DESIGN AND ANALYSIS OF ALGORITHMS
Homework 7

Aman Choudhary
MTech Coursework, CSA 2020
Sr No: 17920

December 5, 2020

1 Problem 1

1.1 Notation

- L_e is the first linear program, while L_P is the second linear program.
- OPT_e and OPT_P are the optimal values returned by L_e and L_P respectively such that,

$$OPT_e = max \sum_{e \in \delta^+(s)} f_e \qquad \text{and} \qquad OPT_P = max \sum_{P \in \mathcal{P}} f_P$$

- $F_e = \{f_e\}_{e \in E}$ is the edge-wise flow computed by L_e.
- $F_P = \{f_P\}_{P \in \mathcal{P}}$ is the path-wise flow computed by L_P.

1.2 Constraints of L$_e$

- $\sum_{e \in \delta^+(v)} f_e - \sum_{e \in \delta^-(v)} f_e = 0$, for every vertex $v \neq s, t$ ...(1)
- $f_e \leq u_e$, for all $e \in E$ ...(2)
- $f_e \geq 0$, for all $e \in E$ ...(3)

1.3 Constraints of L$_P$

- $\sum_{P \in \mathcal{P} : e \in P} f_P \leq u_e$, for all $e \in E$ ...(4)
- $f_P \geq 0$, for all $P \in \mathcal{P}$ ...(5)

1.4 Proof

Result 1: The flow f_e along an edge e, is the sum total of constituent flows of all the $s - t$ paths that pass through edge e. That is,

$$f_e = \sum_{P \in \mathcal{P} : e \in P} f_P, \text{ for all } e \in E$$

Result 2: Each of the $s - t$ paths $P \in \mathcal{P}$ has its source as s. Let, $v_1, v_2, ..., v_d$ be vertices adjacent to s, where d is the degree of s. The first edge of any $P \in \mathcal{P}$ is exactly one of the edges: $(s, v_1), (s, v_2), ..., (s, v_d)$. Let us denote by $\mathcal{P}_i$ the set of paths, whose first edge is (s, v_i). Now, any path P will belong to atmost one $\mathcal{P}_i$. Hence,

$$\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup .... \cup \mathcal{P}_d \text{ , where } \mathcal{P}_i \cap \mathcal{P}_j = \phi, \text{ for } i \neq j$$

Part 1: Let us assume that $OPT_e < OPT_P$

- Now, L_P computes some path-wise flow F_P for G. Let us consider any arbitrary edge $e \in E$. Let us denote the total flow on this edge by f'_e. Using Result 1,

  $$f'_e = \sum_{P \in \mathcal{P} : e \in P} f_P, \text{ for all } e \in E \qquad ...(A)$$

  Thus, we can derive the edge-wise flow $F'_e = \{f'_e\}_{e \in E}$ corresponding to path-wise flow F_P.

- Now,

  $$OPT_P = max \sum_{P \in \mathcal{P}} f_P = \sum_{f_P \in F_P} f_P = \sum_{i=1}^{d} \sum_{P \in \mathcal{P}_i} f_P = \sum_{i=1}^{d} f'_{(s,v_i)} = \sum_{e \in \delta^+(s)} f'_e$$

  Here 4^{th} equality holds by Result 1 2. Now, from our assumption,

  $$OPT_P = \sum_{e \in \delta^+(s)} f'_e > max \sum_{e \in \delta^+(s)} f_e = OPT_e$$

- From (A) and constraint (4), we can say that,

$$f'_e \leq u_e, \text{ for all } e \in E \qquad ...(B)$$

- Result 1 says that the flow along an edge e is the sum total of constituent flows of all the $s-t$ paths that pass through edge e. From constraint (5), it is true that individual flows f_P are non-negative. As a result, f_e which is sum over these flows will also be non-negative, i.e.,,

$$f'_e \geq 0, \text{ for all } e \in E \qquad ...(C)$$

- Consider an arbitrary vertex v. Let us assume that it lies on a path $P \in \mathcal{P}$. Now, the flow value f_P is a constant over the entire path. Hence, the contribution of P to inflow of v is same as its contribution to the outflow from v. As a result, a vertex may be part of any number of paths, but all these paths passing through v ensure that they preserve flow while passing through that vertex. Therefore, the conservation of flow holds true for all vertices except s, t.

$$\sum_{e \,\in\, \delta^+(v)} f'_e - \sum_{e \,\in\, \delta^-(v)} f'_e = 0, \text{ for every vertex } v \neq s, t \qquad ...(D)$$

- From (B), (C) and (D), we conclude that $F'_e = \{f'_e\}_{e \in E}$ satisfies all the necessary constraints for L_e and hence it is a valid edge-wise flow that lies in its feasible region. As a result, L_e would have chosen F'_e and not F_e, as the optimal solution because F'_e provides a more optimal value for its objective function. Hence, we reach a contradiction, proving our assumption to be incorrect.

Part 2: Let us assume that $OPT_e > OPT_P$.

- Now, L_e computes some edge-wise flow F_e for G. Let us consider any arbitrary path $P \in \mathcal{P}$. There would be some amount of flow being routed on this path P, corresponding to the flow F_e. Let us call it f'_P. Hence, $F'_P = \{f'_P\}_{P \in \mathcal{P}}$ is the distribution of flows along the different $s-t$ paths, corresponding to flow F_e. Now,

$$OPT_e = max \sum_{e \,\in\, \delta^+(s)} f_e = \sum_{\substack{f_e \,\in\, F_e \\ e \,\in\, \delta^+(s)}} f_e = \sum_{i=1}^{d} f_{(s,v_i)} = \sum_{i=1}^{d} \sum_{P \,\in\, \mathcal{P}_i} f'_P = \sum_{P \,\in\, \mathcal{P}} f'_P$$

Here 4^{th} equality holds by Result 1 and 2. Now, from our assumption,

$$OPT_e = \sum_{P \,\in\, \mathcal{P}} f'_P > max \sum_{P \,\in\, \mathcal{P}} f_P = OPT_P$$

- Using Result 1, we can say,

$$f_e = \sum_{P \,\in\, \mathcal{P} \,:\, e \,\in\, P} f'_P \qquad ...(E)$$

From (E) and constraint (2), we thus conclude that,

$$\sum_{P \,\in\, \mathcal{P} \,:\, e \,\in\, P} f'_P \leq u_e, \text{ for all } e \in E \qquad ...(F)$$

- Similarly, from (E) and constraint (3), we thus conclude that,

$$\sum_{P \,\in\, \mathcal{P} \,:\, e \,\in\, P} f'_P \geq 0, \text{ for all } e \in E$$

Although the sum of flows over an edge must be non-negative, it does not restrict the individual flows f'_P from being negative. However, if there exist a flow f'_P which has a negative value, then it would mean that we are routing flow from t to s instead of s to t. Since, we have already established that in the general max-flow problem, s does have any incoming edge and t does not have any outgoing edge, routing flow from from t to s is not possible and we reach a contradiction. Hence, it must be true that,

$$f'_P \geq 0, \text{ for all } P \in \mathcal{P} \qquad ...(G)$$

- From (F) and (G), we conclude that $F'_P = \{f'_P\}_{P \in \mathcal{P}}$ satisfies all the necessary constraints for L_P and hence it is a valid path-wise flow that lies in its feasible region. As a result, L_P would have chosen F'_P and not F_P, as the optimal solution because F'_P provides a more optimal value for its objective function. Hence, we reach a contradiction, proving our assumption to be incorrect.

Using conclusions of **Part 1** and **Part 2**, it is clear that, the two linear programs L_e and L_P always have equal optimal objective function value, i.e. $OPT_e = OPT_P$.

2 Problem 2

2.1 Notation

- **MCF** is shorthand notation for multicommodity flow.
- Formally, a flow is a non-negative vector $F = \{f_e\}_{e \in E}$, indexed by the edges of graph $G = (V, E)$. The value of a flow is $\sum\limits_{e \in \delta^+(s)} f_e$, where s is the source vertex.
- A **multicommodity flow**, $M = \{F^{(1)}, F^{(2)}, ..., F^{(k)}\}$ is a set of k flows such that:
  (i) for each $i = 1, 2, ... k$, $F^{(i)}$ is an $s_i - t_i$ flow (in the usual max flow sense); and
  (ii) for every edge e, the total amount of flow (summing over all commodities) sent on e is at most the edge capacity u_e.
- **L** is the equivalent **linear program** for the **MCF** problem.

2.2 Linear Program

1. **Decision Variables**

   Let m be the total number of edges in the input graph $G = (V, E)$. We will index the set of edges from 1 to m in an arbitrary order. L has a total of $k * m$ decision variables of the form f_{ij}. **Here, f_{ij} denotes the flow on the j^{th} edge in the i^{th} flow,** $F^{(i)}$. Intuitively, the set of decision variables $\{f_{i1},\ f_{i2}\ ...,\ f_{im}\}$ represents flow $F^{(i)}$ (proven later).

2. **Linear Constraints**

   Since, each flow $F^{(i)}$ is an $s_i - t_i$ flow (in the usual *max-flow* sense), the constraints of original max-flow problem carry over to the *MCF* problem with few modifications.

   (i) **Conservation Constraint:** For every $s_i - t_i$ flow $F^{(i)}$, the rule of conservation of flow must hold true at all vertices, except for the source and sink. Let n be the number of vertices in the graph. Hence, we have $O(n)$ conservation constraints per flow, $F^{(i)}$. Therefore, the total number of conservation constraints in L is $O(k * n)$. For every $s_i - t_i$ flow $F^{(i)}$,

   $$\sum_{j \in \delta^+(v)} f_{ij} - \sum_{j \in \delta^-(v)} f_{ij} = 0, \text{ for every vertex } v \neq s_i, t_i \qquad ...(1)$$

   (ii) **Capacity Constraints:** For every $s_i - t_i$ flow $F^{(i)}$, the flow on each edge must be non-negative. This puts a constraint on each of our decision variables. Thus we have $O(k * m)$ non-negativity constraints:

   $$f_{ij} \geq 0 \qquad ...(2)$$

   The next set of capacity constraints are a digression from the original max-flow problem, and these are unique to the MCF problem. It must be true that for every edge j, the total amount of flow (summing over all commodities) sent on j is at most the edge capacity u_j. In total, we have $O(m)$ such constraints (one for every edge).

   $$\textstyle\sum_{i=1}^{k} f_{ij} \leq u_j, \text{ for each edge } j \in E \qquad ...(3)$$

3. **Linear Objecive Function**

   The **value of a MCF** is the sum of the values (in the usual max-flow sense) of the flows $F^{(1)}, F^{(2)}, ..., F^{(k)}$, and our objective is to maximize this value.

   $$max\ \textstyle\sum_{i=1}^{k} \displaystyle\sum_{j \in \delta^+(s_i)} f_{ij}$$

2.3 Proof of Correctness

Since the point is to push flow from s_i to t_i for every commodity, we can assume without loss of generality that s_i has no incoming edges and t_i has no outgoing edges.

Claim 1: For any commodity i, the set $\{f_{i1},\ f_{i2}\ ...,\ f_{im}\}$ represents a valid flow $F^{(i)}$ (in the usual *max-flow* sense).

- From (3), we can see that for every edge j, the total amount of flow (summing over all commodities) sent on j is at most the edge capacity u_j. Hence, it follows that the contribution of any particular commodity i to flow over an edge j, i.e. f_{ij} does not exceed u_j. Thus,

$$f_{ij} \leq u_j \qquad ...(4)$$

- From (1), (2) and (4), it is clear that every constraint of the original max-flow problem is being observed for each individual flow $F^{(i)}$. Therefore, every flow $F^{(i)}$ is a valid flow.
- Since, we have established that each flow $F^{(i)}$ is valid, it follows that the flow originating from s_i is equal to the flow that terminates into t_i (as is the case in the usual max-flow problem).

Claim 2: The solution of L is essentially a MCF.

- The solution of L is the set of decision variables $D = \{f_{ij}\}$. Using Claim 1, the set $\{f_{i1},\ f_{i2}\ ...,\ f_{im}\}$ is a valid flow $F^{(i)}$. Now, for each i = 1, 2, ... k, we have a flow $F^{(i)}$, resulting in a set of k flows, $M = \{F^{(1)}$, $F^{(2)}$, ..., $F^{(k)}\}$. This set of flows was computed subjected to constraint (3). Hence, M is a **MCF**, by definition.

Claim 3: The feasible region of solutions of L includes every possible MCF.

- Any arbitrary MCF $M' = \{F^{(1)}$, $F^{(2)}$, ..., $F^{(k)}\}$ can be encoded as follows: For each i = 1, 2, ... k, use $F^{(i)}$ to initialize the values of decision variables in the set $\{f_{i1},\ f_{i2}\ ...,\ f_{im}\}$. Following this process, we generate the set of decision variables $D = \{f_{ij}\}$. Now, D is a point in the R^{k*m} space. We need to show that D lies in the feasible region of L.
- Since, M' is a MCF, each of its constituent flows $F^{(i)}$ is a valid $s_i - t_i$ flow. Each of these flows, would follow conservation constraints. Hence, we can say that D satisfies constraint (1) of L. Moreover each of these flows, would satisfy capacity constraints. In particular, this would mean that constraint (2) (and also condition (4)) is true for all $f_{ij} \in D$. Finally, by definition of MCF, D would also fulfill constraint (3). Since D satisfies all required constraints of L, hence it lies in the feasible region of solutions of L.
- Since, we have shown any arbitrary MCF M' can be encoded into a set of decision variables $D = \{f_{ij}\}$, such that D lies in the feasible region, Claim 3 is hence proved.

Claim 4: The solution returned by L, is a MCF of maximum-possible value.

- Claim 2 tells us that the solution computed by L is first of all, a valid MCF. Then using Claim 3, we established that L was optimizing over all possible MCFs, i.e. its optimization domain was exhaustive. Finally, since the aim of our objective function was to maximize the **value of MCF**, and L computed the result over all possible MCFs, it thus follows that L indeed returns a MCF with the maximum-possible value.

2.4 Time Complexity

The total number of constraints is $O(k*n) + O(k*m) + O(m) = O(k*m)$, which is of polynomial order. Also, the number of terms in each constraint is at most $O(k*m)$, (the total number of decision variables). Hence, it takes polynomial time to specify each constraint. Therefore the overall time complexity to specify the constraints is $O(k^2 * m^2)$. Therefore, the original MCF problem can be reduced to a linear program in polynomial time.
