Theory Assignment-3: ADA Winter-2023

Deepanshu Dabas (2021249) Ankush Gupta (2021232)

April 26, 2023

---

1. The towns and villages of the Island of Sunland are connected by an extensive rail network. Doweltown is the capital of Sunland. Due to a deadly contagious disease, recently, few casualties have been reported in the village of Tinkmoth. To prevent the disease from spreading to Doweltown, the Ministry of Railway of the Sunland wants to completely cut down the rail communication between Tinkmoth and Doweltown. For this, they wanted to put traffic blocks between pairs of rail stations that are directly connected by railway track. It means if there are two stations x and y that are directly connected by railway line, then there is no station in between x and y in that particular line. If a traffic block is put in the track directly connecting x and y, then no train can move from x to y. To minimize expense (and public notice), the authority wants to put as few traffic blocks as as possible. Note that traffic blocks cannot be put in a station, it has to be put in a rail track that directly connects two stations.
   Formulate the above as a flow-network problem and design a polynomial-time algorithm to solve it. Give a precise justification of the running time of your algorithm. **(10 Marks)**

Ans:

Formulation into Network Flow Problem
We can construct a directed graph $G = (V, E)$ as follows:

- Each vertex in V corresponds to a station in the rail network.
- For each railway track that directly connects two stations u and v, we add two directed edge (u, v), (v, u) to E each with capacity 1.

  Note: *If we used $u \to v$ in the augmented path, we will not consider $v \to u$ in this path as it will lead to u again and form loop. However, we can consider any of the paths $u \to v$ or $v \to u$, and therefore we need to add both edges in a graph with the same capacity=1 .*

- We set the source vertex s to be Doweltown and the sink vertex t to be Tinkmoth.

This formulation allows us to apply the max-flow min-cut theorem to find the minimum number of traffic blocks required to disconnect Tinkmoth from Doweltown. Hence,by

this formulation we need to find the minimum number of edges to block to make s-t disconnected from each other which we can find using *"Ford-Fulkerson Algorithm for finding maximum flow"*

Explaination of how maximum-flow value in network is equivalent to the solution to the original problem:

- Since the solution to the original problem is the minimum number of traffic blocks required to disconnect Tinkmoth from Doweltown in our network, it is equivalent to finding the minimum number of edges to block to make the source disconnected from the sink.
- The minimum number of edges to block will equal the maximum flow value in the network. (The max-flow min-cut theorem states that the maximum flow value from source to sink equals the minimum cut value between source and sink in the network. The minimum cut value is defined as the minimum number of edges that need to be removed from the network to disconnect the source and sink.)
- Therefore, from the above points, the maximum flow value from source to sink in our network will correspond to the solution of the original problem.
- We can find the maximum flow value using "Ford- Fulkerson algorithm" and hence the solution of the original problem.

Running Time Analysis

Let F be the maximum flow in the network, E be the number of edges in the graph, and V be the number of vertices in the graph.
Then he running time analysis of this algorithm can be broken down into the following steps:

- Graph Creation Time: $\mathcal{O}(V + E)$
- Algebraic operations $=\mathcal{O}(1)$
- The time complexity of the Ford-Fulkerson algorithm is $\mathcal{O}(E \cdot F)$ where E is the number of edges and F is the maximum flow in the network. This is because each augmenting path can be found in O(E) time and the number of augmenting paths is at most F.
- But since $F \leq E$,the time complexity of Ford-Fulkerson will be $\mathcal{O}(E^2)$.

Therefore, the overall time complexity of this algorithm is $\mathcal{O}(E^2)$. Since E $\leq \frac{V \cdot (V-1)}{2}$, the time complexity of this algorithm is $\mathcal{O}(V^4)$ and hence a polynomial time algorithm.

Psuedo Code Of Algorithm

Algorithm 1 Ford Fulkerson Maximum Flow

```
Input: src, dest, graph, res
Output: maxFlow
parent = integer array of size of vertices array;
 maxFlow ← 0;
 n ← size of graph;
for i = 0 to n - 1 do
    for j = 0 to size of graph[i] - 1 do
        v ← graph[i][j];
         res[i][v[0]] ← v[1];
    end
end
while bfs(src, dest, parent, visited, graph, res) do
    bottleneck ← ∞;
    for v = dest to src do
        u ← parent[v];
         bottleneck ← min(bottleneck, res[u][v]);
    end
    for v = dest to src do
        u ← parent[v];
         res[u][v] ← res[v][u] - bottleneck;
         res[v][u] ← res[v][u] + bottleneck;
    end
    maxFlow ← maxFlow + bottleneck;
end
return maxFlow;
```

Algorithm 2 Get Minimum Cut

```
Input: src, dest, graph
Output: minCut
n ← size of graph;
 res ← integer array of size n by n;
for i = 0 to n - 1 do
    for j = 0 to n - 1 do
        res[i][j] ← 0;
    end
end
for i = 0 to n - 1 do
    for j = 0 to size of graph[i] - 1 do
        v ← graph[i][j];
         w ← v[0], c ← 1;
         res[i][w] ← c;
         res[w][i] ← c;
    end
end
minCut ← maxFlow(src, dest, graph, res);
 return minCut;
```

Acknowledgement:

- Class Lecture 16,17,18
- Algorithms by Jon Kleinberg
- Jon Kleinberg Algorithms Course
- Syamantak Sir YT Playlist