29 Sep 2023 Max Flow Algorithms

Ford – Fulkerson

$O(m)$, BFS

initialize $f = 0$, $G_f = G$

while $\boxed{\exists \text{ an augmenting path } P}$ in G_f:

compute $\delta(P)$ = min residual capacity on P ← $O(n)$

update f to $f + \delta(P) \cdot f^P$ ← $O(n)$

recompute G_f ← $O(n)$

endwhile

output f

Correctness conditional on termination: if we stop, we output a feasible flow with no augmenting path.

[Weds.] $f \leftarrow f + \delta(P) f^P$ preserves feasibility

[Weds] implies f is max flow.

$O(m)$ time per iteration.

If capacities are integer valued, then

iter i

(a) f will always be an integer valued flow

(b) G_f will have integer capacities.

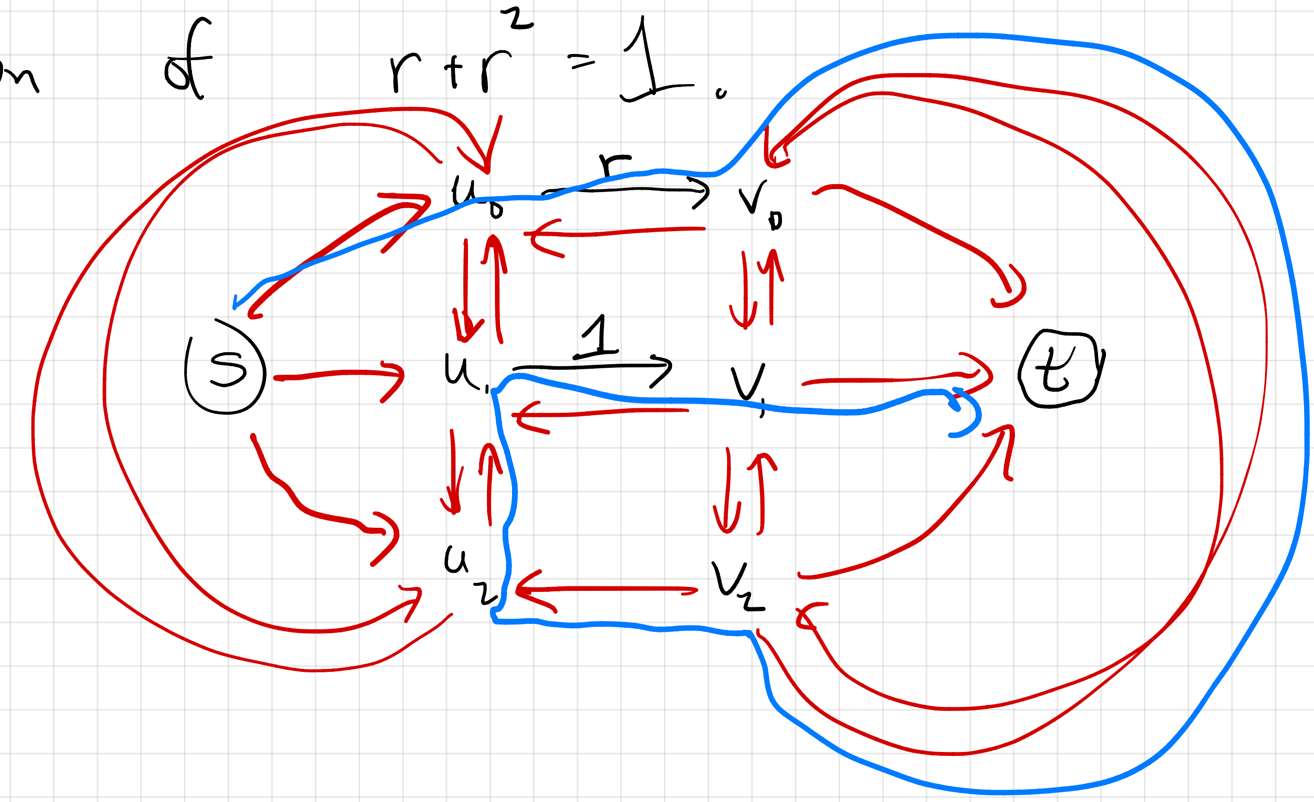
iter $i+1$

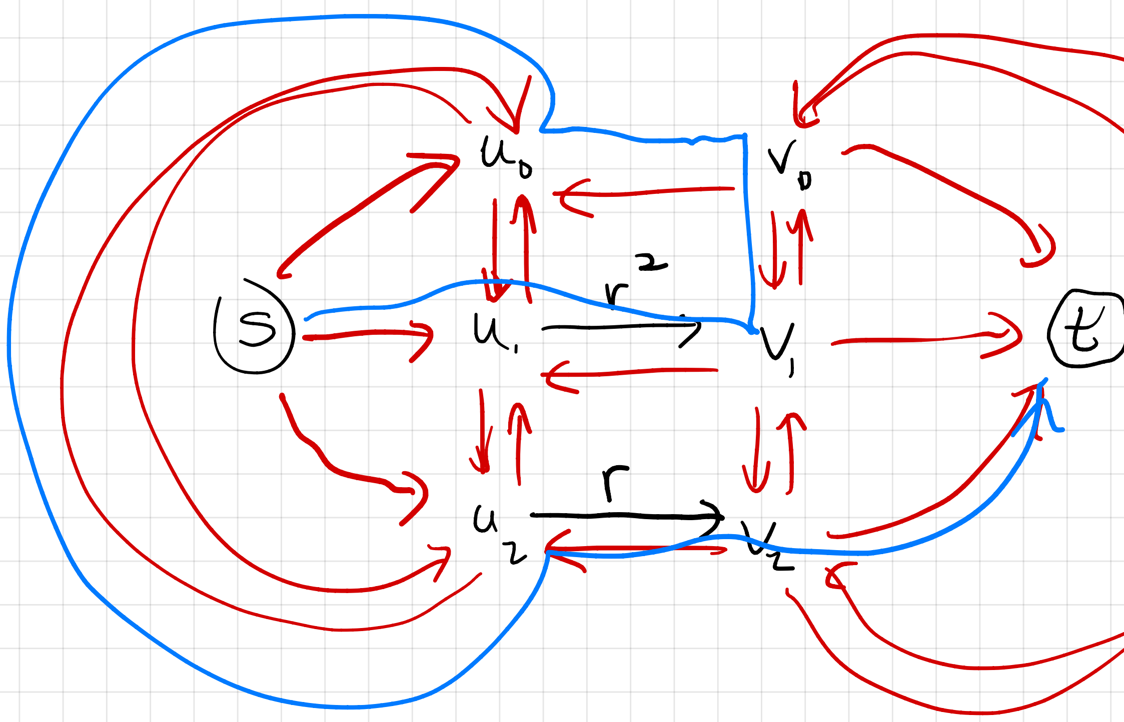
$\text{val}(f)$ increases by at least 1 per iteration, so # iterations bounded by $\text{val}(f^*)$ where f^* is a max flow.

Overall running time $O(m \cdot \text{val}(f^*))$.

Ex. Let $r = \frac{1}{2}(\sqrt{5}-1)$ be the positive solution of $r + r^2 = 1$.

G.





This iterates without end.

Algorithm	Running Time
Ford-Fulkerson	$O(m \cdot \mathrm{val}(f^*))$
Edmonds-Karp #1 augmenting path that maximizes $\delta(P)$	$O(m \cdot \log(n) \cdot \log(n \cdot \mathrm{val}(f^*)))$
Edmonds-Karp #2 augmenting path with fewest edges	$O(m^2 n)$

pseudopolynomial (poly if numbers encoded in unary)

weakly polynomial (poly in input size)

strongly polynomial (polynomial with no dependence on # of digits, as long as arithmetic takes $O(1)$.)

Dinitz	$O(mn^2)$
Push-Relabel	$O(n^3)$
⋮	
Orlin's Algorithm	$O(mn)$ ← fastest known strongly poly
Chen Kyng Liu Peng Probst Gutenberg & Sachdeva (2022)	$O(m^{1+o(1)} \log val(f^*))$ w.h.p. ← fastest known

Push-Relabel Algorithm

Maintains two objects: a preflow and a height function.

<u>Def.</u> f is a preflow if it satisfies

(i) $f(u,v) + f(v,u) = 0 \quad \forall u,v$

(ii) $\underbrace{\sum_{u \in V} f(u,v)}_{\text{"excess" of } v, \text{ net flow arriving into } v.} \geq 0 \quad \forall v \neq s.$