hw9-4102

jh4ctf

November 2021

1

Original/Starting Graph G_f:

	S	U	V	T
S		10	5	
U	0		15	5
V	0	0		10
T		0	0	

Final/Ending Graph G_f:

	S	U	V	T
S		0	0	
U	10		10	0
V	5	5		0
T		5	10	

The final maximum flow of f for this graph is 15

2

The maximum number of iteration can be 2 million. Because in the worst case, we will iterate through the path s,u,v,t and decrease the residual capacity by 1 on the path. However, on the next iteration, we can be iterating through the path s,v,u,t accordingly with the back flow edge. However, the residual capacity of edge (u,t) and (v,t) are in worst case decrease by 1 at each time. The algorithm takes up to 2 million iteration to complete. Thus, the issue is raised that the worst case can slow down the run-time of the algorithm dramatically because of the horrible path choice to iterate through.

3

The first step of our algorithm is to run Ford Fulkerson Algorithm to get the max flow and each edges corresponding flow value. Then we can push back(reversely) find all the augmented path that all the edges can be on. Then we set the flow value to be the capacity of each edge and utilize DFS to find all the path from the source node. On each path, we add the minimum edge's flow up onto other edge's flow. Once an edge reaches its capacity, the edge will be deleted from the path. Then we repeat this step until all the edges are removed from the path(or there is no more augmented path anymore. Since each edge will be removed once, so it won't utilize more than $|E|$ augmenting path.