Ford-Fulkerson Method

Flow maximization in a network (graph) with capacities

Basic idea:
- Find a path from source to target that still has flow capacity *(augmenting path)*
- Add the maximum flow allowed along this path
- Repeat until *no augmenting path*

Issues

1. How do we account for flow by *cancellation*?

2. Does adding an augmenting path lead to a legal flow?

3. Will this process converge?

4. If so, will it lead to a maximum flow?
Problem Formulation

Directed graph $G = (N, E)$
- Two special nodes: $s$ (source) and $t$ (target)
- Assume for any node $v \in N$, there are paths $s \rightarrow v \rightarrow t$

Capacity $c(v, w) \geq 0$
- If $(v, w)$ not an edge, then $c(v, w) = 0$

Flow $f(v, w)$ can be

Detail

Assume no "useless" flows between nodes
- Positive flow in only one direction
Legal Flow $f$

1. $f(v, w) \leq c(v, w)$

2. $f(v, w) = -f(w, v)$

3. For any $v \neq s, t$

$$\sum_{w \neq v} f(v, w) = 0$$

Inputs = Outputs

Consider node 6

$f(6,16) + f(6,12) + f(6,8) + f(6,3)$
**Value of Flow $f$**

Total flow out of source

$$|f| = \sum f(s, w)$$

**Residual Capacity for a Flow**

*Residual Capacity* between nodes $v$ and $w$:

$$r(u, v) = c(u, v) - f(u, v)$$

For example:

$$r(3, 7) = c(3, 7) - f(3, 7)$$

$$r(7, 3) = c(7, 3) - f(7, 3)$$
Residual Graph for Flow $f$

$$R = (N, E')$$

$$E' = \{ (v, w) \mid r(v, w) > 0 \}$$

*Capacities are residual capacities for $f$*
Capacity of a Path

Minimum capacity edge \((v_i, w)\) \(r(v_i, w)\)

Add a flow of \(r(v_i, w)\) along the path

Is the Result a Legal Flow?

- Capacity?

- Skew symmetry?

- Conservation?
Is it a Maximum Flow?

Would seem so:
Have a group of edges that divides s from t and can’t take more flow.
Results

• Net flow across any cut \( |S| \)
• \( |f| \) is bounded above by capacity of any cut
• Max-flow/min-cut theorem
  1. \( f \) is maximum flow
  2. residual graph has no augmenting paths
  3. \( |f| \) is capacity of some \( S \mid T \)

2 \( \Rightarrow \) 3

Consider residual graph \( R \) with no augmenting path
\( S = \{ v \mid s \sim v \text{ in } R \} \)
\( T = N - S \)
Must have \( t \in T \)
Claim that for \( (v, w) \) with \( v \in S, t \in T \),
must have \( r(v, w) = 0 \) \( (f(v, w) = c(v, w)) \)
Suppose not. Then \( r(v, w) > 0 \). Then \( R \) has an edge \( (v, w) \) and so \( s \sim w \sim t \), so \( w \in S \).
Basic Implementation

Start with 0 flow
Repeat
  Add flow along an augmenting path
  Until no such paths

Does it always converge?
  Yes, if capacities are integers. Flow grows by
  at least 1 each time

How long does it take?
  If you pick augmenting paths arbitrarily
  \( O(|E| \cdot f^*) \)
  \( f^* \) maximum flow in the graph

Edmonds-Karp Algorithm

Start with 0 flow
Repeat
  Add flow along an augmenting path with shortest
  length
  Until no paths

Time complexity no longer depends on
value of maximum flow
\( O(|N| \cdot |E|^2) \) time

Intuition: Length of shortest path to a node
in residual graph
  • Each addition of flow increases distance to one node
  • Distance to a node can be increased at
    most \( |E| \) times, since \( |E| \) is length of
    longest possible path
Edmonds-Karp Example