252 COMBINATORIAL OPTIMIZATION

HOMEWORK 3

These exercises are at a mixture of levels. There may be typos or mistakes; please let me know as soon as you find one. I'm happy to discuss your solution attempts in my office hours (Monday 12pm).

Hand in Questions 1,2,4,8,9,10 to the box outside the undergraduate office by Thursday, 13th February, at 2pm. You are strongly encouraged to do the other questions at this time as well, though you should not hand them in. You also are encouraged to work on these problems in groups, though your final write-up should be your own.

(1) Find an augmenting path for the flow shown in Figure 1.
(2) Find a maximal flow for the network of Figure 2.
(3) In the Ford-Fulkerson algorithm we constructed a cut (S, T) at the end of the algorithm by setting S to be all vertices v for which there is an augmenting path from s to v. When the algorithm terminates let T' be the set of all u for which there is an augmenting path from u to t, and let $S' = V \setminus T'$. Show that (S', T') is also a minimum cut for the network N. Do we always have $S = S'$ and $T = T'$?
(4) Construct networks with integral capacities having:

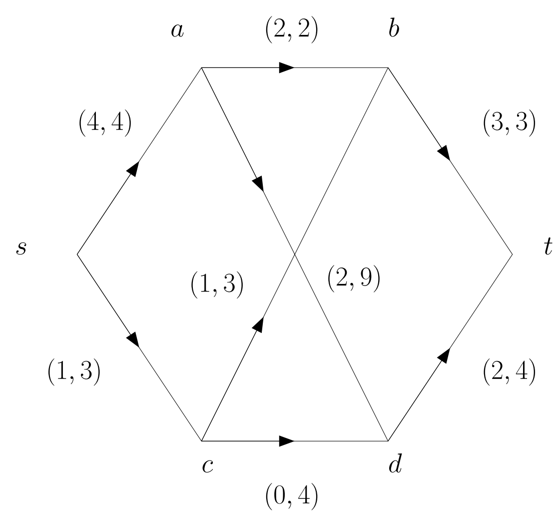


FIGURE 1.

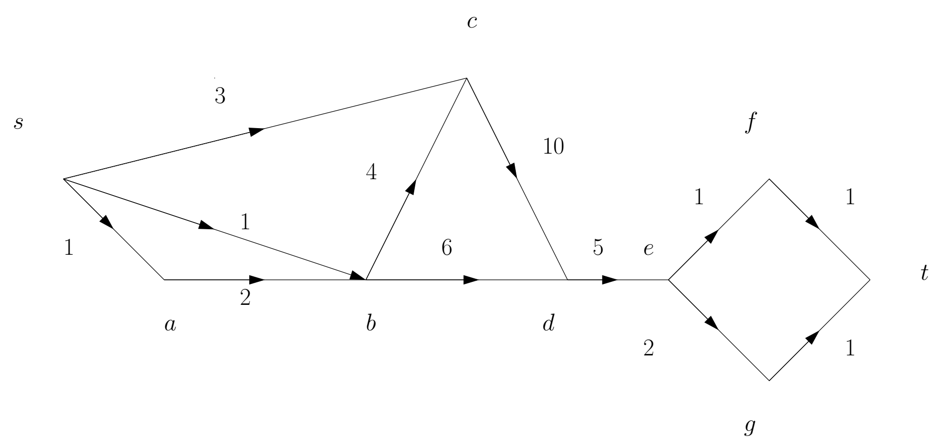


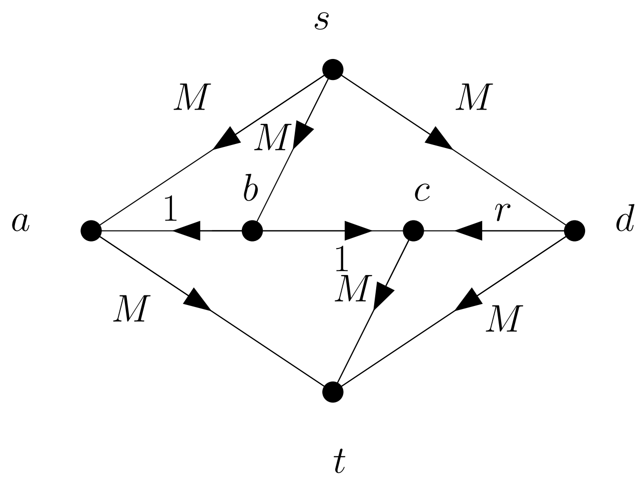
FIGURE 2.

(a) many maximal flows and many minimum cuts,
(b) many maximal flows, and a unique minimum cut,
(c) a unique maximal flow, and many minimum cuts, and
(d) a unique maximal flow, and a unique minimum cut.

(5) One way to find an augmenting path is to create the *residual graph* $G(f)$, which is a weighted directed graph whose vertices are the same as N, and for which there is an edge (v, u) for every edge $(u, v) \in E(N)$. For $(u, v) \in E(N)$ we give $(u, v) \in E(G(f))$ the weight $c(u, v) - f(u, v)$, and (v, u) the weight $f(u, v)$. An augmenting path is then a path from s to t in $G(f)$ using only edges with positive weight. The existence of this can be computed using any of our shortest path algorithms (eg Dijkstra). Use this idea to bound the running time of the augmenting paths algorithm in terms of $|V|$, $|E|$, and $\max_{e \in E} c(e)$. (Warning: if you google for this question, you will see answers that do not use $\max_{e \in E} c(e)$. This requires a careful implementation - most people will learn more by thinking about this question on your own than by reading a more complicated answer.)

(6) A graph $G = (V, E)$ is *k-edge-connected* if it connected whenever any $k-1$ edges are removed. Thus a 1-edge-connected graph is a connected graph, and a 2-edge-connected graph is a connected graph for which there is no edge whose removal disconnects the graph. Given two vertices $x, y \in V(G)$, a collection of paths $P_1, \ldots, P_k$ from x to y are *edge-disjoint* if there is no edge occuring in P_i and P_j for some $i \neq j$.

FIGURE 3.



(a) Draw a 2-edge-connected graph, and a 3-edge-connected graph.
(b) *Menger's theorem* states if G is k edge connected then for all $x, y \in V(G)$ there are k edge disjoint paths from x to y. Prove Menger's theorem using the max-flow/min-cut theorem. Hint: Orient the edges so they go out of x, into y, and both directions if they are not adjacent to x or y (ie add (u, v) and (v, u) if $u, v \neq x, y$). Set $s = x$ and $t = y$. Make the cost $c(e) = 1$ for all edges e. Prove that an flow on this new network will pick out edge disjoint paths from x to y, and use this to deduce Menger's theorem.

(7) Let N be the network shown in Figure 3, where $r = \frac{\sqrt{5}-1}{2}$, and M is an integer at least 2. In this exercise you will show that there is a choice of augmenting paths for which the augmenting paths algorithm does not terminate, and the flow value does not converge to the maximal flow. The key fact about r that you will use is that $r^2 = 1 - r$.
(a) Show that the minimum capacity of a cut on N is $2M + 1$.
(b) Give an example of a flow of value $2M + 1$.
(c) Consider running the augmenting paths algorithm as follows. First augment the path s, b, c, t, then the path $p_1 = s, d, c, b, a, t$, then the path $p_2 = s, b, c, d, t$, then the path p_1 again, then the path $p_3 = s, a, b, c, t$. Now repeat the sequence of augmenting paths p_1, p_2, p_1, p_3 an arbitrary number of times. Show that this is possible; ie show that there

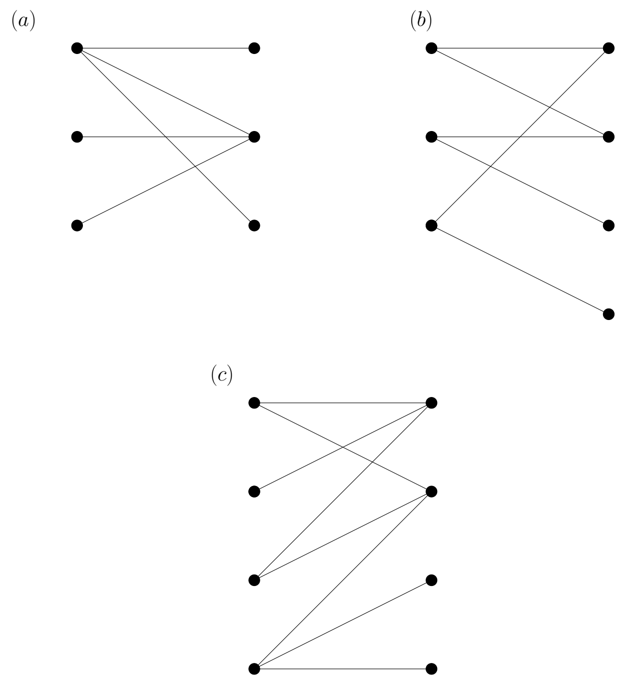


FIGURE 4.

is always spare capacity on the forward edges and positive flow on the negative edges.

(d) Compute the value of the flow after k iterations of the paths p_1, p_2, p_1, p_3. Show that this converges to $3 + 2r$.

This shows that this choice of running the algorithm does not terminate, and does not converge to the correct value. Note that in the proof of the max-flow/min-cut theorem we use the assumption that capacities are integers. This is in fact not necessary, but we will only see this in the last week of term.

(8) For each of the bipartite graphs in Figure 4 find a maximal matching. If it is not complete, illustrate a subset $X \subset A$ violating the conditions of Hall's theorem.

(9) Let $A = \{a, b, c, d\}$ and $B = \{1, 2, 3, 4\}$ have preferences shown in the following table:

1:	a	b	c	d	a:	1	2	3	4
2:	a	b	c	d	b:	3	2	4	1
3:	d	c	b	a	c:	2	4	1	3
4:	d	a	b	c	d:	1	2	3	4

Find a stable matching between A and B.

(10) In class we discussed an algorithm to find a stable matching that involved companies ($b \in B$) offering jobs to applicants ($a \in A$).

We could also run the algorithm so that the roles of A and B are reversed (applicants apply for jobs). For $a \in A$ let $M(a) = \{b \in B : \text{ there is a stable matching containing } (a, b)\}$. Show that the B-offers algorithm matches a with $\min_{\prec_a} M(a)$, while the A-applies algorithm matches a with $\max_{\prec_a} M(a)$. Verify this on the data of the previous question.