COMPSCI 532: Design and Analysis of Algorithms	September 9, 2015
Lecture 5	
Lecturer: Debmalya Panigrahi	*Scribe: Allen Xiao*

1 Overview

In this lecture, we will introduce the minimum-cost flow/circulation problem and an algorithm based on *cycle canceling*.

2 Minimum-cost Flows

Definition 1. *Given parameter $\alpha \geq 0$, graph $G = (V, E)$ with source s and sink t, plus:*
edge capacities,

$$u : E \to \mathbb{N}_{\geq 0}$$

and skew-symmetric edge costs.

$$c : E \to \mathbb{N}$$

$$c(v,w) = -c(w,v) \qquad \forall (v,w) \in E$$

*The **minimum-cost flow** problem is to find a feasible flow of value α with minimum cost. Flow value and feasibility are defined the same way as maximum flow.*

Definition 2. *Let a **circulation** on a network be a an assignment of flow to edges where balance constraints are met at all vertices (there is neither source nor sink). In other words, every vertex has inflow = outflow.*

$$\sum_{w \in V} f(v,w) = 0 \qquad \forall v \in V$$

*Given a graph $G = (V, E)$ with edge capacities $(u(e))$ and skew-symmetric edge costs $(c(e))$, the **minimum-cost circulation** problem is to find a feasible circulation of minimum cost.*

For both problems, the *cost* of a flow/circulation f is:

$$c(f) = \sum_{e \in E} c(e) f(e)$$

Additionally, the cost of any set of edges $S \subseteq E$ is:

$$c(S) = \sum_{e \in S} c(e)$$

Lemma 1. *The minimum-cost flow and minimum-cost circulation problems are equivalent.*

Proof. We demonstrate reductions from each problem to the other.

1. MCC $\to$ MCF:

   Choose any two vertices as s,t, and $\alpha = 0$. An *s-t* flow of value 0 means balance constraints are met at s and t, so the flow is a valid circulation. Thus, min-cost circulation is a special case of min-cost flow.

2. MCF $\to$ MCC:

   Let C be the maximum cost (this must be nonnegative by skew-symmetry). Our plan is as follows: create an edge (t,s) of capacity α and extremely negative cost. (If (t,s) already exists, a parallel edge is fine.) If we make the cost low enough, the minimum cost circulation must saturate this edge. As a consequence, it must create an *s-t* flow of value α, and the cost of this flow will be minimized. This is our minimum cost flow.

   If $c(t,s) \to -\infty$, then the properties above will hold. We need something finite, however, and instead we will make the following argument: if for *all s-t* paths P and *any* cycle Γ in G without (t,s):

   $$c(t,s) + c(P) < c(\Gamma)$$

   Then it is better to send a unit of flow through the (t,s) cycle, and the minimum will send flow on (t,s) plus P instead. Rearranging, we want to set $c(t,s)$ so that:

   $$c(t,s) < c(\Gamma) - c(P)$$

   Γ has at most n edges, and each edge has cost no less than $-C$. Similarly, P has at most n edges, and each has cost no more than C.

   $$c(\Gamma) - c(P) \geq n(-C) - n(C) > -2n(C+1)$$

   One valid cost is therefore:

   $$c(t,s) = -2n(C+1)$$

□

For the remainder of this lecture, we will focus on the minimum-cost circulation problem. The equivalence makes it possible to transform algorithms for one problem to solve the other.

3 Cycle Canceling (Klein)

The cycle canceling algorithm due to Klein [Kle67] is similar to Ford-Fulkerson, in the sense that it is an underspecified framework whose solutions meet some optimality criteria. Before we describe the (short) algorithm, we give the optimality criteria for minimum-cost circulations.

Theorem 2. *Let f be a circulation. The following are equivalent:*

1. *f is minimum cost.*
2. *There is no directed cycle $\Gamma \in G_f$ with negative cost ($c(\Gamma) < 0$).*

Proof. We prove both directions.

1. (1) $\implies$ (2)

   Let f be a minimum cost circulation. Suppose for contradiction that a negative cost cycle Γ exists in G_f. By flow additivity, we can add any flow over Γ to f and to obtain another feasible circulation.

   $$c(f + f(\Gamma)) = c(f) + c(\Gamma)f(\Gamma) < c(f)$$

   This gives us a circulation with lower cost, contradiction our choice of f.

2. (2) $\implies$ (1)

   We prove the second portion by contrapositive. Let f be a non-minimum cost circulation. Then the optimal circulation f^* has lower cost. $(f^* - f)$ is a feasible circulation in the residual graph (additivity of balance constraints). By flow decomposition, $(f^* - f)$ can be decomposed into a set of cycles in G_f, and augmenting f by these cycles gives f^*. Since $c(f^*) < c(f)$:

   $$c(f^* - f) = c(f^*) - c(f) < 0$$

   The average cost of one of the decomposition cycles is negative, so the minimum cost cycle is also negative. Finally, G_f has a negative cycle.

□

Klein's algorithm is as follows:

Algorithm 1 (Klein 1967)

$f \leftarrow 0$
while G_f has negative cycles **do**
    $\Gamma \leftarrow$ a negative cycle in G_f.
    Augment f by a saturating flow on Γ (*cancel* Γ).
end while

Some questions remain. What kind of procedure can we use to find negative cycles? Will this terminate, and in how many iterations?

3.1 Finding Negative Cycles

One algorithm for finding negative cycles is the Bellman-Ford shortest paths algorithm. The algorithm normally (when no negative cycles are present) finds all source-to-vertex shortest paths. At the end of each of its ith iteration, the algorithm maintains the shortest paths of at most i edges. If any path is updated on the nth iteration, then the algorithm found a cycle with lower cost than any path in the algorithm, so this cycle must be negative. This takes $O(mn)$ time.

A "good" choice of cycle would be the *most negative cycle*. Finding such cycles is NP-hard; we can use reductions from Hamiltonian cycle, longest path, etc. In the next section, we modify the algorithm by choosing cycles which have *minimum-mean cost*. As it turns out, these cycles can be found in polynomial time.

3.2 Running Time

Cycles are at most length n, so augmentations can be done in linear time. The search time for a negative cycle dominates the time for an iteration. Every negative cycle reduces the circulation cost by at least one. For integral costs, the running time is $O(mn \cdot c(f^*))$, where f^* is the minimum cost flow.

4 Minimum-mean Cost Cycle Canceling (Goldberg-Tarjan)

In this section, we will describe a modification of Klein's algorithm which ultimately gives us weakly polynomial runtime. First, we define:

Definition 3. *The **mean-cost** of a cycle Γ is the average cost of its edges: $c(\Gamma)/|\Gamma|$.*

Definition 4. *We use $\mu(f)$ to denote the **minimum mean-cost** of all cycles in G_f.*

$$\mu(f) = \min_{\Gamma \in G_f} \frac{c(\Gamma)}{|\Gamma|}$$

Fact 3. *The smallest uniform increase in edge costs leaving G_f without negative cycles is $-\mu(f)$.*

$$\mu(f) = -\inf \left\{ \Delta \mid G_f \text{ has no negative cycles with costs } c'(e) = c(e) + \Delta \right\}$$

Proof. By definition, every cycle had mean cost at least $\mu(f)$. When we make this price adjustment, the average cost per cycle rises by $-\mu(f)$. The mean cost of every cycle is now:

$$\frac{c'(\Gamma)}{|\Gamma|} = \frac{c(\Gamma)}{|\Gamma|} + (-\mu(f)) \geq \mu(f) + (-\mu(f)) = 0$$

And therefore no cycles are negative. □

The algorithm of Goldberg and Tarjan [GT89] cancels the cycle with the most negative mean-cost.

Algorithm 2 (Goldberg-Tarjan 1989)

```
f ← 0
while G_f has negative cycles do
    Γ ← a minimum mean-cost cycle in G_f.
    Cancel Γ.
end while
```

4.1 Running Time

Still using a variant of Bellman-Ford, one can find the minimum mean-cost cycle in $O(mn)$ time. (Hint: the value of $\mu(f)$ can be found in $O(mn)$ time. The cycle corresponding to $\mu(f)$ can be found in $O(n^2)$ time.) To show a bound on the number of iterations, we will have to introduce some new objects.

Definition 5. *A **price function** (or potential function) $p(v)$ is any real-valued function over vertices.*

Definition 6. *The **reduced cost** of an edge (v,w) under price function $p(\cdot)$ is:*

$$c_p(v,w) = c(v,w) + p(v) - p(w)$$

Fact 4. *The reduced cost of a cycle is equal to its actual cost.*

Proof. The vertex prices telescope.

$$
\begin{aligned}
c_p(\Gamma) &= \sum_{(v,w)\in\Gamma} c(v,w) + p(v) - p(w) \\
&= \sum_{(v,w)\in\Gamma} c(v,w) + \sum_{v\in\Gamma} p(v) - p(v) \\
&= \sum_{(v,w)\in\Gamma} c(v,w) \\
&= c(\Gamma)
\end{aligned}
$$

□

One intuition for reduced cost: the price function is the buy/sell price of a good at each vertex, and the edge costs as the transportation cost of one unit of good. The reduced cost is the cost of buying a unit of good at v, moving it to w, and then selling it at w. A negative reduced cost implies profit from making this maneuver.

If you revisit this problem after learning about linear programming duality, you will find that reduced cost comes naturally out of the dual problem to minimum cost circulation. At any rate, reduced cost gives us the next crucial tool for bounding the number of iterations.

Definition 7. *A circulation f is **ε-optimal** with respect to a price function $p(\cdot)$ if, for all edges (v,w),*

$$c_p(v,w) \geq -\varepsilon$$

We will use $\varepsilon(f)$ to denote the minimum ε (choosing over price functions) for which f is ε-optimal.

Fact 5. *Any circulation f (i.e. the initial one) is C-optimal.*

Lemma 6. *For integer costs, if f is ε-optimal and $\varepsilon < 1/n$, then f is optimal.*

Proof. Suppose $\varepsilon < 1/n$, and that f is ε-optimal. We know that for all edges (v,w):

$$c_p(v,w) > -\frac{1}{n}$$

Now consider any cycle Γ in G_f.

$$c(\Gamma) = c_p(\Gamma) > -|\Gamma|\left(\frac{1}{n}\right) \geq -1$$

Since costs are integral, this means no cycle has negative cost. By Theorem 2, f is optimal. □

ε-optimality will be our measure of “closeness” to optimality. At a high level, we will divide the iterations into phases where f goes from ε-optimal to $(1-1/n)\varepsilon$-optimal. Effectively, we analyze the algorithm as a cost-scaling algorithm.

Minimum mean-cost provides a way to manipulate ε-optimality.

Theorem 7. *For any circulation f:*

$$\mu(f) = -\varepsilon(f)$$

Proof. We prove both directions.

1. $\mu(f) \geq -\varepsilon(f)$

   By definition, for all edges (v,w) and any $p(\cdot)$,

   $$c_p(v,w) \geq -\varepsilon(f)$$

   Consider any cycle Γ which has mean cost $\mu(f)$.

   $$c(\Gamma) = c_p(\Gamma) = \sum_{(v,w)\in\Gamma} c_p(v,w) \geq |\Gamma|(-\varepsilon(f))$$

   By definition of $\mu(f)$:

   $$\mu(f) = \frac{c(\Gamma)}{|\Gamma|} \geq \frac{|\Gamma|(-\varepsilon(f))}{|\Gamma|} = -\varepsilon(f)$$

2. $\mu(f) \leq -\varepsilon(f)$

   We construct a price function p for which f is $(-\mu(f))$-optimal; the minimum ε must be at most this. Define new edges costs:

   $$c'(v,w) = c(v,w) - \mu(f)$$

   By Fact 3, the residual graph has no negative cycles under $c'(\cdot)$. Add a new vertex r with cost 0 edges to all other vertices ($c'(r,v) = 0$). Let price function $p(v)$ be the minimum cost distances from r to v under $c'(\cdot)$. This is well defined, since there are no negative cycles under $c'(\cdot)$. Now for any edge (v,w):

   $$\begin{aligned} p(w) &\leq p(v) + c'(v,w) \\ &= p(v) + c(v,w) - \mu(f) \\ c(v,w) + p(v) - p(w) &\geq \mu(f) \\ c_p(v,w) &\geq \mu(f) \end{aligned}$$

   So f is $(-\mu(f))$-optimal with respect to $p(\cdot)$. It follows that $-\varepsilon(f) \geq \mu(f)$.

□

We will need an additional piece for the next major lemma:

Lemma 8. *$\varepsilon(f)$ never increases from canceling a minimum-mean cost cycle.*

Proof. Let f have minimum-mean cost cycle Γ, and be $\varepsilon(f)$-optimal under $p(\cdot)$. Let the circulation after canceling Γ be f'. By Theorem 7,

$$\mu(f) = \frac{c_p(\Gamma)}{|\Gamma|} = -\varepsilon(f)$$

By ε-optimality,

$$c_p(v,w) \geq -\varepsilon(f) \qquad \forall (v,w) \in \Gamma$$

All edges of Γ have reduced cost bounded below by their average, so it must be that:

$$c_p(v,w) = -\varepsilon(f) \qquad \forall (v,w) \in \Gamma$$

Under $p(\cdot)$, f' has the same reduced costs but may have different edges. The only missing edges would be the saturated edges of Γ. These have strictly negative reduced cost. The only new edges would be the reverse edges of Γ. By antisymmetry:

$$c_p(w,v) = \varepsilon(f) \qquad \forall (v,w) \in \Gamma$$

So no negative reduced cost edges are added to f'.

$$c_p(v,w) \geq -\varepsilon(f) \qquad \forall (v,w) \in G_{f'}$$

Since f' is $\varepsilon(f)$-optimal under $p(\cdot)$:

$$\varepsilon(f') \leq \varepsilon(f)$$

□

Lemma 9. *Starting with circulation f, let f' be the circulation after m minimum-mean cost cycles are canceled,*

$$\varepsilon(f') \leq \left(1 - \frac{1}{n}\right) \varepsilon(f)$$

Proof. Let the flow after i cancellations be $f^{(i)}$, and let the ith cycle canceled be $\Gamma^{(i)}$. Suppose any Γ has all edges with strictly negative reduced cost.

$$c_p(e) < 0 \qquad \forall e \in \Gamma$$

After canceling Γ, we saturate at least one of its edges. Pushing flow on Γ adds edges of positive cost (by cost antisymmetry) and removes at least one edge of negative cost. There are at most m of such Γ before the minimum-mean cost cycle has at least one edge with $c_p(e) \geq 0$.

Formally, for some $k \in \{1, \ldots, m\}$, the minimum-mean cost cycle $\Gamma^{(k)}$ has at least one edge of nonnegative reduced cost. Recall that $\varepsilon(f)$ is non-increasing, so all edges in $f^{(k)}$ are still $\varepsilon(f^{(0)})$-optimal. Examining the value of the corresponding minimum-mean cost $\mu(f^{(k-1)})$:

$$\begin{aligned}
\mu(f^{(k-1)}) &= \frac{c_p(\Gamma^{(k)})}{|\Gamma^{(k)}|} \\
&\geq \frac{(1)(0) + (|\Gamma^{(k)}| - 1)(-\varepsilon(f^{(0)}))}{|\Gamma^{(k)}|} \\
&\geq \frac{0 + (n-1)(-\varepsilon(f^{(0)}))}{n} \\
&\geq -\left(1 - \frac{1}{n}\right) \varepsilon(f^{(0)})
\end{aligned}$$

Applying Theorem 7:

$$\varepsilon(f^{(k-1)}) \leq \left(1 - \frac{1}{n}\right) \varepsilon(f^{(0)})$$

□

Finally, we can prove the iteration bound.

Theorem 10. *The minimum-mean cost cycle canceling algorithm runs for at most $O(mn\log(nC))$ iterations on integer costs.*

Proof. Recall that all f are C-optimal, by using the zero prices $p(v) = 0$.

$$c_p(v,w) = c(v,w) + 0 - 0 \geq -C$$

This follows from direct application of Lemma 9. Let $k = mn\log(nC)$, $f^{(i)}$ defined as before, with $f := f^{(0)}$.

$$\begin{aligned}
\varepsilon(f^{(k)}) &\leq \left(1 - \frac{1}{n}\right)^{n\log(nC)} \varepsilon(f) \\
&\leq e^{-\log(nC)}\varepsilon(f) \\
&\leq \left(\frac{1}{nC}\right) C \\
&\leq \frac{1}{n}
\end{aligned}$$

The second inequality used $(1+x) \leq e^x$ and is strict if $x = (-1/n) \neq 0$. It follows that:

$$\varepsilon(f^{(k)}) < \frac{1}{n}$$

and $f^{(k)}$ must be optimal. □

Corollary 11. *The minimum-mean cost cycle canceling algorithm runs in $O(m^2n^2\log(nC))$ time.*

5 Summary

As it turns out, the Goldberg-Tarjan algorithm can be shown to run in *strongly-polynomial* time. This uses a lemma from Tardos [Tar85], which was the first strongly-polynomial min-cost flow algorithm:

Lemma 12. *Let f be an ε-optimal circulation. If an edge e has $|c_p(e)| \geq 2n\varepsilon$, then $f(e)$ is the same for all ε-optimal circulations.*

One instead shows that after enough iterations, every edge becomes *fixed* this way and $f(e)$ is the same as the optimal circulation. This ultimately gives $O(m^2n\log n)$ iterations. An improved version using dynamic trees and slightly different analysis (also by Goldberg and Tarjan) is known as *Cancel-and-Tighten* and runs in $O(mn\log n\log(nC))$ time. This has a similar strongly-polynomial analysis.

The best weakly-polynomial algorithm is due to Ahuja et al. [AGOT92], and uses scaling of both costs and capacities to run in $O(mn\log\log U\log(nC))$ time. A recent result by Goldberg et al. [GKHT15] shows that the unit-capacity blocking flow analysis applies to unit-capacity min-cost flow, running in $O(\min(\sqrt{m}, n^{2/3}))$ blocking flows per cost scale.

One of the major open questions is to find algorithms for min-cost flow which break the $\tilde{O}(mn)$ runtime barrier. ($\tilde{O}(\cdot)$ means to ignore the logarithmic factors.) Recall that Goldberg-Rao was the one to achieve this in max-flow ($\tilde{O}(m^{3/2})$ or $\tilde{O}(mn^{2/3})$ time).

References

[AGOT92] Ravindra K. Ahuja, Andrew V. Goldberg, James B. Orlin, and Robert Endre Tarjan. Finding minimum-cost flows by double scaling. *Math. Program.*, 53:243–266, 1992.

[GKHT15] Andrew V. Goldberg, Haim Kaplan, Sagi Hed, and Robert Endre Tarjan. Minimum cost flows in graphs with unit capacities. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 406–419, 2015.

[GT89] Andrew V. Goldberg and Robert Endre Tarjan. Finding minimum-cost circulations by canceling negative cycles. *J. ACM*, 36(4):873–886, 1989.

[Kle67] Morton Klein. A primal method for minimal cost flows, with applications to the assignment and transportation problems, 1967.

[Tar85] Éva Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, July 1985.