Homework Assigment - 8

Dhayalini Nagaraj
A20359686

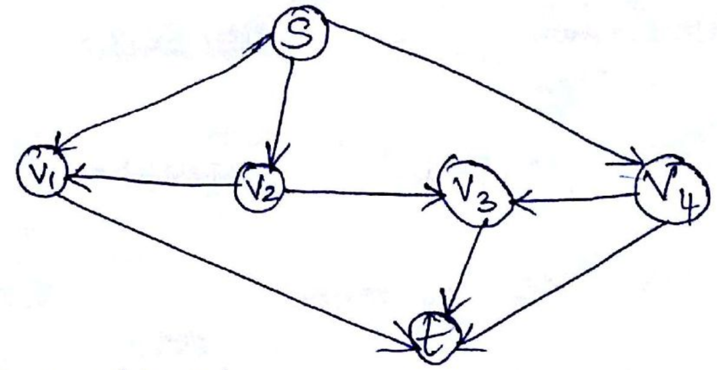
1. a) Prove the claim in the footnote on T25: If the capacities are rational, the Ford-Fulkerson algorithm does not fail. Give an example in which it does fail (clearly you will need to have irrational capacities).

The running time of FORD-FULKERSON depends on how we find the augmenting path p. If we choose it poorly, the algorithm might not terminate. The maximum flow problem arises with integral capacities. If the capacities are rational numbers, we can apply an appropriate scaling transformation to make them all integral. So to transform rational capacities into integers let us multiply all the rational capacities by divisor's least common multiple.

The Ford-Fulkerson algorithm augments the capacity by atleast one for each augmenting ~~as~~ path. So it must terminate. The original max flow can be found by dividing the result by the least common multiple.

This rescaling technique for rational capacities does not apply to irrational capacities. The augmenting path can be arbitarily chosen, ~~it is possible~~ When the augmenting path is arbitarily chose, it is possible to have infinite number of loops.

Let us consider a network with irrational capacities.




~~In this network, consider where~~
Consider this network, where

$$c(V_2, V_1) = c(V_2, V_3) = 1$$
$$c(V_4, V_3) = r = \frac{\sqrt{5}-1}{2}$$

The capacity of all other edges = 2.

Let us choose the series of augmenting paths as

$$P_0, P_1, P_2, P_1, P_3, P_1, P_2, P_1, P_3, \ldots\ldots, P_1, P_2, P_1, P_3 \ldots$$

$P_0 = < s, v_2, v_3, t >$

$P_1 = < s, v_4, v_3, v_2, v_1, t >$

$P_2 = < s, v_2, v_3, v_4, t >$

$P_3 = < s, v_1, v_2, v_3, t >$

The flows of this series of augmenting paths are $1, r, r, r^2, r^2, \ldots,$

This will never terminate. Thus, the algorithm fails to return the max-flow of this network with this choice of augmenting paths.

This network has max-flow as 5 by choosing the paths $< s, v_1, t >$, $< s, v_4, t >$ and $< s, v_{2,3}, t >$. The algorithm also returns

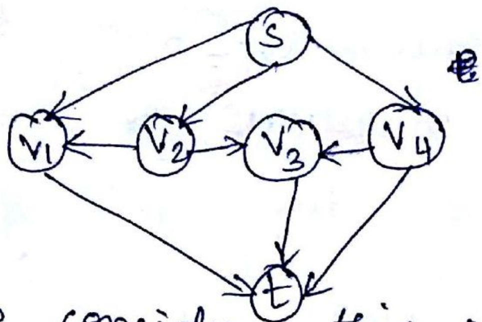
$$1 + r + r + r^2 + r^2 + \ldots\ldots = 1 + 2 \times \frac{r}{1-r}$$

$$1 + 2 \times \frac{r}{1-r} \neq 5.$$

So it does not converge to the maximum flow.

b) Give an example in which it fails and does not converge to the maximum flow.

As seen in the above question rescaling technique ~~does~~ for rational capacities does not apply to irrational capacities.
The augmenting path can be arbitarily chosen and it ~~may have~~ is possible to have infinite no. of loops.




Let us consider this network with irrational capacities ~~and~~ where

$$c(V_2, V_1) = c(V_2, V_3) = 1$$

$$c(V_4, V_3) = r = \frac{\sqrt{5}-1}{2}$$

The other edges capacity $= 2$

Choose series of augmenting path as

$P_0, P_1, P_2, P_1, P_3, P_1, P_2, P_1, P_3, \ldots, P_1, P_2, P_1, P_3 \ldots,$

$P_0 = \langle s, v_2, v_3, t \rangle$

$P_1 = \langle s, v_4, v_3, v_2, v_1, t \rangle$

$P_2 = \langle s, v_2, v_3, v_4, t \rangle$

$P_3 = \langle s, v_1, v_2, v_3, t \rangle$

series of augmenting paths flows are

$1, r, r, r^2, r^2, \ldots,$

This never terminates. So the algorithm fails to return the max flow of this network with this choice of augmenting paths.

This network has max flow as 5 by choosing the paths $\langle s, v_1, t \rangle$, $\langle s, v_4, t \rangle$ and $\langle s, v_2, 3, t \rangle$.

However, the algorithm returns

$$1 + r + r + r^2 + r^2 + \cdots = 1 + 2 \times \frac{r}{1-r}$$

$$1 + 2 \times \frac{r}{1-r} \neq 5.$$

So this shows that Ford-Fulkerson algorithm fails and ~~does~~ does not converge to the maximum flow when we have irrational capacities.

2) Problem 26.2-10 on page 731. The hint given means that you are to prove the existence of the sequence of augmentations, not that they could be found before you have ~~of~~ already found the max flow.

From Edmond's - Karp we can solve this.

If a maximum flow is given, let us first find an edge $(u,v)$ such that $f(u,v)$ is minimum and it is larger than 0. Then find a path from $s$ to $t$ containing $(u,v)$, and reduce the flow on that path by $f(u,v)$.

The edge $(u,v)$ will not be selected again because its flow is now 0.

Repeat the same process until the total flow is reduce to 0. Since one edge is removed each time, we should do this at most $|E|$ times. We reduced flow on each path and those paths could have been an augmenting path. So we could get to our max-flow with at most $|E|$ augmentations.

3. Prove Lemma 26.19 by induction on the number of push/relabel steps.

We will prove that

(1) any overflowing vertex $x$ in the initialization step has a simple path to $s$

(2) suppose the vertex $x$ has a simple path to $s$ and after a RELABEL or PUSH, $x$ and any new overflowing vertex $u$ induced by RELABEL and PUSH still have a simple path to $s$.

<u>Initialisation</u>: In the begining of the algorithm we send the flow from $s$ on height $|V|$ to its neighbors $N_s$ on height 0. This is because every vertex except $s$ is assigned to

height 0 in the initialization. At this point, the overflowing vertices are only those that are in $N_s$. In the residual network there are exactly the same number of edges from $N_s$ to $S$. These edges can be considered as simple paths from ~~overflowing~~ overflowing vertices to $S$, as desired.

Induction: If there is a simple path $p$ from $x$ to $s$ and $x$ is relabeled. Then nothing is changed except for $x.h$. So $p$ is still a simple path from $x$ to $s$. If there is a PUSH at $x$, then $x$ itself ~~it~~ is still connected to $s$ by its previous simple path.

We need to show that any $x$'s downhill vertex that becomes overflowing also has a simple path back to $s$.

Let us consider the PUSH affects this path $x \rightarrow u \rightarrow \dots \rightarrow v$. According to residual network definition, as long as $x$ pushes flow along $u \rightarrow \dots \rightarrow v$, there must be a path $v \rightarrow \dots \rightarrow u$ in $G_f$. That is any new overflowing vertex in the affected path $u \rightarrow \dots \rightarrow v$ has a simple path to $x \rightarrow s$ in $G_f$ as desired.

References:

algo2.iti.kit.edu/documents/Algorithmentechnik/maxflow.pdf
https://books.google.com/books?id=ASXUY604pew
De Brujin's combinatorics.
theory.Stanford.edu
https://lucatrevisan.wordpress.com.
wenku.baidu.com.