Computing the $k$ Shortest Paths

T. M. Murali

*Slides courtesy of Chris Poirel*

March 31, 2014
Motivation

Find the $k$ shortest paths between a pair of nodes $s$ and $t$ in a directed graph, where each edge has a real-valued positive weight.
Two Formulations

- “Finding the $k$ Shortest Loopless Paths in a Network”

- “Finding the $k$ Shortest Paths in a Network”
Two Formulations

- “Finding the $k$ Shortest Loopless Paths in a Network”

- “Finding the $k$ Shortest Paths in a Network”
Shortest Loopless Paths – Basic Idea

Naïve Approaches (time-consuming):
- Enumerate all paths from \( s \) to \( t \) and sort.
- Obtain \( k - 1 \) shortest paths, hide an edge from each path and find a shortest path in the modified network. Test all combinations.
Shortest Loopless Paths – Basic Idea

Naïve Approaches (time-consuming):
- Enumerate all paths from $s$ to $t$ and sort.
- Obtain $k - 1$ shortest paths, hide an edge from each path and find a shortest path in the modified network. Test all combinations.

Basic idea of Yen’s algorithm:
- Compute the shortest path from $s$ to $t$
- The $k^{th}$ shortest path will be a deviation from the previously-discovered shortest path.
Shortest Loopless Paths

- \( \{s, v_2, v_3, \ldots, t\} \) denotes a simple path from \( s \) to \( t \)
- \( P_k = \{s, P^k_2, P^k_3, \ldots, P^k_{|P_k|-1}, t\} \) is the \( k^{\text{th}} \) shortest path from \( s \) to \( t \)
Shortest Loopless Paths

- \{s, v_2, v_3, \ldots, t\} denotes a simple path from \(s\) to \(t\)
- \(P^k = \{s, P^k_2, P^k_3, \ldots, P^k_{|P^k|-1}, t\}\) is the \(k^{th}\) shortest path from \(s\) to \(t\)
- \(D^k_i\) is the “deviation from \(P^{k-1}\) at node \(P^k_{i-1}\)”
  
  More specifically, the shortest \(s \rightsquigarrow t\) path that:
  
  1. coincides with \(P^{k-1}\) from \(s\) to \(P^k_{i-1}\)
  2. deviates to a node \(u\) where \(u\) is not used as this deviation in any of the \(k - 1\) shortest paths
  3. reaches \(t\) by a shortest path from \(u\) without using any node in the first part of the path
Shortest Loopless Paths

- \( \{s, v_2, v_3, \ldots, t\} \) denotes a simple path from \( s \) to \( t \)
- \( P^k = \{s, P^k_2, P^k_3, \ldots, P^k_{|P^k|-1}, t\} \) is the \( k \)th shortest path from \( s \) to \( t \)
- \( D^k_i \) is the “deviation from \( P^{k-1} \) at node \( P^{k-1}_i \)”
  
  More specifically, the shortest \( s \leadsto t \) path that:
  1. coincides with \( P^{k-1} \) from \( s \) to \( P^{k-1}_i \)
  2. deviates to a node \( u \) where \( u \) is not used as this deviation in any of the \( k - 1 \) shortest paths
  3. reaches \( t \) by a shortest path from \( u \) without using any node in the first part of the path

- \( R^k_i = \{s, P^k_2, P^k_3, \ldots, P^k_i\} \) is the root of \( D^k_i \)
- \( S^k_i = \{P^k_i, \ldots, t\} \) is the spur of \( D^k_i \)
Shortest Loopless Paths

- Find the shortest path $P^1$
- For $k = 2, 3, \ldots$, find $P^k$ as follows:
  1. Let $B^k = B^{k-1}$, the set of candidate paths from iteration $k - 1$
  2. for $1 \leq i < |P^{k-1}|$ do
  3. Let $x = P_i^{k-1}$
  4. Hide incoming edges to $x$ for the remainder of iteration $k$
  5. for each $j$ such that the first $i$ nodes in $P^j$ match $P^{k-1}$ do
  6. Hide edge $(x, P_{i+1}^j)$ for the remainder of iteration $k$
  7. $R_i^k$ is the first $i$ nodes of $P^{k-1}$
  8. $S_i^k$ is the shortest path from $x$ to $t$
  9. Join $R_i^k$ and $S_i^k$ to form $D_i^k$
  10. Add candidate path $D_i^k$ to $B^k$
  11. Remove the shortest path from $B^k$ and return it
Example – Find $P^3$

$P^1 = \{s, c, d, t\}$

$P^2 = \{s, a, t\}$

$P^3 = \ ?$
Example – Hide Edges for Root \( \{s\} \)

\[ \begin{align*}
P^1 &= \{s, c, d, t\} \\
P^2 &= \{s, a, t\} \\
P^3 &= \text{?}
\end{align*} \]
Example – Hide Edges for Root \( \{s, a\}\)

\[
\begin{align*}
P^1 &= \{s, c, d, t\} \\
P^2 &= \{s, a, t\} \\
P^3 &= \text{?}
\end{align*}
\]
Example – Find Shortest Spur for Each Root

\[ P^1 = \{s, c, d, t\} \]
\[ P^2 = \{s, a, t\} \]
\[ P^3 = \text{?} \]

\[ S^3_1 = \{s, e, f, t\} \]
\[ S^3_2 = \{a, b, t\} \]
Example – Identify Shortest Deviation

\[ P^1 = \{s, c, d, t\} \]
\[ P^2 = \{s, a, t\} \]
\[ P^3 = ? \]

\[ S^3_1 = \{s, e, f, t\} \]
\[ S^3_2 = \{a, b, t\} \]

\[ D^3_1 = \{s, e, f, t\} \]
\[ D^3_2 = \{s, a, b, t\} \]
How do we find $S^k_i$ efficiently?

For $k = 2, 3, \ldots$, find $P^k$ as follows:

1. Let $B^k = B^{k-1}$, the set of candidate paths from iteration $k-1$
2. for $1 \leq i < |P^{k-1}|$ do
3. Let $x = P_{i}^{k-1}$
4. Hide incoming edges to $x$ for the remainder of iteration $k$
5. for each $j$ such that the first $i$ nodes in in $P^j$ match $P^{k-1}$ do
6. Hide edge $(x, P_{i+1}^j)$ for the remainder of iteration $k$
7. $R_i^k$ is the first $i$ nodes of $P^{k-1}$
8. $S_i^k$ is the shortest path from $x$ to $t$
9. Join $R_i^k$ and $S_i^k$ to form $D_i^k$
10. Add candidate path $D_i^k$ to $B^k$
11. Remove the shortest path from $B^k$ and return it
How do we find $S^k_i$ efficiently?

- For $k = 2, 3, \ldots$, find $P^k$ as follows:
  1. Let $B^k = B^{k-1}$, the set of candidate paths from iteration $k-1$
  2. for $1 \leq i < |P^{k-1}|$ do
  3. Let $x = P^k_{i-1}$
  4. Hide incoming edges to $x$ for the remainder of iteration $k$
  5. for each $j$ such that the first $i$ nodes in $P^j$ match $P^{k-1}$ do
  6. Hide edge $(x, P^j_{i+1})$ for the remainder of iteration $k$
  7. $R^k_i$ is the first $i$ nodes of $P^{k-1}$
  8. $S^k_i$ is the shortest path from $x$ to $t$
  9. Join $R^k_i$ and $S^k_i$ to form $D^k_i$
  10. Add candidate path $D^k_i$ to $B^k$
  11. Remove the shortest path from $B^k$ and return it

- Run Dijkstra’s algorithm.
- What is the running time of each iteration if the graph has $n$ nodes and $m$ edges?
How do we find $S^k_i$ efficiently?

- For $k = 2, 3, \ldots$, find $P^k$ as follows:
  1. Let $B^k = B^{k-1}$, the set of candidate paths from iteration $k - 1$
  2. for $1 \leq i < |P^{k-1}|$ do
  3. Let $x = P_i^{k-1}$
  4. Hide incoming edges to $x$ for the remainder of iteration $k$
  5. for each $j$ such that the first $i$ nodes in $P^j$ match $P^{k-1}$ do
  6. Hide edge $(x, P_{i+1}^j)$ for the remainder of iteration $k$
  7. $R_i^k$ is the first $i$ nodes of $P^{k-1}$
  8. $S_i^k$ is the shortest path from $x$ to $t$
  9. Join $R_i^k$ and $S_i^k$ to form $D_i^k$
  10. Add candidate path $D_i^k$ to $B^k$
  11. Remove the shortest path from $B^k$ and return it

- Run Dijkstra’s algorithm.

- What is the running time of each iteration if the graph has $n$ nodes and $m$ edges? At most $n$ invocations of Dijkstra’s algorithm, i.e., $O(n(m + n \log n))$.

- Therefore total running time is $O(kn(m + n \log n))$. 

T. M. Murali Slides courtesy of Chris Poirel March 31, 2014