Theory Assignment-3: ADA Winter-2023

Chaitanya Arora (2021033)   Raghav Sakhuja (2021274)

1 Question

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.

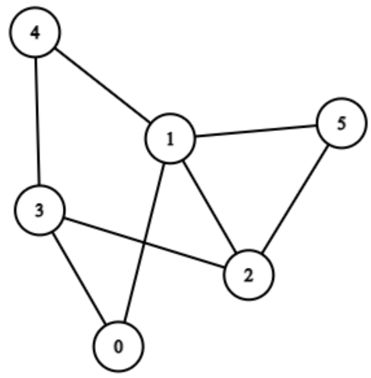
1.1 Answer (a)

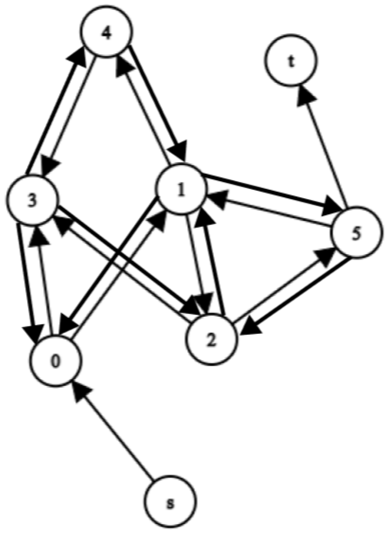
1.1.1 Graph-Description and formulation

To identify this question as a network flow problem, we should identify the source as the city of Tinkmoth and sink as the city of Doweltown. As the railroad connects two cities, we represent a rail connection by an edge, and the railway stations as the nodes in the graph. In the original problem, the railroads are undirected, so the initial layout of the city is a undirected graph. We convert this into a network flow problem by switching undirected edges into two edges each of capacity 1 of opposite directions. If there was a railroad connecting x and y, then we create two edges, one from x to y, and another from y to x each of capacity 1.

We further add 2 more nodes, one being the source and other being the sink. We connect the source s to all the stations in Tinkmoth with edges having infinite capacity and connect all the stations in Doweltown with sink t with edges having infinite capacities.

Here is an example of the contruction. Initially , station 0 is in Tinkmoth and station 5 is in Doweltown. We convert this graph into the graph on the right.





1.1.2 Equivalence of Graph

To prove this equivalence, we can show that the maximum flow value produced by the Ford-Fulkerson algorithm on this graph corresponds to the minimum cut capacity. The algorithm starts with an initial flow of 0 and iteratively finds augmenting paths in the residual network until no such path exists. At this point, the flow produced by the algorithm is at its maximum value.

The minimum cut can be obtained by identifying a set of stations S such that Tinkmoth is in S and Doweltown is not in S. The capacity of this cut is equal to the sum of the capacities of the edges going from S to its complement. Since the capacities of all edges are 1, the capacity of the minimum cut is also equal to the number of edges that cross the cut.

By the max-flow min-cut theorem, the maximum flow value produced by the Ford-Fulkerson algorithm is equal to the capacity of the minimum cut. Thus, the maximum flow value represents the minimum number of traffic blocks that need to be placed to prevent the disease from spreading to Doweltown.

1.1.3 Correctness

The properties of the flow network algorithm can be explained as follows. First, the skew symmetry property ensures that the flow in the reverse direction is always updated with the opposite value whenever the flow is updated. Second, the flow conservation property states that the augmenting path always passes through the nodes in the residual network that are reachable from the source node s and lead to the sink node t. These two properties are proven in the attached paper.

To determine the minimum cut that separates s and t, we can take a cut (R, R') from the residual network after running the maximum flow algorithm. If we can show that this cut is the minimum cut, we have proven that the maximum flow is equal to the capacity of the minimum cut.

To demonstrate that (R, R') is a minimum cut, we can argue that any flow from s to t must pass through R because R contains all the reachable nodes in the residual network. Therefore, the capacity of the cut (R, R') must be equal to the value of the maximum flow. If there were a smaller cut separating s and t, then there would be a path from s to t that does not cross (R, R'), which contradicts the fact that R contains all the reachable nodes from s. Hence, (R, R') is indeed the minimum cut separating s and t, and the maximum flow is equal to the capacity of this cut.

The algorithm makes sure, that if an edge and its reverse both have a positive flow, it can simply be remade into a new flow such that the reverse flow can simply be subtracted from the one forward direction.

The algorithm ensures that flow is updated symmetrically and conserved, and by finding the minimum cut that separates the source and sink nodes, we can determine the maximum flow. Therefore, we can conclude that the maximum flow is exactly equal to the capacity of the minimum cut.

1.1.4 Run-time

The runtime of this algorithm is same as that of Ford Fulkerson as we use that. Thus the time complexity is

$$O(fE)$$

Also, since all the edges have only the capacity of 1, we can only have a max flow on E in the graph.

$$f = O(E)$$

Thus the final time complexity is of order

$$T = O(E.E) = O(E^2)$$

1.1.5 References

We have taken inspiration from the following research paper regarding the working and understanding of the algorithm of Ford Fulkerson in an undirected graph.

$$https://www.inf.ufpr.br/elias/papers/2004/RT_DINF003_2004.pdf$$

Computing the Minimum Cut and Maximum Flow of Undirected Graphs by Jonatan Schroeder, Andr´e Pires Guedes, Elias P. Duarte J.