CSC373 Fall'19
Tutorial 4
Mon. Oct. 7, 2019

Q1 Ford Fulkerson

Consider the following network:

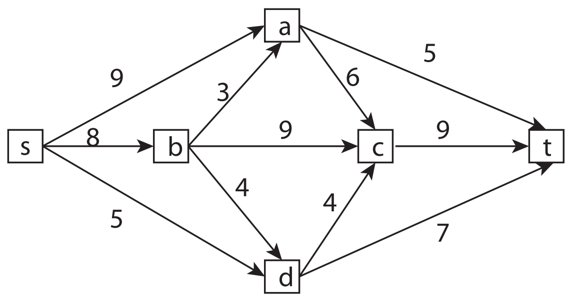
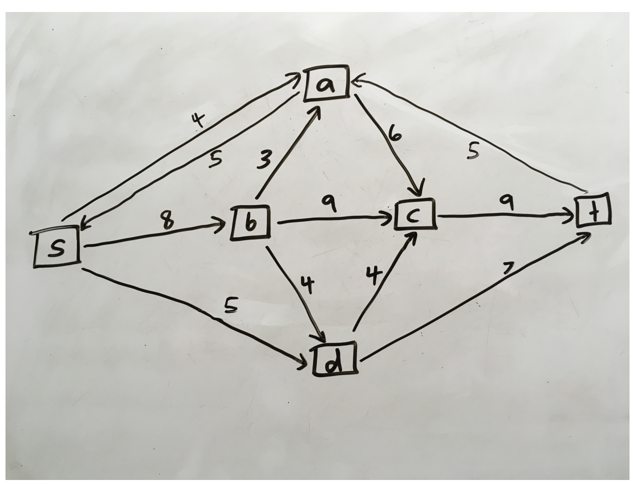


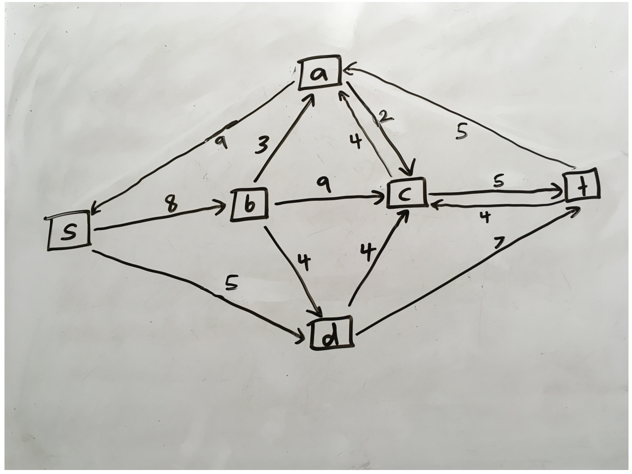
Figure 1:

(a) Compute a maximum flow in this network, using the Ford-Fulkerson algorithm: find augmenting paths and use them to augment the flow, one path at a time. For each augmenting path, take the time to write down the residual capacity and the resulting augmentation in the flow.

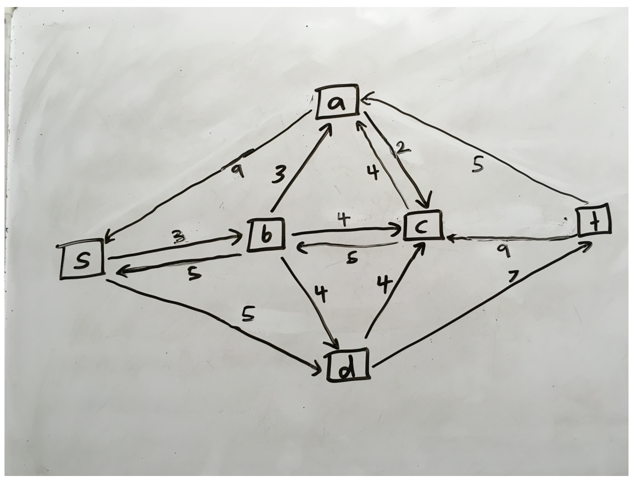
solution:
First push 5 units of flow along the path (s, a, t), resulting in the following residual graph:



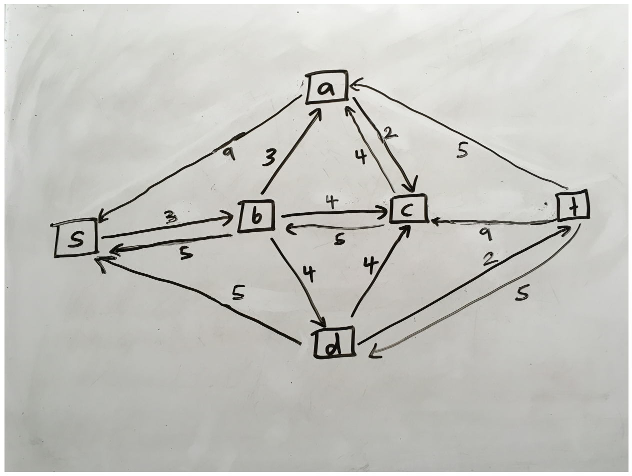
Next push 4 units of flow along (s, a, c, t):



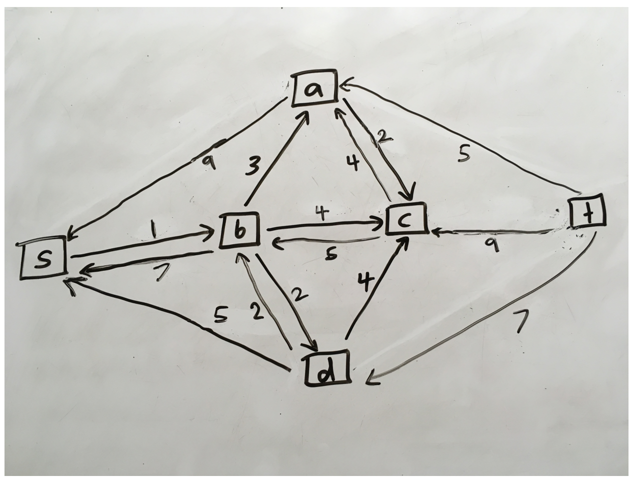
Next push 5 units of flow along (s, b, c, t):



Next push 5 units of flow along (s, d, t):



Finally, push 2 units of flow along (s, b, d, t):



Now we are done because there is no directed path from s to t. We can read the flow off of the final residual graph as follows:

s-a 9/9
s-b 7/8
s-d 5/5
b-a 0/3
a-c 4/6
a-t 5/5
b-c 5/9
b-d 2/4
d-c 0/4
d-t 7/7
c-t 9/9

(b) Consider the cut $X_0 = (\{s, b, c, d\}, \{a, t\})$. Identify all forward and all backward edges across X_0, then compute the capacity and the flow across X_0 (for your maximum flow from part (a)).

solution:
The forward edges across X_0 are (s, a), (b, a), (c, t), and (d, t).
The backward edge across X_0 is (a, c).
The capacity across X_0 is $9 + 3 + 9 + 7 = 28$.
The flow across X_0 is $9 + 0 + 9 + 7 - 4 = 21$.

(c) Find a cut in the network above whose capacity is equal to the value of your maximum flow (this provides a guarantee that your flow really is maximum). Use the algorithm outlined in the proof of the Ford-Fulkerson theorem.

solution: The cut $(\{s, a, b, c, d\}, \{t\})$ has a capacity of $5 + 9 + 7 = 21$, matching the maximum flow. (To find this cut, observe that $\{s, a, b, c, d\}$ is the set of vertices reachable from s in the final residual graph shown above, and t is the only vertex not reachable from s in that graph.)

Q2 Teaching Assignment

Consider the following problem:

Input: Set of profs $p_1, ..., p_n$ with teaching loads $L_1, ..., L_n$, and set of courses $c_1, ..., c_m$ with number of sections $S_1, ..., S_m$, along with subsets of courses that each prof is available to teach – each prof p_i has its own subset $A_i \subseteq \{c_1, ..., c_m\}$.

Output: Assignment of profs to courses such that:

- each prof p_i assigned exactly L_i courses,
- each course c_j assigned exactly S_j profs,
- no prof assigned a course outside their available set,
- no prof teaches multiple sections of the same course.

Show how to represent this problem as a network flow, and how to solve it using network flow algorithms. Justify carefully that your solution is correct and can be obtained in polynomial time.

solution: Create a graph with vertices $p_1, \ldots, p_n, c_1, \ldots, c_m, s, t$, and the following edges:

- (s, p_i) for each p_i, with capacity L_i;
- (c_j, t) for each c_j, with capacity S_j;
- (p_i, c_j) for each p_i and c_j such that p_i can teach c_j (i.e. $c_j \in A_i$), with capacity 1.

Compute a maximum flow f in this network (in poly-time), and assign p_i to teach c_j if $f(p_i, c_j) = 1$. It's possible to teach all sections if and only if $f(c_j, t) = L_j$ for all courses c_j.

To see that this algorithm is correct, first observe that any assignment of profs to courses yields a flow f in the following way:

- Let $f(p_i, c_j) = 1$ if p_i is assigned c_j, and otherwise $f(p_i, c_j) = 0$.
- Let $f(s, p_i)$ be the number of courses assigned to p_i.
- Let $f(c_j, t)$ be the number of profs assigned to c_j.

The value of this flow equals the number of course sections assigned. Therefore the value of the maximum flow is at least as large as the maximum number of course sections that can be assigned.

Similarly, any integer flow f in this network yields an assignment of profs to courses (assign p_i to c_j if $f(p_i, c_j) = 1$) such that no prof teaches more than L_i courses, no course is assigned to more than S_j profs, no prof will be assigned a course they are not available to teach, and no prof teaches multiple sections of a course. Therefore the maximum number of course sections that can be assigned is at least as large as the maximum flow value of the network.

Q3 Mobile Computing

Consider a set of mobile computing clients in a certain town who each need to be connected to one of several possible "base stations". We'll suppose there are n clients, with the position of each client specified by its (x, y) coordinates in the plane. There are also m base stations, each of whose positions is specified by (x, y) coordinates as well.

We wish to connect each client to exactly one base station. 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.

Show how to represent this problem as a network flow, and how to solve it using network flow algorithms. Justify carefully that your solution is correct and can be obtained in polynomial time.

solution: Let the clients and base stations be $c_1, \ldots, c_n$ and $b_1, \ldots, b_m$ respectively. Create a graph with vertices $c_1, \ldots, c_n, b_1, \ldots, b_m, s, t$, and the following edges:

- (s, c_i) with capacity 1, for each c_i;
- (b_j, t) with capacity L, for each b_j;
- for each c_i and b_j do the following. Let (x, y) and (x', y') be the coordinates of c_i and b_j respectively. If $\sqrt{(x - x')^2 + (y - y')^2} \leq r$ then add an edge (c_i, b_j) with capacity at least 1. (Any capacity at least 1 on this edge will yield the same max flow, because (s, c_i) has capacity exactly 1, so only one unit of flow can enter c_i, so only 1 unit of flow can leave c_i.)

Compute a maximum flow f in this network, and assign c_i to b_j if $f(c_i, b_j) = 1$.

The proof of correctness is similar to that in Q2. Given a flow f, assign c_i to b_j if $f(c_i, b_j) = 1$. The capacities of the network imply that each client is assigned to at most a single base station, no client is assigned to a base station beyond a distance of r, and at most L clients are assigned to any base station. Thus the number of clients that can be assigned is at least the max flow.

Conversely, given a legal assignment of clients to base stations, let f be the flow in which $f(s, c_i) = 1$ if c_i is assigned to some base station and $f(s, c_i) = 0$ otherwise, $f(c_i, b_j) = 1$ if c_i is assigned to b_j and $f(c_i, b_j) = 0$ otherwise, and $f(b_j, t)$ is the number of clients assigned to b_j. This is a legal flow, so the max flow is at least the number of clients that can be assigned.