Lecture Notes CS:5350 Edmonds and Karp's Heuristics
Lecture 6: Sep 12, 2019
Scribe: Flannery Currin

1 The Flow Decomposition Theorem

Before we can fully analyze the Edmonds and Karp's heuristics, we will need to understand the flow decomposition theorem.

Theorem. *Let f be an arbitrary feasible flow. There are feasible flows* f_1, f_2, ... f_k *and* $s \to t$ *paths* p_1, p_2, ... p_k *such that:*

1. $k \leq m$*, where m is the number of edges in the graph*
2. f_i *assigns positive values only to edges in* p_i
3. $|f| = \sum_{i=1}^{k} |f_i|$

Essentially, we can decompose f into a combination of paths, but there cannot be more of these paths than we have edges.

Example: Consider the arbitrary flow shown in Figure 1.

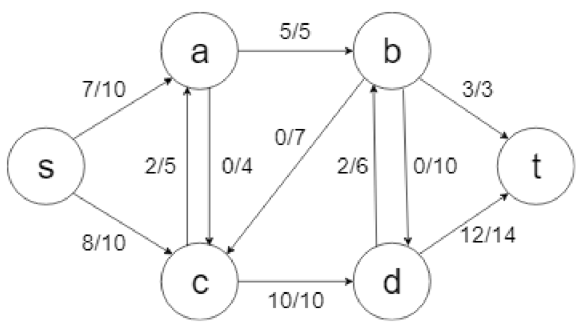

Figure 1: An arbitrary flow to be decomposed.

This flow can be decomposed into the following combination of paths:

- p_1: $s \to a \to b \to t$ (f_1 assigns 3 units to each edge in p_1)
- p_2: $s \to c \to d \to t$ (f_2 assigns 8 units to each edge in p_2)
- p_3: $s \to a \to b \to d \to t$ (f_3 assigns 2 units to each edge in p_3)
- p$_4$: $s \to a \to c \to d \to t$ (f_4 assigns 2 units to each edge in p_4)

For our purposes, we do not need to explore or prove this theorem in detail, but the knowledge that m serves as an upper bound on the number of paths in a decomposed flow leads us to the following corollary which we will use later:

Corollary. *There is a path in G such that every edge on the path has a capacity of at least* $\frac{|f^*|}{m}$.

We know that this corollary is true because if $|f^*|$ is the sum of the flows f_1, f_2, $\dots$ f_k and $k \leq m$, the average flow must be at least $\frac{|f^*|}{m}$, and in order for that to be the average value, there must exist at least one f_i with a value of at least $\frac{|f^*|}{m}$. Having accepted this corollary, we are ready to discuss Edmonds and Karp's heuristics for selecting an augmenting path in the Ford-Fulkerson algorithm.

2 Edmonds and Karp's Heuristics

Provided all capacities are integral, the Ford-Fulkerson algorithm is guaranteed to terminate, but in the worst case it runs in exponential time in the input size. Edmonds and Karp developed two *heuristics* or rules to use when selecting augmenting paths to avoid that worst-case scenario. These heuristics can be used to transform the Ford-Fulkerson algorithm into a polynomial-time algorithm. These heuristics can be summarized by each of the following rules:

1. Choose the fattest augmenting path at each iteration.

2. Choose the shortest augmenting path at each iteration.

In the following sections, we will explore each of these heuristics in more detail.

2.1 The "Fat Pipe" Heuristic

The "fat pipe" or fattest augmenting path heuristic functions as a greedy algorithm. This heuristic picks the augmenting path P with the largest *bottleneck value* in each residual graph. The bottleneck value is the minimum capacity of all the edges in a path – the limiting factor for how much flow can be pushed through that path. By maximizing the the bottleneck value of the augmenting path we pick in each iteration, we maximize the value of F used to augment the flow in that iteration, bringing us closer to the value of f^* in fewer steps than we would need if we chose paths arbitrarily.

2.1.1 Analysis of the "Fat Pipe" Heuristic

Suppose f is the current flow (initially f would have a value of 0). This means that f^r, the flow remaining to be augmented before we can find f^*, has a value of $|f^*| - |f|$. The value of that remaining flow is equal to the value of f', the maximum flow in our residual graph G_f.

Example: Suppose we have the original graph G shown in Figure 2. The value of the max flow in this graph, $|f^*|$, is 40 (found by saturating $s \to a \to t$ and $s \to b \to t$). If we pick the arbitrary path $s \to a \to b \to t$ and push 1 unit of flow through it as our first f (as shown in Figure 3), we will end up with the residual graph G_f shown in Figure 4. The value of the max flow in G_f, f' is 39 (again found by saturating $s \to a \to t$ and $s \to b \to t$). So we see that $|f'| = |f^*| - |f|$.

Let f_0, f_1, f_2, $\dots$ be the sequence of flows produced by the "fat pipe" heuristic. Note that f_0 represents the all-0 initial flow. We will use G_i as shorthand to represent the residual graph for f_i. Let f_i^r represent the max flow in G_i. Our focus is on how this value falls, corresponding to the increase in our calculated f. $G_0 = G$, the original graph, so we know that $|f_0^r| = |f^*|$. Using the flow decomposition theorem, we know f_0^r can be decomposed into m paths or fewer, and the

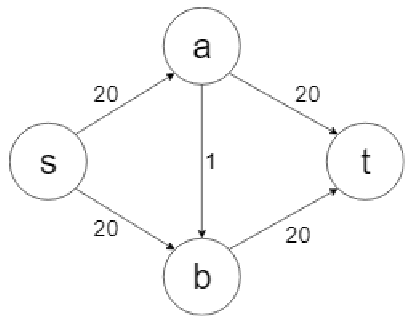

Figure 2: Our original G.

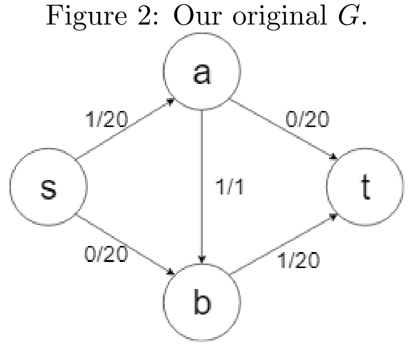

Figure 3: Our original f.

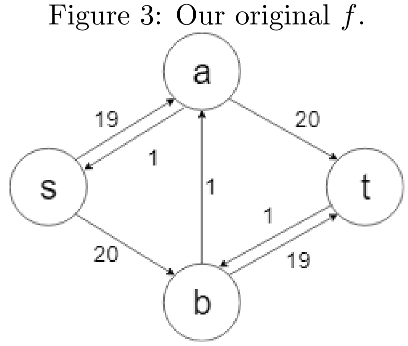

Figure 4: Our G_f.

average flow across all paths in our residual graph must be at least $\frac{|f_0^r|}{m}$. So we can push at least $\frac{|f_0^r|}{m}$ units of flow in the first iteration. This leads us to draw the following conclusion about $|f_1^r|$:

$$|f_1^r| \leq |f_0^r| - \frac{|f_0^r|}{m} = (1 - \frac{1}{m}) \cdot |f_0^r|$$

Expanding this pattern one step further, we see:

$$|f_2^r| \leq (1 - \frac{1}{m}) \cdot |f_1^r| \leq (1 - \frac{1}{m})^2 \cdot |f_0^r$$

Following this pattern, we see that with each iteration, the remaining flow decreases exponentially by a factor of at least $1 - \frac{1}{m}$. So after t iterations, we have:

$$|f_t^r| \leq (1 - \frac{1}{m})^t \cdot |f_0^r| = (1 - \frac{1}{m})^t \cdot |f^*|$$

Using this knowledge, and the knowledge that for all $x \in \mathbb{R}$, $1 + x \leq e^x$, we can see that the maximum flow value in the residual graph will be less than one after $m \cdot \ln|f^*|$ iterations because when $t = m \cdot \ln|f^*|$:

$$|f_t^r| \leq (e^{-\frac{1}{m}})^t \cdot |f^*| < |f^*| \cdot e^{-\ln|f^*|} = 1$$

The only nonnegative, integral flow value less than 1 is 0, which means that $m \cdot \ln|f^*|$ must be an upper bound on the number of iterations it will take to find the maximum flow f^* if we select the fattest augmenting path at each iteration. Using this upper bound, we know that this algorithm runs in $O(m \cdot \log|f^*| \cdot T)$ time, where T is the time spent in each iteration. When we use a Kruskal-like algorithm in reverse order, finding the fattest augmenting path takes $O(m \log n)$ time, where n is the number of vertices in the graph. Taken together, this means the complete algorithm using this heuristic should run in $O(m^2 \cdot \log(m) \cdot \log|f^*|)$ time, which is polynomial in the input size.

2.1.2 Remaining Drawbacks

This heuristic can be used to modify the Ford-Fulkerson algorithm so it runs in polynomial time in the worst case, but only for those cases where all the edge capacities in a graph are integral. In situations where the capacities are arbitrary real numbers, the algorithm still may never halt for inputs with irrational capacities, but will continue to approach the maximum flow.

2.2 The "Fewest Pipes" Heuristic

The "fewest pipes" heuristic allows us to modify the Ford-Fulkerson algorithm such that it will halt within a polynomial-time upper bound in all cases, not just those for graphs with only integral capacities. The goal of this heuristic is to select the augmenting path with the fewest number of edges at every iteration.

2.2.1 Introductory Analysis of the "Fewest Pipes" Heuristic

Theorem. *This heuristic runs in $O(m^2 n)$ time, without any restrictions on capacities.*

To show the termination of this algorithm, we rely on the knowledge that $2m$ is the maximum number of edges in G_f. If $u \to v$ is in the residual graph, and it is saturated, we can get rid of it in the next residual graph. This does not permanently delete the edge – it can reappear when we push flow back in the opposite direction. We will show that an edge can reappear at most $\frac{n}{2}$ times, and we know that with every augmentation, some edge will disappear because the bottleneck edge will be saturated after each iteration. This gives us a budget for the total number of disappearances of $\frac{mn}{2}$. This proof will be continued in the next lecture.