Homework 6 Solution

Due: 3pm, December 7.

Instruction. There are 8 problems in this homework, worth 100 points in total. For all algorithm design problems, you need to provide proofs on the correctness and running time of the algorithms unless otherwise instructed.

Problem 1 [10 pts]

Figure 1 below shows a flow network on which an s-t flow has been computed. The capacity of each edge appears as a label next to the edge, and the numbers in boxes give the amount of flow sent on each edge. (Edges without boxed numbersspecifically, the four edges of capacity 3have no flow being sent on them.)

(a) What is the value of this flow? Is this a maximum (s, t) flow in this graph?

(b) Find a minimum $s - t$ cut in the flow network pictured in Figure 1, and also say what its capacity is.

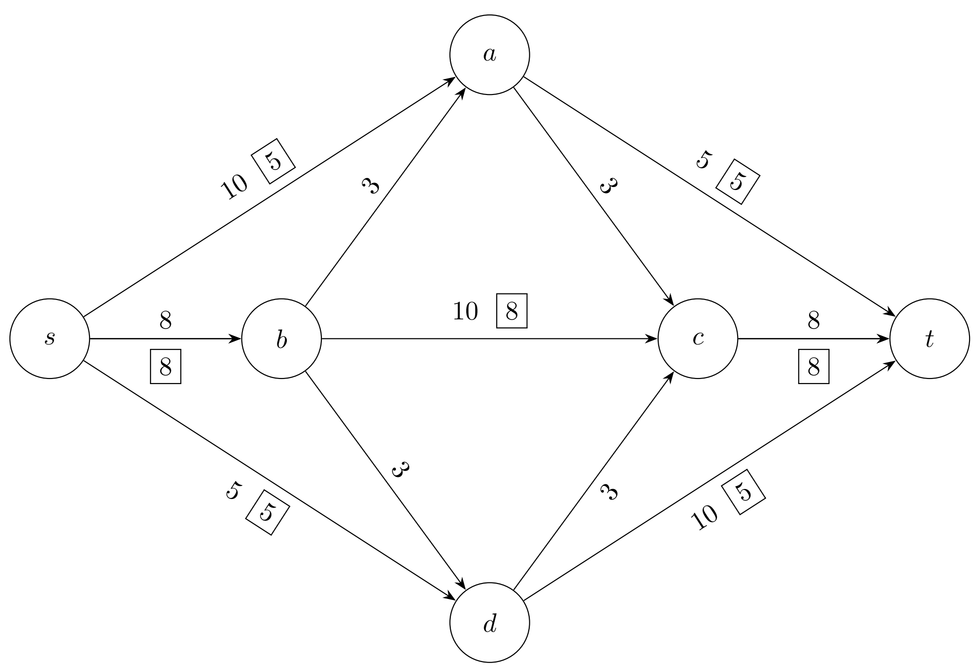


Figure 1: What is the value of the depicted flow? Is it a maximum flow? What is the minimum cut?

Solution:

(a) The flow has value 18. It is not a maximum flow.

(b) The minimum cut is $(\{a\}, \{s, b, c, d, t\})$. Its capacity is 21.

Problem 2 [10 pts]

Figure 2 below shows a flow network on which an s-t flow has been computed. The capacity of each edge appears as a label next to the edge, and the numbers in boxes give the amount of flow sent on each edge. (Edges without boxed numbers have no flow being sent on them.)

(a) What is the value of this flow? Is this a maximum (s, t) flow in this graph?

(b) Find a minimum $s - t$ cut in the flow network pictured in Figure 2, and also say what its capacity is.

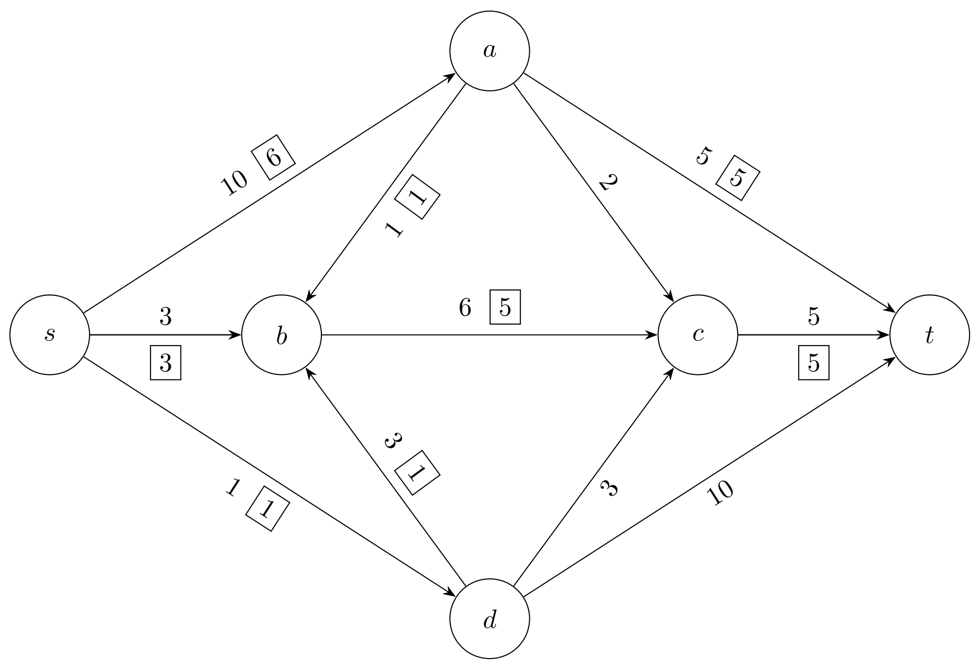


Figure 2: What is the value of the depicted flow? Is it a maximum flow? What is the minimum cut?

Solution:

(a) The value of this flow is 10. It is not a maximum flow.

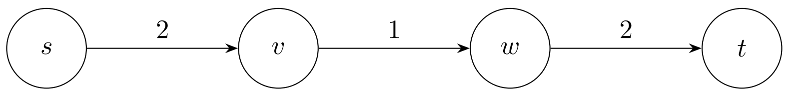
(b) The minimum cut is $(\{s, a, b, c\}, \{d, t\})$. Its capacity is 11.

Problem 3 [10 pts]

Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

> *Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e. If f is a maximum s-t flow in G, then f saturates every edge out of s with flow (i.e., for all edges e out of s, we have $f(e) = c_e$).*

Solution: This is false. Consider the following graph with nodes s, v, w, t, edges (s, v), (v, w), (w, t), capacities of 2 on (s, v) and (w, t), and a capacity of 1 on (v, w). Then the maximum flow has value 1, and does not saturate the edge out of s.

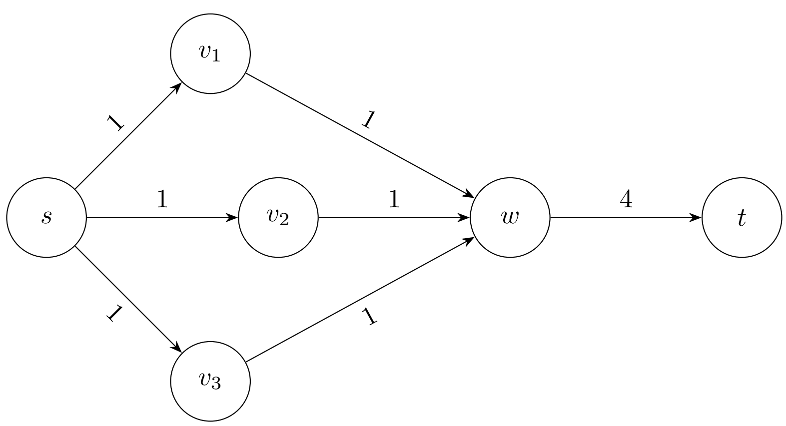


Problem 4 [10 pts]

Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

> *Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e; and let (A, B) be a minimum s-t cut with respect to these capacities $\{c_e : e \in E\}$. Now suppose we add 1 to every capacity; then (A, B) is still a minimum s-t cut with respect to these new capacities $\{1 + c_e : e \in E\}$.*

Solution: This is false. Consider the following graph with nodes s, v_1, v_2, v_3, w, t, edges (s, v_i) and (v_i, w) for each i, and an edge (w, t). There is a capacity of 4 on edge (w, t), and a capacity of 1 on all other edges. Then setting $A = \{s\}$ and $B = V - A$ gives a minimum cut, with capacity 3. But if we add one to every edge then this cut has capacity 6, more than the capacity of 5 on the cut with $B = \{t\}$ and $A = V - B$.



Problem 5 [15 pts]

Consider a set of mobile computing clients in a certain town who each need to be connected to one of several possible base stations. Well suppose there are n clients, with the position of each client specified by its (x, y) coordinates in the plane. There are also k base stations; the position of each of these is specified by (x, y) coordinates as well. For each client, we wish to connect it to exactly one of the base stations. Our choice of connections is constrained in the following ways.

There is a range parameter r a client can only be connected to a base station that is within distance r. There is also a load parameter L no more than L clients can be connected to any single base station. Your goal is to design a polynomial-time algorithm for the following problem. Given the positions of a set of clients and a set of base stations, as well as the range and load parameters, decide whether every client can be connected simultaneously to a base station, subject to the range and load conditions in the previous paragraph.

Solution: This is an application of max flow problem. Have one vertex for every client and one vertex for every base station. Add two extra vertices s and t. Now, add an edge from s to every client vertex and capacity of this edge is 1. Similarly add edges from every base station vertex to t of capacity L. Finally, add an edge from a client vertex to a base station vertex if the client can be assigned to this base station. The capacity is infinity (or 1, does not matter).

Problem 6 [15 pts]

Consider the following problem. You are given a flow network with unit- capacity edges: It consists of a directed graph $G = (V, E)$, a source $s \in V$, and a sink $t \in V$; and $c_e = 1$ for every $e \in E$. You are also given a parameter k. The goal is to delete k edges so as to reduce the maximum s-t flow in G by as much as possible. In other words, you should find a set of edges $F \subseteq E$ so that $|F| = k$ and the maximum s-t flow in $G = (V, E - F)$ is as small as possible subject to this. Give a polynomial-time algorithm to solve this problem.

Solution: If the minimum $s - t$ cut has size $\leq k$, then we can reduce the flow to 0. Otherwise, let $f > k$ be the value of the maximum $s - t$ flow. We identify a minimum $s - t$ cut (A, B), and delete k of the edges out of A. The resulting subgraph has a maximum flow value of at most $f - k$.

But we claim that for any set of edges F of size k, the subgraph $G' = (V, E - F)$ has an $s - t$ flow of value at least $f - k$. Indeed, consider any cut (A, B) of G'. There are at least f edges out of A in G, and at most k have been deleted, so there are at least $f - k$ edges out of A in G'. Thus, the minimum cut in G' has value at least $f - k$, and so there is a flow of at least this value.

Problem 7 [15 pts]

Some of your friends have recently graduated and started a small company, which they are currently running out of their parents garages in Santa Clara. Theyre in the process of porting all their software from an old system to a new, revved-up system; and theyre facing the following problem.

They have a collection of n software applications, $\{1, 2, \cdots, n\}$, running on their old system; and theyd like to port some of these to the new system. If they move application i to the new system, they expect a net (monetary) benefit of $b_i \geq 0$. The different software applications interact with one another; if applications i and j have extensive interaction, then the company will incur an expense if they move one of i or j to the new system but not both; lets denote this expense by $x_{ij} \geq 0$.

So, if the situation were really this simple, your friends would just port all n applications, achieving a total benefit of $\sum_i b_i$. Unfortunately, theres a problem. . . .

Due to small but fundamental incompatibilities between the two systems, theres no way to port application 1 to the new system; it will have to remain on the old system. Nevertheless, it might still pay off to port some of the other applications, accruing the associated benefit and incurring the expense of the interaction between applications on different systems.

So this is the question they pose to you: Which of the remaining applications, if any, should be moved? Give a polynomial-time algorithm to find a set $S \subseteq \{2, 3, \cdots, n\}$ for which the sum of the benefits minus the expenses of moving the applications in S to the new system is maximized.

Solution: This can be solved by the min-cut algorithm. We first create an undirected graph as follows. It has a vertex for every software application, call these $v_1, \cdots, v_n$, where v_i corresponds to application i. We also have a special vertex t. If applications i and j have an associated value x_{ij} , then we have two directed edges between v_i and v_j with capacity x_{ij} . For every vertex v_i, $i \neq 1$, we have an edge from v_i to t with capacity b_{v_i}. Now we argue that a v_1-t min-cut will give the desired solution. Indeed, let (X, Y) be such a cut. So, $v_1 \in X$, $t \notin X$. What is the capacity of (X, Y)? It is

$$\sum_{i,j:v_i \in X, v_j \notin X} x_{ij} + \sum_{i:v_i \in X} b_{v_i} = \sum_{i,j:v_i \in X, v_j \notin X} x_{ij} - \sum_{i:v_i \notin X} b_{v_i} + \sum_{v} b_v.$$

Note that it is exactly the expense minus benefit of moving the applications which are not in the set X (plus a term which is fixed). Thus, finding a min-cut is same as finding the set of applications for which expense minus benefit is minimized, or benefit minus expense is maximized.

Problem 8 [15 pts]

Given a graph $G = (V, E)$, and a natural number k, we can define a relation $\xrightarrow{G,k}$ on pairs of vertices of G as follows. If $x, y \in V$, we say that $x \xrightarrow{G,k} y$ if there exist k mutually edge-disjoint paths from x to y in G. Is it true that for every G and every $k \geq 0$, the relation $\xrightarrow{G,k}$ is transitive? That is, is it always the case that if $x \xrightarrow{G,k} y$ and $y \xrightarrow{G,k} z$, then we have $x \xrightarrow{G,k} z$? Give a proof or a counterexample.

Solution: Recall in class we mentioned that if we set capacity of each edge be set to 1, the maximum number of edge disjoint path from u to v is the maximum u-v flow, which is equivalent to the minimum u-v cut. Now suppose for contradiction that there are less than k edge disjoint paths from x to z, we know that there exists an x-z cut, namely (A, B) with cut capacity $c(A, B) < k$. Now let us focus on whether $y \in A$ or $y \in B$. If $y \in A$, we see that (A, B) is also a y-z cut, and therefore the minimum y-z cut has capacity less than k, contradicting to the fact that there are at least k edge disjoint paths from y to z. Similarly, if $y \in B$, (A, B) is also an x-y cut, and therefore the minimum x-z cut has capacity less than k, contradicting to the fact that there are at least k edge disjoint paths from x to z. We see that in either case, there is a contradiction. Therefore, there are at least k edge disjoint paths from x to z.