Part II
Flow Networks

1 Definitions and basic properties (p. 20)

1.1 Flow networks (p. 21)

1. Assume that the vertices $v \in V$ also have a capacity $c(v) \geq 0$ and assume that we demand that the amount of flow entering v is limited by $c(v)$. Show that we can reduce this more general flow network problem to a standard flow network problem without vertex capacities.

Solution: Every node $v \in V$ is replaced by 2 nodes, v_{in} and v_{out}, or $V' = \{v_{in}, v_{out} | v \in V\}$. Edges are redirected, such that $E' = \{(u_{out}, v_{in}) | (u, v) \in E\} \cup \{(v_{in}, v_{out}) | v \in V\}$.

We set $c'(u_{out}, v_{in}) = c(u, v)$ and $c'(v_{in}, v_{out}) = c(v)$.

2. Consider a flow network in which each vertex v has a (positive or negative) demand $d(v)$. Reduce the following problem to a max flow problem: find a circulation f such that $f(u, v) \leq c(u, v), f(u, v) = -f(v, u)$ and $\sum_{u \in V} f(u, v) = d(v)$ for all v. ☆

Solution: *Definition:* A circulation f satisfies the following conditions:

(a) Capacity constraint: $f(u, v) \leq c(u, v)$ for all $u, v \in V$.

(b) Skew symmetry: $f(u, v) = -f(v, u)$ for all $u, v \in V$.

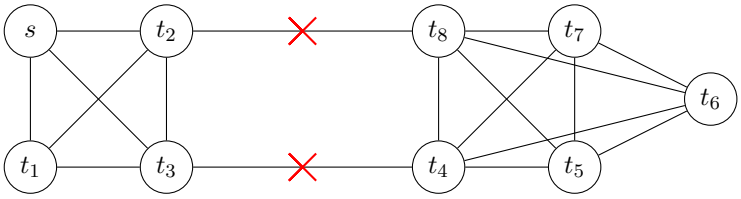
(c) Current conservation: $\sum_{u \in V} f(u, v) = d(v)$ for all $v \in V$.

2 The Ford-Fulkerson method (1956) (p. 22)

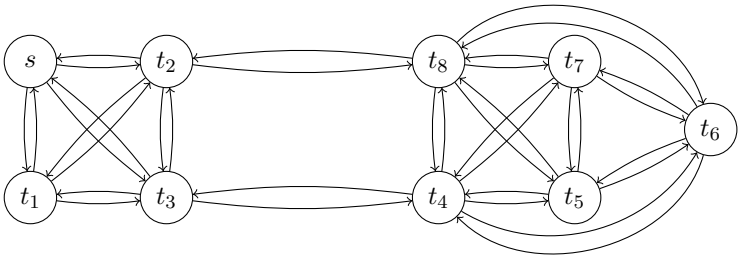
2.1 Flow network problems (p. 25)

1. The edge connectivity number $\lambda(G)$ of a graph is equal to the number of edges one needs to remove to disconnect a graph. Given an undirected graph $G = (V, E)$, show that $\lambda(G)$ can be determined by running at most $|V|$ maximum-flow algorithms on a flow network with at most $O(|V|)$ vertices and $O(|E|)$ edges.

Solution: The edge connectivity number $\lambda(G)$ of the graph below is equal to 2. The graph is separated whenever the 2 edges crossed out in red are removed.



We replace each undirected edge $(u, v) \in E$ by 2 directed edges (u, v) and (v, u) in E'. We assign a capacity $c(u, v) = 1$ to each edge (u, v) in E'. (We now clearly have $\lambda(G) = \min_{u,v \in V, u \neq v} |f^*_{u,v}|$, where $f^*_{u,v}$ denotes a max flow from u to v in $G' = (V, E')$.) We pick an arbitrary node s as source for the max-flow algorithms. Run $|V|-1$ max-flow algorithm with source s, each time with another node $t_i \in V \setminus \{s\}$, each resulting in a maximum flow value $\left|f^*_{s,t_i}\right|$ for source/sink pair (s, t_i).



We now claim that the edge connectivity number can then be computed as $\lambda(G) = \min_{i=1,\ldots,|V|-1} \left|f^*_{s,t_i}\right|$.
Let E^* be a minimal set of edges that disconnects G (thus $|E^*| = \lambda(G)$). Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two subgraphs that we get from G by removing the edges in E^*. Wlog we assume $s \in V_1$. Then, there exists an k such that $t_k \in V_2$. Note, that as $s \neq t_k$ the algorithm above finds f^*_{s,t_k}. Obviously, $|f^*_{s,t_k}| \leq \lambda(G)$ (as the flow cannot be greater than $\lambda(G)$). We now get

$$\min_{i=1,\ldots,|V|-1} \left|f^*_{s,t_i}\right| \leq \left|f^*_{s,t_k}\right| \leq \lambda(G) = \min_{u,v \in V, u \neq v} |f^*_{u,v}| \leq \min_{i=1,\ldots,|V|-1} \left|f^*_{s,t_i}\right|, \tag{1}$$

thus proving the claim.

Extra: Note that we thus have $\left|f^*_{s,t_k}\right| = \lambda(G)$. This implies that (V_1, V_2) is a min cut of f^*_{s,t_k}. We can in fact prove this directly before (1) as follows: Suppose $\left|f^*_{s,t_k}\right| < \lambda(G)$. Let (S, T) be a min cut of f^*_{s,t_k} and let $\tilde{E}$ be the set of edges in $S \times T$. As all capacities are equal to 1, we have $\left|f^*_{s,t_k}\right| = \left|\tilde{E}\right|$, which contradicts the definition of E^* as $\left|\tilde{E}\right| < |E^*|$ and removing undirected edges in G that correspond to the directed edges in the set $\tilde{E}$ disconnects the graph.

2. The vertex connectivity number $\kappa(G)$ of a graph is equal to the number of vertices one needs to remove to disconnect a graph. Given an undirected graph $G = (V, E)$, show that $\kappa(G)$ can be determined by running at most $|V|(\kappa(G) + 1)$ maximum-flow

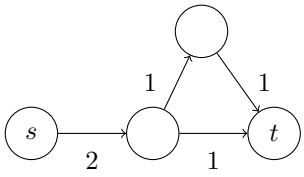
algorithms on a flow network with at most $O(|V|)$ vertices and $O(|E|)$ edges.

Solution: Hint: Split every vertex v in v_{in} and v_{out} and proceed similarly to the previous exercise (the capacity function c is not always 1 though).

3. True or False? ☆

   (a) Let (S, T) be a minimum cut, then (S, T) is still a minimum cut if we increase all the edge capacities by one.

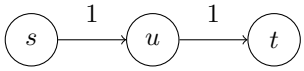
   Solution: This is false. Consider the following graph

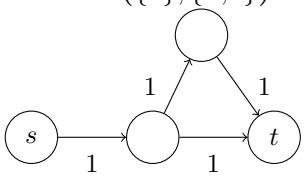
   $(V \setminus \{t\}, \{t\})$ is a minimum cut. If we increase all edge capacities by 1 then it stops being a minimum cut as then $|f| = 3 < c(V \setminus \{t\}, \{t\}) = 4$.

   (b) There is a unique minimum cut if and only if there is a unique maximum flow.

   Solution: Both directions are false. Consider the graph

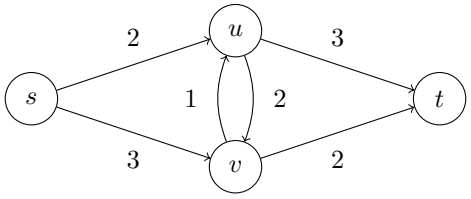
   Clearly the max-flow is unique with $|f| = 1$, but the min-cut is not unique: we can take $(\{s\}, \{u, t\})$ or $(\{s, u\}, \{t\})$ as min-cuts. Now consider the graph

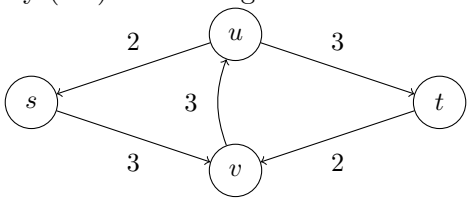
   Clearly, the max-flow f is not unique. However, there is only one min-cut $(\{s\}, V \setminus \{s\})$, as for the two other cuts we have that their capacity is 2, while $|f| = 1$.

   (c) Suppose we always pick a simple path p in G_f such that $c_f(p)$ is maximized, then $c_f(p)$ cannot increase while using the method of Ford and Fulkerson.

   Solution: This is false. Consider the following flow network



We can first choose the following augmenting path s, u, v, t of maximum capacity (= 2) and then get the residual flow network



where we have an augmenting path of residual capacity 3: s, v, u, t.

4. Let (S_1, T_1) and (S_2, T_2) be minimum cuts of G. Argue that $(S_1 \cap S_2, V \setminus (S_1 \cap S_2))$ and $(S_1 \cup S_2, V \setminus (S_1 \cup S_2))$ are also minimum cuts. ☆[2]

Solution: We prove the second case, the first one follows analogously. By the Max-flow Min-cut theorem, a cut (S, T) is minimal if and only if for every edge (u, v) with $u \in S, v \in T$ we have $f(u, v) = c(u, v)$. In our case we have $f(u, v) = c(u, v)$ for every $(u, v) \in S_i \times T_i$ for $i = 1, 2$. So, we must have $f(u, v) = c(u, v)$ for every $(u, v) \in S_i \times (T_1 \cap T_2)$ for $i = 1, 2$. So, we must have $f(u, v) = c(u, v)$ for every $(u, v) \in (S_1 \cup S_2) \times (T_1 \cap T_2)$. The proof is finished by noting that $T_1 \cap T_2 = (V \setminus S_1) \cap (V \setminus S_2) = V \setminus (S_1 \cup S_2)$ (you can easily see this by drawing a Venn diagram).

5. Given the minimum cut (S, T) where S is the set of vertices reachable from s in G_f. How can we check in $O(|V| + |E|)$ time whether this cut is unique? [Hint: use the previous exercise.] ☆[2]

Solution: Hint: Let T' be the set of vertices from which t can be reached in the residual network. What if $T \neq T'$? What if $T = T'$?

6. Indicate how we can find an augmenting path p with $c_f(p)$ maximized in a residual network $G_f = (V, E_f)$ in $O((|V| + |E_f|) \log |E_f|)$ time. Start by sorting the edges in E_f by weight. ☆

Solution: Hint: an algorithm runs in a logarithmic time complexity if after every iteration the amount of remaining work is halved.

7. Find an efficient method to determine a minimum cut (S,T) such that the number of edges between S and T is minimized given that all the edge capacities are integers. [Hint: augment all edge capacities by a constant.] ☆

Solution: We want to transform the capacity function $c : E \to \mathbb{Z}$ into a new capacity funtion $c' : E \to \mathbb{Z}$ such that the following two are true:

- First and foremost, a minimal cut on $G' = (V, E, c')$ is a minimal cut on $G = (V, E, c)$.
- Max flow on G' corresponds to the cut on G with minimal number of edges.

We first note that for the capacity function ac with $a \in \mathbb{N} \setminus \{0\}$ we get the same minimal cuts as for c. We now claim that setting $c'(e) = ac(e)+1$ for $e \in E$ (and zero otherwise), with a sufficiently large, gives a desired capacity function. Let (S,T) be a cut in G' we then have

$$c'(S,T) = \sum_{e \in S \times T} c'(e) = a \sum_{e \in S \times T} c(e) + n,$$

where n is the number of edges in the cut. As $n \leq |E|$ and $c(e)$ is an integer we can choose any $a > |E|$. A minimal cut (S,T) on G' is then also a min-cut on G with minimal number of edges.

Extra: Same exercise but where capacities are not necessarily integers.