Shortest Dipath Problem

CO 351: Network Flow Theory

David Duan, 2019 Fall

Contents

1.6.2 Currency Exchange

1 Shortest Dipath Problem

1.1 Overview

1.1.1 Problem

Given a dipath $D = (N, A)$, arc costs $w \in \mathbb{R}^A$, two distinct nodes $s, t \in N$, we wish to find a minimum cost s, t-dipath.

1.1.2 LP Formulation

We can formulate this as a transshipment problem, where $b_s = -1$ (supply node), $b_t = 1$ (demand node), and $b_v = 0$ for all $v \in N \setminus \{s, t\}$.

$$\begin{aligned} \min \quad & w^T x \\ s.t. \quad & x(\delta(\bar{v})) - x(\delta(v)) = \begin{cases} -1 & v = s \\ 1 & v = t \\ 0 & \text{else} \end{cases} \\ & x \geq \mathbf{0} \end{aligned}$$

Note that solutions to this LP are not guaranteed to be an s, t-dipath (they are s, t-diwalks). However, if D has no negative dicycles, then finding a shortest s, t-dipath is equivalent to finding a shortest s, t-diwalk.

1.1.3 Dual LP

The dual LP is identical to the one for TP.

$$\begin{aligned} \max \quad & y_t - y_s \\ s.t. \quad & y_v - y_u \leq w_{uv} \quad \forall uv \in A \end{aligned}$$

As a remark, feasible potentials correspond to feasible solutions to the dual of the linear programming relaxation of the shortest path problem.

1.1.4 Characteristic Vector of a Path

Let P be an s, t-dipath. We can represent it with $x^P \in \mathbb{R}^A$, called the *characteristic vector* of P, where

$$x_e^P = \begin{cases} 1 & e \in A(P), \\ 0 & e \notin A(P). \end{cases}$$

In other words, $x_e^P = 1$ if and only if e is an arc in P.

Note that:

1. $u \notin N(P) \implies x^P(\delta(u)) = x^P(\delta(\bar{u})) = 0$, as it is not used at all.
2. $u \in N(P) \setminus \{t\} \implies x^P(\delta(u)) = 1$, as exactly one arc is leaving u (except t).

3. $u \in N(P) \setminus \{s\} \implies x^P(\delta(\bar{u})) = 1$, as exactly one arc is entering u (except s).

Moreover, $w(P) = wx^P$, where wx^P denotes the scalar product $\sum_{a \in A} w_a x_a^P$ and is an integer. Thus, every s, t-dipath P correspond to a (integral) feasible solution x^P of (P) of values $w(P)$.

However, not all integral feasible solutions to (P) correspond to s, t-dipath. Each feasible solution to (P) corresponds to an s, t-diwalk.

Theorem If $\bar{x}$ is an integral feasible solution to LP, then $\bar{x}$ is a sum of the characteristic vector of an s, t-dipath and a collection of dicycles.

Proof.

Consider the set of active arcs $F = \{e \in A : \bar{x}_e > 0\}$ in $\bar{x}$.

We first show the existence of an s, t-dipath.

Let $\delta(S)$ be an s, t-cut. The net flow of S is -1 (because the net flow of $s \in S$ is -1 and the net flow of all other nodes in $\delta(S)$ are zero by construction), so there must be at least one arc in $\delta(S)$ with non-zero flow (i.e., it is in F). Since this holds true for every s, t-cut, there exists an s, t-dipath P using arcs of F.

We now show the (possible) existence of a collection of dicycles.

Consider the flow obtained by removing the characteristic vector of P from the integral feasible solution: $x' := \bar{x} - x^P$. Since $\bar{x}, x^P$ both satisfy the flow constraints, we get $x'(\delta(\bar{v})) - x'(\delta(v)) = 0$ for all $v \in N$.

Consider the set of active arcs $F' := \{e \in A : x'_e > 0\}$ in x'.

If $F' = \varnothing$, then we are done, as $\bar{x}$ was an s, t-dipath.

Suppose $F' \neq \varnothing$. Take a longest dipath $v_1, \ldots, v_k$ in F'. Since $x'(\delta(\bar{v})) - x'(\delta(v)) = 0$, there is an arc $v_k v_i$ for some $i < k$. Moreover, v_i cannot be outside of the path since we took a longest dipath. This forms a dicycle C $v_i, v_{i+1}, \ldots, v_k, v_i$.

Removing this cycle from flow x', we get $x'' := x' - x^C$, which also satisfies $x''(\delta(\bar{v})) - x''(\delta(v)) = 0$ for all $v \in N$ and the sum of all flows have decreased by at least 1. By induction, we are done. □

Let $\bar{x}$ be an optimal integral solution to our LP. If $\bar{x}$ is the characteristic vector of an s, t-dipath, then we are done. Else, $\bar{x} = x^P + x^{C_1} + \cdots + x^{C_k}$ where P is an s, t-dipath and C_i's are dicycles.

If there are no negative dicycles, then $\bar{w}^T \bar{x} \geq w^T x^P$, so x^P is an optimal solution in the form of an s, t-path. Otherwise, the LP is unbounded. This can be summarized into the following corollary.

Corollary If there are no negative dicycles, then our LP formulation has an optimal solution that is the characteristic vector of an s, t-dipath.

1.2 Potentials and Optimality Conditions

Recall from TP,

1. The dual solution $y \in \mathbb{R}^N$ is called a *node potential*.
2. Given a node potential y, the *reduced cost* of arc uv is $\bar{w}_{uv} = w_{uv} + y_u - y_v$.
3. A node potential is *feasible* if $\bar{w}_{uv} \geq 0$ for all $uv \in A$.

Given a node potential y, if an arc uv satisfies $\bar{w}_{uv} = w_{uv} + y_u - y_v = 0$, we call it an *equality arc*.

Note that for any dicycle C, $\overline{w}(C) = w(C)$.

Proposition (1) Let $D = (N, A)$ be a digraph with weights $w \in \mathbb{R}^A$ with no negative dicycle. Let Q be an s,t-diwalk with $s \neq t$. Then there exists an s,t-dipath P with $w(P) \leq w(Q)$.

Proof. By theorem, each Q can be decomposed into an s,t-dipath and a collection of dicycles $C_1, \ldots, C_r$ and $w(Q) = w(P) + \sum_{k=1}^{r} w(C_k)$. Since $w(C_k) \geq 0$ for all $k = 1, \ldots, r$, we have $w(P) \leq w(Q)$. □

Proposition (2) Let $D = (N, A)$ be a digraph with weights $w \in \mathbb{R}^A$ with no negative dicycle. Let P be a shortest s,t-dipath and Q be a shortest s,t-diwalk. Then $w(P) = w(Q)$.

Proof. Since s,t-dipath is an s,t-diwalk, $w(Q) \leq w(P)$. By proposition (1), there exists an s,t-dipath P' such that $w(P') \leq w(Q)$. Then $w(P) \leq w(P') \leq w(Q) \implies w(P) = w(Q)$. □

Lemma (3) Let $D = (N, A)$ be a digraph with weights $w \in \mathbb{R}^A$ and feasible potentials $y \in \mathbb{R}^N$. Let Q be an s,t-diwalk. Then $w(Q) \geq y_t - y_s$. Moreover, $w(Q) = y_t - y_s$ iff every arc of Q is an equality arc.

Proof. Suppose $Q = v_1, v_2, \ldots, v_k$ where $s = v_1$ and $t = v_k$. Since y is feasible, $y_{v_{i+1}} \leq y_{v_i} + w_{v_i v_{i+1}}$ for all $i = 1, \ldots, k-1$. Adding them up, we get $y_{v_2} + \cdots + y_{v_k} \leq y_{v_1} + \cdots + y_{v_{k-1}} + w_{v_1 v_2} + \cdots + w_{v_{k-1} v_k}$. Subtracting $y_{v_1} + \cdots + y_{v_{k-1}}$ from both sides, we get $y_{v_k} - y_{v_1} = y_t - y_s \leq w(Q)$. For the second part, observe that $y_t - y_s = w(Q)$ iff $y_{v_{i+1}} = y_{v_i} + w_{v_i v_{i+1}}$ holds for all $i = 1, \ldots, k-1$. □

Let $D = (N, A)$ be a digraph with weights $w \in \mathbb{R}^A$. By lemma (3), every s,t-diwalk has length/cost at least $y_t - y_s$, so to prove that Q is a shortest s,t-diwalk, it suffices to show that $w(Q) = y_t - y_s$, or equivalently, every arc of Q is an equality arc. Hence, we get the following theorem:

Theorem (4) Let $D = (N, A)$ be a digraph with weight $w \in \mathbb{R}^A$. An s,t-dipath P is shortest if there exists feasible potentials y such that all arcs of P are equality arcs.

Lemma (5) Let $D = (N, A)$ be a digraph with weight $w \in \mathbb{R}^A$. If D has a negative dicycle then D has no feasible potentials.

Proof. Let $y \in \mathbb{R}^N$ be feasible potentials and $C = v_1 v_2, \ldots, v_{k-1} v_k, v_k v_1$ be a dicycle. Now C is a v_1, v_1-diwalk, so we must have $w(C) \geq y_{v_1} - y_{v_1} = 0$. Hence, D has no negative dicycle. □

Lemma (6) Let $D = (N, A)$ be a digraph with weight $w \in \mathbb{R}^A$ and suppose all nodes can be reached from $s \in N$. For every $v \in N$, let y_v be the length of the shortest s, v-dipath in D. If D has no negative dicycles, then y are feasible potentials.

Proof. Suppose y are not feasible potentials. Then there exists $uv \in A$ such that $y_u + w_{uv} < y_v$. Let P be a shortest s, u-dipath in D. By definition of y, $w(P) = y_u$. Let Q be the s, v-diwalk obtained by adding uv to the end of P. It follow from proposition (1) that Q contains an s, v-dipath Q' where $w(Q') \leq w(Q)$. But $w(Q) = w(P) + w_{uv} = y_u + w_{uv} < y_v$, so y_v is not the length of the shortest s, v-dipath in D, a contradiction. □

Theorem (7) Let $D = (N, A)$ be a dipath with weights $w \in \mathbb{R}^A$. Then there exists a feasible potential iff there are no negative dicycles in D.

Proof. If D has a negative dicycle, by Lemma (5), there are no feasible potentials. Suppose D has no negative dicycle. Construct D' by adding a new node z and arcs from z to all nodes of D with weight zero. Observe that every node of D' is reachable from z and D' has no negative dicycles. It follows from Lemma (6) that y (obtained by computing the length of the shortest s, v-paths in D') is a feasible potential for D'. Then it is a feasible potential for D as well. □

Theorem (8) Let $D = (N, A)$ be a digraph with weight $w \in \mathbb{R}^A$ and suppose all nodes can be reached from $s \in N$. Suppose D has no negative dicycles. If P is a shortest s, t-dipath, then there exists a feasible potential $y \in \mathbb{R}^N$ such that every arc of P is an equality arc.

Proof. Consider the feasible potential defined in Lemma (6). Then $w(P) = y_t = y_t - 0 = y_t - y_s$ and Lemma (3) implies that every arc of P is an equality arc. □

1.3 Ford's Algorithm

1.3.1 Sub-Paths Optimality

Theorem Let $D = (N, A)$ be a digraph with arc cost $w \in \mathbb{R}^A$ with no negative dicycle. If $v_1, \ldots, v_k$ is a shortest v_1, v_k-dipath, then $v_1, \ldots, v_i$ is a shortest v_1, v_i-dipath.

Proof. Since there are no negative dicycles, the LP has an optimal integral solution corresponding to a characteristic vector of an v_1, v_k-dipath. Then there is a optimal dual solution y where all arcs of P are equality arcs by CS conditions.

Let $P' := v_1, \ldots, v_i$. Then y is still feasible for the dual LP of the shortest v_1, v_i-dipath problem. Any arc in P' is also in P, so all the arcs of P' are equality arcs. Thus the CS conditions are satisfied for v_1, v_i-dipath problem. It follows that P' is an optimal solution. □

1.3.2 Rooted Trees

A tree T is *rooted* at s, if for all $t \in N(T)$, the unique s, t-path in T is an s, t-dipath.

Let $D = (N, A)$ and $s \in N$.

1. There is an s,t-dipath in D for all $t \in N$ iff there exists a spanning tree in D rooted at s.
2. Let T be a spanning tree in D. Then T is rooted at s iff $|\delta(\bar{s})| = 0$ and $|\delta(\bar{v})| = 1$ for $v \in N(T) \setminus \{s\}$.

In a spanning tree T of D rooted at s, for each node v other than s, its *predecessor*, denoted p_v, is the unique node u such that uv is an arc in T.

1.3.3 Ford's Algorithm

Ford's algorithm allows us to find shortest s,v-dipath for all $v \in N$ in one go.

We assume that every node can be reached from s via a dipath. The algorithm tries to produce a feasible potential ($i.e., \forall e : \bar{w}_e \geq 0$) and a rooted spanning tree T at s, so that the arcs of T are all equality arcs.

At each step, we keep track of the potential and predecessor of each node.

Algorithm

1. Initialization.
   a. Set $y_s = 0$ and $y_v = \infty$ for all $v \in N \setminus \{s\}$.
   b. Set predecessor $p_v = \texttt{UNDEF}$ for all $v \in N$.
2. Correction. While y is not feasible, i.e., there exists an arc with negative reduced cost,
   a. Find $uv \in A$ where $\bar{w}_{uv} := w_{uv} + y_u - y_v < 0$.
   b. Set $y_v = y_u + w_{uv}$ (which makes uv an equality arc as $\bar{w}_{uv} = 0$) and $p_v = u$.

An important observation is that y_v never increases: by setting $y_v = y_u + w_{uv}$ where $y_v > y_u + w_{uv}$ to begin with, we are decreasing y_v.

1.3.4 Predecessor Digraph

At any point in the algorithm, the *predecessor digraph*, denoted D_p, is one where $N(D_p) := N$ and $A(D_p) := \{p_v v : v \in N\}$.

Proposition Through the algorithm, $\bar{w}_e \leq 0$ for all arcs $e \in A(D_p)$.

Proof. Let $v \in N$ be arbitrary and p_v be its predecessor. When a correction takes place with an arc whose head is v, $\overline{w}_{p_v v} = 0$. Until the predecessor of v is changed again, the reduced cost stays non-positive, and only y_{p_v} can change (due to connecting other arcs). By observation above, y_{p_v} can only decrease, so $\overline{w}_{p_v v} = w_{p_v v} - y_v + y_{p_v}$ only decreases. $\square$

Lemma Let $D = (N, A)$ be a digraph with weights $w \in \mathbb{R}^A$ and feasible potentials $y \in \mathbb{R}^N$. Let Q be an s,t-diwalk. Then $w(Q) \geq y_t - y_s$. Moreover, $w(Q) = y_t - y_s$ iff every arc of Q is an equality arc.

Proof. Let $Q = v_1, \ldots, v_k$ where $s = v_1$ and $t = v_k$. Since y is feasible, $y_{v_{i+1}} \leq y_{v_i} + w_{v_i v_{i+1}}$ for all $i = 1, \ldots, k-1$. Adding them up, we get $y_{v_2} + \cdots + y_{v_k} \leq y_{v_1} + \cdots + y_{v_{k-1}} + w_{v_1 v_2} + \cdots + w_{v_{k-1} v_k}$. Subtracting $y_{v_1} + \cdots + y_{v_{k-1}}$ from both sides, we get $y_{v_k} - y_{v_1} = y_t - y_s \leq w(Q)$. For the second part, observe that $y_t - y_s = w(Q)$ iff $y_{v_{i+1}} = y_{v_i} + w_{v_i v_{i+1}}$ holds for all $i = 1, \ldots, k-1$. □

Lemma Let $D = (N, A)$ be a digraph with weight $w \in \mathbb{R}^A$. If D has a negative dicycle then D has no feasible potentials.

Proof. Let $y \in \mathbb{R}^N$ be feasible potentials and $C = v_1 v_2, \ldots, v_{k-1} v_k, v_k v_1$ be a dicycle. Now C is a v_1, v_1-diwalk, so we must have $w(C) \geq y_{v_1} - y_{v_1} = 0$. Hence, D has no negative dicycle. □

Proposition If D_p contains a dicycle (at any point in the algorithm), then D contains a negative dicycle and the algorithm does not terminate.

Proof. Suppose we produce a dicycle $C := (v = v_1), v_2, \ldots, (v_k = u), v$ in D_p by connecting the arc uv. Then it must be true that in the previous iteration, $\overline{w_{uv}} = w_{uv} + y_u - y_v < 0$. By proposition above, $\overline{w}_e \leq 0$ for all $e \in A(D_p)$ throughout the algorithm, so
$\overline{w}_{v_i v_{i+1}} = w_{v_i v_{i+1}} + y_{v_i} - y_{v_{i+1}} \leq 0$ for $1 \leq i < k$. Since C is a dicycle, adding up these inequalities cancel out y's and we are left with

$$\sum_{i=1}^{k-1} w_{v_i v_{i+1}} + w_{v_k v_1} < 0,$$

i.e., C is a negative dicycle. It follows from previous lemma that there cannot be a feasible potential, so the algorithm never terminates. □

Proposition If s has a predecessor (at any point in the algorithm), then D contains a negative dicycle and the algorithm does not terminate.

Proof. Exercise.

1.3.5 Termination of Ford's Algorithm

Proposition Suppose the algorithm terminates. Then D_p is a spanning tree of shortest dipaths rooted at s. Furthermore, y_v is the cost of a shortest s, v-dipath.

Proof. Since the algorithm terminates, D_p cannot contain a cycle and s does not have a predecessor. So D_p is a rooted spanning tree. Since all nodes other than s has a predecessor, $|\delta(\bar{v})| = 1$ for all $v \in N \setminus \{s\}$ and D_p is rooted at s. Now, all arcs in D_p are equality arcs, because $\overline{w}_e \leq 0$ for all $e \in A(D_p)$ and $\overline{w} < 0$ is impossible since y is feasible (by termination).

For $v \in N$, let P be the unique s, v-dipath in D_p. Consider the LP formulation of the shortest s, v-dipath problem: x^P is feasible for the primal and y is feasible for the dual. Since all arcs in P are equality arcs, CS conditions hold, so x^P is optimal and the objective of the dual is $y_v - y_s = y_v - 0 = y_v$, i.e., y_v is the cost of a shortest s, v-dipath. □

1.4 The Bellman-Ford Algorithm

The idea is to go through arcs in "passes".

1.4.1 The B-F Algorithm

Algorithm

1. Initialization.
   a. Set $y_s = 0$ and $y_v = \infty$ for all $v \in N \setminus \{s\}$.
   b. Set predecessor $p_v = \texttt{UNDEF}$ for all $v \in N$.
   c. Set the counter $i = 0$.
2. Correction. While $i < |N| - 1$,
   a. For each $uv \in A$, if $\overline{w}_{uv} < 0$, set $y_v = y_u + w_{uv}$ and $p_v = u$.
   b. Increment i.

Note that if a feasible potential is found, then everything from Ford's algorithm applies here, and we have an optimal rooted tree. Else, if we have an infeasible potential after $|N| - 1$ steps, we show that we have a negative dicycle.

1.4.2 Proof of Correctness

Let d_v denote the *cost* of a shortest s, v-dipath.

Proposition Suppose D does not have any negative dicycle. Then at any point in the algorithm $y_v \geq d_v$.

Proof. The claim is clearly true at the initialization. If $y_v \neq \infty$, then there exists a dipath from s to v using D_p. For each of these arcs e, $\overline{w}_e \leq 0$. Adding up all inequalities $\overline{w}_e \leq 0$ for all arcs e in this s, v-dipath, we obtain $w(P) - y_v \leq 0$. Since d_v is the cost of a shortest s, v-dipath, $d_v \leq w(P) \leq y_v$ and we are done. □

Theorem Suppose no negative dicycles exist. After the ith iteration, if there is a shortest s, v-dipath using at most i arcs, then $y_v = d_v$.

Proof. We do an induction on i. When $i = 0$ (initialization), trivial. Assume that this is true after the ith iteration. We want to show this still holds after $(i + 1)$th iteration.

Pick v which has a shortest s, v-dipath that uses at most $(i + 1)$ arcs. If there is a shortest s, v-dipath that uses at most i arcs, by induction hypotheses, $y_v = d_v$. By proposition, y_v will not change.

Suppose there is a shortest s, v-dipath that uses $(i + 1)$ arcs, say $s = v_1, \ldots, v_{i+1} = v$. Since no negative dicycle exists, $v_1, \ldots, v_i$ is a shortest s, v_i-dipath that uses i arcs. By induction, $y_{v_i} = d_{v_i}$ after the ith iteration and this does not change after the $(i + 1)$-th iteration.

Consider $\bar{w}_{v_i v_{i+1}}$.

- If $\bar{w}_{v_i v_{i+1}} = 0$, this means that $y_{v_{i+1}} = y_{v_i} + w_{v_i v_{i+1}} = d_{v_i} + w_{v_i v_{i+1}} = w(P) = d_{v_{i+1}}$.
- If $\bar{w}_{v_i v_{i+1}} > 0$, this means that $y_{v_{i+1}} < y_{v_i} + w_{v_i v_{i+1}} = d_{v_i} + w_{v_i v_{i+1}} = w(P) = d_{v_{i+1}}$. Contradiction. This cannot happen.
- If $\bar{w}_{v_i v_{i+1}} < 0$, this means that the algorithm will correct the arc $v_i v_{i+1}$ in the $(i+1)$th iteration so $y_{v_{i+1}} = y_{v_i} + w_{v_i v_{i+1}} = d_{v_{i+1}}$.

We are done by induction. □

Corollary At the end of Bellman-Ford, if y is feasible, then $y_v = d_v$ for all $v \in N$. Otherwise, you can conclude there exists a negative dicycle.

Proof. Bellman-Ford runs $|N| - 1$ iterations. Any shortest s, v-dipath could use at most $|N| - 1$ arcs. If y is feasible, then there are no negative dicycles. By the theorem above, $y_v = d_v$ for all $v \in N$. If not, then there exists a negative dicycle.

1.5 Dijkstra's Algorithm

When there are no negative costs, we can apply a greedy algorithm.

1.5.1 Motivation

If there are no negative arcs, $y = 0$ is a feasible potential for the dual. We wish to raise potentials by t for non-tree nodes while maintaining feasibility.

1.5.2 Details

Let T be our current tree.

1. $u, v \notin N(T)$: both y_u and y_v increases by t, so $\bar{w}_{uv} = w_{uv} - y_v + y_u$ stays the same.
2. $u, v \in N(T)$: we do not change the potentials, so $\bar{w}_{uv}$ stays the same.
3. $u \notin N(T) \wedge v \in N(T)$: $\bar{w}_{uv}$ increases and that does not affect feasibility of the potentials.
4. $u \in N(T) \wedge v \notin N(T)$: $\bar{w}_{uv}$ decreases by t; thus choose t to be minimum among all such arcs.

Now, the arc which determined the minimum becomes an equality arc and we can add it to T.

1.5.3 The Algorithm

1. Initialize $y_v = 0$ and $T = \{s\}$.
2. While T is not a spanning tree,
   a. Pick $u \in \delta(N(T))$ such that $\bar{w}_{uv} = \min\{\bar{w}_e : e \in \delta(N(T))\}$.
   b. Update $y_z := y_z + \bar{w}_{uv}$ for all $z \notin N(T)$.
   c. Add uv and u to T.

1.5.4 Correctness

By our work above, y is always feasible. This includes at initialization as we do not have negative costs. All arcs in T are equality arcs. In addition, the algorithm produces a spanning tree rooted at S. Thus, the same LP argument gives that it must be a tree of shortest s, v-dipaths for all $v \in N$.

1.5.5 Runtime

By taking advantage of the greedy approach, Dijkstra runs in $O(|N| \log |N| + |A|)$ which is faster than Bellman-Ford.

1.6 Applications

1.6.1 Network Reliability

Given A network $D = (N, A)$ where each arc e is assigned an associated reliability $r_e \in (0, 1]$. Think of this as a probability that r_e is operational.

Goal For a given dipath P, the reliability of P is $r(p) = \prod_{e \in P} r_e$. Our goal is to maximize reliability amongst all s, t-dipaths.

Notice that $\log r(P) = \sum_{e \in P} \log r_e$ and $\log$ is strictly increasing so it suffices to compare logarithms of reliability. We also make this a minimization problem by having negative arc costs: let $-\log r_{ij}$ denote the cost of arc ij.

Exercise. Modify Dijkstra's algorithm to solve this problem without taking logs.

1.6.2 Currency Exchange

Given We have a set of currencies. There is an exchange rate r_{uv} representing how much does 1 unit of currency u converts into currency v.

Goal Exchange a series of currencies back to the original one so that we make a profit.

Solution Since we can make some profit, the following inequality must hold:

$$\prod_{e \in C} r_e = r_{v_1, v_2} r_{v_2, v_3} \ldots r_{v_k, v_1} > 1 \iff \log \prod_{e \in C} r_e > 0 \iff \sum_{e \in C} (-\log r_e) < 0$$

Label each arc with cost $-\log r_e$. We can just run Bellman-Ford.