Graph Theory and Simulation

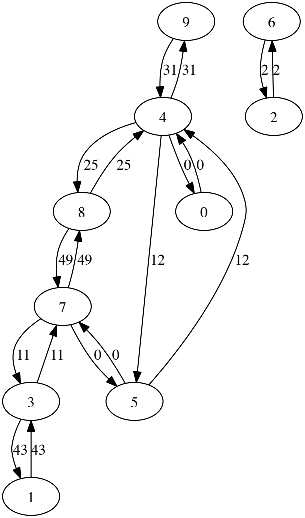
Matthew Hall

February 28, 2013

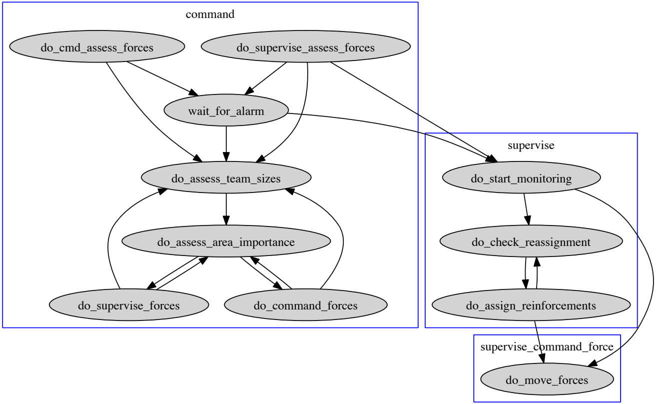
Nodes and Edges

- Nodes
- Edges
  - Weights
  - Direction
  - Only need the immediate neighbors
- http://compgeom.cs.uiuc.edu/ jeffe/teaching/algorithms/
- http://www.win.tue.nl/ nikhil/courses/2012/2WO08/max-flow-applications-4up.pdf

Example



Example



- Motivation: I have nodes and their list of neighbors, what can I do?

Data Representation

- Draw communication networks
- Draw train of thought
- Interactions between players

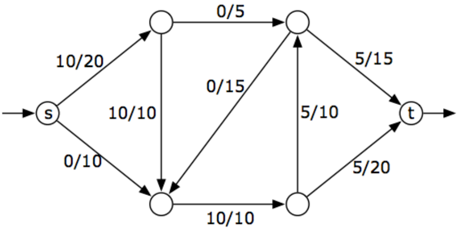
Algorithms

- Visit Every Node (DFS, BFS)
- Shortest Path (A*)
- Spanning Tree
- Coloring
- Connected Components
- Social interaction mining (6 degrees of Kevin Bacon)
- ... Lots more to do ...
- Min Flow / Max Flow

Find the Maximum Flow

- Source (s)
- Sink (t)
- Flow is conserved at each node
- Source Flow = Sink Flow

Example Flow

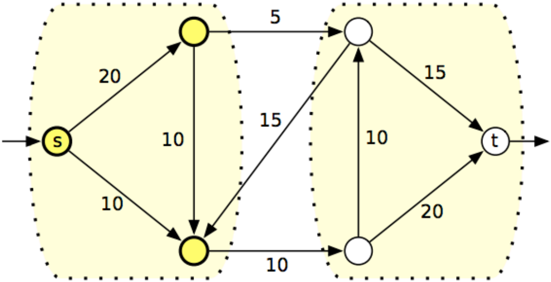


An (s,t)-flow with value 10. Each edge is labeled with its flow/capacity.

Cuts

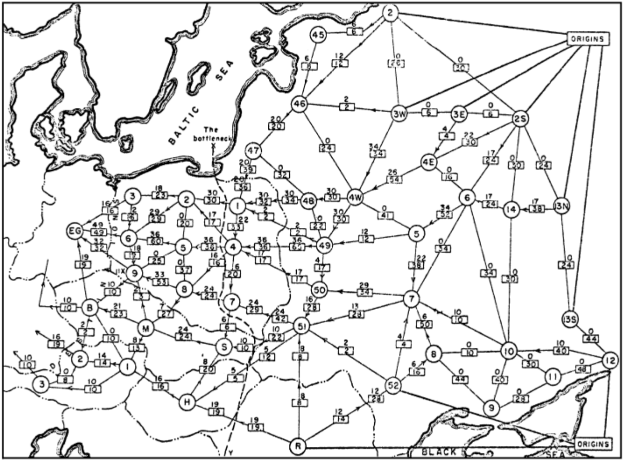
- A cut is a separation of vertexes. For min flow, it is often called the (S,T) cut.
- The capacity of a cut is the sum of the capacities of the edges going from a vertex in S to a vertex in T.
- Maximum flow of a graph is equal to the minimum cut!

Example Cut



An (s, t)-cut with capacity 15. Each edge is labeled with its capacity.

Earliest Usage



Harris and Ross's map of the Warsaw Pact rail network

Algorithms

Technique	Direct	With dynamic trees	Sources
Blocking flow	$O(V^3)$	$O(VE\log V)$	[Dinits; Sleator and Tarjan]
Network simplex	$O(V^2E)$	$O(VE\log V)$	[Dantzig; Goldfarb and Hao; Goldberg, Grigoriadis, and Tarjan]
Push-relabel (generic)	$O(V^2E)$	—	[Goldberg and Tarjan]
Push-relabel (FIFO)	$O(V^3)$	$O(V^2\log(V^2/E))$	[Goldberg and Tarjan]
Push-relabel (highest label)	$O(V^2\sqrt{E})$	—	[Cheriyan and Maheshwari; Tunçel]
Pseudoflow	$O(V^2E)$	$O(VE\log V)$	[Hochbaum]

Several purely combinatorial maximum-flow algorithms and their running times.

- Take it on faith that this problem is solved efficiently
- Lots of approaches with reasonable running times

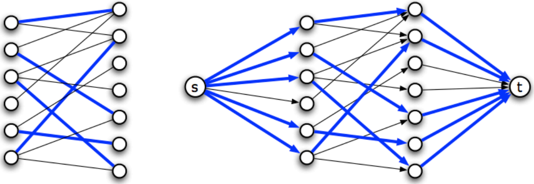
What can we do?

- Find maximum capacity of a network
- Find the bottleneck of a network
- Matching!

What is it?

In the mathematical field of graph theory, a bipartite graph (or bigraph) is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V; that is, and are each independent sets. Equivalently, a bipartite graph is a graph that does not contain any odd-length cycles. (Wikipedia)

Example



A maximum matching in a bipartite graph G, and the corresponding maximum flow in G'.

So what?

- We can convert matching problems into a bipartite graph and then find the maximum flow.
- And??????
- And, we can find the maximal or perfect match.
- And, by changing the way the graph is constructed, we can control the matching.

Multiple Commodity

- Run multiple types of commodities
- Must be fractional flows (integral flows is much harder)

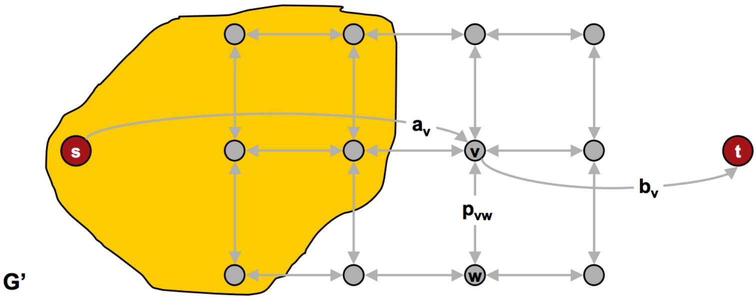
Multiple Tiers of Matching

- Problem: Assign resources to multiple dependent targets
- Solution: Max flow with tiers connected from layer 1 to layer 2 to … layer n

Separation of foreground and background in images

- Construct flow from source (foreground) to sink (background)
- Pixels are connected to minimize transition from foreground to background
- Extra terms to smooth the transition

Separation of foreground and background in images



Matching Sniper Perches to Targets

- Sniper Locations are S
- Targets Locations are T
- Maximal matching is each sniper having enough (or the most targets) and each target not having too much firepower assigned
- We can also assign weapons to each sniper
- What about too many targets to overwhelm the sniper???
- Can we represent movement????

Matching Weapon Systems to Targets in React

- Weapons are S
- Targets are T
- Maximal matching is each weapon having a target and no target is getting excessive firepower
- Is this reasonable for missile defense? people?

Matching People to Coverspots

- People are S
- Cover Locations are T
- Maximal matching is each person having cover and each person walking the shortest distance
- We can also assign people based on the weapon they carry
- Is this reasonable for people to do?
- Would they have planned this?
- What about the defenders?
- What about supervisors?

More Matching

- People to patrol zones
- Ambulances to hospitals
- Teachers to students to classrooms

Traffic Flow

- Pipes, cars, rivers, trains, etc
- Traffic disruption
  - Attack teleco
  - Attack shipping
  - Attack any linear infrastructure

Safety / Risk

- Edge weights are probability of failure or success
- The bottleneck is where the risk is highest
- If I pass this threshold, its smooth sailing.
- If I improve the cut, I have an overall safer system.
- Shortest path is also really useful in this case!

Definition

- a vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex of the set.
- Minimum vertex cover is NP-Hard

Approximation

- Getting a vertex cover within 2n of real cover can be done in polynomial time

Conclusion

Graphs

- Lots of algorithms
- Easy to implement
- Lots of uses

Resources

- Boost Graph Library (BGL)
- The Big Book of Algorithms