The 4th Project Report

Samira Vaez Barenji

Student ID Number: 9921384

In this project, you can find a solution to the maximum number of edge disjoint paths problem in a graph. Two paths are said to be edge disjoint if they don't share any edge.

This problem can be solved by reducing it to maximum flow problem, following the below steps:

1) Consider the given source and destination as source and sink in flow network. Assign unit capacity to each edge.

2) Find the maximum flow from source to sink.

3) The maximum flow is equal to the maximum number of edge-disjoint paths.

When we run Ford-Fulkerson, we reduce the capacity by a unit. Therefore, the edge cannot be used again. So, the maximum flow is equal to the maximum number of edge-disjoint paths. We use Edmonds-Karp implementation of Ford-Fulkerson algorithm to find the maximum flow from source to sink.

So, we can use a maximum flow algorithm to find k edge-disjoint, s-t paths in a graph. Embedded within any flow of value k on a unit-capacity graph there are k edge-disjoint paths. In other words, *the value of the flow or the capacity of the minimum cut gives us the number of edge disjoint paths*.

The idea of Edmonds-Karp is to use BFS in Ford Fulkerson implementation as BFS always picks a path with minimum number of edges. When BFS is used, the worst-case time complexity can be reduced to $O(VE^2)$. This implementation uses adjacency matrix representation though where BFS takes $O(V^2)$ time, the time complexity of the above implementation is $O(EV^3)$, where E is the number of edges and V is the number of vertices.

You can find the resulting output image below for a sample unweighted graph.

```
    [0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 1, 0, 0, 0, 0, 1],
    [0, 1, 0, 0, 0, 0, 0, 1],
    [0, 0, 0, 0, 0, 1, 0, 1],
    [0, 0, 0, 0, 0, 0, 0, 0]]



source = 0
sink = 7
print ("There is maximum %d edge-disjoint paths from %d to %d to save the sheep!" %
      (disjoint_path(graph, source, sink), source, sink))
```

```
There is maximum 2 edge-disjoint paths from 0 to 7 to save the sheep!
```