# Survivable Multi-Cost Networks with *k* Disjoint Paths

Ruchaneeya Leepila, Eiji Oki, and Naoto Kishi

***Abstract*— With the development of network technology, growth of large amount of data increases over networks and a failure mainly affects data loss. As a result, the survivable system is developed as a mechanism to ensure that the data loss can be minimized as possible. A potential of disjoint path routing technique uses to enhance the data loss problem over the communication networks. Therefore, *k* disjoint paths in multi-cost networks, which each network arc may be given *k* different arc costs, is purposed in this paper. This scheme, called the *k* penalty scheme with initial arc cost matrix (KPI), penalizes the use of conflicting arcs found in previously set paths and increases the costs of these arcs in accordance with the *initially given* arc cost matrix. The KPI scheme provides updating the conflicting arcs costs appropriately to avoid overlapping used paths. With our purposed simulations, the results showed that the KPI scheme was able to find *k* disjoint paths faster than the conventional scheme that uses the incrementally updated *auxiliary* arc cost matrix to increases the cost of conflicting arcs. Moreover, the KPI scheme yields *k* disjoint paths with lower total cost than the conventional scheme.**

***Index Terms*— *k* disjoint paths, multi-cost network, network survivability.**

## I. Introduction

THERE is enormous data transfer over a network. A break in a fiber span or node failures can cause a huge damage to users in the network. Therefore, network providers should design survivable networks so that the communication loss can be minimized. Disjoint path routing enhances the survivability of a network [1]-[3]. Disjoint paths, paths that do not sharing the same links or nodes, must be set between source and destination nodes to minimize the damage created by network failure. If *k* disjoint paths are set between source and destination nodes, at least one path is protected against *k*-1 simultaneous failures. It means that a backup path will be used when a working path fails. To utilize efficient resources, the shared backup path protection is applied.

Manuscript received July 8, 2011.

R. Leepila is with Dept. of Information and Communication Engineering, The University of Electro-Communications, Tokyo JAPAN (e-mail: ruchanel@ice.uec.ac.jp).

E. Oki is with Dept. of Information and Communication Engineering, The University of Electro-Communications, Tokyo JAPAN (e-mail: oki@ ice.uec.ac.jp).

N. Kishi is with Dept. of Information and Communication Engineering, The University of Electro-Communications, Tokyo JAPAN (e-mail: kishi@ ice.uec.ac.jp).

The problem of finding disjoint paths in a *single-cost network* has been widely studied [4]-[8]. In a single-cost network, the cost of each network arc is the same for all *k* paths. Several algorithms have been introduced to find *k* disjoint paths. The active path first approach (APF) presented in [4] finds the first shortest path by using a shortest path algorithm such as Dijkstra’s algorithm [5]. Then, it finds the next shortest path after removing the previously found paths. This procedure is repeated until the required number of disjoint paths is obtained. Suurballe's algorithm [6], or its modification, Bhandari's algorithm [7], finds disjoint paths and the total cost of all paths is minimized. The problem of finding such disjoint paths is called the Min-Sum problem.

In [9], J. Rak proposed the *k*-penalty algorithm to find *k*-disjoint paths in a *multi-cost network*. In a multi-cost network, each network arc can have a different cost for all *k* paths. Its applications usually lie in the field of shared backup path protection [10], [11]. In this case, the cost of an arc for a backup path is often a fraction of that for a working path. The Min-Sum problem in a multi-cost network is NP-Complete (NPC) [4], [7], [12]. The *k*-penalty algorithm finds the shortest path as the first path. The arcs on the shortest path and those connected to the transit nodes on this path are considered as forbidden arcs for the next disjoint path to be found. Although the APF algorithm assigns the forbidden arcs infinitely high cost, the *k*-penalty algorithm gives them finite costs to avoid the trap problem [9]. That is, the next path must pay a penalty for using a forbidden arc. Forbidden arc cost is increased by the path cost of the previously found path. Arc costs are incrementally updated and kept in an auxiliary arc cost matrix. If any conflict, i.e. the current path is not disjoint with all previously found paths, occurs, all found paths are deleted. Before starting the process of finding *k* disjoint paths again, the costs of conflicting arcs are incrementally increased by the cost of the last found path in the previous iteration. The path cost is computed using the auxiliary arc cost matrix. Thus, in this paper, we call this algorithm KPA, or *k*-penalty by using the auxiliary arc cost matrix to compute the path cost. This procedure, including the deletion of found paths, is iterated until *k* disjoint paths are found, or the number of iterations reaches a number specified to avoid infinite loops.

We found that KPA sometimes fails even with a large number of iterations, even though disjoint paths actually exist. With every iteration, or conflict, the arc costs in the auxiliary arc cost matrix are increased. In order to avoid traversing arcs with large costs, the algorithm may find a path that overlaps already used paths, including forbidden arcs from the previous paths. This path overlapping causes the deletion of found paths and restarts the process. It takes time to find *k* disjoint paths or sometimes they cannot be found at all. This problem must be solved to find *k* disjoint paths in an efficient manner.

This paper proposes *k* disjoint paths scheme based on the KPA scheme that eliminates the KPA problem. The proposed scheme uses the same penalty process but the cost increases are determined from the *initially given* arc cost matrix. Thus, this scheme is called KPI: *k*-penalty with the initial arc cost matrix. This paper extends the previous work in [13]. The performances of the KPI scheme are investigated in detail with various network topologies and numbers of required disjoint paths of a demand. Moreover, the mathematical formulation of finding *k* disjoint path is described. The computational time complexities of the KPA and KPI schemes are discussed in KPI section. Numerical results show that the KPI scheme is able to find *k* disjoint paths faster than the KPA scheme. Moreover, KPI yields disjoint paths with lower average total cost than the KPA scheme.

The remainder of this paper is organized as follows. Section II shows the formulation of minimizing summation of *k* disjoint paths costs. Section III describes the KPA scheme. Section IV presents the KPI scheme. Section V compares the performance of the KPI scheme to that of the KPA scheme. Section VI summarizes this paper.

## II. Problem Formulation

We consider a graph of directed network $G(V,A)$, where $V$ is a set of network nodes and $A$ is a set of network arcs. Let $P$ be a set of required disjoint paths. An arc from node $u \in V$ to node $v \in V$ is denoted as $(u,v) \in A$. $d_{uv}^{p}$ is the cost of arc from node $u$ to node $v$ on the $p$th path, where $p \in P$. $x_{uv}^{p}$ is equal to 1, if the $p$th path transverses on arc from node $u$ to node $v$ and equal to 0, otherwise. The problem is to find *k* disjoint paths from source node, $s$, to destination node, $t$ so that the summation of disjoint path costs can be minimized. This problem is formulated as an ILP problem in the following.

$$\min \quad \sum_{p\in P}\sum_{u\in V}\sum_{v\in V} x_{uv}^{p} d_{uv}^{p} \tag{1a}$$

$$\text{subject to} \quad \sum_{v\in V} x_{uv}^{p} - \sum_{v\in V} x_{vu}^{p} = \begin{cases} 1, & u = s \\ 0, & u \neq s, u \neq t \end{cases} \tag{1b}$$

$$x_{uk_1}^{1} + x_{uk_2}^{2} + \text{K} + x_{uk_p}^{p} \leq 1, \quad u \neq s \tag{1c}$$
$$\forall u, k_1, k_2, \text{K}, k_p \in V$$

$$x_{k_1 u}^{1} + x_{k_2 u}^{2} + \text{K} + x_{k_p u}^{p} \leq 1, \quad u \neq t \tag{1d}$$
$$\forall u, k_1, k_2, \text{K}, k_p \in V$$

$$x_{uv}^{p} = \{0,1\} \qquad \forall p \in P, \forall u, v \in V \tag{1e}$$

The decision variable is $x_{uv}^{p}$ and the given parameter is $d_{uv}^{p}$.

The objective function in Eq. (1a) minimizes the summation of *k* disjoint path costs. Eq. (1b) is a constraint for flow conservation. To ensure that the paths of each connection do not traverse the common transit nodes, the node disjoint constraints are needed, as shown in Eqs. (1c) and (1d). Eq. (1e) is the binary constraint for the ILP formulation.

Since the problem of finding optimal set of *k* disjoint paths is NP-complete [4], [7], [12], a heuristic algorithm was provided to solve the problem in [9]. This paper proposes another heuristic algorithm to solve the problem efficiently.

## III. KPA: *k*-Penalty with Auxiliary Arc Costs Matrix

This section presents the KPA scheme and its weakness.

### A. *Terminology and Description*

The terminology used in this paper is shown below.

| | |
|---|---|
| $d_r$ | Demand to find a 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 $d_r$ |
| $i_{max}$ | Maximum allowable number of conflicts |
| $p$ | Index of path *1,..,k* |
| $\eta_p$ | *p*th path |
| $a_h$ | *h*th arc, where *h* is denoted as $(u,v) \in$ A |
| $\xi_h$ | Cost of each arc $a_h$ |
| $\xi_h{}^{p}$ | Cost of arc $a_h$ of the *p*th path |
| $\xi^{p}$ | Cost of *p*th path that is a sum of $\xi_h{}^{p}$ over traversed $a_h$ , shown as Eq (2) |
| $\xi_h{}^{aux,p}$ | Auxiliary cost of arc $a_h$ of the *p*th path |
| $\Xi^{p}$ | Initial matrix of arc cost $\xi_h{}^{p}$ |
| $\Xi^{aux}$ | Auxiliary matrix of arc cost $\xi_h{}^{aux}$ |
| $\Xi^{aux,p}$ | Auxiliary matrix of arc cost $\xi_h{}^{aux,p}$ |
| $i_c$ | Conflict counter |

The KPA scheme is shown in Fig.1. Demand $d_r$ to find *k* disjoint paths from source node $s_r$ to destination node $t_r$, the arc cost matrices for each disjoint path, and the maximum allowable number of conflicts, $i_{max}$, are initially given. The arc cost matrix consists of the costs of network arcs for each disjoint path, where each arc from node *u* to node *v* is denoted as $(u,v)$ is in the set of network arcs, A. The KPA scheme outputs the set of *k* disjoint paths and the total costs of the *k* disjoint paths. KPA uses the shortest-path-based algorithm. At Step 1, the conflict counter, $i_c$, is set to 1 and the initial cost matrix of the *p*th path, $\Xi^{p}$, is copied to the auxiliary cost matrix of the *p*th path, $\Xi^{aux,p}$, for all paths, $p$=1,…,*k*. $\Xi^{p}$ is kept to compute the total path cost using Eq. (2) after finding *k* disjoint paths. At Step 2, set $j$=1 to find the first path. In Step 3, $\Xi^{aux,p}$ is copied to $\Xi_h{}^{aux}$ . Step 4 is skipped if $j$=1. To find the next paths $\eta_j$ ($j \neq 1$), path $\eta_j$ has to pay a penalty for using one of the forbidden arcs, i.e. links traversed by previously found paths $\eta_j$ (link disjoint), or links connected to transit nodes used by previously found paths (node disjoint). The costs of the forbidden arcs are increased by the costs of all $j$-1 previously found paths at Step 4. At Step 5, $\eta_j$ is found as the shortest path on the network with auxiliary cost matrix $\Xi^{aux}$. At

| | |
|---|---|
| INPUT: | Demand $d_r$ to find the set of $k$-disjoint paths between a pair of nodes ($s_r$ $t_r$). The initial arc costs matrices $\Xi^1$, $\Xi^2$,..., $\Xi^k$ (one matrix for each path) of a demand. The maximum allowable number of conflicts, $i_{max}$. |
| 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 total path cost of $k$-disjoint paths is $$\xi^{total} = \sum_{p=1}^{k} \sum_{a_h \ on \ path \eta_p} \xi_h^p \quad (2)$$ |
| PROCESS | |
| Step 1 | Set $i_c$ = 1and $\Xi^{aux,p} = \Xi^p$ for $p$ = 1,…, $k$ |
| Step 2 | Set $j$=1. |
| Step 3 | Set $\Xi^{aux} = \Xi^{aux,j}$. |
| 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 $\eta_i$, then increase the arc cost $\xi_h^{aux}$ by path cost $\xi_h^{aux,i}$ of $\eta_i$ on the network with costs matrix $\Xi^{aux,i}$. Set $$\xi_h^{aux} = \xi_h^{aux} + \xi^{aux,i} \quad (3)$$ The path cost is defined by $$\xi^{aux,i} = \sum_{a_h \ on \ path \eta_i} \xi_h^{aux,i} \ \ for \ i = 1, \ldots, j-1 \quad (4)$$ |
| Step 5 | Find the shortest path $\eta_j$ on the network with costs matrix $\Xi^{aux}$. |
| 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 6a) Increase the costs $\xi_h^{aux,1}$,..., $\xi_h^{aux,k}$ of each **conflicting arc $a_h$ of $\eta_j$ by path cost $\xi_h^{aux}$ of $\eta_j$ on the network with cost matrix $\Xi^{aux}$. Set $$\xi_h^{aux,p} = \xi_h^{aux,p} + \xi^{aux} \quad (5)$$ where path cost $\xi^{aux}$ is defined by $$\xi^{aux} = \sum_{a_h \ on \ path \eta_j} \xi_h^{aux} \quad (6)$$ then delete the found paths and set $i_c = i_c + 1$. 6b) If $i_c > i_{max}$ 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. |

- *Forbidden arcs are links traversed by previously found paths $\eta_j$ (link disjoint), or links connected to transit nodes used by previously found paths (node disjoint).
- **Conflicting arcs are links on path $\eta_j$ that are not disjoint with the $j$-1 previously found paths (link disjoint), or links incident to common transit node jointly by $\eta_j$ and by other of previous the $j$-1 previously found paths (node disjoint).

Fig. 1. KPA scheme for finding $k$ disjoint paths given demand ($s_r$ $t_r$).

Step 6, if $\eta_j$ is disjoint with the ($j$-1) previously found paths, the index number of path, $j$, is increased by one and the process goes to Step 7 to check if the required number of $k$ disjoint paths has been obtained. The process terminates if the number of found disjoint paths has reached the required number, $k$. Otherwise, the process will find the next path by reentering Step 3. If $\eta_j$ is not disjoint (link or node) with the $j$-1 previously found paths, a conflict is called and the costs $\xi_h^{aux,1}$ ,…, $\xi_h^{aux,k}$ of each conflicting arc $a_h$, the link shared between the previously found *j-1* paths and path $\eta_j$, or the link connected to the node shared between previously found $j$-1 paths and path $\eta_j$, is increased by the path cost $\xi^{aux}$ of $\eta_j$, which is computed from auxiliary costs matrix $\Xi^{aux}$ (Step 6a) as shown in Eq. (5). After increasing each conflicting arc $a_h$, all found paths are deleted and conflict counter, $i_c$, is increased by one. If $i_c$ is greater than the maximum allowable number of conflicts, $i_{max}$, the process is terminated. If $i_c$ is less than $i_{max}$ the process reenters Step 2.

### B. Example of KPA Scheme

Figures 2 (a1), (a2) and (a3) are multi-cost network that has three sets of arc costs; one for the first path, $\xi_h^1$: the second path: $\xi_h^2$; and the third path: $\xi_h^3$. The KPA scheme is demonstrated with an example in Fig. 3. The example shows how to find the $k$ node-disjoint paths in a multi-cost network with $k$ = 3 for the demand between node 1 to 7. The scheme starts by setting the auxiliary cost matrix as $\Xi^{aux,p} = \Xi^p$ for $p$ = 1, 2, 3 and $i_c$ = 1. The scheme then considers at the first path *j=1* and sets the auxiliary cost matrix $\Xi^{aux} = \Xi^{aux,j}$. The first path $\eta_1$ (1-4-7) is found as the shortest path, shown as Fig. 3 (a1). The costs, $\xi_h^{aux,1}$, for arcs incident to transit nodes of path $\eta_1$ of the set of arc costs $\xi_h^{aux,2}$ are increased by path cost $\xi^{aux,1}$, which is a penalty and equal to 14 in the example, of path $\eta_1$ as shown in Fig. 3 (a2). Then, path

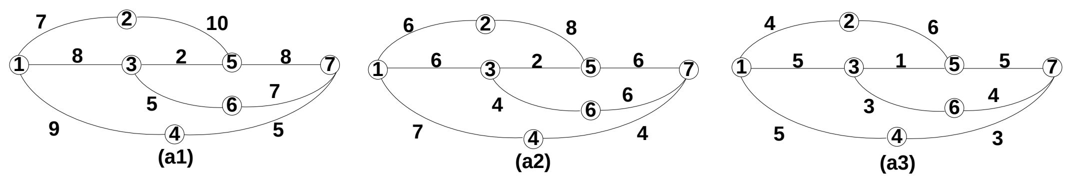


Fig. 2 Multi-cost network with demand $d_r$ = (1, 7) and $k$ = 3.

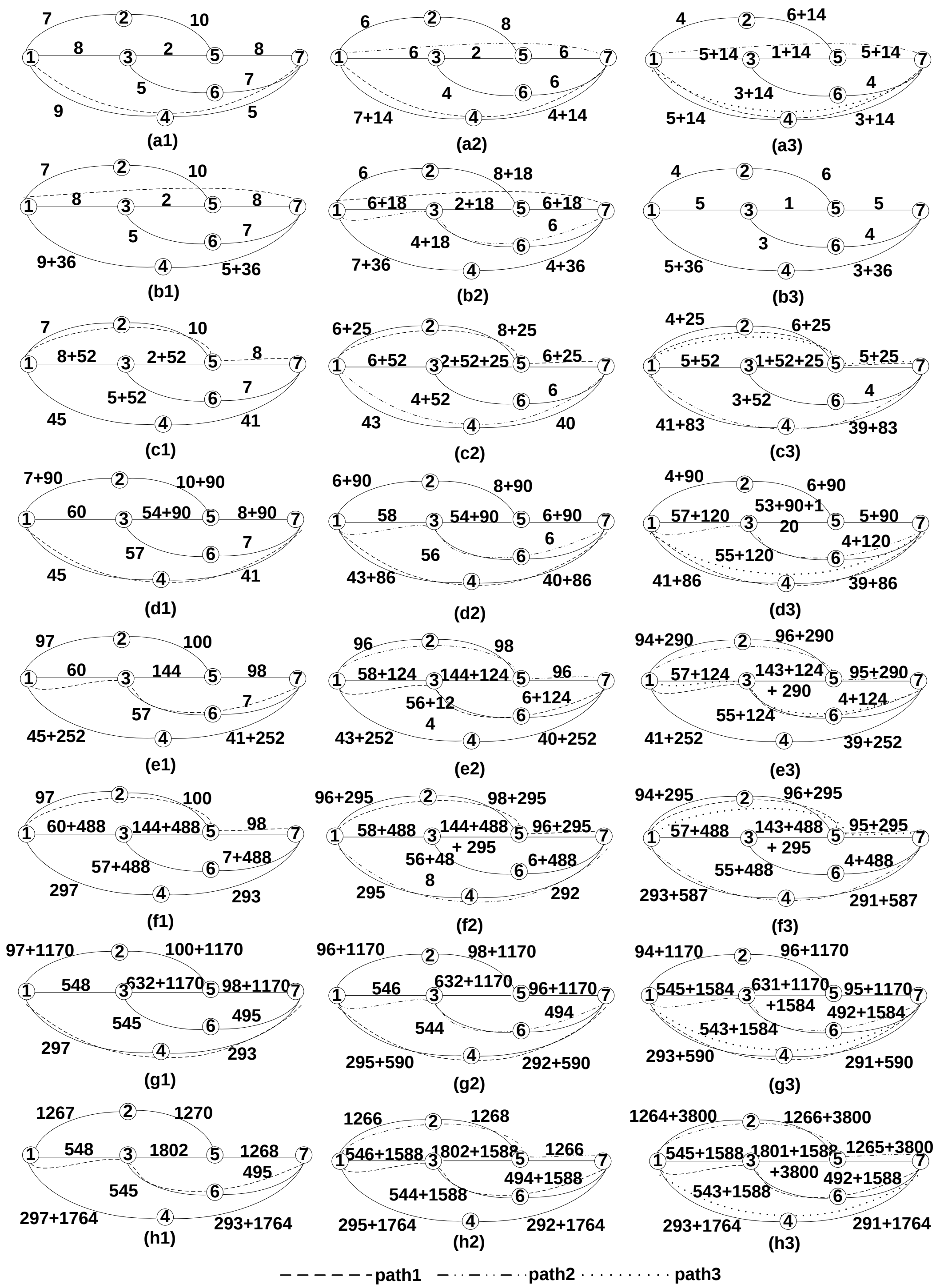


Fig. 3. Example of KPA scheme for multi-cost network with demand $d_r$ = (1, 7) and $k$ = 3.

$\eta_2$ (1-3-5-7) is found. To find the third path, the costs of forbidden arcs of paths $\eta_1$ and $\eta_2$ on the network are increased by path cost $\xi^{aux,1}$ of path $\eta_1$ and path cost $\xi^{aux,2}$ of path $\eta_2$. However, $\eta_3$ (1-4-7), which is not disjoint with $\eta_1$ is found, as

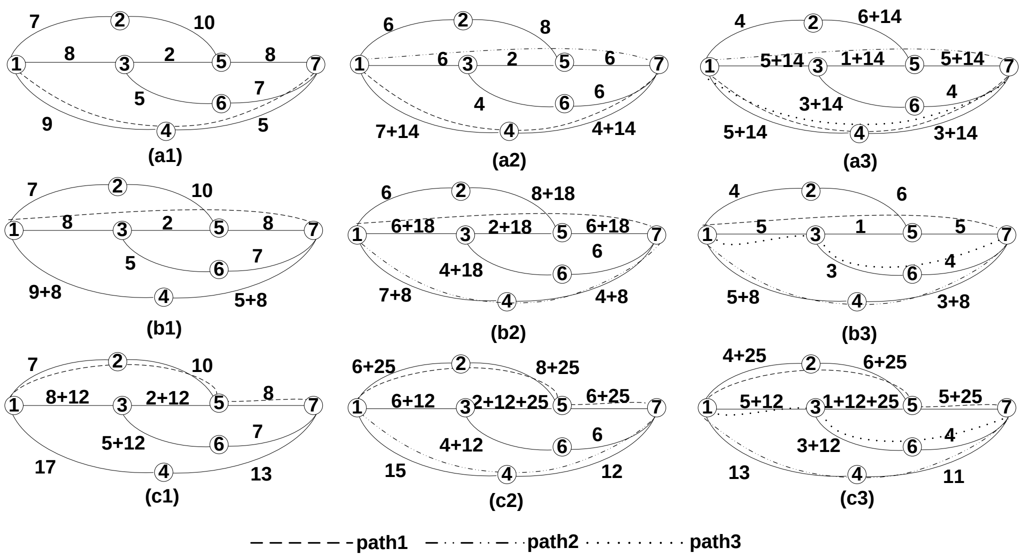



Fig .4. Example of KPI scheme for multi-cost network with demand $d_r$ = (1, 7) and $k$ = 3.

shown in Fig. 3 (a3). Costs $\xi_h^{aux,p}$ of arcs incident to node 4 on $\eta_3$ for all paths, $p$=1,…,$k$, are increased by path cost $\xi^{aux}$, as shown in Eq. (5). The path cost $\xi^{aux}$ is defined by Eq. (6), which is equal to 36 in the example, as shown in Fig.3 (b1). Next, all found paths are deleted and $i_c$ is increased by one. The KPA scheme starts finding $k$ disjoint paths from the beginning again, as shown in Fig. 3 (b1). However, the scheme takes time to find the required set of $k$ node-disjoint paths because it avoids the paths with high cost and this situation leads to overlap with used paths, as shown in Fig. 3 (c3), (d3), (e3), (f3) and (g3). Finally, this scheme finds a set of $k$=3 node-disjoint paths, which are $\eta_1$ (1-3-6-7), $\eta_2$ (1-2-5-7), and $\eta_3$ (1-4-7), at $i_c$ = 8, as shown in Fig. 3 (h3).

## IV. KPI: K-PENALTY WITH INITIAL ARC COSTS MATRIX

The KPI scheme is an extension of the KPA scheme. The KPI scheme uses the same penalty process as the KPA scheme, only the policy of updating $\xi_h^{aux,p}$ is different from the KPA scheme (Step 6a). The KPI scheme increases the costs $\xi_h^{aux,1}$,…, $\xi_h^{aux,k}$ of each conflicting arc $a_h$ of $\eta_j$ by path cost $\xi^j$ of $\eta_j$ using initial costs matrix $\Xi^j$. The Step 6a of the KPA scheme is replaced by the Step 6a of the KPI scheme, which is as follows.

6a) Increase the cost $\xi_h^{aux,1}$,…, $\xi_h^{aux,k}$ of each conflicting arc $a_h$ of $\eta_j$ by path cost $\xi^j$ of $\eta_j$ on the network with cost matrix $\Xi^j$.

$$\text{Set } \xi_h^{aux,p} = \xi_h^{aux,p} + \xi^j \text{ when } p = 1, \ldots, k \tag{7}$$

where path cost $\xi^j$ is defined by

$$\xi^j = \sum_{a_h \text{ on path} \eta_j} \xi_h^j \tag{8}$$

and then delete the found paths and set $i_c$ = $i_c$+1.

The KPA and KPI schemes use Dijkstra's algorithm to examine each of $k$ paths of a demand, which requires O($N^2$) time, where $N$ is the number of network nodes. Since the conflicts are considered in these schemes, the computational time complexities of KPA and KPI schemes are equal to O($MN^2$), where $M$ is the maximum allowable number of conflicts.

### A. Example of KPI Scheme

We reuse the example in demonstrating the KPI scheme, see Fig. 4. The algorithm starts finding the first path $\eta_1$ (1-4-7) by using the shortest-path-based algorithm, Fig. 4 (a1). Before finding path $\eta_2$, the cost $\xi_h^{aux}$ of arcs incident to transit nodes of path $\eta_1$ are increased by the total cost $\xi^{aux,1}$ of path $\eta_1$, which is equal to 14 in the example, Fig. 4 (a2). After that path $\eta_2$ (1-3-5-7) is found. The cost, $\xi_h^{aux}$, of arcs incident to transit nodes of paths $\eta_1$ and $\eta_2$ are increased by path cost $\xi^{aux,1}$ of path $\eta_1$ and path cost $\xi^{aux,1}$ of path $\eta_2$, respectively. However, $\eta_3$ (1-4-7), which is not disjoint with $\eta_1$ and has a common transit node, node 4, is found as shown in Fig. 4 (a3). Cost $\xi_h^{aux,p}$ of arcs incident to node 4 for all paths, $p$=1,...,$k$, are increased by path cost $\xi^3$ computed from the initial arc costs of $\eta_3$ defined by Eq. (8), which is equal to 8 in Fig. 4 (b1). Next, all the found paths are deleted and $i_c$ is increased by one. The algorithm starts from the beginning, as shown in Fig. 4 (b1). Finally, the scheme finds a set of $k$=3 node-disjoint paths, which are $\eta_1$ (1-2-5-7), $\eta_2$ (1-4-7), and $\eta_3$ (1-3-6-7), as shown in Fig. 4 (c3). Since the KPI scheme is more careful in increasing the costs of conflicting arcs, it can find a set of $k$=3 node-disjoint paths at the conflict counter $i_c$ value of 3 in the same way as the KPA scheme. This example shows that the KPI scheme can find a set of $k$ node-disjoint paths faster than the KPA scheme.

## V. PERFORMANCE EVALUATION

We compare the performance of KPI scheme to that of the KPA scheme using computer simulations of various network topologies, see Fig. 5. The node-disjoint paths are considered in this evaluation. The required number of disjoint paths is denoted by $k$. $k$ is set to two and three for the PAN European network, the Italian network and the U.S. long-distance network. Therefore, additional links, shown as dashed lines in three networks, are needed to keep the degree of each node greater than or equal to three. The arc cost matrices of the multi-cost networks are set to three for taking account into finding $k$=2 and $k$=3 disjoint paths. Since the degree of each node in the COST239 network is greater than or equal to four, $k$ is set to two, three and four paths and the cost matrices of the multi-cost networks are set to four. 100 arc cost matrices for each corresponding disjoint path were generated uniformly in a random manner in the range of $0 < \xi_h \leq 1$, where $\xi_h$ is the cost of arc $a_h$. For both KPI and KPA schemes, we examine the average probability that $k$ disjoint paths are successfully found within a specified maximum allowable number of conflicts, $i_{max}$, over all source and destination node pairs for all generated cost matrices. The probability is defined as the success ratio of finding $k$ disjoint paths.

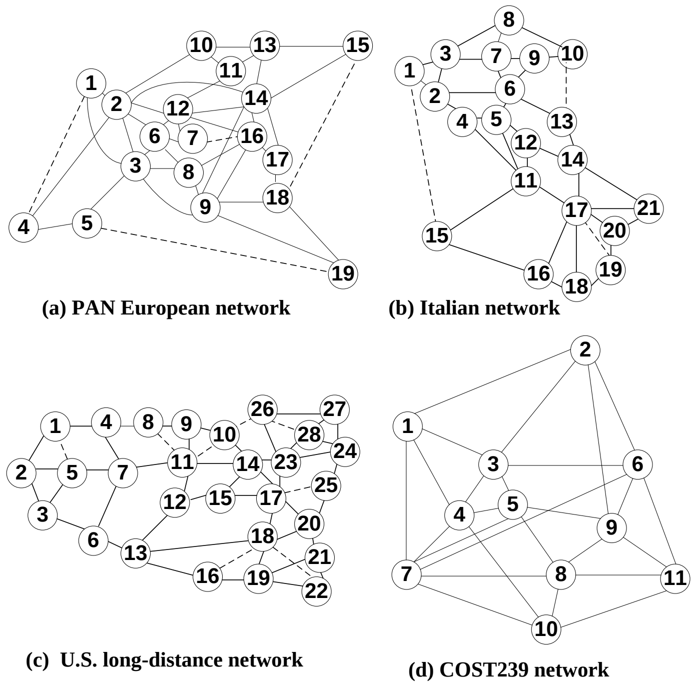


Fig. 5. (a) PAN European network (b) Italian network (c) U.S. long-distance network and (d) COST239 network

Figures 6 (a), (b) and (c) show the successful ratio that the KPI and KPA schemes find $k$ disjoint paths successfully within each number of conflicts on the PAN European network, the Italian network and the U.S. long-distance network, respectively. The results show how many cases each scheme is able to find $k$ disjoint paths successfully within assigned $i_{max}$ compared to all the cases at $i_{max}$ = 30. KPI is able to find a pair of disjoint paths, $k$ = 2, successfully, in the same way as the KPA scheme. Regardless of $i_{max}$, KPI has a higher success ratio than KPA in case of finding $k$ = 3 disjoint paths. In $k$ = 3 case, the KPI scheme yields success ratios of more than 99% with $i_{max}$ =10 for all networks, while the KPA scheme does not reach 99% when $i_{max}$ becomes large for the Italian network and U.S. long-distance network. Since the average node degree of the PAN European network is equal to four but the degree of each node is greater than or equal to three, the difference of the performance of KPI and KPA for finding $k$ = 3 disjoint paths on

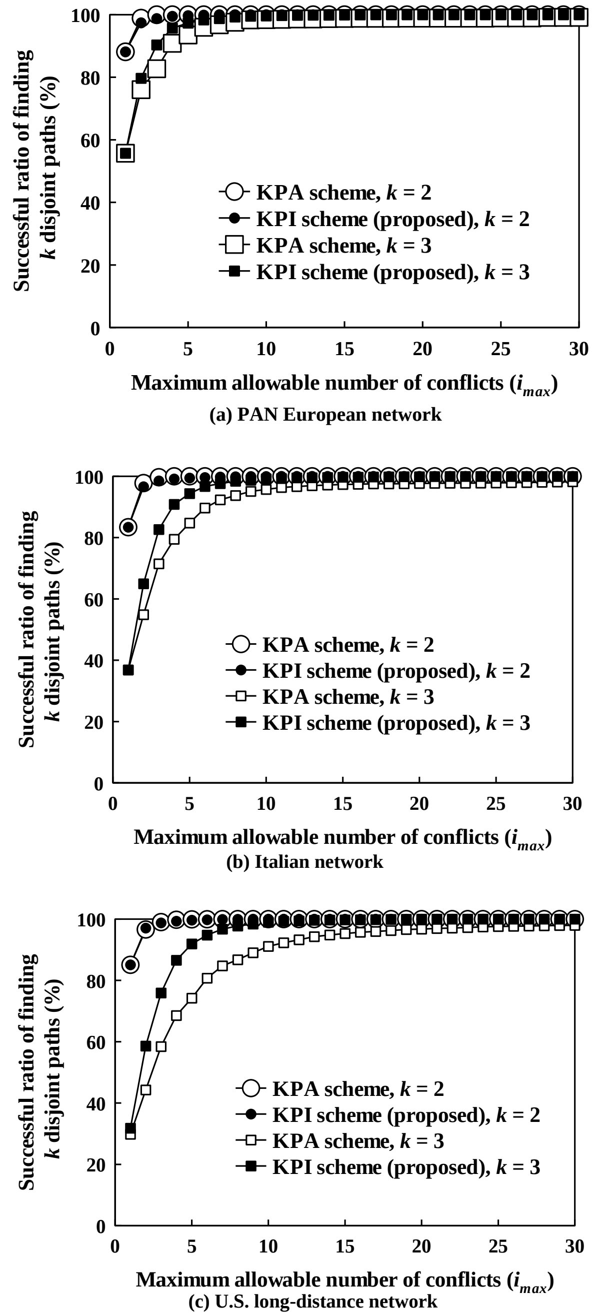


Fig. 6. Successful ratio of finding $k$=2 and $k$=3 disjoint paths (%) within specified maximum allowable number of conflicts, $i_{max}$ on (a) PAN European network (b) Italian network and (c) U.S. long-distance network.

the PAN European network is small. The successful ratio of the KPI and KPA schemes on the COST239 network, which has the degree of each node greater than four, is investigated. The required number of disjoint paths, $k$ =2, $k$=3 and $k$=4 are considered on this network topology as shown in Fig.7. The successful ratio of the KPI scheme has the same trend as the previous three networks. The successful ratio of KPI is the same as that of KPA in cases of finding $k$ =2 and $k$=3 disjoint paths but case of $k$=4, the successful ratio of KPI is higher than that of KPA. In the KPA scheme, the conflict path cost defined in Eq. (6) is set at too large a value, the conflicting arcs are always avoided. On the other hand, as the KPI scheme defines the conflict path cost according to Eq. (8), the conflicting arcs are appropriately utilized. Thus, the *k* disjoint paths are successfully found faster by KPI scheme when *k* is close to the value of degree of each node on each network.

The total cost of *k* disjoint paths, which is defined in Eq. (2), is lower with the KPI scheme than with the KPA scheme. According to the difference of the successful ratio for finding *k* disjoint paths on the PAN European network, the Italian network and the U.S. long-distance network are shown clearly with $k$=3, the normalized total costs of $k$=3 disjoint paths on these three networks are compared in this evaluation. Figure 8 shows comparison of the normalized total costs of $k$=3 disjoint paths, normalized by the total path cost of *k* disjoint paths by Bhandari's scheme. We used Bhandari's scheme, which is a scheme for finding *k* disjoint paths in single-cost network, to find *k* disjoint paths using only the arc cost matrix for the first path. After *k* disjoint paths are found by Bhandari's scheme, the total path cost in a multi-cost network is calculated by using Eq. (2) with three arc costs matrices. The normalized costs for Bhandari's, KPA, and KPI are taken as average values over all source and destination node pairs for all generated cost matrices. The results indicate that the KPI scheme yields lower total path cost of than KPA or Bhandari's scheme for the PAN European, the Italian and the U.S. long-distance multi-cost networks, as shown in Fig. 8. Bhandari's scheme yields the highest path cost among the three schemes, as it considers only the arc cost matrix for the first path to find *k* disjoint paths. Since, in the KPI scheme, the conflicting path cost is suitably estimated and the conflicting arcs are appropriately utilized, it returns the lowest total path costs.

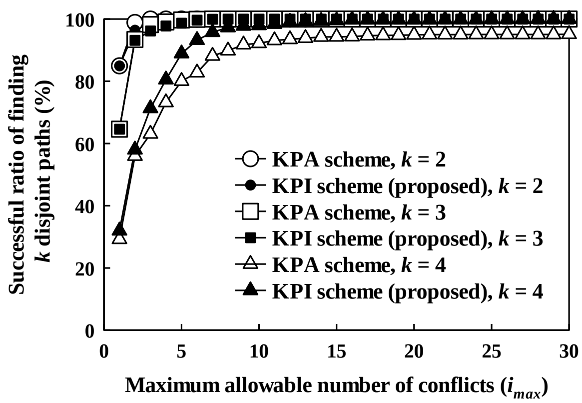


Fig. 7. Successful ratio of finding $k$=2, $k$=3 and $k$=4 disjoint paths (%) within specified maximum allowable number of conflicts, $i_{max}$ on COST239 network

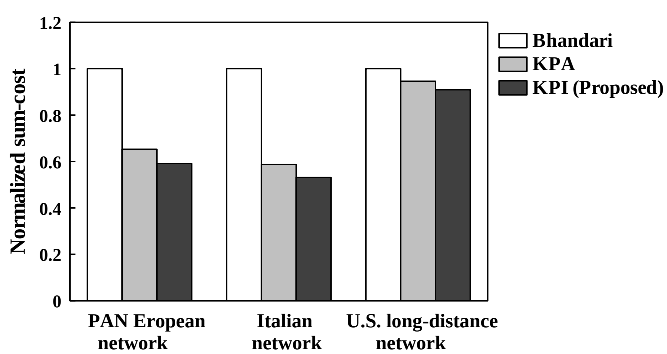


Fig. 8. Normalized summation of $k$=3 disjoint paths costs on PAN European network, Italian network and U.S. long-distance network using Bhandari's, KPA and KPI schemes.

## VI. Conclusion

This paper proposed KPI, *k*-penalty with initial arc costs matrix, to find *k* disjoint paths in a multi-cost network. The KPI scheme uses an initial arc cost matrix in estimating the conflicting path cost. Since the process of updating cost provides given penalty considering with initial arc costs matrix, selecting path that overlaps used paths is avoided. According to the numerical results, the KPI scheme is able to successfully find *k* disjoint paths faster and with lower total path cost than the KPA scheme. The successful ratio of finding *k* disjoint paths of KPI scheme reaches more than 99% at maximum allowable number of conflicts is equal to ten for all networks with the required number of disjoint paths are close to degree of each node on each network. The KPI is an efficient scheme to find disjoint paths. The deviation from the KPI scheme is not large, but the impact on overall performance is significant.

## References


[1] E. Oki and N. Yamanaka, "A Recursive Matrix-Calculation Method for Disjoint Path Search with Hop Link Number Constraints," *IEICE Trans. Commun.*, vol. E78-B, no. 5, 1995, pp. 769-774.

[2] E. Oki, N. Matsuura, K. Shiomoto, and N. Yamanaka, "A Disjoint Path Selection Scheme With Shared Risk Link Groups in GMPLS Networks, " *IEEE Commun. Lett.*, vol. 6, no. 9, Sep. 2002, pp. 406-408.

[3] Y. Guo, F. Kuipers, and P.V. Mieghem, "Link-Disjoint paths for reliable QoS routing, " *Int. J. Commun. Syst.* , Jun. 2003, pp. 779-798.

[4] D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He, "On the Complexity of and Algorithms for Finding the Shortest Path With a Disjoint Counterpart, " *IEEE/ACM Trans. on Networking*, vol. 14, no. 1, Feb. 2006, pp. 147-158.

[5] E.W. Dijkstra, "A note on two problems in connection with graphs, " *Num. Math.*, vol. 1, 1959, pp. 269-271.

[6] J.W. Suurballed, "Disjoint paths in a network," *Networks*, vol. 4, no. 2, 1974, pp. 125-145.

[7] R. Bhandari, *Surviable Networks: Algorithms for Diverse Routing.*, Kluwer Academic Publishers, 1999.

[8] D. A. Dunn, W. D. Grover, and M. H. MacGregor, "Comparison of *k*-Shortest Paths and Maximum Flow Routing for Network Facility Restoration," *IEEE J. Select. Areas Commun.*, vol. 12, Jan. 1994, pp. 88-99.

[9] J. Rak, "*k*-penalty: A Novel Approach to Find *k*-Disjoint Paths with Differentiated Path Costs," *IEEE Commun. Lett.*, vol. 14, no. 4, Apr. 2010, pp. 354-356.

[10] B.G. Jozsa, D. Orincsay, and A. Kern, "Surviving Multiple Network Failures Using Shared Backup Path Prtection," in *Proc. of the 8th IEEE Symp. on Comput. and Commun* , Jul. 2003, pp. 1333-1340.

[11] J. Tapolcai, P.H. Ho, D. Verchere, T. Cinkler, and A. Haque, "A New Shared Segment Protection Method for Survivable Networks with Guaranteed Recovery Time," *IEEE Trans. on Reliability*, vol. 57, no. 2, Jun. 2008, pp. 272-282.

[12] J. Rak and W. Molisz, "A new approach to provide the differentiated levels of network survivability under a double node failure," *11th Int. Conf. on Transparent Optical Networks (ICTON'09)*, Jul. 2009, pp. 1-4.

[13] R.Leepila, E. Oki and N. Kishi, "Scheme to find *k* disjoint paths in Multicost Networks," in *Proc. of IEEE Int. Conf. on Communications (ICC'11)*, Jun. 2011.

**Ruchaneeya Leepila** received B.S. degree in Applied Physics Science in 2002 and M.E. degree in Telecommunication Engineering in 2006 from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand. She is currently a Ph.D. candidate in Information and Communication Engineering at the University of Electro-Communications, Tokyo Japan. Her research interests include network survivability and restoration, network optimization and design.

**Eiji Oki** is an Associate Professor of The University of Electro-Communications, Tokyo Japan. He received B.E. and M.E. degrees in Instrumentation Engineering and a Ph.D. degree in Electrical Engineering from Keio University, Yokohama, Japan, in 1991, 1993, and 1999, respectively. In 1993, he joined Nippon Telegraph and Telephone Corporation's (NTT's) Communication Switching Laboratories, Tokyo Japan. He has been researching IP and optical network architectures, traffic-control methods, high-speed switching systems, and communications protocols. From 2000 to 2001, he was a Visiting Scholar at Polytechnic University, Brooklyn, New York, where he was involved in designing tera-bit switch/router systems. He joined The University of Electro-Communications, Tokyo Japan, in July 2008. He has published more than one hundred peer-reviewed journal, transaction articles, and international conference papers. He is active in standardization of Path Computation Element (PCE) and GMPLS in the IETF. He wrote ten IETF RFCs. He served as a Guest Co-Editor for the Special Issue on "Multi-Domain Optical Networks: Issues and Challenges," June 2008, in IEEE Communications Magazine, Guest Co-Editor for Special Issue on Routing, "Path Computation and Traffic Engineering in Future Internet," December 2007, in Journal of Communications and Networks, and Co-Chair of Technical Program Committee for Workshop on High-Performance Switching and Routing in 2006 and 2010, Track Co-Chair on Optical Networking for ICCCN 2009, and Co-Chair of Technical Program Committee for International Conference on IP+Optical Network (iPOP 2010). Dr. Oki was the recipient of the 1998 Switching System Research Award and the 1999 Excellent Paper Award presented by IEICE, and the 2001 Asia-Pacific Outstanding Young Researcher Award presented by IEEE Communications Society for his contribution to broadband network, ATM, and optical IP technologies. He co-authored two books, “Broadband Packet Switching Technologies,” published by John Wiley, New York, in 2001 and “GMPLS Technologies,” published by RC Press, Boca Raton, in 2005. He is an IEEE Senior Member and an IEICE Senior Member.

**Naoto Kishi** received B.E. and M.E. degree from the University of Electro-Communications, Tokyo, Japan, in 1982 and 1984 and the Ph.D. degree from the University of Tokyo, Tokyo, Japan in 1987, respectively. In 1987, he joined the Department of Electronic Engineering at the University of Electro-Communications as a Research Associate and became a Lecturer in April 1990. Currently, he is a Professor at the same university. From April 1990 to September 1991, he was an Overseas Research Fellow at the Optoelectronics Research Center of the University of Southampton, Southampton, England. His main research fields of interest are active and/or passive lightwave and microwave devices/systems, optical fiber circuits and numerical analysis of electro-magnetic fields. Dr.Kishi is a member of IEEE, the Optical Society of Japan, and the Optical Society of America. In 1989, he was the recipient of the fourth Shinohara Memorial Award by the IEICE.