Linear Formulation for Path Shared Protection

Pin-Han Ho\textsuperscript{1}, Janos Tapolcai\textsuperscript{1}, Hussein T. Mouftah\textsuperscript{2}, and Chi-Hsiang Yeh\textsuperscript{3}

Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada\textsuperscript{1}
\{pinhan,janos\}@bbcr.uwaterloo.ca

School of Information Technology and Engineering, University of Ottawa, Ontario, Canada\textsuperscript{2}
mouftah/site.uottawa.ca

Department of Electrical and Computer Engineering, Queen’s University at Kingston, Ontario, Canada\textsuperscript{3}
yeh@ee.queensu.ca

Abstract—This paper investigates the problem of optimal diverse routing for shared path-based protection in the complete routing information scenario on mesh optical networks, where a novel Integer Linear Programming (ILP) formulation is introduced such that the least-cost link-disjoint working and protection path-pair can be derived in a single step. The proposed ILP formulation is characterized by the facts that it is solvable with the commercially available Linear Programming (LP) solvers and that it can deal with the dependency between working and spare capacity in the network, which is a step ahead of the most state-of-the-art techniques in the design of diverse routing algorithms for shared protection. To verify the proposed ILP, an experiment is conducted to compare it with four reported schemes for end-to-end shared protection on two network topologies, namely APF-PBC, MLR, ITSA, and ILP-2S, where blocking probability is taken as the performance metric with connection requests being dynamically launched into the networks. The simulation results show that the ILP formulation yields the best performance while the ILP-2S scheme investigating less network states yields the worst. We also use the results by the proposed ILP to evaluate the four heuristic-based schemes adopted in the simulation in terms of two performance indexes – the percentage of optimality (denoted as $Q$) and the offset of optimality (denoted as $\%\text{opt}$).

Keywords: survivable routing, shared protection, Integer Linear Programming (ILP), spare link-state.

I INTRODUCTION

Dynamic survivable routing for shared protection has caught extensive research efforts for the past few years due to the ever increasing importance of survivability in the modern communication networks with dynamically changing traffic distribution [1-8]. The difficulty in realizing path shared protection lies in the dependency between the working and protection capacity. In order to deal with the dependency, the most straightforward approach is to use Two-Step-Approach [1] with two suites of custom-defined link-state respectively for solving the link- or node-disjoint working and protection path-pair one after the other. This idea is taken by the study in [2], where a method is provided to find an optimal (or least-cost) shared protection path with the corresponding working path given in advance. To address the allocation of the working path, the study in [3] inspects k-shortest paths between each S-D pair in an ascending order of cost until the least-cost working and shared protection path-pair is derived. The study in [4] further elaborates the Two-Step-Approach by introducing a link metric for solving the protection path considering the probability that the spare capacity along a link is taken by any restoration flow once a failure occurs. In [5], a link metric for solving a working path is defined, which can estimate a proper location for the working path so that a better chance for the protection path to find sharable spare capacity in the network is yielded.

The programming-based solutions for the shared protection problem can be seen in [6,7]. The study in [6] introduces an Integer Programming (InP) under the complete routing information scenario. However, the formulation is nonetheless non-linear and is not solvable by most commercially available LP solvers. The study in [7] provides a heuristic-based ILP formulation, where two scaling parameters are introduced to avoid the non-linearity possibly introduced when multiple states of spare capacity along each link are considered. An ILP formulation that can optimally finding a working and shared protection path-pair under the complete routing information scenario has never been reported before.

In addition to the end-to-end protection, the studies in [6,10,11,12,13,14] suggest to divide the working path into multiple segments, each of which is allocated with a protection segment. In [10], a heuristic algorithm is developed for this purpose, in which the backup bandwidth sharing will not be considered until the physical routes of the backup segments are defined. In [6] and [13], two related dynamic algorithms are proposed to switch over for each link from its immediate upstream node and merge back to the original path at any of the downstream nodes. It is notable that both of the studies do not impose any limitation on the length of the backup paths, which may impair the overall performance. In [11,12], a framework called Short Leap Shared Protection (SLSP) along with a dynamic algorithm called Cascaded Diverse Routing (CDR) is proposed to perform shared segment protection, in which k-shortest paths are enumerated in each segment of the working path to find the coresponding protection segment. CDR is reported to outperform its path-based shared protection counterpart at the expense of taking much larger computation complexity. The study in [14] provides an Active-Path-First based ILP formulation. The algorithm is characterized by the fact that all the possible ways of protection domain allocation are inspected and all the possible numbers of protection domains are iteratively tried. Since the formulation needs to have the working path first before the protection segments can...
be derived, the solution may be far from optimal if the working path is not well selected.

Although the shared segment protection can achieve a better throughput and a shorter restoration time, the implementation of it nonetheless takes extra signaling efforts and requires higher hardware responsiveness. In addition, solving the shared segment protection problem may incur a much higher computation complexity. Therefore, in this paper we turn back to the investigation of shared path-based protection, where an ILP is formulated such that the least-cost working and shared protection path-pair according to the current link-state can be jointly determined for a connection request. This is also the first linear formulation with a suit of link-state for solving the shared protection path (or called spare link-state) that considers the dynamic traffic distribution, in which all the possible states for the protection path to consume spare capacity can be defined. To realize this characteristic, a novel approach of graph transformation is adopted in the formulation, in which an additional residual graph for solving the protection path is introduced. We implement the ILP on two network topologies with dynamically arrived connection requests to verify the correctness. For a comparison purpose, four most recently reported schemes are taken in the experiment, namely Active-Path-First Potential-Backup-Cost (APF-PBC) [5], Iterative Two-Step-Approach (ITSA) [1,11], Maximum Likelihood Relaxation (MLR) [1], and ILP-2S [8].

The paper is organized as follows. Section II introduces the cost function for working and protection paths. Section III introduces the proposed ILP based on the cost function and link metric presented in Section II. Section IV verifies the proposed scheme by comparing it with the proposed schemes for shared path-based protection. Section V concludes the paper.

II DEFINITION OF COST FUNCTION

This section defines the cost function and link metrics for solving each of the working and shared protection paths. Given a network \( G(N,E) \) with \( N \) and \( E \) being the set of nodes and directional links, respectively, the cost function for finding working path \( w \) with a bandwidth \( B(w) \) is as follows:

\[
    c_{w_j} = \begin{cases} 
    \infty & \text{if link } j \text{ is not reservable} \\
    B(w) \cdot c_j & \text{otherwise} 
    \end{cases} 
\]

(1)

where \( c_j \) is the cost for each unit of bandwidth taken by \( w \) along link \( j \). Note that \( c_j \) is custom-designed, and can either be a constant, or take dynamic network traffic into consideration. The fact that link \( j \) is not reservable by \( w \) can be because \( B(w) > F_j \), where \( F_j \) is the free capacity along link \( j \).

For solving the protection path of \( w \) (denoted as \( wp \) with a bandwidth of \( B(wp) \)), we need to define the corresponding link-state (or referred to as spare link-state). In this study three states are identified for a link to be consumed by \( wp \); (1) the case where the link has sufficient sharable spare capacity, in which the protection path can take this link with the smallest cost (i.e., \( \varepsilon \) in this study); (2) the case where the link does not have sufficient (or does not have any) sharable spare capacity, and the protection path must partly (or totally) take free capacity along this link with an extra cost. In this case the spare link-state is \( r_j^w \cdot c_j \). (3) The link does not have sufficient sharable spare capacity and free capacity, in which \( wp \) cannot traverse through this link by any means. In this case the spare link-state is \( \infty \). In the above, \( r_j^w \) is a scaling parameter with a value between 0 and 1 determined by the amount of sharable and non-sharable spare capacity along link \( j \) at the presence of \( w \), and will be defined later in Eq. (2). \( \varepsilon \) is a small number which is designed as \( \varepsilon = \min_{\varepsilon < \infty} c_j \). The motivation for imposing a small cost upon taking spare capacity is to avoid the situation that the protection path is redundantly long, which not only increases the restoration time but also impairs the capacity-efficiency.

Fig. 1 illustrates the capacity distribution along a specific link \( j \). \( V_j \), \( S_j^w \), and \( sh_j^w \) are the amount of spare capacity, non-sharable spare capacity, and sharable spare capacity along link \( j \), respectively. \( V_j \) is defined in the standard link-state library, while \( S_j^w \) and \( sh_j^w \) cannot be determined until the presence of \( w \) due to the dependency between working and protection capacity. Please refer to [1] for detailed descriptions for deriving \( S_j^w \) and \( sh_j^w \) for \( j \in E \).

With the categorization of capacity along link \( j \), \( r_j^w \) is defined as:

\[
    r_j^w = \begin{cases} 
    1 - sh_j^w / B(w) & \text{if } sh_j^w + F_j \geq B(w) > sh_j^w \\
    \infty & \text{if } sh_j^w + F_j < B(w) 
    \end{cases} 
\]

(2)

for any link \( j \in E \). It is clear that \( r_j^w \) is 1 if there is not any sharable spare capacity available along link \( j \). The spare link-state can be expressed as:

\[
    cp_j^w = \begin{cases} 
    c_j \cdot [1 - sh_j^w / B(w)] & \text{if } sh_j^w + F_j \geq B(w) > sh_j^w \\
    \varepsilon \cdot c_j & \text{if } sh_j^w \geq B(w) \\
    \infty & \text{if } sh_j^w + F_j < B(w) 
    \end{cases} 
\]

(3)

for \( j \in E \) and \( j \notin LW \)

where \( LW \) is all the links traversed by \( w \). Fig. 2 shows the four situations defined in Eq. (2). In Fig. 2(a) and Fig. 2(b), \( wp \) may partly take the free capacity and the sharable spare capacity; therefore, the link cost is defined as \( c_j \cdot [1 - sh_j^w / B(w)] \), which is shown in condition (i) of Eq. (2). It is clear that as \( sh_j^w = 0 \) the cost for the protection path taking the link is \( c_j \), which is the same as that taken by the working path. In Fig. 2(c), all bandwidth of \( wp \) can take sharable spare capacity, therefore, the cost is \( \varepsilon \cdot c_j \) as shown in condition (ii) of Eq. (2). In Fig. 2(d), the link cost is infinity because \( wp \) cannot be supported by the residual capacity of the link, which is shown in condition (iii) of Eq. (2). Note that a protection path will take sharable spare capacity along a link whenever it is available. If there is not enough sharable spare capacity, the protection path takes free capacity after consuming all the sharable spare capacity.
Note that $s_{h}^{w}$ and $S_{w}$ are network-wide link-state specific to the presence of $w$. Let $w$ traverse through a set of links, $LW$. Any link $l \in LW$ is, in turn, traversed by a set of working paths denoted as $D_{w}$, which are also the working paths currently involved in the common SRLG of link $l$ with $w$. The derivation of $S_{w}$ and $s_{h}^{w}$ can also be expressed in a matrix form, which is a graceful expression for determining spare capacity along each link by Y. Liu [15]. In this case, we define the working and protection path-link incidence matrices as $A^{w}$ and $B^{w}$, in which $A^{w}$ is a $[D_{w} \times |E|]$ array containing all the working paths that are involved in a common SRLG (i.e., have any common physical link) with $w$, while $B^{w}$ is an $[D_{w} \times |E|]$ array containing all protection paths corresponding to the working paths in $D_{w}$. Here we define $D_{w} = \bigcup_{l \in LW} D_{l}$, where $\bigcup$ is a union operation.

The spare provision matrix for $w$ is defined as $C^{w} = (B^{w})^{T} \cdot A^{w}$, which is a $[|E| \times |E|]$ matrix. Applying a MAX operation upon each row of $C^{w}$ will yield a $1 \times |E|$ vector $S^{w}$, which keeps the amount of non-sharable spare capacity along each link provided with $w$. The $1 \times |E|$ vector $SH^{w}$, which keeps the amount of sharable spare capacity along each link provided with $w$, can thus be derived by the relationship $SH^{w} = V - S^{w}$, where $V$ is a $1 \times |E|$ vector recording the amount of spare capacity along each link.

### III LINEAR FORMULATION FOR SHARED PROTECTION

Let the source and the destination of the connection request be denoted as $s$ and $d$. The graph for solving $w$ is denoted as $G_{w}(N, E_{w})$, which is composed of the links of $B(w) \leq F_{j}$. An additional residual graph is defined to facilitate solving $wp$ and is denoted as $G_{p}(N, E_{p})$. We need this graph to record the spare link-state, which is composed of the links with the amount of free capacity $F_{j}$ plus that of the spare capacity $V_{j}$ larger than or equal to $B(w)$ (i.e., $B(w) \leq F_{j} + V_{j}$ for every link $j$). Please refer to Fig. 3 for an illustration of the graph transformation.

The objective function for the ILP is as follows:

\[
\sum_{(a,b) \in E_{w}} c_{a,b} \cdot x_{a,b} + \sum_{(a,b) \in E_{p}} (c_{a,b} \cdot r_{w} + \varepsilon) \cdot y_{w,v} \tag{3}
\]

where $c_{a,b}$ and $\varepsilon$ have the same definitions as that in Section II. $x_{a,b}$ (with a size of $|E_{w}|$) and $y_{a,b}$ (with a size of $|E_{p}|$) is a 0-1 binary variable indicating flows along the $w$ and $wp$ on link $(a,b)$, respectively, which is 1 if the corresponding flow passes $(a,b)$. It is clear that the concatenation of all the links with $y_{a,b} = 1$ yields $wp$, while the concatenation of all links with $x_{a,b} = 1$ yields $w$. $r_{w,v}^{'}$ is a variable for scaling the cost of the protection path taking spare capacity along link $(a,b)$. We will show later that $r_{w,v}^{'}$ is equivalent to $r_{w,v}$ defined in Section II.

![Fig. 2. The four situations defined in Eq. (2).](image)

![Fig. 3. Link transformations for the edges in $G$, which yields the graphs $G_{w}$ and $G_{p}$ respectively.](image)
The following constraint imposes the link bandwidth limitation upon the consumption of spare capacity.

\[ x_{u,v} + y_{u,v} \leq \forall (u,v) \in E_{p}, v(u,v) \in E_{p}, S^{a,b}_{u,v} + F_{u,v} < B(w) \quad (8) \]

Eq. (8) states that if \( S^{a,b}_{u,v} + F_{u,v} < B(w) \), link \((a,b)\) and \((u,v)\) cannot be used at the same time for \( w \) and \( wp \) since there would be insufficient free and spare capacity for \( wp \) along \((u,v)\) given that \( w \) takes \((a,b)\). At last, the variables \( r'_{u,v} \) are required to be positive, which results in the following constraint:

\[ r'_{u,v} \geq 0 \forall (u,v) \in E_{p} \quad (9) \]

It is clear that the adoption of the second graph has successfully addressed all the three link-state for a protection path to take spare capacity, which are the case of \( sh_{u,v}^{a,b} \geq B(w) \), the case of \( S^{a,b}_{u,v} + F_{u,v} \geq B(w) > sh_{u,v}^{a,b} \), and the case of \( s_{u,v}^{a,b} + F_{u,v} < B(w) \). The former two cases are jointly defined by Eq. (7) and Eq. (9), where \( r'_{u,v} \) is constrained no smaller than 0 and \( 1 - sh_{u,v}^{a,b}/B(w) \) in the two cases, respectively; while the latter case is defined by Eq. (8), in which the traversal of \( wp \) through \((u,v)\) is prohibited if there is no sufficient capacity along the link. Therefore, \( r'_{u,v} \) is equivalent to \( r_w \) defined in Eq. (2), and the dynamic survivable routing problem for shared protection has been formulated with the identical cost function and link-states defined in Section II. It can also be observed that the use of the two residual graphs along with the constraints of Eq. (7), Eq. (8) and Eq. (9) imposes a link bandwidth limitation constraint separately upon the selection of \( w \) and \( wp \).

Note that \( sh_{u,v}^{a,b} \) in the formulation can be defined off-line according the current link-state, which stands for the upper bound of the amount of spare capacity along link \((u,v)\) sharable by the protection path if the corresponding working path passes through link \((a,b)\). According to Fig. 1, we have \( sh_{u,v}^{a,b} = V_{u,v} - S^{a,b}_{u,v} \), where \( V_{u,v} \) is assumed already known, and \( S^{a,b}_{u,v} \) can be derived by:

\[ S^{a,b}_{u,v} = \sum_{p} B(p) \cdot \delta^{a,b}_p \cdot \sigma^{a,b}_p \quad \text{for all working path } p \quad (10) \]

where \( B(p) \) is the bandwidth of working path \( p \), and \( \delta^{a,b}_p \) and \( \sigma^{a,b}_p \) are binary indicators defined as follows:

\[ \delta^{a,b}_p = \begin{cases} 1 & \text{if } p \text{ has its protection path traversing } (u,v) \\ 0 & \text{otherwise} \end{cases} \]

\[ \sigma^{a,b}_p = \begin{cases} 1 & \text{if } p \text{ passes } (a,b) \\ 0 & \text{otherwise} \end{cases} \]

In other words, the demand for spare capacity upon link \((u,v)\) from the working capacity along link \((a,b)\) is the summation of all working bandwidth on link \((a,b)\) that has the corresponding spare capacity along link \((u,v)\). We need to add all the working capacities along link \((a,b)\) together because they are in the same SRLG; as a result, their corresponding protection paths cannot share any spare capacity along link \((u,v)\).

IV VERIFICATION

An experiment is conducted to verify the proposed ILP model and compare it with four reported counterparts, namely APF-PBC [5], ITSA [11, 12], MLR [1], and ILP-2S [8], in terms of blocking probability on two networks topologies. The experiment is arranged as follows. Each directional link in the networks contains 32 units of bandwidth. We consider the capacity efficiency in terms of blocking probability for the dynamically arrived connection requests following the Poisson model and with a holding time defined in an exponential distribution function. The bandwidth for each connection request is randomly distributed among 1–3 units. In the experiment arrangement, node-pair \((i,j)\) has a traffic load \( \rho_{i,j} = \eta \cdot \lambda_{i,j} / \mu_{i,j} \), where \( \lambda_{i,j} \) and \( \mu_{i,j} \) are arrival and departure rate upon the node-pair \((i,j)\), respectively. Without loss of generality, \( \mu_{i,j} \) is set to 1, while \( \lambda_{i,j} \) is a random number between 0.5–1.5 for every node-pair \((i,j)\) such that each node-pair may be subject to different amount of path connection setup demand. The scaling parameter \( \eta \) represents the levels of traffic load in the network with the unit of Erlang. Each data point in the figures is the blocking probability for 50,000 connection requests using a specific survivable routing algorithm. The confidence interval is within 0.1% if we take the result of 100 connection requests as a trial. The network topologies adopted in the simulation have 22 and 30 nodes with 88 and 126 directional links, respectively.

The implementations of each scheme taken for comparison in this experiment is briefly as followed. For APF-PBC, the cost function for the working path is:

\[ L_{w} = 1 + \max_{j} \frac{\max_{j} S_{j}^{l}}{\max_{j} S_{j}^{l}} \]

where \( S_{j}^{l} \) is the total spare capacity required along link \( j \) to protect the working capacity along link \( l \). With ITSA, the program returns a result either when 20 iterations are performed or when the least-cost working and shared protection path-pair is derived. If the algorithm fails to find a feasible working and protection pair within 20 iterations, a blocking is announced for the connection request. The parameter \( c_{j} \) in Eq. (1) is defined as: \( c_{j} = 1 / r c_{j} \), where \( r c_{j} \) is the residual bandwidth along link \( j \). With MLR, an easy link is defined as a network link with sufficient sharable spare capacity to protect a specific working path. The searching of the working path is conducted through a modified Dijkstra’s relaxation process, in which a temporary label of node \( n \) (denoted as \( L[n] \) for node \( n \) given by node \( x \) is defined as the link cost \( c_{x,n} \) divided by the log of the number of Easy Links for \( \pi(s,n) \), where \( \pi(s,n) \) is the working path segment recorded currently from the source to node \( n \). The label replacement at node \( n \) by node \( x \) will be conducted in such a way that \( L(n) \) is the minimal; i.e.,

\[ L[n] = \min \{ L[n], L[x] + c_{x,n}/\log(\text{offset}(x,n) + 1) \} \]

where \( \text{offset}(x,n) \) is the reduction on the amount of easy links due to the newly relaxed link \((x,n)\). Please refer to Appendix B for further descriptions of the algorithm.

It is clear that each MLR and APF-PBC is a scheme aiming to estimate the location of the working path such that the corresponding protection path has a better chance to take sharable spare capacity along each link; while ITSA exhaustively searches among the k-shortest paths to find the
best working and shared protection path-pair. All the above three schemes are operated according to a specific working path given in advance (i.e., each of them is an Active Path First approach) and adopt the same approach for deriving the protection path using the cost function defined above.

The ILP formulation without the additional graph for solving the protection path is denoted as ILP-2S, in which two states for the protection path to consume spare capacity can be addressed (i.e., a link either has or does not have sufficient sharable spare capacity for the protection path). To realize the two-state case, for each connection request, all the links in the network with \( F_j < B(W) \) are excluded from the residual graph before the formulation is solved. Therefore, both working and protection paths are solved in the same graph (i.e., \( E_a \)). In such a circumstance, some links with \( sh^w + F_j \geq B(W) > F_j \) will never be considered in routing the protection path. To make a clear comparison with ILP-3S, the formulation of ILP-2S is presented in Appendix A of the paper. Both ILP-2S and ILP-3S are solved with CPLEX 7.5. All the cases in this study are implemented using a Sun Ultra 80 workstation.

The simulation results are shown in Fig. 4, which demonstrate that the proposed scheme ILP-3S yields the best performance in terms of blocking probability, while the ILP-2S scheme yields the worst. The MLR and APF-PBC schemes achieve a close performance with each other, which are, however, outperformed by the ITSA scheme adopting the policy of exhaustive search.

We also investigate the total cost of the working and protection path-pair for each connection request in all the above cases. It is notable that ILP-3S can always yield a better solution than that by the other cases in the simulation all through the 50,000 connection requests, which can serve as one of the evidences for the optimality that can be achieved by the proposed ILP formulation. The other four schemes are evaluated by the results of ILP-3S in terms of the following two performance indexes: the percentage of optimality (denoted as \( %\text{opti} \)), which is the percentage of the launched connection requests being optimally allocated by a specific scheme; and the offset of the optimality, which is denoted as \( Q \), and is defined as \( Q = C_a / C_{\text{opt}} \), where \( C_a \) is the total cost of a working and protection path-pair for a connection request by using a specific heuristic scheme, while \( C_{\text{opt}} \) is the cost achieved by solving the ILP-3S formulation under the same network environment.

Table I shows the experiment statistics and the average computation time for each connection request in the five cases. The ITSA scheme yields 90.2% of \( %\text{opti} \) and 4.3% of \( Q \) value, which can find approximate optimal solutions with much less computation time (around 1/5) compared with that taken by the ILP-3S scheme. The MLR and APF-PBC schemes yield similar results and take a close amount of computation time. The ILP-2S scheme, however, is outperformed by all the other schemes in both of the performance indexes and the computation time. From the simulation results, it is clear that the optimality of the working and protection path-pair for each connection request is a dominate factor for the overall performance in terms of blocking probability.

**Table I.**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>( %\text{opti} )</th>
<th>( Q )</th>
<th>Time(in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITSA</td>
<td>90.2%</td>
<td>4.3%</td>
<td>1.15</td>
</tr>
<tr>
<td>MLR</td>
<td>80.9%</td>
<td>11.7%</td>
<td>0.21</td>
</tr>
<tr>
<td>APF-PBC</td>
<td>78.6%</td>
<td>13.5%</td>
<td>0.16</td>
</tr>
<tr>
<td>ILP-2S</td>
<td>73.1%</td>
<td>17.2%</td>
<td>0.46</td>
</tr>
<tr>
<td>ILP-3S</td>
<td>100%</td>
<td>0%</td>
<td>5.18</td>
</tr>
</tbody>
</table>

**V CONCLUSIONS**

This paper has demonstrated a novel ILP formulation for solving the shared path-based protection problem in the complete routing information scenario, which is a step ahead of the most state-of-the-art techniques for dynamic diverse routing in optical networks. We first introduce the cost function and the link metric which incorporate with the ILP formulation, based on which the link-disjoint working and shared protection path-pair with the least-cost sum according to the current link-state can be derived for a connection request. We have also shown that the ILP formulation can successfully deal with the dependency of working and spare capacity in the network, in which the three states for routing the protection path can be well defined. To verify the proposed scheme, simulation is conducted to implement the ILP formulation along with four reported heuristic-based counterparts on the two network topologies with dynamically arrived connection requests. It is observed that the proposed ILP achieves the lowest blocking probability under various traffic loads. We conclude that, in addition to serving as an effective on-line algorithm, the proposed ILP can be taken to evaluate some other heuristic-
based diverse routing algorithms. For this purpose, the simulation also examines the computation time along with the simulation statistics for each of the schemes in terms of two performance indexes: the percentage of optimality ($%\text{opti}$) and the offset of optimality ($Q$). It is observed that the optimality of derived working and protection path-pairs behaves as a dominate factor of the overall performance in terms of blocking probability.

REFERENCES