Link Weight Design Model with Traffic-Engineering Links based on Preventive Start-Time Optimization against Link Failures

Mei Nakashima†, Takashi Kurimoto‡, and Eiji Oki†
†Kyoto University, Kyoto, Japan ‡National Institute of Informatics, Tokyo, Japan

***Abstract*—In an Internet protocol network running a link-state routing protocol, determining link weights corresponds to determining a route from a source to a destination that the protocol selects to minimize the sum of the link weights. Previous studies have focused on determining physical link weights to reduce network congestion ratio in case of link failures. However, no study has yet to address a model that determines link weights by including traffic-engineering (TE) links and investigate the effect of including TE links on reducing network congestion. The link-state routing protocol treats a TE link as a logical, direct link between non-adjacent nodes. This paper proposes a link-weight design model with TE links based on preventive start-time optimization (PSO) for handling single link failures, called PSO-TE. The model considers physical and TE links when determining the link weights under the assumption of all single physical link failures. It identifies the set of link weights that minimizes the worst-case network congestion ratio across all considered failure patterns. The numerical findings demonstrate that PSO-TE effectively reduces the worst-case network congestion ratio compared to the baseline model that does not employ TE links, highlighting the advantage of including TE links.**

***Index Terms*—routing, link weights, traffic engineering links, link failure**

I. Introduction

An Internet protocol (IP) network transmits various applications, including streaming videos, online conferences, and cloud services [1], [2]. To ensure these applications maintain high quality, designing and managing an IP network effectively to prevent congestion is crucial. Additionally, an IP network must be resilient against failures caused by fiber cuts or physical interface issues.

IP networks widely use link-state protocols, including open shortest path first (OSPF) [3] and intermediate system to intermediate system (IS-IS) [4]. In a network running a link-state routing protocol, determining link weights corresponds to determining a route from a source to a destination that the protocol selects to minimize the sum of the link weights [5], [6]. Packets are forwarded by referring to the next hop information at each router's routing table fed by the routing protocol.

Given the traffic demand, a network operator needs to determine a suitable set of link weights, considering failures. Three policies exist for determining link weights that reduce network congestion ratio: start-time optimization (SO), run-time optimization (RO), and preventive start-time optimization (PSO).

SO determines link weights at the start of the network operation; SO does not consider the network's dynamic changes caused by link failures. Once a link failure occurs, the network's topology changes and the optimal set of link weights determined by SO may no longer be optimal. SO can cause unexpected network congestion. A more robust strategy than SO is necessary as a network operator's essential task is dealing with the daily link failures [7].

RO is a strategy that considers determining the link weights each time the traffic demand changes to overcome SO's weakness. This strategy minimizes network congestion even during link failures. However, routing information is advertised to all routers whenever the link weights are updated, resulting in network instability.

PSO is a strategy that determines a set of link weights preventively at the start time to minimize the worst-case network congestion ratio over possible link failure scenarios. The PSO strategy was originally introduced in [8]. PSO can suppress increasing network congestion and avoid instability due to link failure, which are the weaknesses of SO and RO, respectively.

Traffic engineering contributes to reducing network congestion and improving overall network resource utilization. In today's Internet, OSPF is a widely used intra-domain routing protocol [3]. Since OSPF always forwards packets on the shortest path, it must be more flexible to facilitate traffic engineering. Even if the shortest path is congested and there is a less congested alternative path, there is no ability to divert packets through the alternative path.

Using the OSPF framework, we can reroute packets through alternate paths by setting logical links, each of which directly connects non-adjacent nodes, called traffic-engineering (TE) links [9]. Adopting TE links can potentially improve the network's performance by controlling traffic flow. The document in [9] describes extensions that primarily use opaque link state advertisements (LSAs) to describe the TE topology and provide a way to distribute this information.

However, previous studies have focused on determining physical link weights based on PSO to reduce network congestion in the event of link failures. No studies have yet incorporated TE links into PSO to optimize link weights. A question arises: *Can network congestion be further reduced by incorporating TE links under PSO?*

This paper proposes a link-weight design model with TE links based on PSO for handling single link failures, called PSO-TE. The model considers physical and TE links when

This work was supported in part by ROIS NII Open Collaborative Research 24S0102.

determining the link weights under the assumption of all single physical link failures. It identifies the set of link weights that minimizes the worst-case network congestion ratio across all failure patterns. The numerical results demonstrate that PSO-TE effectively reduces the worst-case network congestion ratio compared to PSO without TE links. We also examine the effect of PSO, including TE links, by comparing the network congestion ratio of SO with TE links (SO-TE) and PSO-TE. PSO-TE introduces a practical method for working within the existing framework of static link weights without modifying the routing protocols or the routers themselves by using the TE links.

II. Problem descriptions

A. Network definitions

A directed graph $G(V,E)$ represents a network, where V is a set of nodes and E is a set of links. $(i,j) \in E$ denotes a link from node $i \in V$ to node $j \in V \setminus \{i\}$. $e \subseteq E$ also denotes a link including both directions, where $e = \{(i,j),(j,i)\}$. c_{ij} is the capacity of $(i,j) \in E$. $F = \{0, 1, \cdots, \frac{|E|}{2}\}$ is the set of link failure indices l. Failures occur in both directions, so link $l = \{(i,j),(j,i)\}$ get to be out of service. $l = 0$ means no link failure and $l (\neq 0)$ means the failure of link $e = l \subseteq E$, which can also express $l \in F$ when we focus a failure index. G_l denotes G has link failure at $e = l\ (\neq 0)$. $C = \{c_{ij}|(i,j) \in E\}$ denotes a set of link capacity c_{ij} of $(i,j) \in E$. u_{ij} denotes the traffic volume of passing through $(i,j) \in E$. We call $\frac{u_{ij}}{c_{ij}}$ a link utilization ratio of $(i,j) \in E$. $W = \{w_{ij}|(i,j) \in E\}$ is the link weight matrix of network G, where w_{ij} is the weight of $(i,j) \in E$, where $w_{ij} \in [1, w_{\max}]$ is a positive integer number. $x_{ij}^{pq}(W)$ is the portion of traffic from node $p \in V$ to node $q \in V \setminus \{p\}$ routed through $(i,j) \in E$, where $(p,q) \in P$. P is a set of source and destination node pairs in the network, where $|P| = |V|(|V|-1)$. If a link weight set W is given, routing and $x_{ij}^{pq}(W)$ are determined based on the shortest path routing; we adopt equal-cost multi-path routing (ECMP). ECMP evenly splits traffic among equal-cost paths [10]. $x_{ij}^{pq}(W)$ is a load-distribution variable under link weights set W, where $0 \leq x_{ij}^{pq}(W) \leq 1$. t_{pq} represents the traffic demand of source and destination node pair $(p,q) \in P$. $T = \{t_{pq}|(p,q) \in P\}$ denotes a traffic matrix.

u_{ij} denotes the total traffic demands passing through $(i,j) \in E$. u_{ij} is given by:

$$u_{ij} = \sum_{(p,q) \in P} t_{pq} x_{ij}^{pq}. \tag{1}$$

B. Introducing TE links

There are two link types: physical and TE links. Directed graph $G'(V,E')$ denotes a network including physical and TE links, where E' is a set of physical links and TE links. Physical link $(i,j) \in E$ is the same as that of Section II-A. The newly introduced type of link is a TE link established between node pair $(u,v) \in P_{\mathrm{T}} \subseteq P$; (u,v) is a node pair of $u \in V$ and $v \in V \setminus \{u\}$. P_{T} is a set of node pairs, each with at least one TE link.

The union set of physical link set E and the TE link set E_{T} is the total link set of the network; $E' = E \cup E_{\mathrm{T}}$. We set P_{T} to satisfy $|P_{\mathrm{T}}| = m$. We can set up to k TE links in each node pair $(u,v) \in P$. The set of TE links between node pair $(u,v) \in P$ is E_{T}^{uv}, where $E_{\mathrm{T}}^{uv} \subseteq E_{\mathrm{T}}$ and $|E_{\mathrm{T}}^{uv}| \leq k$. E_{T} is the union of E_{T}^{uv} for all $(u,v) \in P_{\mathrm{T}}$, i.e., $\bigcup_{(u,v) \in P_{\mathrm{T}}} E_{\mathrm{T}}^{uv}$.

We introduce the following two policies for selecting the set of node pairs P_{T} and routing each TE link over physical links. First, we chose the $|P_{\mathrm{T}}| = m$ node pairs whose traffic demand t_{uv}, where $(u,v) \in P$, is within the m highest from non-neighboring node pairs with the shortest hop score H or more; we set H to two. Second, we select at most k paths with the smallest number of hops in each node pair $(u,v) \in P_{\mathrm{T}}$. We use Yen's algorithm to find the k shortest hop paths [11]. If more than k paths with the smallest number of hops exist, we arbitrarily select k paths with the least overlapping score of physical links through which they pass. We define the overlapping score as the sum of the numbers of overlapping paths for all $(i,j) \in E$. For example, if there are three overlapping paths on links $(i,j) = (1,2)$ and two overlapping paths on links $(i,j) = (2,4)$, respectively, the overlapping count is five.

We update the definition of link utilization u_{ij}, which denotes the total traffic demands passing through $(i,j) \in E$. Each traffic through a TE link consumes the link capacity of the physical link, so u_{ij} is given by:

$$u_{ij} = \sum_{(p,q) \in P} \sum_{e' \in E' : e' \text{uses } (i,j)} t_{pq} {x'}_{e'}^{pq}. \tag{2}$$

$T = \{t_{pq}|(p,q) \in P\}$ denotes a traffic matrix. ${x'}_{e'}^{pq}$ is the portion of traffic from node $p \in V$ to node $q \in V \setminus \{p\}$ routed through $e' \in E'$, where $(p,q) \in P$. If a link weight set W is given, routing and ${x'}_{e'}^{pq}$ are determined based on the shortest path routing adopting ECMP. ${x'}_{e'}^{pq}$ is a load-distribution variable under link weights set W, where $0 \leq {x'}_{e'}^{pq} \leq 1$.

We define link failure as follows. We consider only a failure of at most one physical link including both directions $l \subseteq E$, where $l = \{(i,j),(j,i)\}$. The set of physical link failures is F, where $F = \{0, 1, \cdots, \frac{|E|}{2}\}$. We express a physical failure index as $l \in F$. $l = 0$ means no link failure and $l\ (\neq 0)$ means the failure of link $l \subseteq E$, which can also express $l \in F$ when we focus a failure index. $E'_l \subseteq E'$ denotes the set of links in E' when link l fails. When a physical link l fails, TE links in E_{T} fail each of which passes through the physical link l. $G'_l(V, E'_l)$ expresses a network with link failure $l\ (\neq 0)$.

Figure 1 shows an example of a network, where $|P_{\mathrm{T}}| = m = 1$ and $k = 3$. A link without an arrow indicates a bidirectional link and a link with an arrow indicates a directed link. We have $(u,v) = (0,2) \in P_{\mathrm{T}}$. Figure 1(a) shows $G(V,E)$ representing a physical network. Figure 1(b) shows physical and TE links. There are three paths with a minimum hop path of two between $(u,v) =$

$(0,2)$. Figure 1(c) shows $G'(V,E')$, which puts the physical and TE links together. Figure 1(d) shows $G'_l(V,E'_l)$, where failed physical link $l = \{(2,5),(5,2)\}$; the TE link between node pair $(0,2) \in P_{\mathrm{T}}$ passing through the failed physical link l also fails.

C. Link weight design

The network congestion ratio, r, denotes the maximum value of all link utilization ratios in the network; $r = \max_{(i,j)\in E} \frac{u_{ij}}{c_{ij}}$, where $0 \le r \le 1$.

W_{cand} is the set of candidate W for which the worst-case congestion is calculated. $r(W,l)$ is a function that returns the congestion ratio for G_l with OSPF-based shortest path routing using link weights in W. $R(W)$ mentions the worst-case congestion ratio in W among all link failure scenarios $l \in F$.

$R(W)$ is defined by:

$$R(W) = \max_{l\in F} r(W,l). \tag{3}$$

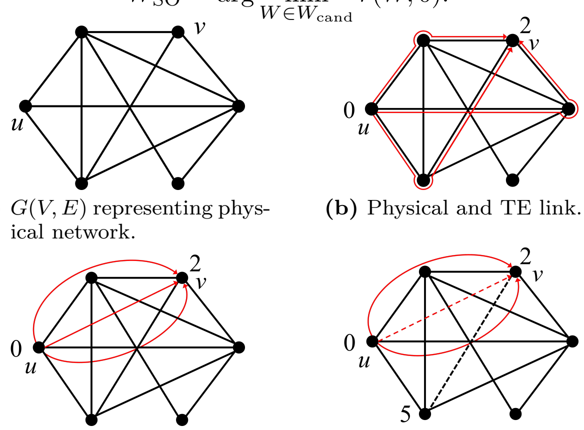
PSO aims to find a link weight set W_{PSO} that minimizes $R(W)$ for a network $G(V,E)$. W_{PSO} is defined by:

$$W_{\mathrm{PSO}} = \arg \min_{W\in W_{\mathrm{cand}}} R(W). \tag{4}$$

The network congestion ratio achieved by using W_{PSO} is $R(W_{\mathrm{PSO}})$, which represents the upper bound of congestion for a single link failure scenario in the network.

On the other hand, the SO's goal is to find a link weight set W_{SO} that minimizes $r(W,0)$ for a network $G(V,E)$. W_{SO} is defined by:

$$W_{\mathrm{SO}} = \arg \min_{W\in W_{\mathrm{cand}}} r(W,0). \tag{5}$$



(a) $G(V,E)$ representing physical network. (b) Physical and TE link. (c) $G'(V,E')$. (d) $G'_l(V,E'_l)$, where $l = \{(2,5),(5,2)\}$. A dashed line expresses a failed link.

Fig. 1: Network model. A link without an arrow indicates a bidirectional link. A link with an arrow indicates a directed link.

III. PSO-TE algorithm

This section describes the algorithm of PSO-TE using tabu search. The detailed description is as follows.

- Step 1 (setting TE links and making $G'(V,E')$): We set TE links in the network. At first, for all node pairs $(u,v) \in P$, we find m non-neighboring node pairs in the order of increasing traffic volume t_{pq}. Next, we select at most k paths with the smallest number of hops in each node pair by using Yen's algorithm [11]. If there are more than k paths with the smallest number of hops, we arbitrarily select k paths with the least overlapping score of physical links through which they pass. We create E' by adding TE links for m node pairs to E; we obtain $G'(V,E')$.
- Step 2 (initialization): The iteration counter is set to $i = 0$. The initial value of each element in link weight set W_{itr} is a random value not registered in the tabu list. Add W_{itr} into the tabu list. W_{itr} is used to find the network congestion ratio.
- Step 3 (identifying bottleneck link with every single failure): Identify the link contributing to the highest link congestion ratio by using (3) for all single link failure patterns. We call this highest congested link a marked link and the failure link contributing to the highest network congestion ratio a marked failure link. At the first time of iteration, $W_{\mathrm{PSO}} \leftarrow W_{\mathrm{itr}}$ and $r_{\mathrm{PSO}} \leftarrow r_{\mathrm{itr}}$.
- Step 4 (moving to next candidate): In this step, we consider a topology with the single marked failure link. Increase the link weight of the marked link until at least one flow passing through the marked link changes. Here, the link weight that has the most flow among the physical links and TE links is increased. If the link utilization ratio of the marked link decreases, update the new link weight set as W_{tmp} and register in the tabu list. Otherwise, the link utilization ratio of the marked link does not change and exceeds the maximum allowed link weight $w_{\max}$, move to Step 7. If the link weight of W_{tmp} exceeds $w_{\max}$, move to Step 7. Also, if W_{tmp} has already been in the tabu list, move to Step 7.
- Step 5 (candidate evaluation): The link weight set updated in Step 4 is evaluated by (3). All single-link failures are considered to calculate the worst-case network congestion. Then, find the network congestion ratio for the new link weight set as $r_{\mathrm{tmp}} \leftarrow R(W_{\mathrm{tmp}})$. If it is smaller than r_{itr}, $W_{\mathrm{itr}} \leftarrow W_{\mathrm{tmp}}$, $r_{\mathrm{itr}} \leftarrow r_{\mathrm{tmp}}$. If the network congestion ratio r_{itr} is not updated $J_{\max}$ times in a sequence, go to Step 7.
- Step 6 (updating currently found best solution): If the network congestion ratio of W_{itr} is lower than that of W_{PSO}, set $W_{\mathrm{PSO}} \leftarrow W_{\mathrm{itr}}$ and $r_{\mathrm{PSO}} \leftarrow r_{\mathrm{itr}}$.
- Step 7 (moving to next iteration): If the maximum number of iterations $I_{\max}$ is not exceeded, set $i \leftarrow i+1$ and go back to Step 2. If $I_{\max}$ is exceeded, the algorithm terminates and returns W_{PSO} and r_{PSO} as a solution. The maximum number of iterations $I_{\max}$ is tailored to the network's size, available computational resources, and the desired solution precision.

IV. Numerical Results

A. Experimental Environments

We investigate the performance of PSO-TE via simulation, compared with PSO without TE links and SO-

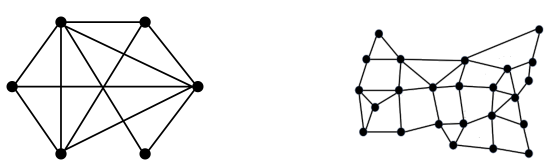
TE. We present the networks and parameters used in the simulations. We use two networks, as shown in Fig. 2. We use two networks used in [12].

We set input parameter values as follows. We generate traffic demand $t_{pq} \in T, (p, q) \in P$, randomly by following a uniform distribution in the range of (0, 100 U_{D}), where U_{D} is a given constant positive value. We set link capacity c_{ij} for all $(i, j) \in E$ randomly by following a uniform distribution in the range of ($10U_{\mathrm{C}}$, 100 U_{C}), where U_{C} is a given constant positive value. $U_{\mathrm{D}}/U_{\mathrm{C}}$ is set so that we can get feasible solutions. To compare the worst-case network congestion ratios of the different models with different failure scenarios, we normalize them divided by the worst-case network congestion ratio among all link failure scenarios $l \in F$ obtained by SO without TE. This normalization prevents us from discussing the effects of $U_{\mathrm{D}}/U_{\mathrm{C}}$. $C_{\max}$ is set to 30. We set $w_{\max}$ to 30 and 50 for networks (a) and (b) in Fig. 2, respectively.

Since we randomly generate traffic demands and link capacity in our evaluation, we conduct n trials and take the resulting metric average over n trials. We set $n = 10$ for all simulations. Let ξ and σ denote a resulting average value and a standard deviation over n trials, respectively. $\frac{\sigma}{\xi}$ for each result is in the range from 0.07 to 0.35.

$I_{\max}$ expresses the maximum number of iterations in the tabu search. $I_{\max}$ is set by considering the network size, available computational resources, and the desired solution precision. The larger the value of $I_{\max}$ is, the more accurate the solution is, and the longer the computation time is required. Thus, we set a suitable value of $I_{\max}$ by observing the trade-off between the solution accuracy and the computation time. In the simulation of SO-TE, we set $I_{\max}$ to 30 and 40 for networks (a) and (b), respectively. In the simulation of PSO-TE and PSO without TE links, we set $I_{\max}$ to 10 and 20 for the networks in Figs. 2(a) and (b), respectively.

We evaluate PSO-TE, PSO without TE links, and SO-TE by performing the following steps. For randomly generated traffic T and link capacity C, we calculate the worst-case network congestion ratios r_{PSO} and r_{SO}, for each network type. We code the algorithms for PSO-TE, PSO without TE links, and SO-TE using C++. We use a computer configured with AMD EPYC Rome 7502P 32C/64T 128GB memory to run the algorithms.



(a) (b)

Fig. 2: Networks.

B. Results and Discussions

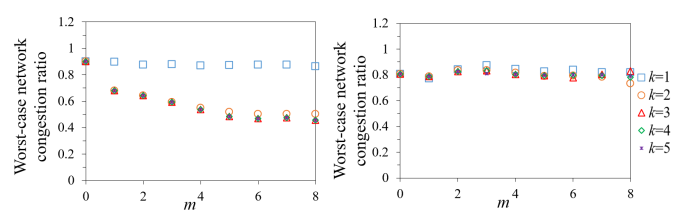
1) Effect of installing TE links on PSO-TE: We investigate the effect of installing TE links on PSO-TE by obtaining the worst-case congestion ratio among all link failure scenarios $l \in F$.

Figures 3(a) and (b) show the worst-case network congestion ratios of PSO-TE depending on m and k for networks (a) and (b) presented in Figs. 2(a) and (b), respectively. $m = 0$ indicates PSO without TE links.

Based on Fig. 3, we discuss the effect of m on the worst-case congestion ratio among all link failure scenarios $l \in F$. The worst-case congestion ratio tends to decrease as m increases for PSO-TE. It suggests reducing the worst-case congestion ratio by adding TE links. Increasing m makes the route design more flexible, reducing the worst-case congestion ratio. On the other hand, we observe that, in some cases, the worst-case congestion ratio increases even as m increases. This is because even if the weight of the link with the largest flow amount on the most congested link increases, the flow may remain due to ECMP. The remaining flows may cause a slight increase in the worst-case congestion ratio compared to before increasing m, i.e., adding TE links.

Next, we discuss the effect of k on the worst-case congestion ratio among all link failure scenarios $l \in F$. Figure 3 shows that the worst-case congestion ratio tends to decrease as k increases. This suggests that increasing the number of physically different paths that TE links take can reduce the worst-case congestion ratio. On the other hand, in some cases, we observe an increase in the worst-case congestion ratio even when k increases. This is because, for the same reason as the observation on increasing m, increasing k may increase the number of flows passing through a specific link, where ECMP does not distribute the flows effectively. The decrease in the worst-case congestion ratio stops at a certain k since only the shortest hop paths are selected as a TE link.

As described above, the worst-case congestion ratio among all link failure scenarios $l \in F$ can be reduced by appropriately setting m and k. This underscores the importance of the network operator's role in determining these parameters to optimize network performance.



(a) Network (a). **(b)** Network (b).

Fig. 3: Worst-case network congestion ratio of PSO-TE dependent on m and k.

We discuss how much the worst-case congestion ratio among all link failure scenarios $l \in F$ can be reduced by selecting the appropriate m and k based on Fig. 3. Let $(m, k) = (m', k')$ be the pair of m and k when the worst-case congestion ratio is best reduced. $r_{(m',k')}$ is the worst-case congestion ratio with $(m, k) = (m', k')$.

TABLE I: Reduction ratio γ in PSO-TE.

Network type	$r_{(0,0)}$	$r_{(m',k')}$	(m', k')	γ
Network (a)	0.902	0.458	(8, 3)	0.493
Network (b)	0.806	0.730	(8, 2)	0.094

The baseline model is the model without TE links, i.e., $(m,k)=(0,0)$. $r_{(0,0)}$ is the worst-case congestion ratio with $(m,k)=(0,0)$. Table I shows the reduction ratio γ when the worst-case congestion ratio is best reduced by setting $(m,k)=(m',k')$. γ is defined by:

$$\gamma = \frac{r_{(0,0)} - r_{(m',k')}}{r_{(0,0)}}. \quad (6)$$

Table I shows the reduction ratios γ of PSO-TE. The range of γ is 0.094–0.493 for our examined networks. Network (a) in Fig. 2(a) has larger γ than network (b) in Fig. 2(b). When the network size is smaller, the routing flexibility becomes limited without TE links; the effect of installing TE links is more significant.

We provide the computation times of PSO-TE. The computation times with $(m,k)=(0,0)$ and $(8,3)$ for network (a) are 1.8 [s] and 6.6 [s], and those with $(m,k)=(0,0)$ and $(8,2)$ for network (b) are 978 [s] and 2078 [s], respectively. The computation time increases as the network size increases and m and k increase since the number of link weights increases.

2) Comparison of PSO-TE with SO-TE: We compare PSO-TE to SO-TE. We set TE links $(m,k)=(m',k')$ for both of PSO-TE and SO-TE.

TABLE II: Comparison of worst-case network congestion ratios among all link failure scenarios with TE links.

Network type	$\max_{l\in F} r(W_{\text{PSO-TE}}, l)$	$\max_{l\in F} r(W_{\text{SO-TE}}, l)$	α
Network (a)	0.46	0.84	0.45
Network (b)	0.73	0.85	0.14

Table II shows the network congestion ratios with a link failure. In terms of the worst-case network congestion ratio among all link failure scenarios $l \in F$, the following relationship is satisfied:

$$\max_{l\in F} r(W_{\text{PSO-TE}}, l) \leq \max_{l\in F} r(W_{\text{SO-TE}}, l). \quad (7)$$

This indicates that the worst-case network congestion ratio of PSO-TE over $l \in F$ is smaller than or equal to that of SO-TE. The reduction ratio of the worst-case network congestion ratio with TE links is defined by:

$$\alpha = \frac{\max_{l\in F} r(W_{\text{SO-TE}}, l) - \max_{l\in F} r(W_{\text{PSO-TE}}, l)}{\max_{l\in F} r(W_{\text{SO-TE}}, l)}. \quad (8)$$

The range of α is 0.14–0.45 for our examined networks. This result shows that PSO-TE can reduce network congestion ratios caused by link failures with TE links.

TABLE III: Comparison of worst-case network congestion ratios without link failure with TE links.

Network type	$r(W_{\text{PSO-TE}}, 0)$	$r(W_{\text{SO-TE}}, 0)$	β
Network (a)	0.39	0.29	0.37
Network (b)	0.59	0.24	1.44

Next, we evaluate the performance of PSO-TE in terms of the network congestion ratio without link failure. Table III shows the network congestion ratios without link failure. In terms of the network congestion ratios without link failure, the following relationship is satisfied:

$$r(W_{\text{SO-TE}}, 0) \leq r(W_{\text{PSO-TE}}, 0). \quad (9)$$

The deviation between $r(W_{\text{PSO-TE}}, 0)$ and $r(W_{\text{SO-TE}}, 0)$, β is defined by:

$$\beta = \frac{r(W_{\text{PSO-TE}}, 0) - r(W_{\text{SO-TE}}, 0)}{r(W_{\text{SO-TE}}, 0)}. \quad (10)$$

The range of β is 0.37–1.44 for our examined networks. This result shows that the network congestion ratio of SO-TE without link failure is smaller than that of PSO-TE.

In summary, compared with SO-TE, PSO-TE can suppress increasing network congestion due to link failures expressed by α at the expense of penalty β, i.e., a larger network congestion ratio in a non-failure scenario.

V. Conclusion

This paper proposed a link-weight design model with traffic-engineering (TE) links based on preventive start-time optimization (PSO) for handling single link failures, called PSO-TE. The model considers physical and TE links when determining the link weights, assuming all single physical link failures. It determines the link weights to minimize the worst-case network congestion ratio against possible single-link failure patterns. Numerical results observed that PSO-TE effectively reduces the worst-case network congestion compared to PSO without TE links, demonstrating the benefits of including TE links. Compared with SO-TE, PSO-TE can suppress increasing network congestion due to link failures. PSO-TE provides a practical solution within the existing framework of static link weights without modifying routing protocols or routers, leveraging OSPF-TE.

References

[1] M. Ali, A. Anjum, O. Rana, A. R. Zamani, D. Balouek-Thomert, and M. Parashar, "RES: Real-time video stream analytics using edge enhanced clouds," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 792–804, 2022.

[2] L. Yang, J. Liu, S. Li, D. Zhang, and Z. Xia, "Enhancing user experience in ultra HD cloud performing arts live streaming: A QoS-to-QoE mapping approach," *IEEE Trans. Broadcast.*, vol. 70, no. 2, pp. 413–428, 2024.

[3] J. Moy, "OSPF version 2," RFC 2328, Apr. 1998.

[4] T. Li and H. Smit, "IS-IS extensions for traffic engineering," RFC 5305, Oct. 2008.

[5] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 4, pp. 756–767, 2002.

[6] E. Oki, "Preventive start-time optimization against network failure," in *Proc. 2014 10th Int. Conf. the Des. of Reliable Commun. Netw. (DRCN)*, 2014, pp. 1–8.

[7] G. Iannaccone, C.-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an IP backbone," in *Proc. 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW 2002)*, 2002, pp. 237–242.

[8] I. M. Kamrul and E. Oki, "PSO: preventive start-time optimization of OSPF link weights to counter network failure," *IEEE Commun. Lett.*, vol. 14, no. 6, pp. 581–583, 2010.

[9] D. M. Yeung, D. Katz, and K. Kompella, "Traffic engineering (TE) extensions to OSPF version 2," RFC 3630, Oct. 2003.

[10] C. Hopps and D. Thaler, "Multipath issues in unicast and multicast next-hop selection," RFC 2991, Nov. 2000.

[11] J. Y. Yen, "Finding the k shortest loopless paths in a network," in *Proc. Manage. Sci.*, vol. 17, no. 11, 1971, pp. 712–716.

[12] Y. Hirano, F. He, T. Sato, and E. Oki, "Preventive start-time optimization to determine link weights against probabilistic link failures," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 2275–2293, 2021.