

Assignment 3

Navneel Singhal

2018CS10360

For the first part, the proof will be as follows:

Showing that an MST gives a feasible solution for the integer program:

If x_e is the indicator variable for whether we choose the edge e , then the last condition is satisfied trivially. Since an MST has $n - 1$ edges, the second last condition is also satisfied. Now consider any subset A of E . The edges of the MST that are also in A will not have any cycle, so they form a forest. The number of edges in the forest is the number of vertices in the forest minus the number of components in the forest. Note that removing some edges (possibly 0) from A to get the set of edges of the MST that are also in A would not decrease the number of components, and the number of vertices is n . Hence, the number of edges of the MST in A is at most $n - \alpha(A)$, which means that the first inequality is also satisfied. This means that an MST is a feasible solution for the integer program.

Showing that any feasible solution to the integer program corresponds to a spanning tree:

Since the value of x_e is always in $\{0, 1\}$, we can interpret it as the indicator variable for inclusion of the edge e into the solution. We claim that the set S of edges e with $x_e = 1$ corresponds to a spanning tree. Note that the number of edges is $n - 1$ from the second last constraint. Now suppose that there is a cycle in S . Now consider the first constraint with A being the set of edges in the cycle. In (V, A) , the number of edges that are in A as well as the MST is $|A|$, and the number of vertices in the component corresponding to the cycle is $|A|$. There are $n - |A|$ other components, all having a single vertex. So the LHS is $|A|$, and the RHS is $n - (n - |A| + 1) = |A| - 1$, which is a contradiction. So there is no cycle in S . Since (V, S) is an acyclic graph and has $|V| - 1$ edges, (V, S) is a spanning tree, as required.

Now note that since the MST is a feasible solution and there is no smaller objective function value (for that would give a spanning tree with a lesser cost than the MST), we are done.

1 Part 1

The integer program is as follows:

$$\begin{aligned} \max \quad & \sum_e -c_e x_e \\ \text{subject to} \quad & \sum_{e \in A} x_e \leq n - \alpha(A) \quad A \subset E \\ & \sum_e x_e = n - 1 \\ & x_e \in \{0, 1\} \quad e \in E \end{aligned}$$

The LP relaxation is as follows:

$$\begin{aligned}
& \max \sum_e -c_e x_e \\
& \text{subject to} \\
& \sum_{e \in A} x_e \leq n - \alpha(A) \quad A \subseteq E \\
& \sum_e x_e = n - 1 \\
& x_e \geq 0 \quad e \in E
\end{aligned}$$

We don't need $x_e \leq 1$, since it is automatically implied by the first constraint when it is applied to $A = \{e\}$.

The dual LP is as follows:

$$\begin{aligned}
& \min \sum_{A \subseteq E} (n - \alpha(A)) y_A \\
& \text{subject to} \\
& \sum_{A \subseteq E | e \in A} y_A \geq -c_e \quad e \in E \\
& y_A \geq 0 \quad A \subseteq E
\end{aligned}$$

2 Part 2

Consider the set $S = \{e_{i_1}, \dots, e_{i_{n-1}}\}$ of edges chosen by Kruskal's algorithm. We will consider sets of the form $A_i = \{e_1, \dots, e_i\}$ for $1 \leq i \leq m$. For A_i , we let $y_{A_i} = c_{e_{i+1}} - c_{e_i}$, where $c_{e_{m+1}} = 0$, and for the remaining subsets of E , we will set $y_A = 0$. For the primal solution, we will have $x_e = 1$ iff Kruskal's algorithm picks edge e .

Firstly we claim that the constructed dual solution is a feasible solution of the dual LP.

Since the indexing of the edges is in sorted order of their costs, we have $c_{e_{i+1}} - c_{e_i} \geq 0$, so $y_{A_i} \geq 0$ for all i . For the other subsets of E , we have $y_A = 0$ identically, so it satisfies the last condition as well.

Consider the second constraint for a fixed edge e . Any subset of E having e that is not of the form A_i for some i has a contribution of 0 to the sum, so it suffices to look at sets of the form A_i . If $e = e_{i_k}$, then e is in $A_{i_k}, A_{i_k+1}, \dots, A_m$, and no other A_i 's. Hence the LHS is $\sum_{i=i_k}^m c_{e_{i+1}} - c_{e_i} = c_{e_{m+1}} - c_{e_i} = -c_{e_i}$, which satisfies the constraint.

Now we show that the complementary slackness condition is satisfied.

Note that for sets not of the form A_i , we have $y_A = 0$, so the complementary slackness condition is satisfied for the constraints on the variables corresponding to these subsets of E .

Now consider the set A_i . We shall show that we have $\sum_{e \in A_i} x_e = n - \alpha(A_i)$. By an earlier observation, it suffices to show that the number of components in (V, A_i) is exactly the same as the number of components in (V, A'_i) , where A'_i is the set of edges from A_i chosen by Kruskal's algorithm.

Suppose that this isn't the case. Then there exists a component in the sets of components of the graph (V, A_i) such that not all vertices of this component are in the same component in the graph (V, A'_i) (this is true since the number of components of (V, A_i) is strictly less than that of (V, A'_i) from our earlier observation, as they aren't equal). Hence, there exist two adjacent vertices in (V, A_i) which are in different components in (V, A'_i) (else we would have that any two adjacent vertices in (V, A_i) are in the same component in (V, A'_i) , and by induction, we would have that the number of components in (V, A_i) would be at least the number of components in (V, A'_i) , which would be a contradiction).

Consider the edge that joins those two vertices, say $e_r = (u, v)$. Consider what would have happened when Kruskal's algorithm was considering adding this edge into the solution. Since they are not in the same component in (V, A'_i) , they would not have been in the same component at that step either. So Kruskal's algorithm

must have made the greedy decision of taking that edge into the solution. This would mean that u, v are adjacent in (V, A'_i) , which is a contradiction.

Hence, we have shown that the complementary slackness conditions hold true for the dual variables.

Now we show the complementary slackness conditions for the primal variables.

Note that by our earlier observation, we always have $\sum_{A \subseteq E | e \in A} y_A = -c_e$ for all edges $e \in E$, so we are done.

3 Part 3

Note that we only need to show that the first inequality in both LPs are equivalent under the conditions of the other two constraints that are common to both LPs. We will do this part in two parts.

Showing that a feasible solution to the original LP is also a feasible solution to the new LP:

Let x be the feasible solution to the original LP. Consider a set $S \subset V$. Let A be the set of edges of the subgraph of (V, E) induced by S . Then we have $\sum_{e \in S} x_e \leq n - \alpha(A)$. It suffices to show that $n - \alpha(A) \leq |S| - 1$.

Note that the number of connected components in (V, A) is at least $n - |S| + 1$: this is because the vertices not in S are in a component of their own, and hence form $n - |S|$ components, and there is at least one component in S , since S is non-empty (else the condition for $S = \emptyset$ in the original LP formulation is itself a contradiction).

This implies that $\alpha(S) \geq n - |S| + 1$, and on rearranging we get $n - \alpha(A) \leq |S| - 1$. This completes the proof of this direction.

Showing that a feasible solution to the new LP is also a feasible solution to the original LP:

Suppose x is a feasible solution to the new LP. Consider any set $A \subset E$ of edges, and let $C_1, \dots, C_{\alpha(A)}$ be the connected components of the graph (V, A) .

By adding up the first constraint in the new LP for the sets $S = C_1, \dots, C_{\alpha(A)}$, we get $\sum_{e \in A} x_e = \sum_{i=1}^{\alpha(A)} \sum_{e \in C_i} x_e \leq \sum_{i=1}^{\alpha(A)} |C_i| - 1 = n - \alpha(A)$. This shows that x is indeed a feasible solution for the original LP as well.