

CS5800: Algorithms — Spring '21 — Virgil Pavlu

Homework 12

Submit via [Gradescope](#)

Name: Zerun Tian

Collaborators:

Instructions:

- Make sure to put your name on the first page. If you are using the \LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- Please review the grading policy outlined in the course information page.
- You must also write down with whom you worked on the assignment. If this changes from problem to problem, then you should write down this information separately with each problem.
- Problem numbers (like Exercise 3.1-1) are corresponding to CLRS 3rd edition. While the 2nd edition has similar problems with similar numbers, the actual exercises and their solutions are different, so make sure you are using the 3rd edition.

1. (20 points) Exercise 26.1-3.

Suppose that a flow network $G = (V, E)$ violates the assumption that the network contains a path $s \rightsquigarrow v \rightsquigarrow t$ for all vertices $v \in V$. Let u be a vertex for which there is no path $s \rightsquigarrow u \rightsquigarrow t$. Show that there must exist a maximum flow f in G such that $f(u, v) = f(v, u) = 0$ for all vertices $v \in V$.

Solution:

For vertices v who are not immediately connected to u , i.e. $(u, v) \notin E$ and $(v, u) \notin E$, it is trivial to show that $f(u, v) = f(v, u) = 0$ in any case, including the case of maximum flow. According to the observation at the end of page 709 of CLRS, we know $f(u, v) = 0$ for $(u, v) \notin E$.

For vertices v who are immediately connected to u , i.e. $(u, v) \in E$ or $(v, u) \in E$, let's show that in a maximum flow, $f(u, v) = f(v, u) = 0$. If flows on those edges were ever non-zero, to preserve the flow conservation property, there will be flow out when there is flow in at u , and vice versa. In this case, given the network contains a path $s \rightsquigarrow v \rightsquigarrow t$ for all vertices $v \in V - \{u\}$, there would also be a path $s \rightsquigarrow u \rightsquigarrow t$: a contradiction.

Since we know the vertex u is not in any path from s to t , there will be no augmenting path flowing through the vertex u . By running the FORD-FULKERSON algorithm, we can find a maximum flow without ever updating (u, v) and (v, u) edges. Hence, those edges incoming to u and outgoing from u have the initial zero flow.

Therefore, we showed that for all vertices $v \in V$, $f(u, v) = f(v, u) = 0$.

2. (20 points) Exercise 26.1-4.

Let f be a flow in a network, and let α be a real number. The *scalar flow product*, denoted αf , is a function from $V \times V$ to \mathbb{R} defined by

$$(\alpha f)(u, v) = \alpha \cdot f(u, v).$$

Prove that the flows in a network form a *convex set*. That is, show that if f_1 and f_2 are flows, then so is $\alpha f_1 + (1 - \alpha)f_2$ for all α in the range $0 \leq \alpha \leq 1$.

Solution:

To show $\alpha f_1 + (1 - \alpha)f_2$ is a flow, we test if it satisfies the flow properties: capacity constraint and flow conservation.

Because f_1 and f_2 are valid flows, for all $u, v \in V$, we have that $f_1(u, v) \leq c(u, v)$ and $f_2(u, v) \leq c(u, v)$. Then, we derive,

$$\begin{aligned} \alpha f_1 + (1 - \alpha)f_2 &= \alpha f_1(u, v) + (1 - \alpha)f_2(u, v) \\ &\leq \alpha c(u, v) + (1 - \alpha)c(u, v) \\ &= c(u, v) \end{aligned}$$

Now that the new flow satisfies the capacity constraint, we check if it maintains flow conservation. Both f_1 and f_2 satisfy the capacity constraint where $\sum_{v \in V} f_1(v, u) = \sum_{v \in V} f_1(u, v)$ and $\sum_{v \in V} f_2(v, u) = \sum_{v \in V} f_2(u, v)$ for every $u \in V - \{s, t\}$. Then, the new flow has the form for every vertex u ,

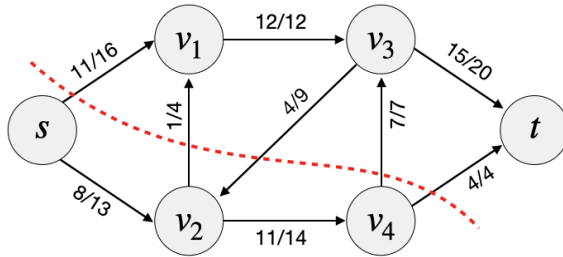
$$\begin{aligned} \sum_{v \in V} \alpha f_1(v, u) + (1 - \alpha)f_2(v, u) &= \alpha \sum_{v \in V} f_1(v, u) + (1 - \alpha) \sum_{v \in V} f_2(v, u) \\ &= \alpha \sum_{v \in V} f_1(u, v) + (1 - \alpha) \sum_{v \in V} f_2(u, v) \\ &= \sum_{v \in V} \alpha f_1(u, v) + (1 - \alpha)f_2(u, v) \end{aligned}$$

The flow conservation holds because the flow in equals flow out w.r.t every vertex $u \in V - \{s, t\}$. Therefore, the flows form a convex set.

3. (20 points) Exercise 26.2-2.

In Figure 26.1(b), what is the flow across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$? What is the capacity of this cut?

Solution:



The flow of the cut is:

$$\begin{aligned} \sum_{u \in S; v \in V} f(u, v) &= f(s, v_1) + f(v_2, v_1) + (-f(v_3, v_2)) + f(v_4, v_3) + f(v_4, t) \\ &= 11 + 1 - 4 + 7 + 4 = 19 \end{aligned}$$

The capacity of the cut is:

$$\begin{aligned} \sum_{u \in S; v \in V} w(u, v) &= w(s, v_1) + w(v_2, v_1) + w(v_2, v_3) + w(v_4, v_3) + w(v_4, t) \\ &= 16 + 4 + 0 + 7 + 4 = 31 \end{aligned}$$

4. (*Extra Credit*) Exercise 26.2-10.

Solution:

5. (30 points) Implement Push-Relabel for finding maximum flow.

Extra Credit: use relabel-to-front idea from Chapter 26.5 with the Discharge procedure.

Solution:

Code is not yet complete. Will do the demo after the final.

6. (15 points) Explain in a brief paragraph the following sentence from textbook page 737: “To make the preflow a legal flow, the algorithm then sends the excess collected in the reservoirs of overflowing vertices back to the source by continuing to relabel vertices to above the fixed height $|V|$ of the source”.

Solution:

When we can no longer send flows to the sink, i.e. maximum flow is reached using the push-relabel method, we will have some overflowing vertices. A vertex u is overflowing if it has some excess in it, i.e. $e(u) > 0$, which is caused by having flow out less than flow in. Since a legal flow needs to satisfy the flow conservation property, we will have to send the excess back to where it is originated until the source through the “reverse” edges. To be able to send flows back to the source, we have to increase the height of those overflowing vertices above $|V|$.

7. (*Extra Credit*) Exercise 26.4.4.

Solution: