COMPSCI 532: Design and Analysis of Algorithms	September 2, 2015
Lecture 3	
Lecturer: Debmalya Panigrahi	*Scribe: Allen Xiao*

1 Overview

In this lecture, we discuss the maximum flow algorithm of Goldberg and Rao [GR98]. This is a capacity scaling algorithm which runs in $O(\Lambda m \log n \log nU)$ time, where $\Lambda = \min\{\sqrt{m}, n^{2/3}\}$. The algorithm itself uses a *binary length function* and applies the results from unit capacity blocking flow. As one might suspect, this is the reason for the similarity to the bounds for unit capacity blocking flow.

For similar reasons to blocking flow, using better data structures and analysis can give us $O(\Lambda m \log \frac{n^2}{m} \log U)$ instead. Goldberg-Rao remains the fastest weakly polynomial algorithm for maximum flow, since its publication in 1998.

2 Goldberg-Rao

We will begin with a description of the Goldberg-Rao algorithm with some terms undefined. We will explain them as we progress. Let F is an upper bound to the amount of residual flow. Let $\overline{f}$ be the residual flow.

Algorithm 1 (Goldberg-Rao 1998)

$f(v,w) \leftarrow 0 \quad \forall (v,w) \in E$
$F \leftarrow nU$
while $F \geq 1$ **do** ▷ (capacity scale)
    $\Delta \leftarrow F/\Lambda$.
    while $\overline{f} > F/2$ **do** ▷ (phases)
        Compute $L(f,\Delta)$, the layered graph for G_f with threshold Δ.
        Find a blocking flow in $L(f,\Delta)$ or a flow of capacity Δ, whichever is smaller.
        Augment f by the flow found.
    end while
    $F \leftarrow F/2$
end while

The number of phases is $O(\log nU)$. Within each phase, we will show that the number of iterations is $O(\Lambda)$.

2.1 Unit Capacity Blocking Flow

Recall the argument we used in unit capacity blocking flow. There, we defined distances from s using the following length function over edges:

$$\ell(v,w) = 1 \qquad \forall (v,w) \in E$$

The length function effectively defines the layered graph, which is why we did not explicitly define lengths before.

Definition 1. *For any length function $\ell()$, let the **volume** of G be:*

$$vol(G) = \sum_{(v,w)\in E} u(v,w)\ell(v,w)$$

Let F be an upper bound on the amount of remaining flow, i.e. the maximum flow in the residual graph. Then, the crucial bound to prove Λ iterations was:

$$\overline{f} \leq \frac{vol(G_f)}{d_\ell(s,t)} = \frac{m}{d_\ell(s,t)}$$

Blocking flows raised $d(s,t)$ by at least 1 each iteration, and sent at least one unit of flow each iteration. For some choice of threshold distance Λ, the iteration count is:

$$\Lambda + \frac{m}{\Lambda}$$

The two values of Λ minimize this sum, and we get $O(\Lambda)$ iterations.

2.2 Iterations for Binary Lengths

The previous bound holds in general for any volume we define.

Lemma 1. *For any length function $\ell(\cdot)$ and arbitrary capacities,*

$$\overline{f} \leq \frac{vol(G_f)}{d_\ell(s,t)}$$

Proof. Consider the flow decomposition of the residual flow into paths P. Let $\overline{f}(p)$ be the residual flow on path $p \in P$.

$$\begin{aligned}
vol(G_f) &= \sum_{e\in E} u(e)\ell(e) \\
&\geq \sum_{e\in E}\sum_{p:e\in p} \overline{f}(p)\ell(e) \\
&= \sum_{p\in P}\sum_{e\in p} \ell(e)\overline{f}(p) \\
&\geq \sum_{p\in P} d_\ell(s,t)\overline{f}(p) \\
&= d_\ell(s,t)\sum_{p\in P}\overline{f}(p) \\
&= d_\ell(s,t)\overline{f}
\end{aligned}$$

Rearranging:

$$\overline{f} \leq \frac{vol(G_f)}{d_\ell(s,t)}$$

□

For some capacity Δ we will choose, define the following length function:

Definition 2. *The **binary length function** used by Goldberg-Rao is:*

$$\ell(v,w) = \begin{cases} 0 & u_f(v,w) \geq 2\Delta \\ 1 & otherwise \end{cases}$$

Contrast to the length function used in blocking flow (Dinitz) and Edmonds-Karp, which was a unit length function. The idea is to make high-capacity edges stay in the same level of the layered graph, and not contribute to any layered cut (length 0). We must be careful with how many edges we give length 0 to, however, since we still want to show that distances increase with blocking flows.

Definition 3. *An edge (v,w) is **admissible** under a length function $\ell(\cdot)$ if $d_\ell(s,w) = d_\ell(s,v) + \ell(v,w)$. Again, we use G_a to denote the admissible subgraph.*

The layered graph under this length function has the property that edges from L_i to L_{i+1} have residual capacity at most 2Δ. The volume in this case is:

$$vol(G_f) \leq 2m\Delta$$

Admissible edges (of length 0) may also be found within layers, which was not the case for the unit length function.

In order to use the blocking flow iteration bound, we we must show that $d_\ell(s,t)$ increases monotonically.

Lemma 2. *Blocking flows under $\ell(\cdot)$ always increase $d_\ell(s,t)$.*

We will prove this in a later section.

Theorem 3. *The number of iterations of each phase is $O(\Lambda)$.*

Proof. Before the residual flow drops below $F/2$:

$$d_\ell(s,t) \leq \frac{2m\Delta}{F/2}$$

With Lemma 2, the number of blocking flows to reach this point is $4m\Delta/F$. Beside blocking flows, we may send flows of value Δ. This we can repeat at most F/Δ times. The total number of iterations is no more than the sum of iterations for these two types of flows.

$$\frac{F}{\Delta} + \frac{4m\Delta}{F}$$

We can minimize this sum by choosing $\Delta = F/\Lambda$:

$$\Lambda + \frac{4m}{\Lambda}$$

For both values of Λ, we get an iteration count of $O(\Lambda)$. □

The two remaining things we have to do for the time bound to hold:

1. Provide an efficient algorithm for finding blocking flows under $\ell(\cdot)$. The unit-length algorithms will not work immediately, since the admissible graph can have cycles from length 0 edges.
2. Prove Lemma 2, showing that $d_\ell(s,t)$ increases with each blocking flow.

2.3 Finding Blocking Flows with a Binary Length Function

Existing blocking flow algorithms require the admissible graph to be acyclic, which may fail due to cycles of length 0 edges. The solution is as follows: we contract strongly connected components of length 0 edges, then run any of the original blocking flow algorithms on the acylic contracted G_a. After all, all graphs are simply directed acyclic graphs on strongly connected components.

Subsequently, we must convert a flow on the contracted graph into a flow on the original graph. We do this by routing the flow through each strongly connected component C:

1. Select an arbitrary $r \in C$ as the *root*.
2. Build two spanning trees over C with r as root: an *in tree* and an *out tree*. The in tree has every edge directed towards r, while the out tree has every edge directed away from r.
3. For any flow which uses the contracted component C, route flow to r using the in tree edges, then route out of r using the out tree edges.

The edges of C are used at most twice in the routing process. Since the flow sent in any iteration is at most Δ, the flow routed through any of the contracted edges is at most 2Δ. Since all length 0 edges have residual capacity over 2Δ, C has enough capacity to route the flow.

In the previous lecture, we showed a DFS-style approach which foudn one in $O(mn)$ time. Blocking flows can be computed in $O(m\log(n^2/m))$ time for unit capacity graphs. This dominates the time of an iteration – recomputing residuals and lengths, contracting/expanding edges can be done in $O(m)$ time each.

2.4 Increasing *s*-*t* Distance

Let $\ell(\cdot)$ and $\ell'(\cdot)$ be the length functions before and after a blocking flow (with flows f and f' respectively). We want to show that any s-t path in f' must be longer than $d_\ell(s,t)$. Let p be a shortest s-t path in f', and consider its embedding in the layered graph under $\ell(v)$.

Lemma 4. *Let $\ell'(p)$ be the length of shortest s-t path p in f'. For every $(v,w) \in p$:*

$$\ell'(v,w) \geq d_\ell(s,w) - d_\ell(s,v) \qquad d_\ell(s,w) \leq d_\ell(s,v)$$

Proof. By definition, this inequality holds for length function $\ell(\cdot)$.

$$\ell(v,w) \geq d_\ell(s,w) - d_\ell(s,v)$$

When $\ell'(v,w) \geq \ell(v,w)$, we are done. Also, when $d_\ell(s,w) \leq d_\ell(s,v)$, we are done. Alternatively, when $\ell'(v,w) < \ell(v,w)$ and $d_\ell(s,w) > d_\ell(s,v)$,

$$\begin{aligned} \ell'(v,w) &= 0 \\ \ell(v,w) &= 1 \end{aligned}$$

$$1 \geq d_\ell(s,w) - d_\ell(s,v) > 0$$

$$d_\ell(s,w) = d_\ell(s,v) + 1$$

So the residual capacity on (v,w) must have increased in f', meaning that we pushed flow on (w,v). However, since $d_\ell(s,w) > d_\ell(s,v)$, (w,v) is not admissible and no flow could have been pushed on it. We conclude that $\ell'(v,w)$ satisfies the inequality. □

Corollary 5. *The sum over p telescopes:*

$$\ell'(p) \geq d_\ell(s,t)$$

Not quite what we need, since we wanted strict inequality. For the proof of Lemma 2, we will show that equality cannot occur:

Proof. Some $(v,w) \in p$ must be inadmissible by the properties of blocking flow. Assume for contradiction that the distance on (v,w) stays preserved, in other words that $\ell'(v,w) = d(s,w) - d(s,v)$. We use a case analysis on $d_\ell(s,v)$ and $d_\ell(s,w)$.

1. $d_\ell(s,w) < d_\ell(s,v)$:

   $\ell(v,w)$ is always nonnegative, so equality cannot hold over (v,w). This case cannot occur.

2. $d_\ell(s,w) > d_\ell(s,v)$:

   By the previous lemma's argument, implies that (w,v) had flow pushed on it but was simultaneously inadmissible. A contradiction.

3. $d_\ell(s,w) = d_\ell(s,v)$:

   Since (v,w) was not admissible, it must be that $\ell(v,w) = 1$. After the blocking flow, (v,w) became admissible: $\ell'(v,w) = 0$. As before, it must be that there was flow along (w,v) and so $\ell(w,v) = 0$. By the length function, $u_f(w,v) \geq 2\Delta$ and $u_f(v,w) < 2\Delta$. The blocking flow routed at most Δ flow, so $u_f(v,w)$ must be no more than Δ less than $u_{f'}(v,w)$. Combined:

   $$\Delta \leq u_f(v,w) < 2\Delta$$
   $$u_f(w,v) \geq 2\Delta$$

The final case does not yet show inequality. Instead, we will modify the length function so that the algorithm avoids this case. The idea will be to give these edges length 0. However, we will have a problem if one of these *special edges* (or its reverse) has $\leq 2\Delta$ residual capacity – we will not be guaranteed enough capacity to route after contraction. To solve this problem, we bump up all the thresholds by $+\Delta$.

Definition 4. *Let a* ***special edge*** (v,w) *be one where:*

1. $d_\ell(s,w) = d_\ell(s,v)$
2. $2\Delta \leq u_f(v,w) < 3\Delta$
3. $u_f(w,v) \geq 3\Delta$

Definition 5. *The true* ***binary length function*** *used in Goldberg-Rao is:*

$$\ell^*(v,w) = \begin{cases} 0 & u_f(v,w) \geq 3\Delta \text{ or } (v,w) \text{ special} \\ 1 & \text{otherwise} \end{cases}$$

Reconsider the last case:

3. $d_\ell(s,w) = d_\ell(s,v)$:

   We encounter the case from before (inadmissible under $\ell(\cdot)$). Be careful here – the length $\ell'(v,w) = 0$ is also under the old length function, not $\ell^*(\cdot)$. The other cases still work the same way (only the $d_\ell(s,w) = d_\ell(s,v)$ one is affected). However, (v,w) is now a special edge, and therefore admissible under $\ell^*(\cdot)$. A contradiction our choice of (v,w).

□

3 Summary

Goldberg-Rao was the first augmentation algorithm to bridge the perceived runtime gap between general capacity ($\tilde{O}(mn)$) and unit capacity graphs ($\tilde{O}(m\Lambda)$).

It is still an open problem to generalize the capacity scaling technique of Goldberg-Rao to minimum-cost flows, a closely related problem. We will introduce minimum cost flows in a later lecture, but at a high level many minimum-cost flow algorithms use a *cost* scaling whose set of admissible edges ("low cost") is not quite compatible with that of Goldberg-Rao ("low capacity").

References

[GR98] Andrew V Goldberg and Satish Rao. Beyond the flow decomposition barrier. *Journal of the ACM (JACM)*, 45(5):783–797, 1998.