CS 580	Algorithm Design, Analysis, And Implementation	
Fall 2021	Vassilis Zikas	HW 5

Problem 1

Collaborators: None

a.

Primal: The primal has a variable x_e associated with each edge in G(V,E).

The constraint tells us that for each s-t cut S, we pick at least one edge. This makes sure that the output path goes from s to t for sure.

The objective minimizes the total weight of a bunch of selected edges.

The variable x_e tells us if the edge e is chosen in the s-t minimum path or not (Can be fractional)

Dual: The dual has a variable y_s for each cut S in S'.

The variable y_S is a measure of how much the cut S contributes to the minimum distance path.

The constraint states that for all edges, the sum of all y_S is less that the edge weight. This is the edge weight constraint. It makes sure we don't use more edge weight that we have. This constraint also makes sure that we choose the minimum weight path.

The objective is to maximize the sum of variables for all cuts. This makes sure there is a path from s to t.

---

b.

Iteration 1: C_1 **is {s}**
$\delta(C_1)$ is (s,a), (s,b)
y_{C_1} can be incremented from 0 to 13 until it becomes equal to a edge weight (s,b). y_{C_1} **= 13** The edge e_1 **= (s,b)** chosen and x_{e_1} is set to 1.

Iteration 2: C_2 **is {s, b}**
$\delta(C_2)$ is (s,a), (a,b), (b,a), (b,c), (b,d)
y_{C_2} can be incremented from 0 to 3 until we get $y_{C_1} + y_{C_2} = 16 = c_{(s,a)}$ y_{C_2} **= 3** The edge e_2 **= (s,a)** chosen and x_{e_2} is set to 1.

Iteration 3: C_3 **is {s,a,b}**
$\delta(C_3)$ is (a,c), (b,c), (b,d)
y_{C_3} can be incremented from 0 to 6 until we get $y_{C_2} + y_{C_3} = 9 = c_{(b,c)}$ y_{C_3} **= 6** The edge e_3 **= (b,c)** chosen and x_{e_3} is set to 1.

---

c.

Let C be the connected component formed by the edges whose x_e = 1 at iteration t

By the end of iteration, an edge e' in δ(C) is added to the edges in the connected component in C.

We know that e' is in δ(C) =)= |{(u,v) in E : |{u,v} $\cap$ S =1}|

Let e' = (u,v). One vertex of e' is in C and the other is not. It means that adding this edge e' does not create a cycle or back edge!

So the set of edges F_t at the end of any iteration is a tree.

---

d.

Lets prove this by contradiction.

Statement 1: Assume for a y_S |P $\cap$ $\delta(s)$| $>$ 1.

It means that for a cut S in S', the s-t path consists of 2 edges from the cut.

Let the two edges be (a,c) and (b,d). They are in δ(S). a and b belong to S cut and c and d belong to t cut. Our algorithm increments y_S until it $\Sigma_{SinS':ein\delta(S)}$ y_S = c_e for some edge e in δ(S).

It would mean that there is a path from **s to a to c to t**

And another path from **s to b to d to t**.

This will clearly lead to the cycle in the edges e: $x_e > 0$

This contradicts the results of (c) which prove that there cannot be cycle in the set of edges e: $x_e > 0$ at the end of the algorithm

So Statement 1 is wrong.

Statement 2: Statement 1: Assume for a y_S |P $\cap$ $\delta(s)$| = 0.

This is a trivial case to prove. If |P $\cap$ $\delta(s)$| = 0, then path P from s to t cannot pass through the cut S of the graph G. It would mean there is no s-t path.

So Statement 2 is also wrong.

So |P $\cap$ $\delta(s)$| = 1

---

e.

The algorithm returns s-t path P. P has some or all edges from e: $x_e > 0$.

(d) states that at the end of the algorithm, for all S in S' if $y_S > 0$, then |P $\cap$ δ(S)| = 1.

It means there is exactly one edge through a cut S that the s-t path uses. It means that there is a simple path from s to t. The s-t path exists but we have to prove that it is minimum.

Proof by contradiction.

Statement 3: Assume that for some $y_S > 0$, the non optimal weight edge e" is chosen

For every $y_S > 0$, an edge would be added to the existing set of edges that have x_e = 1. Assume we added non optimal edge e'' during some iteration because $\Sigma_{SinS':e''in\delta(S)}$ y_S = $c_{e''}$

If we had a better edge e' which would minimize the total s-t distance, then y_S would have stopped increasing at that point. And e' would be chosen to be put into connected component.

This is a contradiction to Statement 3. So at every cut, the optimal edge is chosen that minimizes the s-t path.

Problem 2

Collaborators: None

a. The problem has an efficient algorithm that is polynomial in time.

The problem is to check if any spanning tree of a graph G(V,E) has weight lesser than 42 (At most 42).

Let us construct a minimum spanning tree (MST) for G using Kruskal's algorithm (Time complexity is O(|E|log|V|). Let s be equal to sum of the weights of all the edges in the MST (Can be found in O(n)).

If s <= 42, then the problem returns yes (Means, there is a spanning tree with weight lesser than or equal to 42), else the problem returns no (There is no spanning tree with weight lesser than or equal to 42)

Time complexity: O(|E|log|V|)
Space complexity: O(|E| + |V|)

---

b. The problem is NP-hard. We will prove it by reducing another NP-hard problem to this problem using Karp reduction. We can reduce Undirected Hamiltonian Path problem to our problem.

Note: Hamiltonian Path can be reduced to Hamiltonian Circuit problem by adding an edge between the start and end vertices of the path. Hamiltonian circuit can be reduced to Hamiltonian path problem by removing any edge. So they are equivalent problems.

In class, we learnt that 3-SAT can be reduced to directed Hamiltonian cycle problem which can be reduced to undirected Hamiltonian cycle problem. 3-SAT is NP-Hard. So problem of finding HAM cycle in undirected graph is also NP-hard due to properties of transitivity and reduction.

Let us call the problem of finding if G has a spanning tree with exactly 2 leaves as G-ST-2.

Proving G-ST-2 is NP-Hard:

1. Showing G-ST-2 is in NP: We have to show that the problem has a polynomial certifier to show it is in NP. Given a spanning tree of G that gives output "yes" for the problem, we can traverse the tree using BFS and find all the nodes that are leaves. The traversal takes O(|E|+|V|) time which is polynomial. So G-ST-2 has a polynomial certifier. It is in NP.

2. Reduce undirected Hamiltonian path(HAM) to G-ST-2
Claim: HAM $\leq_p$ G-ST-2

Proof:
a. Proving that "yes" instance of HAM will mean "yes" instance of G-ST-2:

Assume G has a Hamiltonian path P. P is a simple path by definition of Hamiltonian path. I means P visits all vertices V of G and it visits them in such a way that for vertices (v1, v2, v3, v4 ... vn) in G, path has edges (vi, vj) such that j = i+1, i = 1....n-1.

So P does not have any cycles. It is a tree which visits all nodes (Spanning tree). All nodes in P are internal nodes, expect v1 and vn. Therefore there is exactly 2 leaves. So G-ST-2 has output "yes" as well.

b. Proving that "yes" instance of G-ST-2 will mean "yes" instance of HAM:

Assume G has a spanning tree T with exactly 2 leaves. T visits all vertices V of G by definition of a spanning tree. The number of leaves is 2 which means it is actually a path such that path has edges (vi, vj) such that j = i+1, i = 1....n-1. So it is a Hamiltonian Path!

Hence proved that HAM $\leq_p$ G-ST-2

Since HAM is NP-Hard, G-ST-2 is also NP hard!

---

c. The problem is NP-hard. We will prove it by reducing another NP-hard problem to this problem using Karp reduction. We can reduce Undirected Hamiltonian Path (HAM) problem to our problem.

Lets call the problem of checking if graph G has a spanning tree with maximum degree 2 as **G-MD-2**

Proving G-MD-2 is NP-Hard:

1. Showing G-MD-2 is in NP: We have to show that the problem has a polynomial certifier to show it is in NP. For each vertex, we get its degree by counting the number of edges connected to it. If G is represented using Adjacency list, then we just traverse the list corresponding to each vertex and count the number of edges connected to it (which is its degree) and check if it is less than or equal to 3. The overall traversal takes O(|E|+|V|) time which is polynomial. So G-MD-2 has a polynomial certifier. It is in NP.

2. Reduce undirected Hamiltonian path(HAM) to G-MD-2
Claim: HAM $\leq_p$ G-MD-2

Proof:
a. Proving that "yes" instance of HAM will mean "yes" instance of G-MD-2:

Assume G has a Hamiltonian path P. P is a simple path by definition of Hamiltonian path. I means P visits all vertices V of G and it visits them in such a way that for vertices (v1, v2, v3, v4 ... vn) in G, path has edges (vi, vj) such that j = i+1, i = 1....n-1.

So P does not have any cycles. It is a tree which visits all nodes (Spanning tree). All nodes in P have degree 2 except v1 and vn, which have degree 1. So degrees of all vertices in P have a maximum value of 2. So G-MD-2 gives is a "yes" output as well.

b. Proving that "yes" instance of G-MD-2 will mean "yes" instance of HAM:

Assume G has a spanning tree T with maximum degree of 2. T visits all vertices V of G by definition of a spanning tree. Maximum degree of any node in the spanning tree T is 2. SO it can have e edges connected to it or 1. If 1 edge is connected to it, it is a leaf. If 2 edges are connected, it has one child and 1 parent. The only possible spanning tree of this form is a skewed tree. Which is actually a simple path from one leaf to another. Therefore it is a Hamiltonian path.

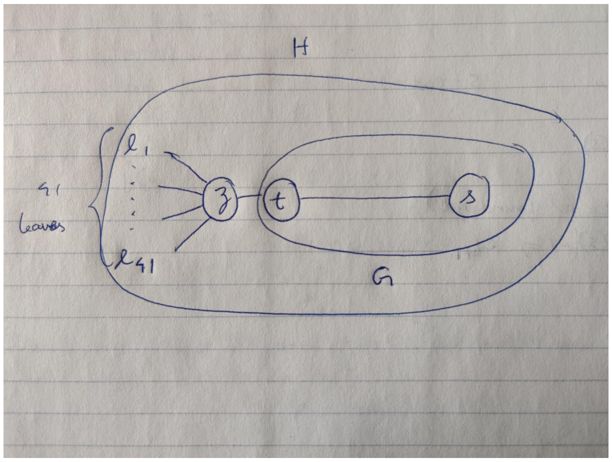
Hence proved that HAM $\leq_p$ G-MD-2

Since HAM is NP-Hard, G-MD-2 is also NP hard!

---

d. The problem is NP-hard. We will prove it by reducing another NP-hard problem to this problem using Karp reduction. We can reduce Undirected Hamiltonian Path (HAM) problem to our problem after slight modification and addition of a gadget.

Gadget: Let G be the original graph. To a node "t" connect another node "z" with an edge. Also add 41 nodes l1, l2, l3 ... l41 with edges connecting them to only z as shown in the diagram below. Let the modified graph be H.



Proving H-ST-42 is NP-Hard:

1. Showing H-ST-42 is in NP: We have to show that the problem has a polynomial certifier to show it is in NP. Given a spanning tree of H that gives output "yes" for the problem, we can traverse the tree using BFS and find all the nodes that are leaves. The traversal takes O(|E|+|V|) time which is polynomial. So H-ST-42 has a polynomial certifier. It is in NP.

2. Reduce undirected Hamiltonian path(HAM) of G to H-ST-42

Claim: G contains a Hamiltonian path (HAM) if and only if H has a spanning tree with at most 42 leaves (H-ST-42)

HAM $\leq_p$ H-ST-42

a. Proving that "yes" instance of HAM will mean "yes" instance of H-ST-42:

Assume G has a Hamiltonian path P from s to t. P is a simple path by definition of Hamiltonian path. Let T be the spanning tree of graph H. T will have the edges from all leaves to z, edge from z to t, and all the edges in the path P. T will have exactly 42 leaves (41 l nodes and s node). So it has atmost 42 leaves. So it outputs yes for H-ST-42.

b. Proving that "yes" instance of H-ST-42 will mean "yes" instance of HAM:

Assume H has spanning tree T of atmost 42 leaves. the 41 l nodes will be leaves in T. So the spanning tree would have the edges from z to 41 l nodes, and a simple path from z to s through t. s would be the 42nd leaf node. Any other case, and the number of leaves would be more than 42 which contradicts our assumption that T has atmost 42 leaves.

So if we remove all edges from z to l nodes and from z to t, we get a simple path from t to s in graph G. So HAM outputs true.

Hence proved that HAM $\leq_p$ H-ST-42

Since HAM is NP-Hard, H-ST-42 is also NP hard!