Network Flows (Cont...)

RAMESH K. JALLU

LEC-12

DT. 14/02/22

Algorithm(Recap)

FORD-FULKERSON-METHOD(G, s, t)

initialize flow f to 0
while there exists an augmenting path p in the residual network G_f
    augment flow f along p
return f

FORD-FULKERSON(G, s, t)

for each edge $(u, v) \in G.E$
    $(u, v).f = 0$
while there exists a path p from s to t in the residual network G_f
    $c_f(p) = \min\{c_f(u, v) : (u, v) \text{ is in } p\}$
    for each edge (u, v) in p
        if $(u, v) \in E$
            $(u, v).f = (u, v).f + c_f(p)$
        else $(v, u).f = (v, u).f - c_f(p)$

Designing a Faster Flow Algorithm

- If we choose augmenting paths with large bottleneck capacity, we can make a lot of progress
- Having to find such paths can slow down each individual iteration by quite a bit
- We can improve the bound on FORD-FULKERSON by finding the augmenting path p with a BFS more cleverly
- That is, we choose the augmenting path as a *shortest* path from s to t in the residual network, where each edge has unit distance
- We call the Ford-Fulkerson method so implemented the ***Edmonds-Karp algorithm***

More tighter analysis

- The analysis depends on the distances to vertices in the residual network *Gf*
- We denote by $\delta f\,(u, v)$ for the shortest-path distance from u to v in *Gf* , where each edge has unit distance
- For all vertices $v \in V$ - $\{s, t\}$, let the shortest-path distance be $\delta_f(s, v)$
- An edge (u, v) in a residual network G_f is ***critical*** on an augmenting path p if the residual capacity of p is the residual capacity of (u, v)
- After we have augmented flow along an augmenting path, any critical edge on the path disappears from the residual network

Cont …

- Observe that, at least one edge on any augmenting path must be critical
- We can show that each of the $|E|$ edges can become critical at most $|V| / 2$ times
- Let u and v be vertices in V that are connected by an edge in E
- Since augmenting paths are shortest paths, when (u, v) is critical for the first time, we have $\delta_f(s, v) = \delta_f(s, u) + 1$
- Once the flow is augmented, the edge (u, v) disappears from the residual network
- It cannot reappear later on another augmenting path until after the flow from u to v is decreased, which occurs only if (v, u) appears on an augmenting path

Cont …

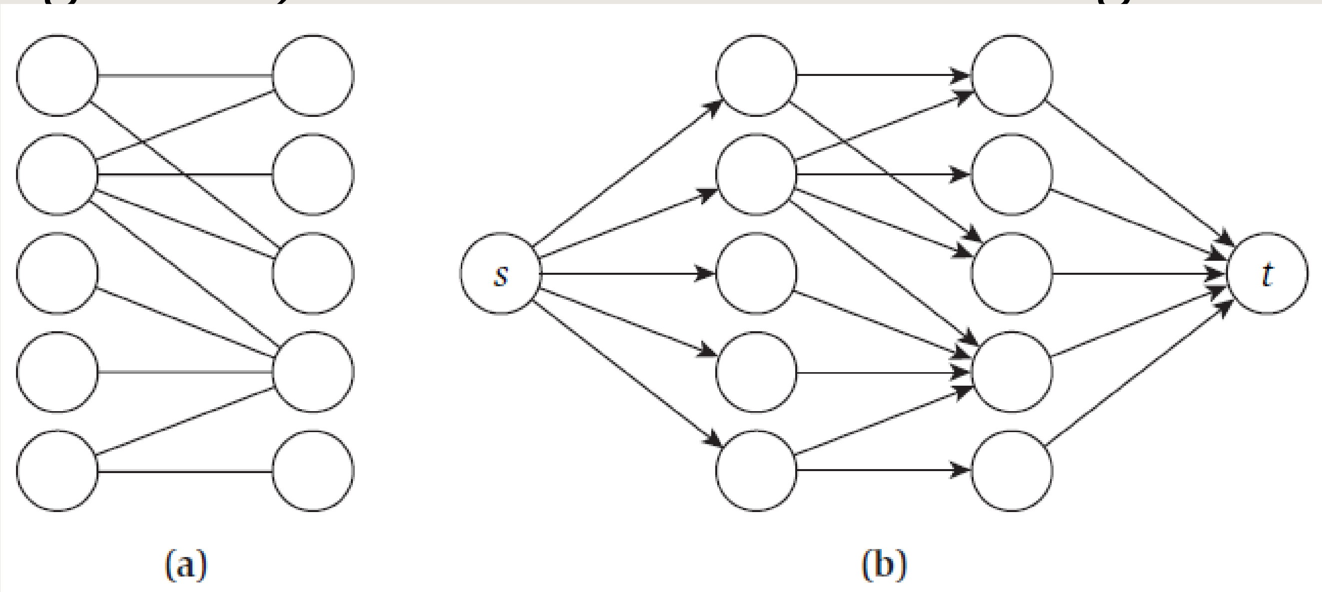
- If f' is the flow in G when this event occurs, then we have $\delta_{f'}(s, u) = \delta_{f'}(s, v) + 1 \geq \delta_f(s, v) + 1 = \delta_f(s, u) + 2$
- The intermediate vertices on a shortest path from s to u cannot contain s, u, or t
- Therefore, until u becomes unreachable from the source, if ever, its distance is at most $|V| - 2$
- Thus, after the first time that (u, v) becomes critical, it can become critical at most $(|V| - 2)/2 = |V|/2 - 1$ times more, for a total of at most $|V|/2$ times
- Since there are $O(|E|)$ pairs of vertices that can have an edge between them in a residual network, the total number of critical edges during the entire execution of the Edmonds-Karp algorithm is $O(|V||E|)$

Running time

- **Theorem**: If the Edmonds-Karp algorithm is run on a flow network $G = (V, E)$ with source s and sink t, then the total number of flow augmentations performed by the algorithm is $O(|V||E|)$
  - Hence, running time of the algorithm is $O(|V||E|^2)$
- This is the first polynomial time algorithm for the problem
- There has since been a huge amount of work devoted to improving the running times of maximum-flow algorithms

Finding a maximum bipartite matching

- The ***bipartite matching problem*** is that of finding a matching in G of largest possible size
- We construct a flow network $G' = (V', E')$ from G (by assigning a unit capacity for each edge in E') and run Ford-Fulkerson algorithm on G'



Analysis

- If M is a matching in G, then there is an integer-valued flow f in G' with value $|f| = |M|$
- ***Proof***: If $(u, v) \in M$, then $f(u, v) = f(s, u) = f(v, t) = 1$
- For all other edges $(u, v) \in E'$, we define $f(u, v) = 0$
- It is simple to verify that f satisfies the capacity constraint and flow conservation properties
- Intuitively, each edge (u, v) in M corresponds to one unit of flow in G' that traverses the path $s \rightarrow u \rightarrow v \rightarrow t$
- Moreover, the paths induced by edges in M are vertex-disjoint, except for s and t
- The net flow across cut $(L \cup \{s\}, R \cup \{t\})$ is equal to $|M|$; thus, the value of the flow is $|f| = |M|$

Analysis (Cont …)

- If f is an integer-valued flow in G', then there is a matching M in G with cardinality $|M| = |f|$
- ***Proof***: let f' be an integer-valued flow in G', and let $M = \{(u, v) \mid u \in L, v \in R,$ and $f(u, v) > 0\}$
- Each vertex $u \in L$ has only one entering edge, namely (s, u), and its capacity is 1
- Thus, each $u \in L$ has at most one unit of flow entering it, and if one unit of flow does enter, by flow conservation, one unit of flow must leave
- Furthermore, since f is integer-valued, for each $u \in L$, the one unit of flow can enter on at most one edge and can leave on at most one edge

Analysis (Cont …)

- Thus, one unit of flow enters u if and only if there is exactly one vertex $v \in R$ such that $f(u, v) = 1$, and at most one edge leaving each $u \in L$ carries positive flow
- A symmetric argument applies to each $v \in R$
- The set M is therefore a matching
- To see that $|M| = |f|$, observe that for every matched vertex $u \in L$, we have $f(s, u) = 1$, and for every edge $(u, v) \in E \setminus M$, we have $f(u, v) = 0$
- Consequently, the flow value across cut $(L \cup \{s\}, R \cup \{t\})$ is equal to $|M|$
- Therefore, we have that $|f| = f(L \cup \{s\}, R \cup \{t\}) = |M|$

Running time

- Construction of $G' = (V', E')$ can be done in $O(|V|)$ time as $|E| \geq |V|/2$ and $|E| \leq |E'| = |E| + |V| \leq 3|E| = O(|E|)$

- WKT, the running time of Ford-Fulkerson is $O(C|E|)$

- As $C \leq |V|$, we can conclude that a matching of maximum size in a bipartite graph $G = (V, E)$ can be computed in $O(|V||E|)$ time

- Thank you!