Solutions to Practice Problems 8

Solution 1:

(a) Many sequences of augmenting paths are possible; here is one, starting from the zero flow.

augmenting path	bottleneck
$s \to a \to c \to t$	2
$s \to a \to d \to t$	2
$s \to a \to b \to c \to t$	4
$s \to b \to d \to t$	3
total	11

After the fourth augmentation the residual network contains no s-t path (see part (b)), so the algorithm halts with $|f| = 11$. The resulting flow is shown in Fig. 1, where each edge is labeled $f(e)/c(e)$.

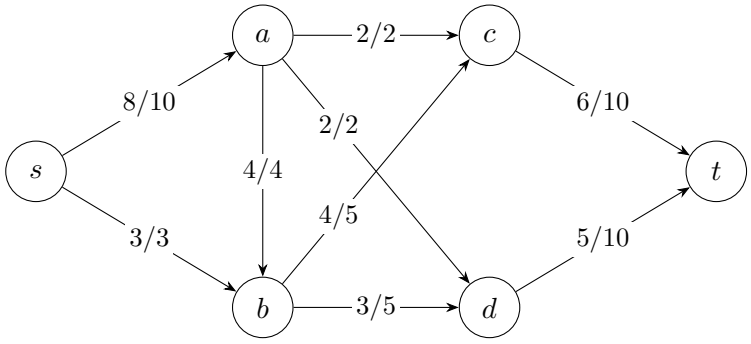


Figure 1: A maximum flow, of value $|f| = 11$.

As a check, conservation holds at each interior vertex: a receives 8 and sends $4 + 2 + 2 = 8$; b receives $3 + 4 = 7$ and sends $4 + 3 = 7$; c receives $2 + 4 = 6$ and sends 6; and d receives $2 + 3 = 5$ and sends 5. The sink receives $6 + 5 = 11 = |f|$.

(b) The residual network G_f is shown in Fig. 2. Each edge (u, v) of G contributes a forward residual edge of capacity $c(u, v) - f(u, v)$ when this is positive, and a backward residual edge (v, u) of capacity $f(u, v)$ when the flow is positive.

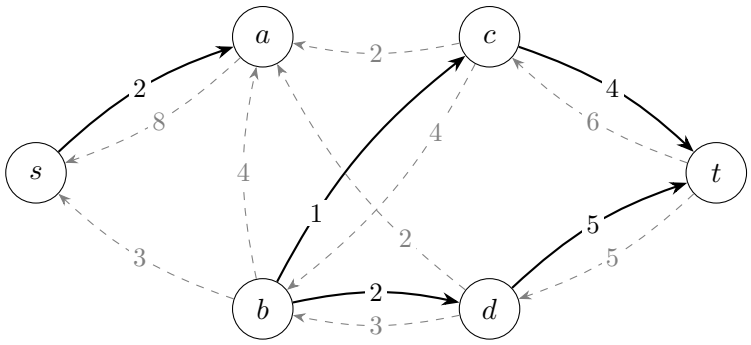


Figure 2: The residual network G_f. Solid edges are forward residual edges, dashed edges are backward ones.

Starting from s, the only residual edge leaving s is (s, a), of residual capacity 2. From a, every outgoing edge of G is saturated, so a has no forward residual edge at all, and the only residual edge leaving a is the backward edge (a, s). Hence

$$S \;=\; \{s, a\}, \qquad V \setminus S \;=\; \{b, c, d, t\}.$$

The capacity of this cut is

$$c(S, V \setminus S) \;=\; c(s, b) + c(a, b) + c(a, c) + c(a, d) \;=\; 3 + 4 + 2 + 2 \;=\; 11,$$

which equals $|f|$, as expected.

(c) The edges crossing forward, from S to $V \setminus S$, are

$$(s, b) : 3/3, \qquad (a, b) : 4/4, \qquad (a, c) : 2/2, \qquad (a, d) : 2/2,$$

and each is indeed saturated. There are *no* edges crossing backward, from $V \setminus S$ to S: the only edges entering a or s in G are (s, a), whose tail s lies in S as well. So the second condition holds vacuously here, and the total flow on backward-crossing edges is 0.

These two conditions force equality for the following reason. The value of a flow equals the net flow across any s-t cut,

$$|f| \;=\; \sum_{\substack{(u,v)\in E \\ u\in S,\, v\notin S}} f(u, v) \;-\; \sum_{\substack{(v,u)\in E \\ v\notin S,\, u\in S}} f(v, u).$$

Saturation of the forward edges turns the first sum into $c(S, V \setminus S)$, and the vanishing of the flow on backward edges makes the second sum 0. Hence $|f| = c(S, V \setminus S) = 11$. Since no flow can exceed the capacity of any cut, f is a maximum flow and $(S, V \setminus S)$ is a minimum cut.

(d) Let f be a flow admitting no augmenting path, and let S be the set of vertices reachable from s in G_f.

1) The pair $(S, V \setminus S)$ is an s-t cut. Certainly $s \in S$, by the empty path. And $t \notin S$: a directed path from s to t in G_f is precisely an augmenting path, and by hypothesis none exists.

2) Every forward-crossing edge is saturated. Let $(u, v) \in E$ with $u \in S$ and $v \notin S$, and suppose $f(u, v) < c(u, v)$. Then G_f contains the forward residual edge (u, v), with positive residual capacity $c(u, v) - f(u, v)$. Since u is reachable from s in G_f, so is v, contradicting $v \notin S$. Hence $f(u, v) = c(u, v)$.

3) Every backward-crossing edge carries no flow. Let $(v, u) \in E$ with $v \notin S$ and $u \in S$, and suppose $f(v, u) > 0$. Then G_f contains the backward residual edge (u, v), of positive residual capacity $f(v, u)$. Again u is reachable from s, hence so is v, contradicting $v \notin S$. Hence $f(v, u) = 0$.

Substituting these two facts into the identity for the net flow across a cut, exactly as in part (c),

$$|f| \;=\; \sum_{u \in S,\, v \notin S} f(u, v) \;-\; \sum_{v \notin S,\, u \in S} f(v, u) \;=\; \sum_{u \in S,\, v \notin S} c(u, v) \;-\; 0 \;=\; c(S, V \setminus S).$$

This one equation proves both claims at once. Recall that for *every* flow f' and *every* s-t cut (A, B) we have $|f'| \leq c(A, B)$, since the net flow across the cut is at most the total capacity of its forward edges. Now take any flow f': then $|f'| \leq c(S, V \setminus S) = |f|$, so f is maximum. And take any cut (A, B): then $c(A, B) \geq |f| = c(S, V \setminus S)$, so $c(S, V \setminus S)$ is minimum.

Solution 2:

(a) Take two sources s_1, s_2, one sink t_1, and one interior vertex v, with edges

$$(s_1, v) \text{ of capacity } 1, \qquad (s_2, v) \text{ of capacity } 1, \qquad (v, t_1) \text{ of capacity } 1.$$

There are two source-sink pairs. The maximum s_1-t_1 flow is 1, and the maximum s_2-t_1 flow is 1, so the proposal reports $1 + 1 = 2$. But in any flow the single edge (v, t_1) carries at most 1 unit, and all flow must pass through it, so the true maximum value is 1.

The proposal ignores the fact that the sources *compete* for shared capacity: each pairwise computation is free to use the whole of edge (v, t_1), whereas a single flow must divide that capacity among all sources at once.

(b) **The reduction.** Add a new *super-source* s^* with an edge $s^* \to s_i$ to every source, and a new *super-sink* t^* with an edge $t_j \to t^*$ from every sink, all of capacity ∞. Call the result G^*. Run any max-flow algorithm on G^* from s^* to t^*, then delete s^* and t^* together with their edges.

Think of s^* as a single reservoir feeding all the factories, and t^* as a single drain collecting from all the warehouses. The new edges have unlimited capacity, so they never restrict anything; they only collect the many sources and sinks into one of each.

Correctness. We show that every flow in either network corresponds to a flow *of the same value* in the other. Then neither maximum can exceed the other, so they are equal.

- *From G to G^*.* Given a flow in G, keep it as is, and on each new edge $s^* \to s_i$ send exactly the amount that s_i sends out. Each source now passes along what it receives, as conservation requires; the same trick on the edges $t_j \to t^*$ handles the sinks. Everything s^* sends goes out through the sources, so the value is unchanged.

- *From G^* to G.* Given a flow in G^*, simply delete s^*, t^* and their edges. The remaining vertices still satisfy conservation, since we removed no edge touching them. Whatever s^* sent into s_i now leaves s_i instead, so the total flow out of the sources equals the old value.

The second direction is the useful one in practice: it tells us that a maximum flow in G^*, with the extra vertices deleted, *is* a maximum flow in G.