Computer Science C73 November 23, 2022
Scarborough Campus University of Toronto

Solutions for Homework Assignment #7

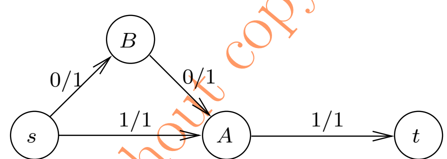
Answer to Question 1.

a. This statement is true. We prove the contrapositive: If some edge e that crosses the minimum cut (S, T) is not saturated by flow f, then f is not a maximum flow. Let f^* be a maximum flow.

$$\begin{aligned}
\mathcal{V}(f) &= \sum_{e' \in out(S) \cap in(T)} f(e') - \sum_{e' \in out(T) \cap in(S)} f(e') && \text{[by the flow lemma]} \\
&\leq \sum_{e' \in out(S) \cap in(T)} f(e') \\
&< \sum_{e' \in out(S) \cap in(T)} c(e') && \text{[since } f(e') \leq c(e') \text{ for every edge } e' \text{ and } f(e) < c(e)\text{]} \\
&= \mathcal{V}(f^*) && \text{[by the max-flow-min-cut lemma]}
\end{aligned}$$

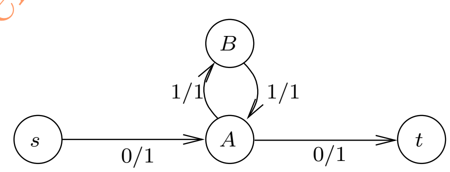
Since $\mathcal{V}(f) < \mathcal{V}(f^*)$, f is not a maximum flow.

b. This statement is false. Below is a counterexample:



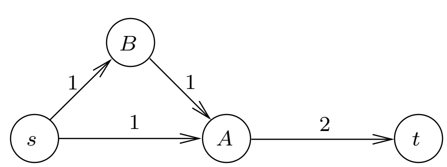
The edge (s, A) is saturated but the only cuts it crosses are $(\{s\}, \{A, B, t\})$ and $(\{s, B\}, \{A, t\})$. These have capacity 2, so neither is a minimum cut since $(\{s, A, B\}, \{t\})$ has capacity 1.

c. This statement is false. Below is a counterexample:



The value of the flow is zero, but some edges (forming a cycle) contain non-zero traffic.

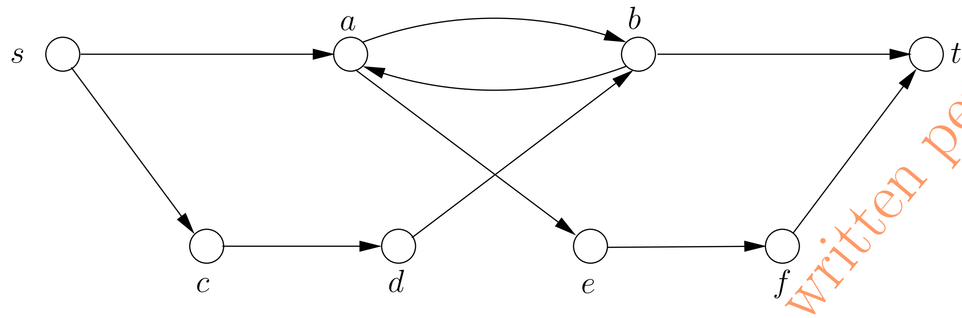
d. This statement is false. Below is a counterexample:



$(\{s\}, \{A, B, t\})$ is a minimum cut: it has capacity 2, the same as the value of the flow that saturates every edge. However, if we increase the capacity of every edge by 1, $(\{s\}, \{A, B, t\})$ is no longer a minimum cut: it has capacity 4, while $(\{s, A, B\}, \{t\})$ has capacity 3.

Answer to Question 2. a. Consider the flow network shown below, and assume that all capacities are 1.

In the first augmentation the algorithm must choose to augment along the path s, a, b, t, and in the second augmentation it is obliged to use the remaining augmenting path s, c, d, b, a, e, f, t. The resulting maximum flow has a cycle (a, b, a).



b. Given a flow f of flow network $\mathcal{F} = (G, s, t, c)$, where $G = (V, E)$, let $\mathcal{U}(f)$ be the number of edges used by f, i.e., $\mathcal{U}(f) = |\{e \in E : \ f(e) > 0\}|$.

Let f be any flow in $\mathcal{F}$, and let $v = \mathcal{V}(f)$. Let f' be a flow with value v that uses a minimum number of edges; that is, $\mathcal{V}(f') = v$ and $\mathcal{U}(f') \leq \mathcal{U}(f'')$ for every flow f'' such that $\mathcal{V}(f'') = v$. (Flow f' is well-defined because, although the number of flows with value v may be infinite (since v is not necessarily an integer), the number of edges they use is a non-negative integer.) We claim that f' is acyclic, which implies the result we want.

Suppose, for contradiction, that f' is not acyclic. Thus there is a cycle $u_1, u_2, \ldots, u_k$, such that $u_k = u_1$ and $f'(u_i, u_{i+1}) > 0$ for each i, $1 \leq i < k$. Without loss of generality, we can assume that this cycle is simple. (If not, it contains a simple cycle, obtained by following the nodes of this cycle until a node is repeated; we can therefore apply our argument to that simple cycle.) Let b be the minimum flow under f' among all edges in the cycle; i.e., $b = \min\{f'(u_i, u_{i+1}) : 1 \leq i < k\}$. Note that $b > 0$. Now remove b units of flow from each edge on the cycle. That is, define a function f'' on the edges as follows:

$$f''(e) = \begin{cases} f'(e) - b, & \text{if } e = (u_i, u_{i+1}), \text{ for some } i,\ 1 \leq i < k \\ f'(e), & \text{otherwise} \end{cases}$$

The function f'' satisfies the capacity constraint: This is obvious for edges not on the cycle $u_1, u_2, \ldots, u_k$, since the flow has not changed for them. For every edge e on the cycle, this is because $f''(e) = f'(e) - b < f(e) \leq c(e)$ (here we use the fact that $b > 0$); and $f''(e) = f(e) - b \geq f(e) - f(e) = 0$ (here we use the fact that $b \leq f(e)$). The function f'' also satisfies the conservation constraint: For every node u that is not on the cycle, the flows in and out of u do not change in f'' relative to f'; and, since the cycle is simple, for every node u that is on the cycle the flows into u decrease by b and the flows out of u also decrease by b, so the flows into u equal the flows out of u under f'', as under f'. Since f'' satisfies the capacity and conservation constraints, it is a flow.

The node s is not on the cycle because s has no incoming edges. Therefore $\sum_{e \in out(s)} f''(e) = \sum_{e \in out(s)} f'(e) = v$. Furthermore, $\mathcal{U}(f'') < \mathcal{U}(f')$ because every edge on the cycle that carries flow b under f' (of which there is at least one) has zero flow under f''. So f'' is a flow with value v and uses fewer edges than f', contradicting the definition of f'. Therefore, f' is an acyclic flow of value v.

Answer to Question 3.

a. Obviously, increasing the capacity of one edge cannot decrease the value of the maximum flow. Increasing the capacity of one edge by one unit can increase the capacity of the minimum cut by at most one, and therefore (by the max-flow-min-cut theorem) it can increase the value of the maximum flow by at most one.

Since all capacities are integers, by the integrality theorem the maximum flow has integer value. Therefore increasing the capacity of one edge by one unit either leaves the value of the maximum flow unchanged (if it does not increase the capacity of the minimum cut) or increases the value of the maximum flow by one (if it increases the capacity of the minimum cut).

The argument for decreasing the capacity of an edge by one unit is symmetric (replace "increase" by "decrease" and vice versa in the above).

b. The algorithm is as follows:

IncreasedCapacityMaxFlow($\mathcal{F}, f, e$)
1 construct the residual graph G_f of flow network $\mathcal{F}^+$ with respect to f
2 **if** G_f has no $s \to t$ path **then return** f
3 **else**
4   augment f along an $s \to t$ path of G_f resulting in flow f'
5   **return** f'

Correctness: We first note that f is a flow in $\mathcal{F}^+$, (though possibly not a maximum one): it satisfies both the capacity and conservation constraints. Thus line 1 makes sense: we can construct the residual graph G_f of $\mathcal{F}^+$ with respect to f. Furthermore, by the correctness of the Ford-Fulkerson algorithm, if G_f has no $s \to t$ path, f is a maximum flow of $\mathcal{F}^+$, justifying line 2. It remains to show that if G_f does have an $s \to t$ path, then a single augmenting path along that path is sufficient to produce a maximum flow in $\mathcal{F}^+$. This is because this augmentation will result in a flow f' such that $\mathcal{V}(f') \geq \mathcal{V}(f) + 1$ (because the capacities are integers and the flow f is integral). By part (a) the maximum flow in $\mathcal{F}^+$ has value at most $\mathcal{V}(f) + 1$. So, $\mathcal{V}(f') = \mathcal{V}(f) + 1$ and f' is a maximum flow in $\mathcal{F}^+$, justifying line 5.

Running time: Constructing the residual graph in line 1 takes $O(n + m)$ time, finding an $s \to t$ path (of one exists) in line 2 using BFS or DFS takes $O(n + m)$ time, and augmenting f in line 4 takes $O(n)$ time. So the algorithm's running time is $O(n + m)$.

c. At a high level (with details to be explained later), the algorithm is as follows:

ReducedCapacityMaxFlow($\mathcal{F}, f, e$)
1 **if** $f(e) < c(e)$ **then return** f
2 **else**
3   **if** there is an $s \to t$ path P that contains e and all its edges carry positive flow in f **then**
4   let f' be the function on edges s.t. $f'(e') = f(e') - 1$ if e' is on P, and $f'(e') = f(e')$ if e' is not on P
5   construct the residual graph $G_{f'}$ of flow network $\mathcal{F}'$ with respect to f'
6   **if** $G_{f'}$ has no $s \to t$ path **then return** f'
7   **else**
8   augment f' along an $s \to t$ path of $G_{f'}$ resulting in flow f''
9   **return** f''
10   **else**
11   find a cycle C of edges carrying positive flow that contains e
12   let f' be the function on edges s.t. $f'(e') = f(e') - 1$ if e' is on C, and $f'(e') = f(e')$ if e' is not on C
13   **return** f'

We must explain the following points:

(a) How to implement line 3 (i.e., determining if an $s \to t$ path P that contains e exists, and if so, finding such a path).

(b) Why the function f' defined in line 4 is a flow (i.e., that it satisfies the capacity and conservation constraints). This is necessary so that it makes sense to construct the residual graph in line 5, and augment f' to obtain the new flow f'' that is returned in line 9.

(c) How to implement line 11 (i.e., how to find a cycle C that contains e and consists of edges that carry positive flow, assuming that the algorithm reaches line 11).

(d) Why the function f' defined in line 12 is a flow. This is necessary so that the constructing the residual graph in line 5, and augmenting f' to obtain a new flow f'' make sense.

- For (a), let $e = (u, v)$. We define a graph G^- by deleting from G all edges that do not carry flow in f. This can be done in $O(n + m)$ time, where n is the number of nodes and m is the number of edges in G. We then use BFS or DFS on G^- to determine if there is an $s \to u$ path P_1 and a $v \to t$ path P_2. If so, P consists of the edges on P_1, followed by e, followed by the edges in P_2. Otherwise there is no $s \to t$ path that contains e. Since BFS and DFS both take $O(n + m)$ time, (a) can be done in $O(n + m)$ time.
- For (b), since f is integral and all edges on P carry positive flow, each edge on P carries at least flow of 1. Thus, reducing the flow by one along these edges satisfies the lower bound of the capacity constraint. Since f is a flow, the upper bound of the capacity constraint is obviously satisfied. So f' satisfies the capacity constraint. It also satisfies the conservation constraint because the only nodes involved in edges whose flow has changed are those on P; for every node on P other than s and t, the in-flow was reduced by 1 and the out-flow was reduced by 1. Since f satisfied the conservation constraint, so does f'. Note that defining f' takes $O(m)$ time (constant amount per edge).
- For (c), note that we reach line 11 only if $f(e) > 0$ (see line 1) and there is no $s \to t$ path that contains e consisting of edges that carry positive flow (see line 3). Then by the conservation property of f, e must lie on a cycle C of G^- (i.e., the graph G with edges carrying no flow removed), all nodes of which are not reachable from s and from which t cannot be reached in G^-. Thus, by applying a BFS on G^- starting from the head of e we will eventually reach the tail of e, and so we will have discovered a cycle C containing e consisting of edges that carry positive flow. Note that this step takes $O(n + m)$ time.
- (d) is similar to (b).

Running time: From the above discussion it is clear that the running time of the algorithm is $O(n+m)$.

Correctness: If $f(e) < c(e)$ then f is a valid flow in $\mathcal{F}^-$, and so by part (a), it is a maximum flow in $\mathcal{F}'$; this justifies the flow returned in line 1. So, suppose $f(e) = c(e)$, i.e., f saturates e. There are two cases:

CASE 1: There is an $s \to t$ path P that contains e and all edges of P carry positive flow. In this case, f is no longer a valid flow, as its traffic on e exceeds the new capacity of e. This does not necessarily mean that the maximum flow in $\mathcal{F}'$ has lower value: It may still be possible to carry the same amount of traffic from s to t through an alternate flow. So, we first find a valid flow f' in $\mathcal{F}'$ by reducing the traffic on each edge of P by one, and leaving the traffic of all other edges as before. It is clear that f' satisfies the capacity constraint (since the traffic through e has been reduced and so it satisfies the new capacity constraint); and the conservation constraint (since the traffic in and out of each node on P other than s and t has been reduced by one, and the traffic in and out of nodes not on P was not changed). Furthermore, $\mathcal{V}(f') = \mathcal{V}(f) - 1$, since the traffic of only one edge out of s was reduced by one. If the residual graph $G_{f'}$ has no $s \to t$ path then, by the correctness of the Ford-Fulkerson algorithm, f' is a maximum flow in $\mathcal{F}'$; this justifies the flow returned in line 6. (This is the case where the reduction of the capacity of e reduces the value of the maximum flow.) Otherwise, by augmenting f' along this path with we obtain a flow f'' such that $\mathcal{V}(f'') \geq \mathcal{V}(f') + 1 = \mathcal{V}(f)$. By part (a), $\mathcal{V}(f'') \leq \mathcal{V}(f)$, But since a flow in $\mathcal{F}'$ cannot have value greater so f'' is a maximum flow of $\mathcal{F}'$; this justifies the flow returned in line 9. (This is the case where, although e is saturated by f, the reduction of the capacity of e does not affect the value of the maximum flow, as there is an alternative way through which we can channel $\mathcal{V}(f)$ units of traffic from s to t.)

CASE 2: There is no $s \to t$ path P that contains e all of whose edges carry positive traffic in f. In this case, the traffic on e is sloshing around in a cycle C of nodes that are not reachable from s and from which t is not reachable; this is useless traffic that contributes nothing to the value of the maximum flow. Because f is integral, there is at least one unit of (useless) flow around every edge of this cycle, so if we reduce the traffic on each edge on C (including e) by one we get a flow f' that satisfies the new capacity contraint (since the traffic through e was reduced by one) as well as the conservation constraint (since the traffic into each node of C and out of each node on C was reduced by one), and it has the same value as f (since the edges out of s are not on this cycle and so their traffic was not reduced). Thus, f' is a maximum flow of $\mathcal{F}'$; this justifies the flow returned in line 13.