thus the average degree (averaging over all vertices of the tree) is $\frac{2(n-1)}{n} < 2$. On the contrary, on a cycle, the degree of the vertices (and the average degree) equals 2. Therefore, in order to exclude the cycles, the number of edges with $b_e = 1$ should be strictly smaller than the number of vertices in the subgraph.

Problem 6 (`4 points`): As we discussed in class, the following LP

$$\begin{aligned} \text{maximize} \quad & \sum f_i \\ \text{subject to} \quad & \sum_{i:\, e \in P_i} f_i \le c_e, \ \forall \text{ edges } e \\ & f_i \ge 0 \end{aligned} \tag{5}$$

is the max-flow problem formulation with flows associated with paths, where $P = \{P_i\}$ is the set of all paths from source to destination, and f_i is the flow associated with the ith path.

- Prove that the LP in (5) is equivalent to the max-flow problem formulation with flows associated to each edge.
- In a graph with unit capacity edges (i.e., $c_e = 1, \ \forall$ edges e), prove that if the the solution of the max-flow problem is k then we can find k edge-disjoint paths from s to t in the graph. (Two paths are said to be edge-disjoint if they do not share a common edge.)
- Derive the dual of (5) and give an interpretation to the dual variables and the objective function.

Solution: We will prove the equivalence by showing that, from an optimal solution in one problem, we can construct a feasible solution to the other problem which achieves the same objective function and vice versa. Let us denote the path-based max-flow problem as P1, and the edge-based max-flow problem as P2. In the following discussion, we will repeatedly use the following facts:

Fact 1. For any feasible solution of P1, the following holds

$$\sum_{\text{All paths which pass through } i} f_m = \sum_{k:(k,i)\in\mathcal{E}} \sum_{m:(k,i)\in P_m} f_m = \sum_{j:(i,j)\in\mathcal{E}} \sum_{m:(i,j)\in P_m} f_m, \quad \text{for all } i \in \mathcal{V} \setminus \{s, d\} \tag{6}$$

Fact 2. For any feasible solution of P2, the following holds

$$\sum_{k:(k,i)\in\mathcal{E}} f_{ki} = \sum_{j:(i,j)\in\mathcal{E}} f_{ij}, \quad \text{for all } i \in \mathcal{V}. \tag{7}$$

Next, we show the equivalence.

(1). P2 $\Rightarrow$ P1: Assume an optimal solution f_{ij}^* exists for the max-flow problem. Then construct the set of path flows f_m for all m in such a way that the following holds

$$\sum_{m\colon (i,j)\in P_m} f_m = f^\star_{ij}, \qquad \text{for all } (i,j)\in\mathcal{E}. \tag{8}$$

If such a construction of f_m is possible, then we can proceed by proving the equivalence. It is easy to see that this selection of f_m for all m is greater than or equal to 0, and satisfies the capacity constraint in (5). Therefore, that construction gives a feasible point in (5). Note that each path ends at the destination, and therefore for each P_m, there exists i such that $(i,d)\in P_m$. Therefore, the objective function can be expressed as

$$\sum f_m = \sum_{i:(i,d)\in\mathcal{E}} \sum_{m:(i,d)\in P_m} f_m = \sum_{i:(i,d)\in\mathcal{E}} f^\star_{id} = f^\star_{ds}$$

where the second equality follows from (6) and last equality follows from (7). What remains is to try to prove that such a construction of f_m with condition (8) is possible. One way to construct such a solution is using the following algorithm

1. Initialize $\hat{f}_{ij} = f^\star_{ij}$ for all $(i,j)\in\mathcal{E}$ and $f_m = 0$ for all paths m.
2. Select a path P_m. For path P_m, select the edge $(i,j)^{\min}\in P_m$ such that $(i,j)^{\min} = \arg\min_{(i,j)\in P_m} \hat{f}_{ij}$ (the minimum edge flow across the path). Set the flow of the path to be equal to that edge flow, *i.e.*, $f_m = \hat{f}_{(i,j)^{\min}}$.
3. Subtract $\hat{f}_{(i,j)^{\min}}$ from $\hat{f}_{ij}$ for all $(i,j)\in P_m$ (i.e., subtract it from all edge flows of all edges in the path).
4. Repeat the previous steps until in every path there is at least one edge with $\hat{f}_{ij} = 0$.

Note that the termination criterion dictates that, for each path, there is at least one edge with $\hat{f}_{ij} = 0$. Note also that the set of flows $\hat{f}_{ij}$ maintained within the algorithm is always a feasible (but not optimal) point in P2. Therefore, it always satisfy condition (7). Finally, note that the following is always maintained within the execution of the algorithm

$$\sum_{m:(i,j)\in P_m} f_m + \hat{f}_{ij} = f^\star_{ij}, \qquad \text{for all } (i,j)\in\mathcal{E}. \tag{9}$$

We need to show that, once the algorithm terminates, then condition (8) is satisfied, which is equivalent to prove that $\hat{f}_{ij} = 0$ in (9). We show this by contradiction, that is, assume that $\hat{f}_{ij} > 0$ for some $(i,j)\in\mathcal{E}$. Since $\hat{f}_{i,j}$ is a feasible point in P2, then it satisfies (7). Therefore, there exists at least one k such that $\hat{f}_{k,i} > 0$. One can repeat the previous argument to show that there is a consecutive sequence of edges from s to i for which $\hat{f}_{ij} > 0$. Similar argument can also be made to show that there is a consecutive sequence of edges from i to d with the same property. The two sequences for edges, from s to i and from i to d form a path with strictly positive flows in $\hat{f}_{ij}$. However, this contradicts the termination condition of the algorithm. Therefore, upon termination, all $\hat{f}_{ij} = 0$, and therefore condition (8) is met.

P2 $\Leftarrow$ P1: Assume an optimal solution f_i^* exists for (5). Then construct

$$f_{ij} = \sum_{k:(i,j)\in P_k} f_k^\star$$

It is clear that $0 \leq f_{ij} \leq c_{ij}$. Note also that for all $i \in \mathcal{V}$

$$\sum_{j:(j,i)\in\mathcal{E}} f_{ji} - \sum_{k:(i,k)\in\mathcal{E}} f_{ik} \;\; = \sum_{j:(j,i)\in\mathcal{E}} \; \sum_{m:(j,i)\in P_m} f_m^\star - \sum_{k:(i,k)\in\mathcal{E}} \; \sum_{m:(i,k)\in P_m} f_m^\star = 0$$

where the last equality holds from (6). Therefore feasibility is preserved. Finally, the objective function becomes

$$f_{ds} = \sum_{i:(i,d)\in\mathcal{E}} f_{id} = \sum_{i:(i,d)\in\mathcal{E}} \; \sum_{m:(i,d)\in P_m} f_m^\star = \sum_m f_m^\star$$

where the last equality follows from (6) and by noting that, by definition, all paths go through d.
(2). First we note that the solution of the max-flow problem with unit edge capacities is integral (i.e., all extreme points have integer elements). Since $c(e) \in \{0,1\}$ for all edges e (except (t,s)), then $f_{ij}^\star \in \{0,1\}$.
We now apply the procedure discussed in part 1.
Since $f_{ij}^\star \in \{0,1\}$, then also $f_{e^{min}} \in \{0,1\}$ for each path. Whenever $f_{e^{min}} = 1$, then we are removing all edges that belong to that path. Any subsequent path that shares an edge with a preceding path of $f_{e^{min}} = 1$ will have a capacity of zero. Therefore, we know that our construction gives us unit-capacity paths with disjoint edges. Since the optimal solution of the max-flow problem is k and the two problems are equivalent, this implies that an optimal integral point in the max-flow gives us a unit-capacity edge-disjoint paths that have a total capacity of k. This proves the required property.
(3). We can write the LP in (5) in a matrix form as

$$\begin{array}{ll} \text{maximize} & \mathbf{1}_{|P|}^T f \\ \text{subject to} & \begin{bmatrix} -I_{|P|} \\ A_{|\mathcal{E}|\times|P|} \end{bmatrix} f \leq \begin{bmatrix} \mathbf{0}_{|P|} \\ c \end{bmatrix} \end{array} \tag{10}$$

where A is the edge-path adjacency matrix (the columns correspond to paths and the rows to edges, column i has ones at the entries corresponding to edges in path i and zeros otherwise). The dual of (10) can be derived as

$$\begin{array}{ll} \text{minimize} & c^T\lambda \\ \text{subject to} & \sum_{e:e\in P_i} \lambda_e \geq 1, \; \forall P_i \in P \\ & \lambda \geq 0 \end{array} \tag{11}$$

This can be interpreted as a min-path-cut. Each variable λ_e corresponds to an edge in the graph, and $\lambda_e = 1$ corresponds to a cut in that edge. The constraints ensure that in each path, at least one edge is cut. The LP tries to find the path-cut with the maximum capacity.