CSC373 Summer '22
Tutorial 4
June 9-16, 2022

Q1 Ford Fulkerson

Consider the following network:

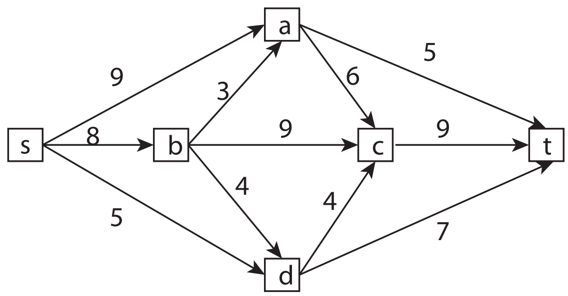

Figure 1:

(a) Compute a maximum flow in this network using the Ford-Fulkerson algorithm. For each iteration, write down the augmenting path, its bottleneck/residual capacity, and the value of the flow at the end of the iteration.

(b) Consider the cut $X_0 = (S = \{s, b, c, d\}, T = \{a, t\})$. Identify all forward and all backward edges across X_0. Compute the capacity of X_0.

(c) Find a cut in the network whose capacity is equal to the value of the flow you computed in part (a). (This provides a guarantee that your flow is indeed maximum.) Use the idea outlined in the proof of correctness of the Ford-Fulkerson algorithm.

Q2 Graph Modifications

In this problem, we will consider what happens to the maximum flow when the flow network G is modified slightly.

(a) TRUE/FALSE: In any network G with integer edge capacities, there always exists an edge e such that increasing the capacity of e increases the maximum flow value in G.

(b) Suppose we are given a network G with n nodes, m edges, and integer edge capacities, and we are also given a flow f in G of maximum value. We now increase the capacity of a specific edge e by one. Give an $O(m + n)$ time algorithm to find a maximum flow in the updated network.

Q3 Teaching Assignment

Suppose there are m courses: $c_1, \ldots, c_m$. For each $j \in \{1, \ldots, m\}$, course c_j has s_j sections. There are n professors: $p_1, \ldots, p_n$. For each $i \in \{1, \ldots, n\}$, professor p_i has a teaching load of ℓ_i and likes to teach the subset of courses $A_i \subseteq \{c_1, \ldots, c_m\}$.

Your goal is to use the network flow paradigm to design an algorithm, which either finds an assignment of professors to courses satisfying the following constraints or reports that no such assignment exists.

- Each professor p_i must be assigned exactly ℓ_i courses.
- Each course c_j must be assigned to exactly s_j professors.
- No professor should be be assigned a course that they do not like to teach.
- No professor can teach multiple sections of the same course.

(a) Describe your full algorithm. That is, describe the network flow instance created (nodes, edges, and edge capacities) and how your algorithm uses a maximum flow in this instance to determine if a valid assignment of professors to courses exists, and output one if it does.

(b) Prove that your reduction is correct. That is, prove that there exists a valid assignment of professors to courses if and only if your algorithm finds one.

(c) What is the worst-case running time of your full algorithm if you use the naïve Ford-Fulkerson algorithm to solve the network designed in part (a)?