CMPUT 675: Topics in Combinatorics and Optimization **Fall 2016**

Lecture 9 (Sept 26): The Mean Cycle Canceling Algorithm

Lecturer: Zachary Friggstad *Scribe: Jacob Denson*

We begin this lecture by solving the problem of minimal cost bipartite matching, using the techniques we have developed to solve the minimal cost flow problem. After this warmup problem, we move onto the more involved problem of analyzing the asymptotics of the 'minimum mean cycle cancelling algorithm'. With some work, we can establish a very rough bound establishing the polynomial time complexity of the mean cycle cancelling algorithm (a possibly better bound will be established next lecture).

9.1 Minimum-Cost Bipartite Matching

We now take a brief aside from the main topic of this lecture to discuss how minimal cost flows can be used to solve the analogous problem in bipartite graphs. Given the name, the minimal cost bipartite matching problem should be fairly self explanatory. We take a particular bipartite graph $\mathcal{G} = (L \cup R; E)$ with $|L| = |R|$ and edge costs $c : E \to \mathbb{R}_{\geq 0}$. The goal is to find a minimum-cost perfect matching on G in polynomial time. That is, our goal is a perfect matching M which minimizes

$$\text{cost}(M) = \sum_{e \in M} c(e)$$

The similarity to the minimal cost flow problem should immediately suggest a connection between the two problems.

By modifying the standard formulation of the maximum matching problem as a max flow problem, we can reformulate the minimal cost perfect matching problem as a minimal cost max flow problem. The original process constructs a directed bipartite graph $\mathcal{H}$, whose vertices are obtained from $\mathcal{G}$ by adding two new vertices s and t, orienting all edges in E so that they point from L to R, and adding new edges sv, for $v \in L$, and wt, for $w \in R$.

If we define a assign all edges a capacity of 1, then there is a one-to-one correspondence between integral valued $s - t$ flows and matchings in the graph. To see this, note that $f(e) + f(e') \leq 1$ for any e, e' that share an endpoint. So any integral flow has $f(e) \in \{0, 1\}$ and $f(e') \in \{0, 1\}$ and at most one of these is selected. If we set $A_f = \{e \in E : f(e) = 1\}$, we get a matching in $\mathcal{G}$ of size $\text{val}(f)$ (as each unit of flow crosses some directed version of an edge in E). Correspondingly, one can reverse this process easily to convert a matching M on the edges of $\mathcal{G}$ to an integral flow f_M in $\mathcal{G}$ with $\text{val}(f_M) = |M|$.

Now we define a cost function c' on $\mathcal{H}$ in a way which mirrors the cost function c on $\mathcal{G}$. We let

$$c'(sv) = c'(wt) = 0 \qquad c'(vw) = c(vw)$$

Then we see that the correspondence described above preserves the cost function. That is, if a matching M corresponds to a flow f, then $\text{cost}(M) = \text{cost}(f)$, because f is obtained from M by adding edges of the form sv and wt, which add no value to the cost of the flow.

Thus, finding a minimum-cost integral flow f with $\text{val}(f) = |L|$ in $\mathcal{H}$ is equivalent to finding a minimum-cost perfect matching in $\mathcal{G}$. We can find such flow using the successive shortest paths algorithm, and we need only compute $|L| = |R|$ successive paths as each iteration of the algorithm improves the length of the candidate path. This is an algorithm for the minimum-cost perfect matching problem that runs in $O(n(m + n \log n))$ time.

9.2 Minimum Mean Cycle Cancelling

Recall that we let $\mathcal{G} = (V; E)$ be a directed graph with edge costs $c : E \to \mathbb{R}_{\geq 0}$ and capacities $\mu : E \to \mathbb{R}_{\geq 0}$. We want to find a minimum-cost $s - t$ flow of value γ.

We already know that a flow f has minimal cost if $\mathcal{G}_f$ contains no negative-cost cycles. This suggests that a strategy for finding minimal cost flows is obtained from successively augmenting along negative-cost cycles from $\mathcal{G}_f$ until there are none left. As with the maximum flow algorithms, we must be careful which cycles we choose in order to guarantee a good asymptotic speed to our algorithm. It turns out that fast asymptotic results are available if we prune negative cycles with 'minimum mean'.

Algorithm 1 The Minimum Mean Cycle Canceling Algorithm

1: $f \leftarrow$ any (s, t) flow of value γ (found in $O(n^3\sqrt{m})$ time)
2: **while** $\mathcal{G}_f$ contains negative weight cycles **do**
3: $\quad C \leftarrow$ a cycle in $\mathcal{G}$ which minimizes its mean cost $c(C)/|C|$.
4: $\quad$ Augment f along C.
5: **return** f

The algorithm certainly terminates, since the cost of f decreases for every iteration of the algorithm, at a rate bounded by the rational values of the edge costs. It remains to be seen that the algorithm terminates in a polynomial amount of time (and this is not immediately obvious, and is not necessarily true if we do not take cycles which minimize the mean cost).

First, we argue why we can find a minimum ratio cycle C in polynomial time.

Lemma 1 *Let $\mathcal{H} = (W, F)$ be a multigraph[1] with edge costs c where each vertex has equal indegree and outdegree. Then $\mathcal{H}$ contains a cycle C such that*

$$\frac{c(C)}{|C|} \leq \frac{c(F)}{|F|}$$

Proof. Perform a cycle decomposition on $\mathcal{H}$; writing $F = C_1 \cup \cdots \cup C_n$ for cycles C_i where $C_i \cap C_j = \emptyset$ for $i \neq j$ we find

$$\frac{c(F)}{|F|} = \frac{c(C_1) + \cdots + c(C_n)}{|C_1| + \cdots + |C_n|}$$

If $c(C_i)/|C_i| > c(F)/|F|$ for each i, then $c(C_i)|F| > c(F)|C_i|$ also holds for each i, and so, summing up we find

$$c(F)|F| = \sum_i c(C_i)|F| > \sum_i c(F)|C_i| = |F|c(F)$$

a clear contradiction which proves the existence of a C_i satisfying the constraints of the lemma. ■

Given any circuit (i.e. walk that starts and ends at the same location) W, the corresponding multigraph (which contains duplicates of repeated edges in the walk) has an equal number of edges entering and exiting each node, so W can thus be broken up into disjoint cycles.

The lemma above shows that the minimum of $\frac{c(W)}{|W|}$ taken over all closed walks is achieved at a simple cycle.

Theorem 2 *We can find a cycle C with minimum mean cost $\frac{c(C)}{|C|}$ in time $O(mn^2)$.*

[1] Meaning we may have multiple copies of the same edge

Before starting the proof, we note the textbook by Korte and Vygen contains an even faster method running in time $O(mn)$ that uses the "Bellman-Ford" table (mentioned in the proof below) in a more subtle way.

Proof. For $k \leq n$, define $\delta_k(v)$ to be the cheapest closed walk from v to itself using exactly k (not necessarily distinct) edges. If there is no such walk, let $\delta_k(v) = \infty$. If we let $\delta_k(v, w)$ be the cheapest path from v to w using k edges, then we find

$$\delta_k(v, w) = \min_{uw \in E} \; c(v, u) + \delta_{k-1}(u, w)$$

(This is just the Bellman Ford algorithm in disguise). Computing all of these values can be done in $O(mn^2)$ time: for each k and each v the total number of "iterations" of the recurrence for calculating $\delta_k(v, w)$ terms is bounded by m. Let v^* and k^* minimize $\delta_{k^*}(v^*)/k^*$; among all such minimizers choose k^* to be smallest. Then the walk from v^* in k^* steps must be a cycle, from the above remark, and thus is a minimum ratio cycle. ■

Before bounding the number of iterations of the minimum mean cycle cancelling algorithm, let $c_{\max} := \max_{e \in E} c(e)$ and $c_{\min} := \min_{e \in E : c(e) \neq 0} c(e)$ (if all costs are equal to 0, then any flow is a minimum-cost flow so this entire discussion is unecessary). Note that

$$\log_2 \frac{c_{\max}}{c_{\min}}$$

is polynomially bounded by the number of bits required to represent the numerator and denominators of $c_{\max}$ and $c_{\min}$. That is, if $c_{\max} = \frac{a}{b}$ and $c_{\min} = \frac{x}{y}$ with $a, b, c, d \in \mathbb{Z}$ then $\log_2 \frac{c_{\max}}{c_{\min}} \leq \log_2 a + \log_2 y$.

So, the following theorem really shows the number of iterations is bounded by a polynomial in the input size, when we account for the size required for representing all values in the input.

Theorem 3 *The number of iterations of the minimum mean cycle cancelling algorithm is* $O(mn \log_2(n \cdot \frac{c_{\max}}{c_{\min}}))$.

Proof. Consider the sequence of flows

$$f_1, f_2, \ldots, f_N$$

which are found in the MMCC algorithm, together with the cycles

$$C_1, \ldots, C_{N-1}$$

which are augmented to produce the sequence of flows. So f_N is desired the minimum-cost flow.

For $i < j$, define

$$\alpha(i, j) = \{e, \overleftarrow{e} : e, \overleftarrow{e} \in C_i \cup C_j\}.$$

This is the set of possible edges e in the residual graph such that $C_i \cup C_j$ contain both e and its reverse.

Lemma 4 *Let $i < j$ be such that $\alpha(k, j) = \emptyset$ for $i < k < j$. Then*

$$\frac{c(C_i)}{|C_i|} \leq \frac{n}{n - |\alpha(i, j)|} \; \frac{c(C_j)}{|C_j|}$$

Proof. Let $H = (C_i \dot{\cup} C_j) - \alpha(i, j)$. That is, keep multiple copies of edges in the union but exclude edges e where $e, \overleftarrow{e}$ appear between the two cycles.

Then every vertex has equal indegree and outdegree using edges H because this is true in $C_i \dot{\cup} C_j$, and removing edges along with their reverse changes both the in degree and out degree of a vertex by the same amount.

Furthermore, we claim each edge in H is in E_{f_i}. Otherwise, consider some $e \in H$ was in $E_{f_j} - E_{f_i}$. As $e \notin E_{f_i}$ yet it is in E_{f_j}, there must be some $i \leq k < j$ with $\overleftarrow{e} \in C_k$. We also know $e \in C_j$ (as it is in H but not E_{f_i}) so

$e \in \alpha(k,j)$. By assumption, we then have $k = i$, but this contradicts the fact that H is obtained by removing $\alpha(i,j)$ from $C_i \dot{\cup} C_j$.

Therefore, edges of H lie in E_{f_i}. By Lemma 1, this means

$$\frac{c(C_i)}{|C_i|} \leq \frac{c(H)}{|H|}.$$

Therefore,

$$\begin{aligned}\frac{c(C_i)}{|C_i|}(|C_i| + |C_j|) &= \frac{c(C_i)}{|C_i|}(|H| + |\alpha(i,j)|) \\ &\leq c(H) + |\alpha(i,j)|\frac{c(C_i)}{|C_i|} \\ &= c(C_i) + c(C_j) + |\alpha(i,j)|\frac{c(C_i)}{|C_i|}\end{aligned}$$

where we have used the fact that $c(H) = c(C_i) + c(C_j)$ since the cost of reverse edges cancels out. This can be rephrased as

$$\frac{c(C_i)}{|C_i|}\left(|C_j| - |\alpha(i,j)|\right) \leq \frac{c(C_j)}{|C_j|}|C_j|$$

And dividing out, we find

$$\frac{c(C_i)}{|C_i|} \leq \frac{|C_j|}{|C_j| - |\alpha(i,j)|} \cdot \frac{c(C_j)}{|C_j|}$$

and this inequality can be weakened to the form above, because $x/(x-y)$ is a decreasing function of x for $y > 0$, and $n/(n - |\alpha(i,j)|) \geq 0$. ■

Returning to our original proof:

Observation 1: $\frac{c(C_i)}{|C_i|} \leq \frac{c(C_{i+1})}{|C_{i+1}|}$, because $j = i+1$ satisfies the requirements of Lemma 4.

Observation 2: If i, j satisfy the requirements of Lemma 4 and $\alpha(i,j) \neq \emptyset$, then $\frac{c(C_i)}{|C_i|} \leq \frac{n}{n-2} \cdot \frac{c(C_j)}{|C_j|}$. This is because $|\alpha(i,j)|$ must be even as it includes edges along with their reversed counterpart.

Observation 3: For any i with $i + m \leq N - 1$ we must have $\frac{c(C_i)}{|C_i|} \leq \frac{n}{n-2} \cdot \frac{c(C_{i+m})}{|C_{i+m}|}$. To see this, note that each augmentation removes at least one edge from the current residual network (namely, an edge that is critical for the augmenting cycle: its residual capacity us used up). If C_j for $j > i$ uses some edge $e \notin E_{f_i}$ then it must be $\overleftarrow{e} \in C_k$ for some $i \leq k < j$ (in order for e to be introduced to the residual network). Then we would have $k < j$ satisfying the requirements of Lemma 4 and $e, \overleftarrow{e} \in \alpha(k,j)$.

We claim this happens for some $j \leq i + m$, namely that C_j uses an edge not in E_{f_i}. This is because each augmentation removes at least one edge from the residual graph (namely a critical edge: one who's residual capacity is used up). So if $C_i, C_{i+1}, \ldots, C_{i+m-1}$ uses only edges in E_{f_i} then $E_{f_j} \cap E_{f_i} = \emptyset$ so C_{i+m} must use an edge not in E_{f_i}.

To complete this observation, note that for such a pair $i \leq k < j \leq i + m$ we have $\alpha(k,j) \neq \emptyset$. So $\frac{c(C_k}{|C_k|} \leq \frac{n}{n-2} \cdot \frac{c(C_j)}{|C_j|}$ by Observation 2. By Observation 1, this then means $\frac{c(C_i)}{|C_i|} \leq \frac{n}{n-2} \cdot \frac{c(C_{i+m})}{|C_{i+m}|}$.

To complete the proof of the theorem. We iterate Observation 3 n times and see for any i with $i + mn < N$ that

$$\frac{c(C_i)}{|C_i|} \leq \left(1 + \frac{2}{n-2}\right)^n \frac{c(C_{i+nm})}{|C_{i+nm}|} \leq e^2 \frac{c(C_{i+nm})}{|C_{i+nm}|} \leq 2\frac{c(C_{i+nm})}{|C_{i+nm}|}.$$

This follows from, say, the bound $(1 + a/x)^{x+a/2} \geq e^a$ for any $a, x > 0$ (recall the cycles have negative cost).

We know $-c_{\max} \leq \frac{c(C)}{|C|} \leq \frac{-c_{\min}}{n}$ for any negative-cost cycle C. So the number of iterations of the algorithm must be at most $nm \cdot \log_2 \left(n \cdot \frac{c_{\max}}{c_{\min}} \right)$. This finishes our initial analysis of the mean cycle canceling algorithm. ■

The bound for the number of iterations of this algorithm will be analyzed to be $O(m^2 n \log n)$ in the next lecture. Note this is independent of the edge costs. This is potentially a better than the bound shown above especially if costs can be exponentially large compared to n and m.