

CSC 301: Analysis of Algorithms

Week 10: Flow

William Rebelsky
[rebelskyw]

Grinnell College

Monday 15th April, 2024

Table of Contents

- 1 Monday: Network Flow
 - Flow
 - Ford-Fulkerson
- 2 Wednesday: Bipartite Matching
 - Bipartite Matching
 - Summary
- 3 Friday: Discussion

CSC 301: Analysis of Algorithms

Class 31: Max Flow, Min Cut

William Rebelsky
[rebelskyw]

Grinnell College

Monday 15th April, 2024

Reminders

- Homework 7 is due Wednesday
- Quiz 7 is due tonight
- Token opportunities: Please fill out this form:
https://grinnell.co1.qualtrics.com/jfe/form/SV_2mhIpFACg0FH4YDc. If 75% of the class fills it out: 1 token, if 95% of the class fills it out: 2 tokens for everyone
- Other than Homework 6, Everything should be graded. If you see something that I didn't grade, let me know.
- Schedule:
 - This week: Flow
 - Next week: Dynamic Programming
 - The week after: Final Project plus more in depth review of topics (look for a survey from your mentors)

Essay

Essays have been graded:

- I appreciated the detailed, in depth essays that I received
- I did not see many comments about what I personally could do better in future courses (more generally, not just with the week on bias, there were some very good suggestions for how I could approach this topic more effectively)
- I will probably set up an anonymous survey for you to take regarding ways I could improve the course or my teaching more generally to take towards the end of the semester
- As I mentioned, this is my first time as the instructor of record and I would appreciate any thoughts you have on what went well or not so well in this course

Flow Basics

- Interpret directed graph as flow network
- Source to the rest of the graph to a sink
- Can be used to model many problems
- Edges have capacity $c(u,v)$ that is nonnegative
- Flow into a vertex equals flow out
- Goal: Maximize Flow from Source to Sink

Flow Requirements

- Directed Graph
- Each edge has a nonnegative capacity
- Every vertex lies on a path from the source, s , to the sink, t
- There are no reverse or antiparallel edges (i.e. if (u,v) exists, (v,u) does not)
- Flow is a function that satisfies two properties
 - Capacity constraint: The flow along every edge is non negative and not larger than the capacity of the edge
 - Flow conservation: Flow into a vertex equals the flow out of it

Fixing Flow Graphs

- If you have reverse edges: split one of the nodes into two, have the reverse edge flow through it
- If you have multiple sources or sinks: point from a supersource or to a supersink

Flow Practice

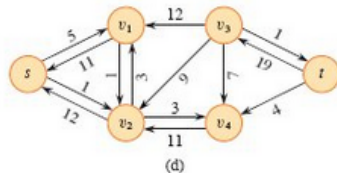
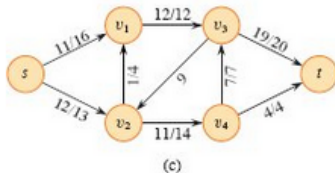
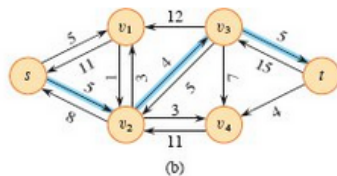
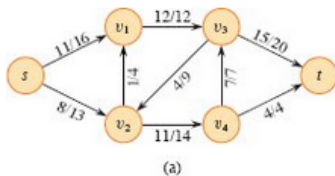
- 24.1-6: Professor has two children who don't get along and will not walk along the same path to school. They are ok crossing at a corner. Given a map of the town, formulate the problem of determining whether both children can go to the same school as a maximum-flow problem.

Ford-Fulkerson Method

- Method not algorithm: many implementations
- Relies on three ideas:
 - Residual Networks
 - Augmenting Paths
 - Cuts
- Iteratively increases the flow by finding an augmenting path in the residual network

Residual Network

- G_f is a network of edges whose capacities represent how the flow can change on edges of G
- Contains edges not in G
- Forward edge: Residual capacity (cap-flow)
- Backwards edge: Flow



Augmenting Path

- A simple path from s to t in the residual network G_f

Cut

- As with graphs, it is a partition of the vertices into S and $T=V-S$. Additional restriction: s in S , t in T
- Net flow across the cut: sum of the flow going from S to T across the cut minus the sum of the flow from T to S
- Capacity of the cut: Total capacity of the edges going from S to T
- Minimum Cut: a cut whose capacity is minimal over all cuts of the network

Max-Flow, Min-Cut

Theorem 24.6: The following are equivalent

- f is a maximum flow in G
- the residual network G_f has no augmenting paths
- $|f|=c(S,T)$ for some cut (S,T) of G

Ford-Fulkerson

- Set all flows to 0, while there is a path, add flow
- Finding the path is the major contributor to runtime
- If irrational edge capacities: may never terminate
- If rational edge capacities: convert to integer capacities
- If Integer capacities (which is most common): Find an augmenting path in linear time

Ford-Fulkerson: time proportional to flow

- s-u-t cap 1m, s-v-t cap 1m, u-v cap 1: see board

Edmonds-Karp Algorithm

- Goal: Fix the prior issue
- Use BFS to find an augmenting path with fewest edges
- $O(VE^2)$

- Unique flows and cuts: give an example of a flow network with a unique max flow but no unique min cut. Give an example with a unique min cut but no unique max flow
- 24.2-2 In figure 24.1b what is the net flow across the cut $S=s,v_2,v_4$ $T=v_1,v_3,t$? What is the capacity of the cut?
- 24.2-3 Show the execution of Edmonds-Karp on Figure 24.1a
- 24.2-4 What is the min cut corresponding to the maximum flow shown in figure 24.6? Of the augmenting paths, which one cancels flow?
- 24.2-13 Suppose you want to find the minimum cut with the smallest number of edges. Show how to modify the the capacities of G to create a new flow network in which the minimum cut in the new graph is a minimum cut with the smallest number of edges in G .

Table of Contents

- 1 Monday: Network Flow
 - Flow
 - Ford-Fulkerson
- 2 Wednesday: Bipartite Matching
 - Bipartite Matching
 - Summary
- 3 Friday: Discussion

CSC 301: Analysis of Algorithms

Class 32: Bipartite Matching

William Rebelsky
[rebelskyw]

Grinnell College

Wednesday 10th April, 2024

Reminders

- Homework 7 is due Tonight
- Homework 8 is released and due next Wednesday
- Quiz 8 will be released Friday

Maximum Bipartite Matching

- Given an undirected Graph, a matching is a subset of edges such that for all vertices, v , in V , at most one edge of M is incident on v
- A maximum matching is a matching of maximal cardinality, i.e has the most edges
- A bipartite graph is a graph with two sets L and R where all edges are between L and R
- Many practical applications: Machines with tasks, teachers with classes, classes with rooms

Bipartite Matching

Finding a maximum matching

- Using Ford-Fulkerson:
 - ➊ Add a source pointing to all of the left nodes
 - ➋ Add a sink pointing from all the right nodes
 - ➌ Make all the edges directed from L to R
 - ➍ Give all edges Capacity 1
 - ➎ Run FF

Summary

Summary

- Flow graphs are directed graphs from source to sink where the edges have capacity
- This has many practical applications
- MaxFlow/MinCut
- You can convert graphs to flow graphs and run max flow algorithms
- You can convert problems to graphs

Practice

- Given the residual graph on the board: Draw the flow graph that leads to this residual graph and justify whether or not it is a max flow
- 24.2-10: Show how to find a maximum flow in a flow network by a sequence of at most $|E|$ augmenting paths
- 24-1 Escape Problem: see book
- 24-2 Minimum Path Cover: see book

Table of Contents

- 1 Monday: Network Flow
 - Flow
 - Ford-Fulkerson
- 2 Wednesday: Bipartite Matching
 - Bipartite Matching
 - Summary
- 3 Friday: Discussion

CSC 301: Analysis of Algorithms

Class 33: Discussion

William Rebelsky
[rebelskyw]

Grinnell College

Friday 19th April, 2024

Reminders

- Homework 8 is due Wednesday
- Quiz 8 is released

Problems

- 24-3 Hiring Consulting Experts
- Course Management: You have m courses, r rooms with capacities. Each room can have at most one course, each course has exactly one TA and Professor. There are p professors with a list of courses that they can teach. Each professor has a list of 5 TAs they will work with. TAs can work in any course but are limited to only working one course. Figure out how to solve this problem, prove it works, and give a runtime analysis.

HW

- 1) 24.1-1 Show that splitting an edge in a flow network yields an equivalent network
- 2) 24.1-7 Suppose that a flow network has vertex capacities as well. Show how to convert it into an equivalent flow network without edge capacities. How many edges and vertices will the new network have?
- 3) 24.2-5 Prove that adding the supersources and supersinks doesn't change the flow
- 4) 24.2-8 No edges into s
- 5) 24.2-11 Edge Connectivity
- Bonus: Course Management: You have m courses, r rooms with capacities. Each room can have at most one course, each course has exactly one TA and Professor. There are p professors with a list of courses that they can teach. Each professor has a list of 5 TAs they will work with. TAs can work in any course but are limited to only working one