Maximum Network Flow

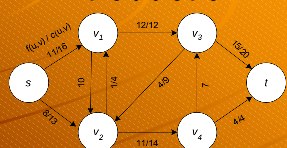
Ford-Fulkerson Method

Dave Miller

Presentation Overview

- Introduction
- Basic Method
- Things To Remember
- Key Concepts
- Basic Algorithm
- Example
- Analysis
- References

Introduction



Flow network G=(V,E) with flow f(u,v) and capacity c(u,v)

- Maximum flow problem
  - Given a flow network G = (V,E) determine the greatest possible flow *f(u,v)* from source (*s*) to sink (*t*) without violating capacity *c(u,v)*
- Ford-Fulkerson is the classical method for solving the maximum flow problem

Introduction – Cont.

- Tidbits
  - Co-developed in 1956 by Mathematicians Lester Randolph Ford Sr. and Delbert Ray Fulkerson
  - Referred to as a method because there are several algorithm implementations with differing running times

Basic Method

- Iterative process with the flow *f* initial value set to 0
- On each iteration, increase flow *f* by finding an "augmenting path" and augmenting the flow along this path
- Repeat process until no augmenting path can be found, process terminates yielding maximum flow

Basic Method – Cont.

- Pseudo code

```
FORD-FULKERSON-METHOD(G, s, t)
    initialize f to 0
    while there exists an augmenting path p
        do augment flow f along p
    return f
```

where,

G is the flow network G = (V,E)
s is the source of the flow
t is the sink
f is the flow from s to t
p is the path

Things To Remember

- Properties of flows
  - Capacity constraint:
    - A flow cannot be greater than the capacity of the edge that is f(u,v) ≤ c(u,v)
  - Skew symmetry
    - The flow from vertex u to vertex v is the negative of the flow in the reverse direction that is f(u,v) = -f(v,u)
    - Notional convenience
  - Flow conservation
    - The total flow entering a vertex must equal the total flow leaving the vertex
    - More formally at a given vertex ∑ f(u,v) = 0

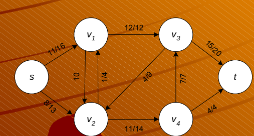
Key Concepts

- Residual Network
- Augmenting Path

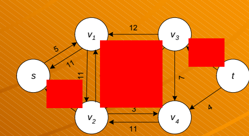
Residual Network

- Consists of edges that can admit more flow
- Def. ***residual capacity*** $c_f(u,v)$ is the additional flow that can be pushed on to an edge without exceeding the capacity, *residual capacity is given by* $c_f(u,v) = c(u,v) - f(u,v)$
- Residual network G_f is a flow network with capacities c_f

Residual Network – Cont.



A flow network G with *f*=19



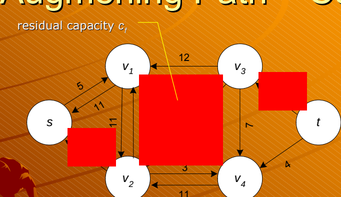
A residual network G_f with the augmenting path highlighted

Augmenting Path

- Is a path *p* from *s* to *t* in the residual network G_f
- Max. amount the flow can be increased on each edge in the augmenting path *p* is called the residual capacity of *p*, *given by*

$$c_f = \min\{c_f(u,v) : (u,v) \text{ is on } p\}$$

Augmening Path – Cont.



Augmenting path (highlighted in red) *p* in G_f with the residual capacity c_f = 4

Basic Algorithm

- Pseudo code

  Ford-Fulkerson(G,s,t)

  for each edge (u,v) ∈ E[G]

  do f[u,v] ← 0

  f[v,u] ← 0

  while there exists a path p from s to t in the residual network G_f

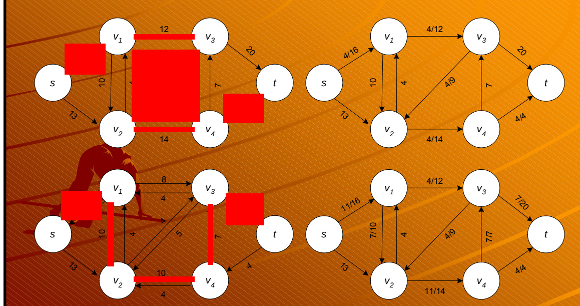
  do $c_f(p) \leftarrow \min\{c_f(u,v) : (u,v) \text{ is in } p\}$

  for each edge (u,v) in p

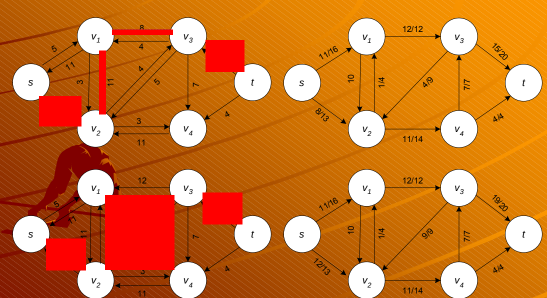
  do f[u,v] ← f[u,v] + $c_f(p)$

  f[v,u] ← -f[u,v]

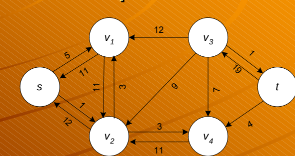
Example



Example – Cont.



Example – Cont.



No further augmenting paths are found, algorithm exits with max. flow f=23

Analysis

- Running time depends on how the augmenting path is chosen
- If capacities are integral values and using a breadth-first or depth-first search O(E) to find the augmenting path, the algorithm runs in O(E |f*|) where f* is the maximum flow

References

- Thomas H. Cornmen, Charles E. Leiserson, and Clifford Stein. *Introduction to Algorithms.* MIT Press, 2001
- Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993
- http://en.wikipedia.org/wiki/Ford-Fulkerson_algorithm, 10 April, 2005