17 Mar 2021

Running time of Ford-Fulkerson + Push-relabel algorithm (§7.4)

Recap:

G dir graph

s, t source, sink vertices

$c(e)$ capacity of e

$f(e)$ flow on e

G_f "residual graph"

forward edge: (u,v) s.t. $f(u,v) < c(u,v)$

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

backward edge: (v,u) s.t. $f(u,v) > 0$

$$c_f(v,u) = f(u,v)$$

Augmenting path: path $s \to t$ in G_f.

augment(f, P) is an operation that takes f, P and modifies f to increase its value by

$$b(f,P) = \min \{ c_f(e) \mid e \in E(P) \}.$$

FORD-FULKERSON ALGORITHM

initialize $f(e) = 0 \quad \forall e$

repeat {
  compute G_f
  find augmenting path P
  $f \leftarrow$ augment (f, P)
} until G_f has no augmenting path.

MAX-FLOW MIN-CUT THEOREM:

For every flow network, the value of a maximum flow equals the minimum capacity of an s-t cut.

Furthermore every terminating execution of Ford-Fulkerson finds a maximum flow, and a min s-t cut can be found by taking

$A = \{$ vertices reachable from s by a path in $G_f \}$

$B = V - A.$

Unfinished business: bound the running time.

Key observation: every "augment" operation increments $v(f)$ by a positive amount. Try to bound the # of such increments.

Lemma: If all residual capacities $c_f(e)$ are integers and we apply augment(f, P) to an augmenting path, afterward all residual capacities remain integers.

Proof. If f is the old flow, f' is the new flow, and $b = b(f, P)$ then

$$\forall e \quad f'(e) - f(e) \in \{-b, 0, b\}.$$

Since b is one of the resid capacities, all three of $-b, 0, b$ are integers.

Corollary. When Ford-Fulkerson is run on a flow network with integer capacities the flow at the end of every iteration is an integer-valued flow, and the flow value increases by at least 1 in each iteration.

Let $C = \sum_{e \text{ out of } s} c(e)$.

Then every flow must satisfy $v(f) \leq C$, hence Ford-Fulkerson never takes more than C iterations.

n = # vertices
m = # edges

Running time per iteration:

- Compute $c_f(e)$ $\forall e$ — $O(m)$
- Search for s-t path, P — $O(m)$ BFS/DFS
- augment (f, P) — $O(n) \leq O(m)$

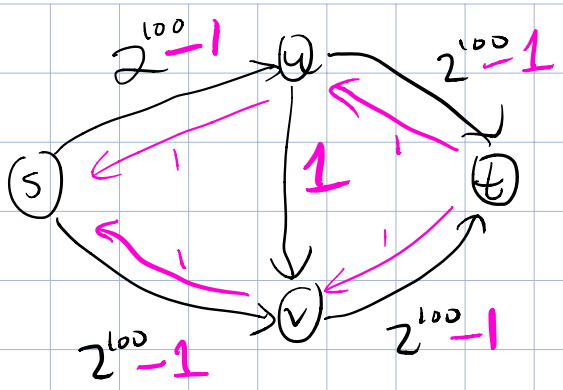
Total $O(m)$.

Ford-Fulkerson takes $O(mC)$.

"pseudopolynomial"

Running time bounded by a polynomial function of the magnitudes of the input #'s.

Potentially exponential in # of bits in the input description.



Max-flow = $2 \cdot 2^{100}$

An execution of Ford-Fulkerson with obnoxiously bad choices of augmenting paths may take $2 \cdot 2^{100}$ iterations to terminate.

Just take aug path with min # of hops?
It's possible to prove that heuristic always leads the algo to finish in poly time. (Edmonds-Karp Algorithm)

FLOW INTEGRALITY THEOREM: In a flow network with integer capacities there always exists an integer-valued max flow.

Def. A **preflow** in a flow network is a function $f : E \to \mathbb{R}$ s.t.

(a) [Capacity constraints]

$\forall e \quad 0 \le f(e) \le c(e)$

(b) [non-negative excess]

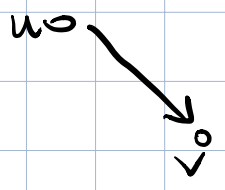
Let $x(v) := \sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e)$.

For all $v \neq s$, $x(v) \geq 0$.

The push-relabel algorithm maintains a label $h(v)$ at each vertex V. These labels always satisfy a **steepness condition** with respect to f:

if $e = (u,v) \in E(G_f)$ (meaning $c_f(e) > 0$)

then $h(u) \le h(v) + 1$.



PUSH-RELABEL.

initialize $f(e) = \begin{cases} c(e) & \text{if } e \text{ is edge out of } s \\ \phi & \text{otherwise} \end{cases}$

initialize $h(v) = \begin{cases} n & \text{if } v = s \\ \phi & \text{otherwise.} \end{cases}$

Compute $c_f(e)$ $\forall\ e \in E(G_f)$

Compute $x(v)$ $\forall\ v$

while $\exists v \neq t$ with $x(v) > 0$:

  if G_f contains an edge (v,w) such that $h(v) > h(w)$:

    PUSH(v,w): let $\delta = \min\{x(v), c_f(v,w)\}$

      if (v,w) is forward edge

        $f(v,w) \leftarrow f(v,w) + \delta$

      if (v,w) is backward edge

        $f(w,v) \leftarrow f(w,v) - \delta$

      update $c_f(v,w)$, $c_f(w,v)$, $x(v)$, $x(w)$

  else:

    RELABEL(v): $h(v) \leftarrow h(v) + 1$.

endwhile

output f.

Obs 1. When alg terminates, f is a flow.

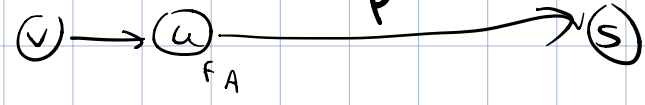
Lemma. At any time, if $x(v) > 0$ then G_f contains a path from v to s.

Proof. Let $A = \{ v \mid G_f \text{ contains a path } v \text{ to } s \}$

$B = V - A$.

We know $\forall\, u \in A,\ v \in B,\ f(u,v) = 0$.

Because if $u \in A$, $f(u,v) > 0$ then $E(G_f)$ contains (v,u).

Then 

so $v \in A$.

We've shown that no (pre)flow leaves A.

This implies that $v \notin A$ cannot have $x(v) > 0$.

(Where did the preflow come from that leads to excess at v?)