ORIE 633 Network Flows	October 4, 2007
Lecture 10	
Lecturer: David P. Williamson	*Scribe: Kathleen King*

1 Efficient algorithms for max flows

1.1 The Goldberg-Rao algorithm

Recall from last time:

Dinic's Algorithm

> $f \leftarrow 0$
> while $\exists s-t$ path in G_f
>   Compute distances d_i to t in A_f.
>   Compute a blocking flow $\tilde{f}$ on admissible arcs (i.e., $(i,j) \in A_f$ and $d_i = d_j + 1$).
> $f \leftarrow f + \tilde{f}$.

We proved that Dinic's Algorithm finds a max flow efficiently. This proof relied on the following lemma.

Lemma 1 *d_s increases at each iteration.*

We can use routines that find blocking flows in acyclic graphs in $O(mn)$ time, or $O(m \log n)$ time, using fancy data structures. Now, recall that we defined Λ as below:

Definition 1 *Let* $\Lambda = \min\{m^{\frac{1}{2}}, n^{\frac{2}{3}}\}$.

We showed that running a blocking flow algorithm for $O(\Lambda)$ iterations allows us to find a cut in the residual graph with $O(\Lambda)$ arcs across it. For unit capacity graphs, this means that at most $O(\Lambda)$ more units of flow can be sent in the residual graph, meaning that the algorithm will finish in at most $O(\Lambda)$ more iterations.

How can we use this to help us in the case of general capacities?

Idea: Suppose that the arcs from D_k to D_{k-1} all have residual capacity no more than Δ for all k. Then after Λ blocking flows we'll have a cut with residual capacity no more than $\Delta\Lambda$. This seems like it is useful; the amount of remaining flow is reduced significantly with a relatively few blocking flow computations. Then we will reduce Δ and repeat.

How can we ensure that all arcs from D_k to D_{k-1} to have residual capacity at most Δ? We'll need to modify our notion of distance. Let

$$l_{ij} = \begin{cases} 1 & \text{if } u^f_{ij} \le \Delta \\ 0 & \text{otherwise} \end{cases}$$

and define d_i to be the distance from i to t using these lengths l. We then change our definition of admissible arcs.

Definition 2 *An arc is* ***admissible*** *if* $(i, j) \in A_f$ *and* $d_i = d_j + l_{ij}$.

This new idea of lengths causes its own set of problems, however. In particular we have:

Problem 1: In order for the blocking flow style proof to work, we need to have d_s increase in each iteration, and it's not obvious that it will under the new definitions.

Problem 2: The graph of admissible arcs might have cycles (since some l_{ij} may equal 0). This is an issue since the efficient algorithms we know for blocking flows only run on acyclic graphs.

First we'll address Problem 2. "Shrink" (contract) every strongly-connected component of length 0 arcs to a single node and then run the blocking flow algorithm. See Figure 1.

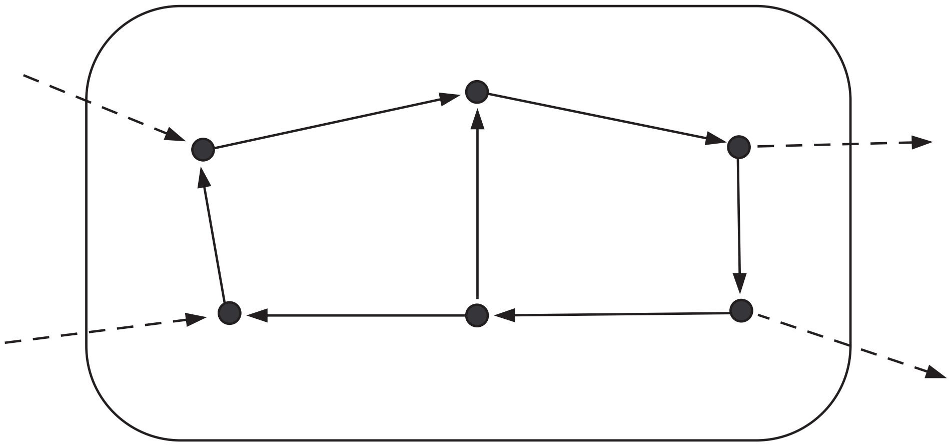

Figure 1: A set of strongly-connected components of admissible arcs to be contracted.

But now we have a new problem. How do we route flow in the "unshrunken" graph, so that if we find a flow in the shrunken graph, we can also find a flow in the unshrunken graph?

It is helpful to us that all arcs in each strongly connected components have capacity at least Δ (since they had length zero). Suppose we limit the flow entering and leaving the component to be at most $\frac{\Delta}{4}$. Here's how we do this:

(i) For each of the unshrunken strongly-connected components, pick some root node, r;

(ii) Build 2 trees: an *intree* to r, and an *outtree* from r (see Figure 2);

(iii) Use the intree to route incoming flow to r and the outtree to route outgoing flow from r.

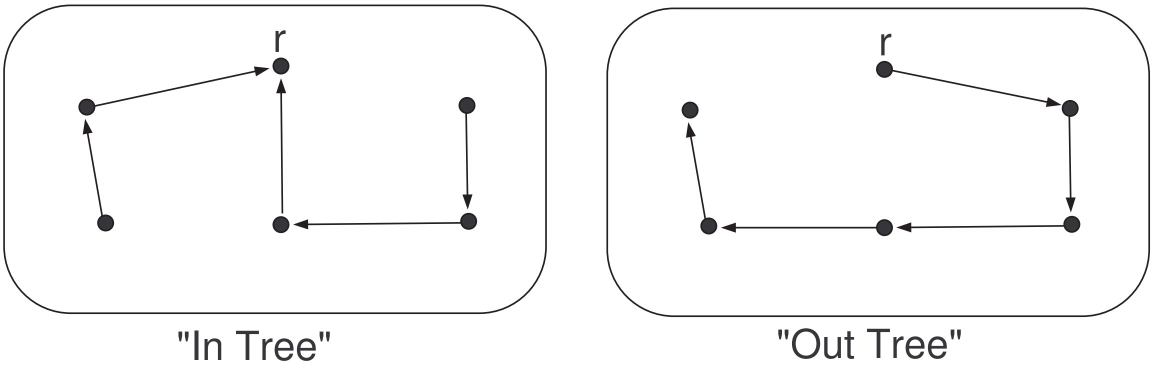

Figure 2: In and out trees.

Each edge is then used at most twice (at most once in the intree and at most once in the outtree) so by routing at most $\frac{\Delta}{4}$ flow on each of these trees, we're only using $\frac{\Delta}{2}$ capacity of the the arcs, which is okay, since each of these has capacity at least Δ.

To limit the flow incoming/outgoing from each shrunken node to be at most $\frac{\Delta}{4}$, we will change the goal of each iteration of the algorithm. We either find a blocking flow, or we find a flow of value $\frac{\Delta}{4}$. In either case, we're making progress. Now we can present the algorithm, in its almost complete state.

[Almost] Goldberg-Rao (1998)

$F \leftarrow mU$ where $U = \max_{(i,j)\in A} u_{ij}$
$f \leftarrow 0$
While $F \geq 1$
  $\Delta \leftarrow \frac{F}{2\Lambda}$
  Repeat 5 Λ times:
    $l_{ij} \leftarrow \begin{cases} 1 & \text{if } u_{ij}^f < \Delta \\ 0 & \text{otherwise} \end{cases} \quad \forall (i,j) \in A_f$
    Compute distances d_i to sink t using lengths l
    Shrink strongly-connected components of admissible arcs
    Find a flow $\tilde{f}$ in the shrunken graph that is either a blocking flow or has value $\frac{\Delta}{4}$
    $\hat{f} \leftarrow \tilde{f}$ with flows routed in the shrunken components;
    $f \leftarrow f + \hat{f}$
  $F \leftarrow \frac{F}{2}$

We now prove a lemma that we will need to bound the running time.

Lemma 2 *F is an upper bound on the maximum flow value in G_f.*

Proof: We prove the statement by induction on the algorithm. First note that it's true initially; $F = mU$ is an upper bound on the total amount of flow.

Now consider the repeat loop. After 5Λ times, either:

1. we find a flow of value $\frac{\Delta}{4}$ at least 4Λ times, or
2. we find a blocking flow at least Λ times.

If (1) is true, then the flow has increased by $4\Lambda\frac{\Delta}{2} = \Lambda\Delta = \frac{F}{2}$. Since we knew before that there was at most F flow in the residual graph, the flow remaining after these iterations is at most $F - \frac{F}{2} = \frac{F}{2}$.

If (2) is true, then, assuming the Lemma to be true, we know $d_s \geq \Lambda$, which implies that there exists a cut in the residual graph of capacity at most $\Lambda\Delta = \frac{F}{2}$.

Thus, in either case we can reduce our estimate of the remaining flow in G_f to $\frac{F}{2}$. □

Running time: What is the running time of the algorithm? We have $\log(mU)$ iterations of the outer loop, because F starts off as mU and is cut in half at each iteration. For each outer loop iteration there are 5Λ iterations. Each inner loop requires $O(m \log n)$ work to find the blocking flow.

Together, this gives a running time of

$$O((\log(mU))(\Lambda)(m \log n)) = O(\min(m^{\frac{1}{2}}, n^{\frac{2}{3}})m(\log(mU))(\log n)) = o(mn).$$

Now, we will show that under the new definitions of distances d given the lengths l, the blocking flow analysis goes through as before; namely, if we compute a blocking flow, then the distance from the source to the sink has strictly increased.

Lemma 3 *If we compute a blocking flow, then d_s strictly increases.*

Proof: Let f' be the current flow and let f be the previous flow. Let l' be the current lengths and let l be the previous lengths. We wish to show that for any shortest s-t path P in A_f that both of the following hold:

1. For all $(i,j) \in P$ we have $d_i \leq d_j + l'_{ij}$; that is, the old distance labeling is still valid.
2. $\exists (i,j) \in P$ such that $d_i < d_j + l'_{ij}$.

If both (1) and (2) are true, then the distance to s in the current iteration is

$$\sum_{(i,j)\in P} l'_{ij} > \sum_{(i,j)\in P} (d_i - d_j) = d_s - d_t = d_s - 0 = d_s.$$

So now we need to prove (1) and (2). We'll start with (1).

First we want to show that $d_i \leq d_j + l_{ij}$. We know that $(i,j) \in A_{f'}$. So either $(i,j) \in A_f$, in which case $d_i \leq d_j + l_{ij}$ by the definition of distances, or $(j,i) \in A_f$ and we pushed flow on it. So (j,i) was admissible and hence $d_j = d_i + l_{ij}$ which means $d_i \leq d_j - l_{ji} \leq d_j + l_{ij}$.

When could we have $d_i \leq d_j + l_{ij}$, but $d_i \not\leq d_j + l'_{ij}$? This could only occur when $l_{ij} > l'_{ij}$, which means $l_{ij} = 1$ and $l'_{ij} = 0$ and $d_i = d_j + l_{ij}$. Note that $l_{ij} = 1$ and $l'_{ij} = 0$ means that the capacity on arc (i, j) has increased. This situation could only occur if we pushed flow on (j, i). But (j, i) was not admissible since $d_j < d_i$. This proves (1).

Now consider (2). For any s-t path P, $\exists (i, j)$ that was not admissible in the previous iteration, by the properties of a blocking flow. Thus either $(i, j) \notin A_f$ or $(i, j) \in A_f$ and $d_i < d_j + l_{ij}$. If $(i, j) \notin A_f$, then for (i, j) to exist in $A_{f'}$, we must have pushed flow on (j, i). So (j, i) was admissible, and following the discussion above, we must have had $d_i \leq d_j - l_{ji}$. How could we then have had $d_i \not< d_j + l_{ij}$? This can only be true if $d_i = d_j$, $l_{ji} = 0$ and $l'_{ij} = 0$. Then we had $u^f_{ij} = 0$ and $u^{f'}_{ij} \geq \Delta$, which implies we pushed more than Δ units of flow, which is a contradiction.

Now suppose that $(i, j) \in A_f$ and $d_i < d_j + l_{ij}$. Given this, how could we have $d_i \not< d_j + l'_{ij}$? It must be that $l'_{ij} = 0$, $l_{ij} = 1$, and $d_i = d_j$. This can only occur if (j, i) is admissible, since $l'_{ij} = 0$ and $l_{ij} = 1$ implies that the residual capacity of (i, j) has increased. We now need to use a trick.

Trick: We're going to change our definitions to make this case go away. We had

$$l_{ij} \leftarrow \begin{cases} 1 & \text{if } u^f_{ij} < \Delta \\ 0 & \text{otherwise} \end{cases}$$

We will define a *special arc* to be an arc (i, j) such that $d_i = d_j$ and $\frac{\Delta}{2} \leq u^f_{ij} < \Delta$ and $u^f_{ji} \geq \Delta$. Set $l_{ij} = 0$ for all special arcs (i, j). Notice that special arcs are admissible, but distances are unchanged. However, since $l_{ij} = 0$, special arcs could be part of shrunken components. But that's okay because $u^f_{ij} > \frac{\Delta}{2}$.

Now we can finish the proof:

Because (i, j) is not admissible, the arc (i, j) is not a special arc, so $u^f_{ij} < \frac{\Delta}{2}$. To make $l'_{ij} = 0$ (i.e., $u^{f'}_{ij} \geq 0$), we must have pushed more than $\frac{\Delta}{2}$ flow on (j, i). This is a contradiction, because we can only push at most $\frac{\Delta}{4}$ flow per iteration. Hence, we must have $d_i < d_j + l'_{ij}$, which proves (2), and this completes the proof. □