Finding $k$ Disjoint Paths in Multi-Cost Networks

Leepila Ruchaneeya
Department of Information and Communication Engineering
The University of Electro-Communications
Outline

- Introduction
- Motivation
- $k$-penalty with auxiliary cost matrix (KPA) scheme
- $k$-penalty with initial cost matrix (KPI) scheme
- Illustrative Numerical Examples
- Numerical Results
- Conclusion
Introduction

- Survivability networks
  - A failure of any network element may lead to significant data and revenue losses.
  - A disjoint path routing enhances the survivability of a network
  - Survivability is provided by backup paths that used to restored the effected traffic after failure of working (active) paths.
Introduction

Fig. 1. Active and backup path in a network
Introduction

- Single-cost networks
  - The cost of each network arc $a_h$ remains the same for each of $k$ paths of a demand

Fig. 2. Single-cost network
Introduction

- Multi-cost networks

  - The cost of any arc $a_h$ may be different for each of $k$ paths

    - Applying the backup path sharing.
      - The cost of an arc $a_h$ in a backup path is often the fraction of its original cost, used in working path computation.

Fig. 3. Multi-cost network
Motivation

- KPA scheme is a scheme to find k-disjoint paths in multi-cost network.
- Sometimes KPA cannot find disjoint paths even though the disjoint paths of a demand exist.
- The KPI scheme is developed from KPA scheme and proposed to avoid the problem as mentioned above.
- The summation of total path costs is minimized by using KPI scheme.
KPA Scheme

**INPUT**

- A demand $d_r$ to find the set of end-to-end $k$-disjoint paths between a pair of nodes $(s_r, t_r)$
- The matrices $\Xi_1$, $\Xi_2$, ..., $\Xi_k$ of arc costs (one matrix for each path of a demand)
- The upper bound $i_{upper}$ on the number of allowed conflicts

**OUTPUT**

- The set of $k$ disjoint paths $\eta_1$, $\eta_2$, ..., $\eta_k$ - all between a given pair of demand source and destination nodes $(s_r, t_r)$
- The summation of $k$ disjoint paths costs
KPA Scheme

- **Terminology**
  - $d_r$: demand to find the set of end-to-end $k$ disjoint paths between a pair of nodes $(s_r, t_r)$
  - $s_r$: source node of demand $d_r$
  - $t_r$: destination node of demand $d_r$
  - $i_{upper}$: the number of allowed conflicts
  - $p$: index of path $1, \ldots, k$
  - $\eta_p$: $p$-th path
  - $a_h$: $h$-th arc in network
  - $\xi_h$: cost of each network arc $a_h$
  - $\xi_h^p$: cost of arc $a_h$ of $p$-th path
  - $\xi^p$: path cost of arcs that are transversed by $p$-th path
  - $\Xi^p$: matrix of arc costs $\xi_h^p$
  - $\Xi^{tmp}$: auxiliary matrix of arc costs $\xi_h^{tmp}$
  - $i_c$: conflict counter
KPA Scheme

- **Step 1** Set $i_c = 1$ and $\xi_{h}^{tmp,p} = \xi_{h}^{p}$ for $p=1,\ldots,k$ for each network arc $a_h$.
- **Step 2** Set $j = 1$.
- **Step 3** For each network arc $a_h$, set $\xi_{h}^{tmp} = \xi_{h}^{tmp,j}$.
- **Step 4** Consider each path $\eta_i$ from the set of previously found $j-1$ paths and for each arc $a_h$, if $a_h$ is a forbidden arc* of the path, then increase the arc cost $\xi_{h}^{tmp}$ by path cost $\xi_{h}^{tmp,i}$ of $\eta_i$ on the network with costs matrix $\Xi_{tmp,i}$.
- **Step 5** Find the shortest path $\eta_j$ on the network with costs matrix $\Xi_{tmp}$.
- **Step 6** If $\eta_j$ is disjoint with the previously found $j-1$ paths then set $j = j + 1$ and go to Step 7.
  - else
    - a) Increase the costs $\xi_{h}^{temp,1}$, $\ldots$, $\xi_{h}^{temp,k}$ of each conflicting arc** $a_h$ of $\eta_j$ by path cost $\xi_{h}^{tmp}$ of $\eta_j$ on the network with costs matrix $\Xi_{tmp}$; delete the found paths and set $i_c = i_c + 1$.
    - b) if $i_c > i_{upper}$ then terminate and reject the demand, else go to Step 2.
- **Step 7** If $j > k$ then terminate and return the found set of paths else go to Step 3.
Fig. 4. Example of the KPI scheme for a demand $d_r=(1,7)$ and $k = 3$ in the multi-cost network.
Illustrative Numerical Example

Find the common node between 2 paths (conflicting node)
Illustrative Numerical Example

Path cost = 11

Path cost = 29

Path cost = 27

Path cost = 64

Path cost = 67

1-3-6-7

1-2-5-7

1-4-7
KPI Scheme

Step 6 If $\eta_j$ is not disjoint with the previously found $j$-1 paths then

a) Increase the costs $\xi_{h_{\text{tmp},1}}$, ..., $\xi_{h_{\text{tmp},k}}$ of each conflicting arc $a_h$ of $\eta_j$ by path cost $\xi_j$ of $\eta_j$ on the network with initial costs matrix $\Xi_j$
Illustrative Numerical Example

Path cost = 3

Path cost = 4

Path cost = 7

Path cost = 7

Path cost = 11

Path cost = 15

Path cost (org) = 5

1-2-5-7

1-4-7

1-3-6-7
Numerical results

The simulation is to find $k$ disjoint paths in case of $k = 3$, so additional links, shown as dashed lines in both networks, provide the degree of each node greater than or equal to 3.
Numerical Results

Fig. 6. Cumulative $k$ disjoint paths finding success rate (%) at each $i_{upper}$ of finding $k$ disjoint paths on U.S. long-distance network.
Numerical Results

Fig. 7. Cumulative $k$ disjoint paths finding success rate (%) at each $i_{\text{upper}}$ of finding $k$ disjoint paths on Italian network.
Numerical Results

Fig. 8. Normalized summation of $k$ disjoint paths costs on U.S. long-distance network and Italian network by using Bhandari’s, KPA and KPI scheme
Conclusion

- KPI scheme is able to find the required number of $k$ disjoint paths faster than the KPA scheme.

- Consider the summation of $k$ disjoint paths costs, the KPI scheme gets the summation of costs lower than costs of $k$ disjoint paths that are found by KPA scheme.
Thank you for your attention