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

Lecture 7 (Sept. 21): Graph Potentials and Minimum-Cost Flow

Lecturer: Zachary Friggstad *Scribe: Zachary Friggstad*

We discussed graph potentials, which are a short certificate that a directed graph with edge costs has no negative cost cycles. They are also useful for finding minimum-cost paths in graphs with negative edge costs using Dijkstra's algorithm (which normally does not work in this setting).

We then began a discussion of minimum-cost flows.

7.1 Minimum-Cost Paths

First, a brief review of what we may recall about minimum-cost paths from our undergraduate algorithms classes.

Let $\mathcal{G} = (V; E)$ be a directed graph and $c : E \to \mathbb{R}$ **edge costs**. They may be negative. A **negative-cost cycle** is a cycle C in $\mathcal{G}$ whose total edge cost $c(C)$ is strictly negative.

Let s, t be distinct vertices in G. We want to know about the cheapest $s - t$ **path** (doesn't repeat vertices). We distinguish between algorithms that compute **Single-Source Shortest Paths** (SSSP) and **All-Pairs Shortest Paths** (APSP). An SSPS algorithm will compute the cost of a cheapest $s - v$ path from a given s for all $v \in V$. An APSP algorithm will compute the costs of the cheapest paths between all pairs of nodes. Furthermore, enough information will be stored to recover a minimum-cost path in $O(n)$ time between any queried pair of nodes for which the minimum-cost path cost was computed.

- In general this is NP-hard if the graph contains negative-cost cycles (from the Hamiltonian Path problem).
- If there are no negative-cost cycles, the Bellman-Ford computes SSPS in $O(n \cdot m)$ time and the Floyd-Warshall algorithm computes APSP in $O(n^3)$ time.
- If there are no negative-cost edges, Dijkstra's algorithm computes SSPS in $O(m + n \log n)$ time.

The Bellman-Ford algorithm can also be used to detect if the graph has negative-cost cycles in $O(m \cdot n)$ time.

You may have only seen $O(m \log n)$ running time for Dijkstra's in your undergraduate classes, but $O(m+n \log n)$ is possible using more sophisticated data structures. In this class, you may take these running times for granted without reproving them.

7.2 Potentials

We fix a directed graph $\mathcal{G} = (V; E)$ with costs $c : E \to \mathbb{R}$.

Definition 1 *A* **potential** *for* $(\mathcal{G}, c)$ *is a mapping* $\phi : V \to \mathbb{R}$ *such that if we define* $c_\phi : E \to \mathbb{R}$ *by*

$$c_\phi(uv) := c(uv) + \phi(u) - \phi(v)$$

then $c_\phi(uv) \geq 0$ *for each* $uv \in E$. *We call the* c_ϕ*-costs when* ϕ *is a potential the* **potential costs**.

Potential costs are useful when discussing paths and cycles.

Lemma 1 *For any $s, t \in V$ (not necessarily distinct), any $s-t$ walk W, and any potential ϕ for $(\mathcal{G}, c)$ we have $c_\phi(W) = c(W) + \phi(s) - \phi(t)$.*

Proof. Let the nodes of the walk be given (in order) as $s = v_1, v_2, \ldots, v_k = t$. Then

$$c_\phi(W) = \sum_{i=1}^{k-1} c(v_i v_{i+1}) + \phi(v_i) - \phi(v_{i+1}).$$

But each v_i with $1 < i < k$ has $\phi(v_i)$ appearing positively in one term and negatively in the previous term, so the sum telescopes to simply $c(W) + \phi(s) - \phi(t)$. ■

Corollary 1 *For any $s-t$, the set of minimum-cost $s-t$ paths under costs c is the same as the set of minimum-cost $s-t$ paths under costs c_ϕ.*

So if we are given a potential, we can use Dijkstra's algorithm to compute minimum-cost paths in graphs with negative-cost edges $O(m + n\log n)$ time.

When does a graph have a potential, and how can we find one?

Theorem 1 *There is a potential for $(\mathcal{G}, c)$ if and only if $\mathcal{G}$ contains no negative-cost cycles. This can be decided and a potential can be computed (if it exists) in $O(n \cdot m)$ time.*

Proof. If there is a potential ϕ then $c_\phi(C) \geq 0$ for every cycle C because $c_\phi(e) \geq 0$ for each edge e.

Conversely, suppose $\mathcal{G}$ has no negative cost cycle. Construct the auxiliary graph $\mathcal{H} = (V \cup \{r\}; E \cup \{rv : v \in V\})$ where r is new vertex not in V. Keep the same costs c for $e \in E$ and set $c(rv) = 0$ for each $v \in V$. Let $\phi(v)$ be the cost of a cheapest $r-v$ walk in $\mathcal{H}$. Note this is well-defined as $\mathcal{H}$ has no negative cost cycles[1].

Then for each edge $uv \in E$ we have $c(uv) + \phi(u) \geq \phi(v)$, or else we could get a cheaper walk to v by first taking a cheapest walk to u and then using the edge uv. Thus ϕ is a potential for $(\mathcal{G}, c)$.

The Bellman-Ford algorithm applied to $\mathcal{H}$ with start vertex r will compute these ϕ values or indicate that $\mathcal{H}$ (thus $\mathcal{G}$) has a negative-cost cycle. ■

7.3 Minimum-Cost Flows

Throughout this section, we fix the following.

- $\mathcal{G} = (V; E)$ a directed graph
- $c : E\mathbb{R}_{\geq 0}$ nonnegative edge costs
- $\mu : E\mathbb{R}_{\geq 0}$ edge capacities
- s, t distinct vertices in V

[1] This takes a short proof: any walk with repeated vertices contains a cycle, so removing it will not increase the cost. Therefore, for any walk there is a path that is no more expensive, so the cost of a cheapest walk is well-defined as we only have to look at the finite set of all $r-v$ paths.

- $\gamma \geq 0$ a **target flow value**

For any $f : E \to \mathbb{R}$ we define $\text{cost}(f) = \sum_{e \in E} c(e) \cdot f(e)$. It is helpful to view $c(e)$ as the cost of pushing one unit of flow across e.

Definition 2 *Given the above input, the* **minimum-cost flow problem** *is to find an $s-t$ flow f with* $\text{val}(f) = \gamma$ *such that* $\text{cost}(f') \geq \text{cost}(f)$ *for all $s-t$ flows f' with* $\text{val}(f) = \gamma$ *(or determine no such flow f exists).*

In other words, we want to push γ units of flow from s to t and to do it as cheaply as possible.

We know how to efficiently determine if any such flow exists by computing a maximum flow f in polynomial time. If $\text{val}(f) < \gamma$ then no such flow exists. Otherwise, scaling each $f(e)$ by $\frac{\gamma}{\text{val}(f)}$ will produce a flow with value γ. Alternatively, we could create a new source s', attach s' to s with an edge with capacity γ, and simply see if the maximum $s'-t$ flow has value γ or not.

We define the residual network $\mathcal{G}_f = (V; E_f)$ for a flow f and residual capacities $\mu_f : E_f \to \mathbb{R}_{\geq 0}$ as before. The edge costs $c_f : E_f \to \mathbb{R}$ of $\mathcal{G}_f$ are defined as follows: for $e \in E_f$,

$$c_f(e) = \begin{cases} c(e) & \text{if } e \in E \\ -c(\overleftarrow{e}) & \text{if } \overleftarrow{e} \in E \end{cases}$$

7.3.1 Augmenting Cycles

Let f be a flow and C a cycle in $\mathcal{G}_f$. To **augment** f along C is to create a new flow f' in the following way.

Let $\alpha = \min_{e \in C} \mu_f(e)$ and note $\alpha > 0$. For $e \in E$ let

$$f'(e) = \begin{cases} f(e) + \alpha & \text{if } e \in C \\ f(e) - \alpha & \text{if } \overleftarrow{e} \in C \\ f(e) & \text{otherwise} \end{cases}$$

One can verify the following:

- f' is an $s-t$ flow (i.e. it satisfies flow conservation at $v \neq s, t$, is nonnegative, and obeys capacities).
- $\text{val}(f') = \text{val}(f)$
- $\text{cost}(f') = \text{cost}(f) + \alpha \cdot c(C)$

An example of augmenting along a cycle is depicted in Figure 7.1

7.3.2 Certificate of Optimality

Before we present our first algorithm, let us consider how to certify that a given flow is in fact a minimum-cost flow. To be precise, we call a flow f a **minimum-cost flow** if $\text{cost}(f') \geq \text{cost}(f)$ for all flows f' with $\text{val}(f') = \text{val}(f)$.

Theorem 2 *A flow f is a minimum-cost flow if and only if $\mathcal{G}_f$ contains no negative-cost cycles.*

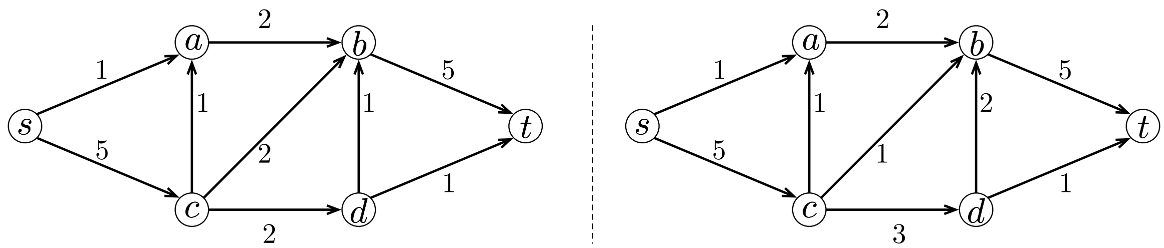

Figure 7.1: Left: a flow (capacities and costs not depicted). If $\mu(cb) = 2, \mu(cd) = 4, \mu(db) = 2$ then the edges cd, db, bc form a cycle in $\mathcal{G}_f$. If $c(cb) = 6, c(cd) = 3, c(db) = 2$ then the cost of this cycle is $3 + 2 - 6 = -1$. We can augment the flow along this cycle by 1 unit (as $\mu_f(db) = 1$ and db is the critical edge). Right: the resulting flow. It's cost is 1 cheaper than the cost of the flow on the left.

Equivalently, by Theorem 1, a flow f is a minimum-cost flow if and only if there is a potential for $(\mathcal{G}_f, c_f)$.

Proof. We just saw that if $\mathcal{G}_f$ has a negative-cost cycle c, then augmenting f along C produces a strictly cheaper flow f' with $\text{val}(f') = \text{val}(f)$.

The other direction will be proven next lecture. ∎