Theory of Computing

Lecture 10

MAS 714

Hartmut Klauck

Network Flows

- A *flow network* is a directed graph G=(V,E) with nonegative edge weights C(u,v), called *capacities*
  - nonedges have capacity 0
- There is a *source* s and a *sink* t
- We assume that for all v there is a path from s to v and from v to t
- A *flow* in G is a mapping f from $V \times V$ to the reals:
  - For all u,v: $f(u,v) \leq C(u,v)$ [capacity constraint]
  - For all u,v: $f(u,v) = -f(v,u)$ [skew symmetry]
  - For all $u \neq s,t$: $\sum_v f(u,v)=0$ [flow conservation]
- The *value* of a flow f is $|f|=\sum_v f(s,v)$
- The *Max Flow problem* consists of finding the maximum value of any flow for G,C,s,t

Motivation

- The Max Flow problem models the situation where commodities need to be transported through a network with limited capacities
- Application to other problems

Application: Matching

- For simplicity we consider the bipartite matching problem
- $G=(L \cup R,E)$ is a bipartite graph
- A matching is a set $M \subseteq E$, in which no two edges share a vertex
- A perfect matching (assuming $|L|=|R|$) has $|L|$ edges
- Edges have weights $W(u,v)$
- A *maximum matching* is a matching with the maximum total edge weight

Flow network for bipartite matching

- $G=(L \cup R,E)$ is a bipartite graph (unweighted)
- Add two extra vertices s, t
  - connect s to all vertices of L, weight 1
  - keep the edges in E (directed from L to R), weight 1
  - connect all vertices in R to t, weight 1
- Then the maximum flow in the new graph is equal to the maximum matching
  - We will prove this later, it is obvious that a matching leads to a flow but not the other way around

Back to flows: remarks

- Having several sources and sinks can be modeled by using extra edges
- Nonadjacent u,v have $C(u,v)=0$
- Notation: For vertex sets X,Y denote $f(X,Y)=\sum_{x\in X, y\in Y} f(x,y)$
- Similarly define $C(X,Y)$
- $C(u,v)\neq C(v,u)$ is possible!

An observation

- $N=(G,C,s,t)$ is a flow network, f a flow. Then
  - For all $X \subseteq V$: $f(X,X)=0$
  - For all $X,Y \subseteq V$: $f(X,Y)=-f(Y,X)$
  - For all X,Y,Z with $X \cap Y=\emptyset$:

    $f(X \cup Y,Z)=f(X,Z)+f(Y,Z)$

The Ford Fulkerson Method

- To start set $f(u,v)=0$ for all pairs u,v
- We will look for augmenting paths. i.e., paths in G along which we can increase f
- Repeat until there is no augmenting path

Augmenting paths

- Setting $C(u,v)=0$ for nonedges allows us to assume that the graph is complete
- Consider *simple* paths from s to t
- Definition: If $C(u,v)-f(u,v)>0$ for all edges on the path then the path is *augmenting*
  - Note: $C(u,v)=0$ and $f(u,v)<0$ possible
- **Definition:** capacity $C(p)$ of a path p is the minimum capacity of any edge on the path

Residual Network

- Given: flow network G,C,s,t, and flow f
- „Remove“ f, to get the residual network
- Formally: Set $C_f(u,v)=C(u,v)-f(u,v)$

- G with capacities C_f is a new flow network
- Note: $C_f(u,v)>C(u,v)$ is possible

Residual Network

Lemma:

- $N=(G,C,s,t)$ flow network, f flow
- $N_f=(G,C_f,s,t)$ the residual network
- f‘ a flow in N_f
- f+f‘ defined by $(f+f')(u,v)=f(u,v)+f'(u,v)$
- Then f+f‘ is a flow in N with value $|f+f'|=|f|+|f'|$

Proof:

- $(f+f')(u,v)=f(u,v)+f'(u,v)$
- Statement about $|f+f'|$ trivial
- Have to check that $f+f'$ is really a flow
  - Skew Symmetry: $(f+f')(u,v)=f(u,v)+f'(u,v)$
    $=-f(v,u)-f'(v,u)=-(f+f')(v,u)$
  - Capacity: $(f+f')(u,v)=f(u,v)+f'(u,v)$
    $\leq f(u,v)+C_f(u,v) = f(u,v)+C(u,v)-f(u,v)=C(u,v)$
  - Flow conservation:
    $\sum_v (f+f')(u,v)=\sum_v f(u,v)+\sum_v f'(u,v) =0+0=0$

Augmenting Paths

- Flow network N and flow f
- Find the residual network N_f
- For a path p from s to t the residual capacity is $C_f(p)=\min\{C_f(u,v): (u,v) \text{ on } p\}$
- Search an augmenting path,
  - i.e. path s to t with $C_f(p)>0$
- Idea: remove edges with capacity 0, perform BFS from s until t is found

Augmenting Paths

- Let p be an augmenting path
- Set $f_p(u,v)$=
  - $C_f(p)$ if (u,v) on p
  - $-C_f(p)$, if (v,u) on p
  - 0 otherwise
- **Claim:** then f_p is a flow in the residual network
- And $f+f_p$ is a flow with value $|f|+|f_p|$ in N
- Proof by checking flow conditions

Ford Fulkerson

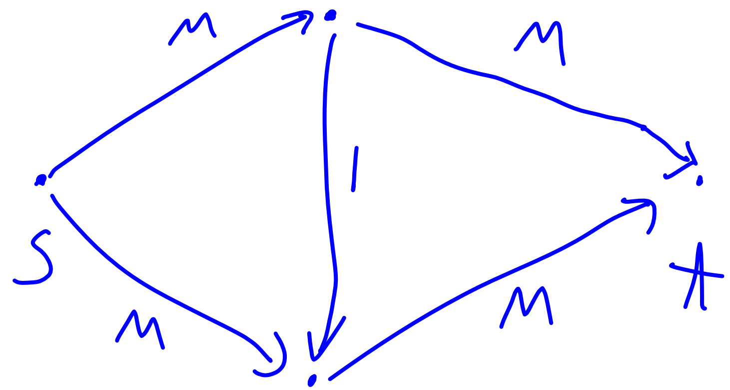
1. Input: flow network N=(G,C,s,t)
2. Set f(u,v)=0 for all pairs u,v
3. While there is an augmenting path p in N_f:
    1. Compute $C_f(p)$
    2. For all edges u,v on p:
        1. set $f(u,v):=f(u,v)+C_f(p)$
        2. set $f(v,u):=-f(u,v)$
4. Output f

Running time

- Computing augmenting paths takes time O(m+n)
- What is the number of iterations?
- Depending on the choice of paths (and the capacities) FF need not terminate at all!
- Running time can be $\Theta(|f|\ m)$ where f is the max flow and capacities are integers
  - For integer weights no more than $O(|f|)$ iterations are needed (augmenting paths add flow 1 at least)
  - Example that $\Omega(|f|)$ iterations can happen

Running time

- Network where 2M iterations can happen:



Improving the running time

- Improvement, choose the augmenting paths with BFS
- **Claim:**
  - If in each iteration an augmenting path in N_f is chosen by BFS then running time is $O(nm^2)$
  - Choosing paths by BFS means choosing augmenting paths with the *smallest number of edges*
- Proof: Later
- Edmonds-Karp algorithm

Correctness of Ford Fulkerson

- Tool: the Max-flow Min-cut Theorem
- This will imply that a flow is maximum iff there is no augmenting path
- Hence the output of Ford Fulkerson is correct

- The theorem is an example of duality/min-max theorems

Min Cuts

- s-t Min Cut problem: input is a flow network
- An s-t cut is a partition of V into L,R with $s \in L$ and $t \in R$
- The capacity of the cut is $C(L,R)=\sum_{u,v} C(u,v)$, where $u \in L$ and $v \in R$
- The output is an s-t cut of minimum capacity

Max-Flow Min-Cut Theorem

- **Theorem**
  - N=(G,C,s,t) flow network, f a flow
  - The following statements are equivalent:
    1. f is a maximum flow
    2. N_f has no augmenting path
    3. |f|=C(L,R) for some s-t cut L,R

Proof

- **Lemma**
  - Let f be a flow and L,R an s-t cut
  - Then f(L,R)=|f|
- Proof:
  - f(L-{s},V)=0 by flow conservation
  - f(L,R)=f(L,V)-f(L,L) [from earlier observation]
  - =f(L,V)
  - =f({s},V)+f(L-{s},V)
  - =f({s},V)
  - =|f|

Proof

- **Lemma**

  $|f| \leq C(L,R)$ for every flow f and every s-t cut L,R
- Proof
  - $|f| = f(L,R) \leq C(L,R)$

- Hence the maximum value of $|f|$ is upper bounded by $C(L,R)$ for any s-t cut L,R

Proof of the theorem

- 1 to 2:
  - Assume f is max but N_f has an augmenting path p, so $f+f_p$ is larger, contradiction
- 2 to 3:
  - Assume N_f has no augmenting path
  - $|f| \leq C(L,R)$ by the Lemma
  - Construct a cut:
    - L: vertices reachable from s in N_f
    - R=V-L
  - $f(L,R)=|f|$
  - All edges in G from L to R satisfy:
    - $f(u,v)=C(u,v)$, otherwise they would be edges in N_f
    - Hence $|f|=f(L,R)=C(L,R)$
- 3 to 1:
  - $|f| \leq C(L,R)$. IF $|f|=C(L,R)$, then f must be maximum flow

Correctness of Ford-Fulkerson

- This proves that Ford Fulkerson computes a maximum flow

Duality

- The value of related maximization and minimization problems coincides
- Similar example: Linear Programming
- Useful to prove bounds:
  - To show that there is no larger flow than f it is enough to point out a cut with small capacity

Running Time

- We assume that augmenting paths are found by BFS
- Then:
  - Number of iterations is at most O(mn)
- Ford-Fulkerson finds maximum flows in time $O(m^2n)$

Number of iterations

- **Lemma:**
  - Let N be a flow network
  - For all $v \neq s,t$:
    - The distance from s to v (number of edges) increases monotonically when changing N to a residual network

Proof

- Assume the distance $\delta(s,v)$ decreases in an iteration
- f is the flow before the iteration, f' afterwards
- $\delta(s,v)$ is the distance in N_f; $\gamma(s,v)$ in $N_{f'}$
- v has min. $\delta(s,v)$ among all v with $\gamma(s,v)<\delta(s,v)$ (*)
- p: path $s\rightarrow u\rightarrow v$ shortest path in $N_{f'}$
  - $\gamma(s,u) = \gamma(s,v)-1$
  - $\delta(s,u) \leq \gamma(s,u)$ (*)
  - (u,v) is no edge in N_f
    - Assume $(u,v)\in N_f$
    - $\delta(s,v)\leq\delta(s,u)+1$
    - $\leq\gamma(s,u)+1$
    - $=\gamma(s,v)$ contradiction

Proof

- (u,v) edge in $N_{f'}$ but no edge in N_f
- In the iteration the flow from v to u is increased
- There is an augmenting path path in N_f with edge (v,u)
- I.e., a shortest path from s to u with edge (v,u) at the end exists in N_f
- Hence:
  - $\delta(s,v)$

    $= \delta(s,u)-1$

    $\leq \gamma(s,u)-1$

    $= \gamma(s,v)-2$
- But we assumed $\gamma(s,v)<\delta(s,v)$

Number of iterations

- **Theorem:**
  - There are at most mn iterations
- Proof:
  - Edges in N_f are *critical*, if the capacity of the augmenting path p is the capacity of (u,v) in N_f
  - Critical edges are removed, i.e. are not in $N_{f'}$
  - Every augmenting path has at least 1 critical edge
  - **Claim:** an edge can be critical at most n/2-1 times
  - Since at most 2m edges are used there can be at most nm iterations

Proof of the claim

- (u,v) critical in N_f for the first time: $\delta(s,v)=\delta(s,u)+1$
- (u,v) vanishes in the iteration
- (u,v) can only reappear if the flow from u to v is reduced
- Then (v,u) is on an augmenting path; f' the flow at this time
- $\gamma(s,u)=\gamma(s,v)+1$
- $\delta(s,v)\leq\gamma(s,v)$
- Hence:
  - $\gamma(s,u)=\gamma(s,v)+1$
  - $\geq\delta(s,v)+1$
  - $=\delta(s,u)+2$
- Distance s to u increases by 2, before (u,v) can be critical
- Distance is integer between 0 and n-1
- (u,v) can be critical at most n/2-1 times.

Conclusion

- The Edmonds-Karp implementation of the Ford-Fulkerson approach computes maximum flows in time $O(m^2n)$