Ex 1

We will describe a decision problem fitting to the optimization problem seen in the given question.

Input:

* Graph .
* .
* A path length .

Output:

If there is a s.t and also .

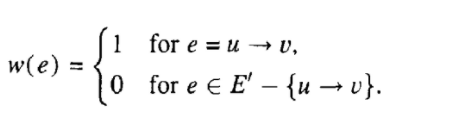
Section 2

Now we will prove that the decision problem we have formulated, is in NP. In order to prove so, we need to find a fitting function that can be computed in a polynomial time by a non-deterministic Turing machine.   
First we find the shortest path between and using Dijkstra. Now we set .   
Next, our function takes as input a graph G and and traverse on the graph from to any adjacent node to , and from each of those to their adjacent nodes (giving that this node not already on the current path) and so on and so forth. Each path that reaches to and the current path is different from we add it to , and set  
 .   
Now we prove that our function run in a polynomial time. We can see that the length of the longest path between two vertices in the graph that our function would check is as long as . And is finite.

Now we will show, that given a witness, we can verify its correctness in a polynomial time.   
Given a witness We can check in a polynomial time if the path is legal () , .

Section 3

Lemma 3 in the article ‘Cumputing strictly-second shortest path’ state that the ‘edge-connecting simple-cycle problem’ is NP-Hard.

Its possible to prove that the decision problem above is NP-Hard by reduction to is the edge-connecting simple-cycle problem.   
Given a directed graph G = <V,E> and 2 edges e1 :=(u,v), e2 :=(a,b) (**belong to G** ) we construct a new graph G’ := <V’, E, w> with V’ = V, , and 

By construction, the shortest path from b to a is the path p1 that include only the added edge (b,a) while w(p) = 0.  
its straightforward that every other path’s weight from b to a that not include the edge (u,v) is 0, therefore the if there is a path from b to a that include (u,v) its must be the second shortest path, with weight 1.

Its follows that, given paths , the cycle solves the strictly-second-shortest simple-path problem from b to a in G’.

Section 3

The defined problem for undirected graph is indeed NP-hard as well.  
in order to prove so we will convert the given undirected graph G to a directed graph G’ by switching each of the edges (u,v) in G with 2 edges .  
doing so we haven’t change any of G’s paths, therefore we can use the same proof on this case.