Homework 4
Colorado CSCI 5454

Alex Book

September 21, 2021

People I studied with for this homework: Cole Sturza
Other external resources used: N/A

Problem 1

Part a

The flow along edge (u, t) exceeds the capacity of the edge (flow of 4, capacity of 3). This does not respect the capacity constraint property.

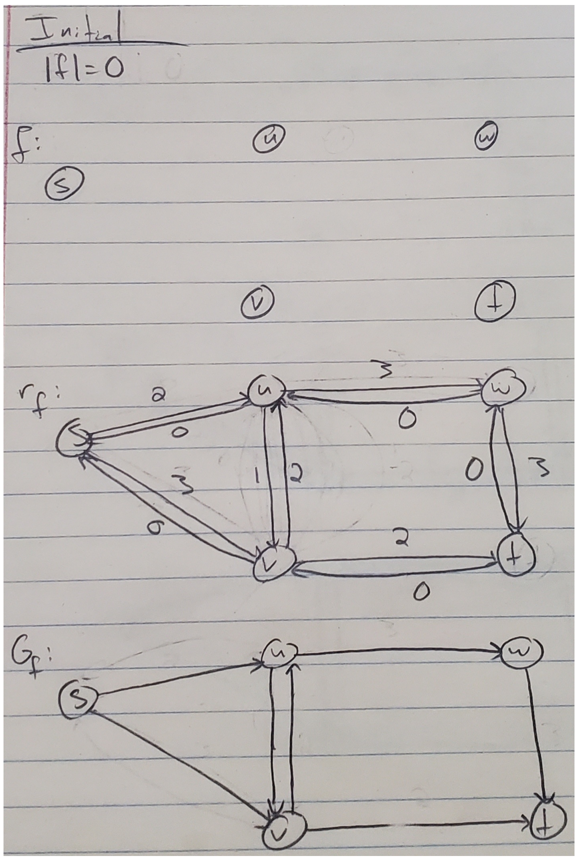
Part b

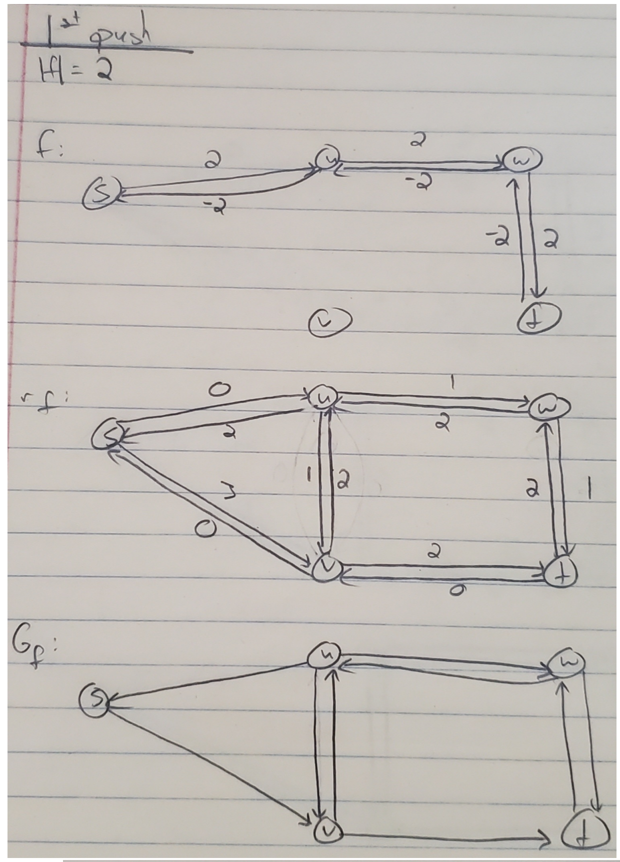
Conservation of flow is not observed, as the flow along edge (s, u) is 3, but the flow along edge (u, t) is 4. This does not respect the net flow constraint property.

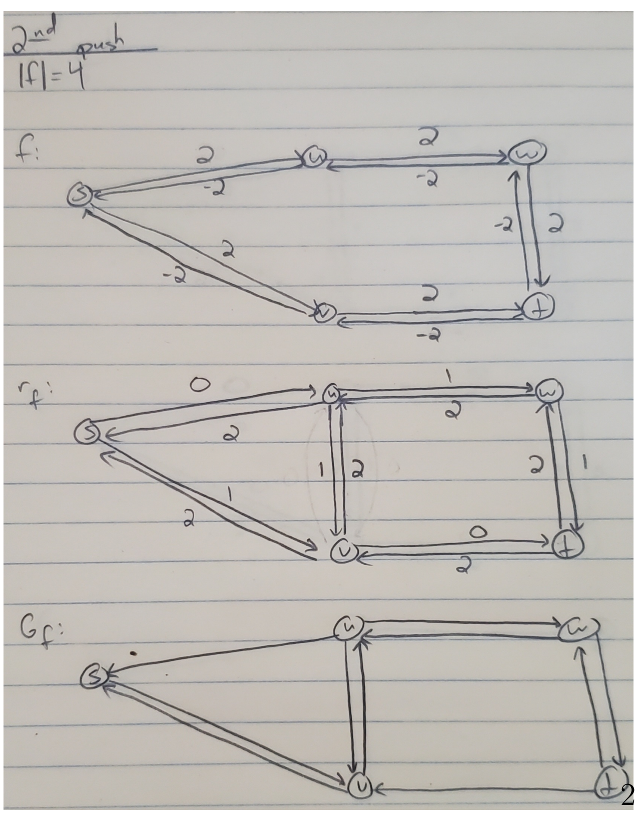
Part c

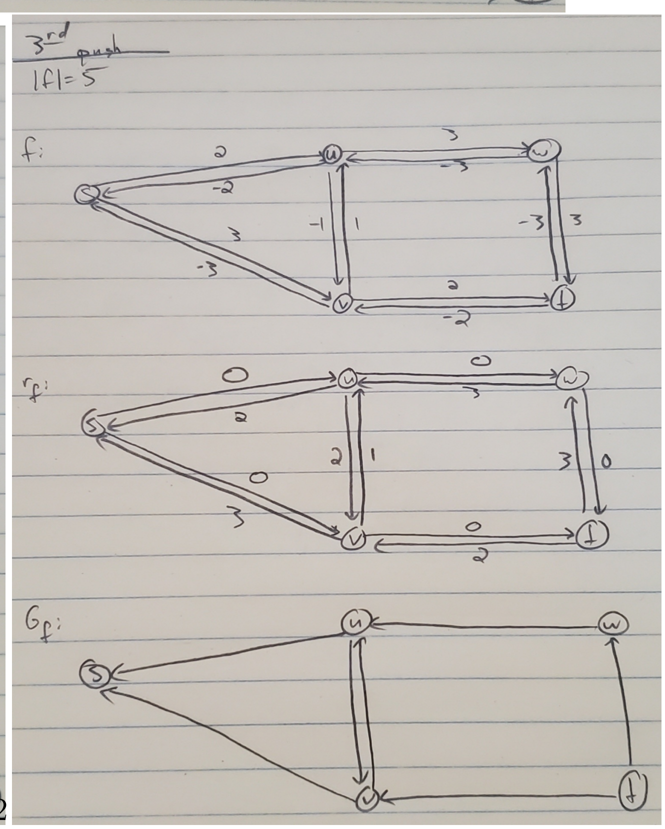
The flow along edge (s, u) is not equal to the negative flow along edge (u, s). This does not respect the skew-symmetric property.

Problem 2


Initial
|f|=0
f:
rf:
Gf:


1st push
|f| = 2
f:
rf:
Gf:


2nd push
|f|=4
f:
rf:
Gf:


3rd push
|f|=5
f:
rf:
Gf:

Problem 3

Part a

Set all vertex capacities to ∞. This is equivalent to the original problem where only edge capacities are specified, as the vertices can handle any amount of flow. This would allow VertCapAlg to solve the original problem.

Part b

First, we split each non-source and non-sink vertex v into two new vertices, labeled v_{in} and v_{out}, and draw an edge between them of capacity w_v. We then reroute all incoming edges (u, v) into v_{in} and all outgoing edges (v, w) out of v_{out}. We then remove all vertex capacities. Let us call this new graph G'. This new problem can then be reduced to the original problem with only edge weights specified, as the constraints presented by the vertex capacities are handled by the "interior" edges (v_{in}, v_{out}), as all flow through a single vertex v in the given graph G must pass through the edge (v_{in}, v_{out}) in the new graph G'.

Thus, we can simply use the Ford-Fulkerson framework to solve this problem. We reference the correctness proofs and running time analysis presented in Lecture 5 notes (on graph $G' = (V', E')$, Edmonds-Karp has a running time of $O(|V'| \cdot |E'|^2)$).

Part c

Set all edge capacities to ∞. Since VertCapAlg can solve the problem when edge and vertex capacities are given, the problem is solved.

Part d

Consider the process of creating the graph G' as described in Part b. We use that to create such a graph G' for this problem. We then set the capacities of all "non-interior" edges (all edges that are not of the form (v_{in}, v_{out}) to ∞. We then use an algorithm of the Ford-Fulkerson framework to push flow through the graph (e.g. Edmonds-Karp). Once that algorithm concludes, we can use the final G_f to find the minimum edge cut (we would use BFS to find the correct partition of edges, which has been proven to be correct and efficient, traversing only as far is as required, as we reach the correct paritioning as soon as there are no remaining edges on which to reach more vertices). Since the only non-∞ edge capacities are interior edges, those must be the only possible forward edges not in G_f (as G_f stores only the edges with remaining capacity, and edges with infinite capacity will never reach a capacity of 0). Thus, the minimum edge cut must only be through interior edges. Once this cut is found, we can say that the minimum vertex cut includes only the vertices that are connected by the interior edges that were included in the minimum edge cut (e.g. if edge

(x_{in}, x_{out}) is included in the minimum edge cut, vertex x must be included in the minimum vertex cut)