Chapter 16

Network Flow V - Min-cost flow

CS 573: Algorithms, Fall 2013
October 22, 2013

16.1 Minimum Average Cost Cycle

16.1.0.1 Minimum Average Cost Cycle

(A) \( G = (V, E) \): a digraph, \( n \) vertices, \( m \) edges.
(B) \( \omega : E \rightarrow \mathbb{R} \) weight on the edges.
(C) directed cycle: closed walk \( C = (v_0, v_1, \ldots, v_t) \), where \( v_t = v_0 \) and \( (v_i \rightarrow v_{i+1}) \in E \), for \( i = 0, \ldots, t - 1 \).
(D) average cost of a directed cycle is \( \text{AvgCost}(C) = \frac{\omega(C)}{t} \).
(E) \( d_k(v) \): min length of walk with exactly \( k \) edges, ending at \( v \).
(F) \( d_0(v) = 0 \) and \( d_{k+1}(v) = \min_{e=(u \rightarrow v) \in E} \left( d_k(u) + \omega(e) \right) \).
(G) Compute \( d_i(v) \), for \( \forall i, \forall v \in V \).

In \( O(nm) \) time using dynamic programming.

16.1.0.2 Computing the Min-Average Cost cycle

Cost of minimum average cost cycle is

\[
\text{MinAvgCostCycle}(G) = \min_{C \text{ is a cycle in } G} \frac{\text{AvgCost}(C)}{n - 1} \max_{k=0}^{n-1} \left( \frac{d_n(v) - d_k(v)}{n - k} \right).
\]

Theorem 16.1.1. The minimum average cost of a directed cycle in \( G \) is equal to

\[
\alpha = \min_{v \in V} \max_{k=0}^{n-1} \left( \frac{d_n(v) - d_k(v)}{n - k} \right).
\]

Namely, \( \alpha = \text{MinAvgCostCycle}(G) \).

16.1.0.3 Proof

Proof

(A) Adding \( r \) to weight of every edge increases the average cost of a cycle \( \text{AvgCost}(C) \) by \( r \).
(B) \( \alpha \) also increases by \( r \).
(C) Assume price of min. average cost cycle = 0.
(D) ... all cycles have non-negative (average) cost.
(E) Prove: \( \text{MinAvgCostCycle}(G) = 0 \implies \alpha = 0 \).

(Implies theorem by shifting prices by \( r \)).
16.1.1 Proof continued

16.1.1.1 \( \text{MinAvgCostCycle}(G) = 0 \implies \alpha \geq 0 \)

Proof continued

(A) \( \alpha = \min_{u \in V} \beta(u) \), where \( \beta(u) = \frac{\max_{k=0}^{n-1} d_u(n) - d_u(k)}{n-k} \).

(B) Assume \( \alpha \) realized by vertex \( v \); \( \alpha = \beta(v) \).

(C) \( P_n \): \( n \) edges walk ending at \( v \), of length \( d_n(v) \).

(D) \( P_n \) must contain a cycle.

(E) Break \( P_n \): a cycle \( \pi \) (length \( n-k \)) and path \( \sigma \) (length \( k \)).

(F) \( d_n(v) = \omega(P_n) = \omega(\pi) + \omega(\sigma) \geq \omega(\sigma) \geq d_k(v) \),

16.1.2 Proof continued

16.1.2.1 Continue proving: \( \text{MinAvgCostCycle}(G) = 0 \implies \alpha \geq 0 \)

(A) \( \omega(\pi) \geq 0 \): since \( \pi \) is cycle + by assumption \( \forall \) cycle cost \( \geq 0 \).

(B) \( \implies d_n(v) - d_k(v) \geq 0 \). As such, \( \frac{d_n(v) - d_k(v)}{n-k} \geq 0 \).

Now, \( \alpha = \beta(v) \geq 0 \), by the choice of \( v \).

(C) QED for this direction.

16.1.3 Proof for other direction

16.1.3.1 \( \text{MinAvgCostCycle}(G) = 0 \implies \alpha \leq 0 \):

(A) \( C = (v_0, v_1, \ldots, v_t) \): directed cycle of weight 0.

(B) \( \min_{j=0}^{\infty} d_j(v_0) \) realized by index \( r < n \).

(Otherwise remove non-negative cycles.)

(C) \( \xi \) = walk of length \( r \) ending at \( v_0 \).

(D) \( w \in C \) = walk \( n-r \) edges on \( C \) from \( v_0 \).

(E) \( \tau \) is this walk (i.e., \( |\tau| = n-r \)).

(F) \( d_n(w) \leq \omega(\xi || \tau) = d_r(v_0) + \omega(\tau) \),

(G) \( \rho \) = walk on \( C \) from \( w \) back to \( v_0 \).

(H) \( \tau || \rho \) goes around \( C \) several times.

(I) \( \omega(\tau || \rho) = 0 \), as \( \omega(C) = 0 \).
16.1.4 Proof for other direction

16.1.4.1 MinAvgCostCycle(G) = 0 \implies \alpha \leq 0: continued

(A) For any \( k \): extend \( k \) edges shortest path ending at \( w \) to a path to \( v_0 \) (concatenating \( \rho \))

(B) \( d_k(w) + \omega(\rho) \geq d_{k+|\rho|}(v_0) \geq d_r(v_0) \geq d_n(w) - \omega(\tau) \),

(C) \( \omega(\rho) \geq d_n(w) - \omega(\tau) - d_k(w) \).

(D) \( 0 = \omega(\tau \parallel \rho) = \omega(\rho) + \omega(\tau) \geq (d_n(w) - \omega(\tau) - d_k(w)) + \omega(\tau) = d_n(w) - d_k(w) \)

(E) \( \Rightarrow \beta(w) = \max_{k=0}^{n-1} \frac{d_n(w) - d_k(w)}{n-k} \leq 0. \)

(F) \( \alpha = \min_{v \in V(G)} \beta(v) \leq \beta(w) \leq 0 \)

(G) \( \Rightarrow \alpha = 0. \) QED

16.1.4.2 Computing \( \alpha \):

(A) \( \forall k, \forall v \, d_k(v) \) : longest path with \( k \) edges ending at \( v \). Computed in \( O(nm) \) time.

(B) \( \alpha = \min_{v \in V} \max_{k=0}^{n-1} \frac{d_n(v) - d_k(v)}{n-k} \).

(C) Compute \( \alpha \) in \( O(n^2) \) after \( d_i(\cdot) \) computed.

16.1.4.3 Finding min average cost cycle...

(A) Proved: Minimum avg cost of cycle in \( G \) is \( \alpha = \min_{v \in V} \max_{k=0}^{n-1} \frac{d_n(v) - d_k(v)}{n-k} \).

(B) Compute \( v \) that realizes \( \alpha \).

(C) Add \( -\alpha \) to all the edges in the graph.

(D) Looking for cycle of weight 0.

(E) Recompute \( d_i(\cdot) \) to agree with the new weights of the edges.

(F) For \( v \) above: \( 0 = \alpha = \max_{k=0}^{n-1} \frac{d_n(v) - d_k(v)}{n-k} \)

(G) \( \Rightarrow \forall k \in \{0, \ldots, n-1\} \frac{d_n(v) - d_k(v)}{n-k} \leq 0 \)

(H) \( \Rightarrow \forall k \in \{0, \ldots, n-1\} d_n(v) - d_k(v) \leq 0. \)

(I) \( \Rightarrow \forall i \, d_i(v) \leq d_i(v) \), for all \( i \).

16.1.4.4 Finding min average cost cycle...

(A) Repeat proof of theorem...

(B) \( P_n \) : path with \( n \) edges realizing \( d_n(v) \).

(C) \( P_n = \sigma || \pi \)

\( \sigma : \) a path of length \( k \), \( \pi \) is a cycle.

(D) \( \omega(\pi) \geq 0 \)

(E) \( \omega(\sigma) \geq d_k(v) \)

(F) \( \omega(\pi) = d_n(v) - \omega(\sigma) \leq d_n(v) - d_k(v) \leq 0 \)

(G) \( \pi \) is a cycle and \( \omega(\pi) = 0. \) Done!

(H) Note - the reweighting is not really necessary.

16.1.4.5 Finding min average cost cycle...

Corollary 16.1.2. A direct graph \( G \) with \( n \) vertices and \( m \) edges, and a weight function \( \omega(\cdot) \) on the edges, one can compute the cycle with minimum average cost in \( O(nm) \) time.
16.2 Potentials

16.2.0.6 Shortest path with negative weights...

(A) Dijkstra algorithm works only for graphs with non-negative weights.
(B) If negative weights, then one can use the Bellman-Ford algorithm.
(C) Bellman-Ford is slow... $O(mn)$.
(D) Show how to use Dijkstra algorithm for some cases.

16.2.0.7 Potential

A potential $p(\cdot)$ is a function that assigns a real value to each vertex of $G$, such that if $e = (u \rightarrow v) \in G$ then $w(e) \geq p(v) - p(u)$.

16.2.0.8 Lemma (i)

Lemma 16.2.1. $\exists p(\cdot) \text{ potential for } G \iff G \text{ has no negative cycles (for } w(\cdot)).$

Proof:

$\Rightarrow$: Assume $\exists p(\cdot)$ potential. For any cycle $C$:

$$w(C) = \sum_{e \in C} w(e) \geq \sum_{e \in C} (p(v) - p(u)) = 0.$$ 

$\Leftarrow$: Assume no negative cycle. $p(v)$: shortest walk that ends at $v$.

Claim: $p(v)$ is a potential.

(A) No negative cycles: $p(v)$ is well defined.
(B) $\forall (u \rightarrow v) \in E(G)$: $p(v) \leq p(u) + w(u \rightarrow v)$
(C) $p(v) - p(u) \leq w(u \rightarrow v)$, as required.

16.2.0.9 Lemma (ii)

Lemma 16.2.2. $p(\cdot): \text{potential. } \forall e = (u \rightarrow v) \in E(G): \ell(e) = w(e) - p(v) + p(u)$

(A) $\ell(\cdot)$ is non-negative for all edges.
(B) $\forall s, t \in V(G)$: shortest path $\pi$ of $d_\ell(s, t)$ also s.p. $d_\omega(s, t)$.

Proof:

Proof of (A): $w(e) \geq p(v) - p(u) \implies w(e) - p(v) + p(u) \geq 0$.

Proof of (B): $\forall s - t \text{ path } \pi \in G$: $\ell(\pi) = \sum_{e=(u \rightarrow v) \in \pi} (w(e) - p(v) + p(u)) = w(\pi) + p(s) - p(t)$, 

$\implies d_\ell(s, t) = d_\omega(s, t) + p(s) - p(t)$.

16.2.0.10 Lemma (iii)

Lemma 16.2.3. $G$: graph. $p(\cdot): \text{potential.}$

Compute the shortest path from $s$ to all vertices of $G$ in $O(n \log n + m)$ time, where $G$ has $n$ vertices and $m$ edges.

Proof:

(A) Use Dijkstra algorithm on the distances defined by $\ell(\cdot)$.
(B) The shortest paths are preserved under this distance by Lemma (ii), and this distance function is always positive.
16.3 Minimum cost flow

16.3.0.11 Min cost flow

Input:
\[ G = (V, E): \text{directed graph.} \]
\[ s: \text{source.} \]
\[ t: \text{sink} \]
\[ c(\cdot): \text{capacities on edges,} \]
\[ \phi: \text{Desired amount (value) of flow.} \]
\[ \kappa(\cdot): \text{Cost on the edges.} \]

Definition - cost of flow \( \text{cost of flow} f: \text{cost}(f) = \sum_{e \in E} \kappa(e) * f(e). \)

16.3.0.12 Min cost flow problem

Min-cost flow \( \text{minimum-cost s-t flow problem:} \) compute the flow \( f \) of min cost that has value \( \phi \).

min-cost circulation problem Instead of \( \phi \) we have lower-bound \( \ell(\cdot) \) on edges.
(All flow that enters must leave.)

Claim 16.3.1. If we can solve min-cost circulation \( \implies \) can solve min-cost flow.

HERE: All demands on vertices are zero!

16.3.0.13 Residual graph...

The \textit{residual graph} of \( f \) is the graph \( G_f = (V, E_f) \) where

\[ E_f = \left\{ e = (u \rightarrow v) \in V \times V \middle| \begin{array}{l}
        \text{f}(e) < \text{c}(e) \\
        \text{or f}(e^{-1}) > \ell(e^{-1})
    \end{array} \right\}. \]

where \( e^{-1} = (v \rightarrow u) \) if \( e = (u \rightarrow v) \).

Assumption 16.3.2. \( \forall u, v \quad (u \rightarrow v) \in E(G) \implies (v \rightarrow u) \notin E(G). \)

Cost function is anti-symmetric:
\[ \forall (u \rightarrow v) \in E_f \quad \kappa((u \rightarrow v)) = -\kappa((v \rightarrow u)). \]

16.3.0.14 Some definitions

Definition 16.3.3. \textit{Cycle sign} Directed cycle \( C \) in \( G_f \).

\[ e = (u \rightarrow v) \in E(G): \chi_C(e) = \begin{cases} 
1 & e \in C \\
-1 & e^{-1} = (v \rightarrow u) \in C \\
0 & \text{otherwise};
\end{cases} \]

Pay 1 if \( e \) is in \( C \) and \(-1 \) if we travel \( e \) in the “wrong” direction.

Definition 16.3.4. \textit{Cycle cost} The \textit{cost} of a directed cycle \( C \) in \( G_f \) is
\[ \kappa(C) = \sum_{e \in C} \kappa(e). \]
16.3.0.15 Even more definitions

(A) Circulation comply with capacity and lower-bounds constraints is **valid**.
(B) flow function that only comply with conservation property is a **weak circulation**.
(C) Weak circulation might violate capacity and lower bounds.
(D) Weak circulation might not be a valid circulation.

16.3.0.16 Another lemma

**Lemma 16.3.5.** \( f, g: \) two valid circulations in \( G = (V, E) \). Let \( h = g - f \).

(A) \( h \) is a weak circulation,
(B) if \( h(u \to v) > 0 \) then \( (u \to v) \in G_f \).

Proof...

(A) \( h \) is clearly a weak circulation (conservation of flow - verify).
(B) If \( h(u \to v) \) is negative, then \( h(v \to u) = -h(u \to v) \).
(C) For \( e = (u \to v) \), \( h(u \to v) > 0 \):
   (i) If \( e = (u \to v) \in E \), and \( f(e) < c(e) \implies e \in G_f \).
      If \( f(e) = c(e) \implies h(e) = g(e) - f(e) \leq 0 \). Contradicts \( h(u \to v) > 0 \).

16.3.0.17 Proof continued...

Proof continued: For \( e = (u \to v) \), \( h(u \to v) > 0 \), and \( e = (u \to v) \notin E \):

(A) \( \implies e^{-1} = (v \to u) \in E \). Otherwise \( h(u \to v) = 0 \).
(B) \( 0 > h (e^{-1}) = g (e^{-1}) - f (e^{-1}) \).
(C) \( \implies f (e^{-1}) > g (e^{-1}) \geq \ell (e^{-1}) \).
(D) Flow by \( f \) on \( e^{-1} \) larger than lower bound.
(E) Can return this flow in the other direction.
(F) \( \implies e \in G_f \).
Bibliography


