Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2022-14. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres. ISSN 1587–4451.

Polynomial-Time Algorithm for the Regional SRLG-disjoint Paths Problem

Balázs Vass , Erika Bérczi-Kovács , Ábel Barabás , Zsombor László Hajdú, and János Tapolcai

December, 2022

Polynomial-Time Algorithm for the Regional SRLG-disjoint Paths Problem

Balázs Vass ⋆, Erika Bérczi-Kovács ⋆⋆, Ábel Barabás ⋆⋆⋆,
Zsombor László Hajdú, and János Tapolcai

Abstract

The current best practice in survivable routing is to compute link or node disjoint paths in the network topology graph. It can protect single-point failures; however, several failure events may cause the interruption of multiple network elements. The set of network elements subject to potential failure events is called Shared Risk Link Group (SRLG), identified during network planning. Unfortunately, for any given list of SRLGs, finding two paths that can survive a single SRLG failure is NP-Complete. In this paper, we provide a polynomial-time SRLG-disjoint routing algorithm for planar network topologies and a large set of SRLGs. Namely, we focus on regional failures, where the failed network elements must not be far from each other. We use a flexible definition of regional failure, where the only restrictions are that i) the topology is a planar graph, ii) each SRLG forms a set of connected edges in the dual of the planar graph, and iii) for each node v, the links incident to v are part of an SRLG. The proposed algorithm is based on a max-min theorem. Through extensive simulations, we show that the algorithm scales well with the network size, and one of the paths returned by the algorithm is only 4% longer than the shortest path on average.

1 Introduction

Disjoint path computation is the essence of any strategy for networks to survive failures. The current best practice is to utilize network flow algorithms, such as Suurballe's algorithm [37], to efficiently compute link or node disjoint paths in the network topology graph. However, several papers studied [29, 12, 14, 16, 10, 15, 20, 25, 36] that the

⋆Balázs Vass, Zsombor László Hajdú, and János Tapolcai are with MTA-BME Future Internet Research Group and ELKH-BME Information Systems Research Group, Department of Telecommunication and Media Informatics, Faculty of Electrical Engineering and Informatics (VIK), Budapest University of Technology and Economics (BME). {balazs.vass,hajdu,tapolcai}@tmit.bme.hu.

⋆⋆Department of Operations Research, Eötvös Loránd University, Budapest, Hungary, Alfréd Rényi Institute of Mathematics and MTA-ELTE Egerváry Research Group on Combinatorial Optimization. erika.berczi-kovacs@ttk.elte.hu

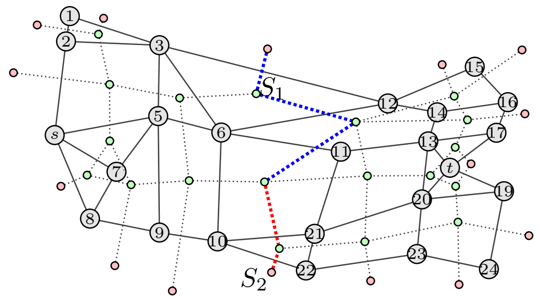
⋆⋆⋆Department of Operations Research, Eötvös Loránd University, Budapest, Hungary.barabasabel@gmail.com

networks have severe outages when almost every equipment in a vast physical region gets down as a result of a disaster, such as earthquakes, hurricanes, tsunamis, tornadoes, etc. These types of failures are called *regional failures*, which are simultaneous failures of nodes/links located in specific geographic areas. The set of network elements subject to potential failure events is called Shared Risk Link Group (SRLG), identified during network planning [47, 8, 32, 46, 38].

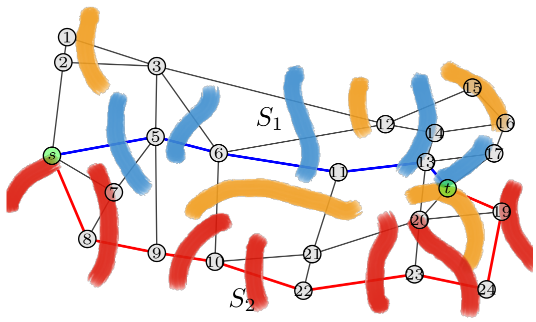
Unfortunately, for any given list of SRLGs and topology graph, finding two paths that can survive a single SRLG failure is NP-Complete [17, 11]. The proof is a reduction to 3SAT where each SRLG corresponds to a clause in the formula. Roughly speaking, a very artificial topology graph and SRLG settings are needed to show the high computational complexity of the problem, and many believe SRLG-disjoint routing is a well-solvable problem in practice. For example, Kobayashi-Otsuki provided [18] a routing algorithm for circular disk failures of fixed radius in a planar graph topology where the links are straight lines. Circular disk failures of the fixed radius are the most well studied regional failure model, see [29, 38]. Naturally arises the question: *Is there another set of regional SRLGs for which the SRLG-disjoint routing problem is solvable in polynomial time? Can we define a simple and general property of the regional SRLGs to have efficient routing algorithms?* The paper provides a positive and surprisingly simple answer as follows.

This study assumes the network topology is a planar graph. In backbone optical networks, it is rare that cables cross each other without having an optical cross-connect at the intersection. Planarity is an essential assumption to have a polynomial-time algorithm for an otherwise NP-hard problem (see Sec. 6 how to extend our algorithm for "almost" planar graphs). Apart from that, we adopt a very general model, here we may consider the network is somehow embedded on the Earth's surface, the links are curved lines between the endpoints, and an SRLG is resulting from a connected disaster area. We assume the list of SRLGs is defined in the service level agreement (SLA) [35] at network planning. The list of SRLGs typically involves physically close network nodes and parallel links, might be computed by any regional failure model [41, 38, 39, 40], or based on historical data of natural disasters, such as earthquakes [42], tornadoes, tsunamis, electromagnetic pulse (EMP) attacks, etc [26, 16, 34]. Protecting single network element failures (link or node failures) is the current best practice (e.g., Huawei [4, Sec. 4.5.4], Alcatel-Lucent [2, pp. 46-50], Cisco Systems[3, Chpt. 19], Juniper [5, Chpt. 3], Infinera[33]), so we add all link and node failures to the SRLG list.

Furthermore, the proposed routing algorithms do not even require knowing the geometry of the network, such as node coordinates and route of the cables. It is necessary because the router's routing engine cannot have such geographic information. The exact location of the network equipment is sensitive information for military and economic reasons, which will never be widely distributed on the internet. Note that, often, the network operators do not have any information about the route of the links or the physical coordinates of the intermediate routing nodes because the links are hired as a service from an independent company [1], called the Physical Infrastructure Provider. After all, information on the routes of the links is not part of any network protocol so far. So the key idea of our approach is that knowing the dual of the planar



(a) The US network topology graph (G) with its dual (G^*). The dual nodes are drawn with small green, and the outer region is the red dual node, split on the illustration into multiple nodes. The dual-edges are drawn with dotted lines and intersect the corresponding network links. The duals of two SRLGs, S_1 and S_2, are highlighted.



(b) The regional SRLGs ($\mathcal{S}_{region}$) are hand drawn with brush, and colored with the same color of the path traversed by, otherwise orange. The full list of SRLGs also include every single link or node failures as well. Two SRLG-disjoint paths between the source (s) and the target (t) node are drawn with red and blue links.

Input: a planar graph $G = (V, E)$, for every node the cyclic order of incident links in a planar drawing, two distinct nodes $s, t \in V$, and a set $\mathcal{S} \subseteq 2^{|E|}$ of dual-connected SRLGs with $\mathcal{S}_V \subseteq \mathcal{S}$.
Maximum Regional SRLG-disjoint Paths Problem (MRSDP): **Find:** maximum cardinality set of pairwise $\mathcal{S}$-disjoint s-t paths.

Figure 1: Illustration of the problem. Dual-edges corresponding to a regional SRLG are connected in the dual graph, for example, SRLG S_1 on (b) is mapped to blue dual-edges on (a). Note that SRLGs S_1 and S_2 forms an s-t cut, thus, there can be at most two SRLG-disjoint s-t paths.

topology graph is sufficient for the routing computations, and also we will define only combinatorial properties that the SRLGs must meet.

Fig. 1a shows such an example input: a planar topology graph with its dual graph. The nodes of the dual graph are the faces, and there are edges between the adjacent faces. Thus, each link e of the topology graph has a corresponding dual-edge, whose endpoints are the dual vertices corresponding to the faces on either side of e. Therefore, an SRLG as a set of links can be mapped to a set of dual-edges.

To mitigate the above problem, we assume the routing engine knows the dual graph of the planar network topology with the mapping between the links and dual-edges. The only assumption we have for SRLGs, that the corresponding dual-edges are connected. Note that it is a very loose restriction and covers all SRLGs that correspond to a connected disaster area. Here the disaster area is the geographic (connected) region in which the network elements are subject to fail simultaneously. A regional failure disconnects a link if it contains at least one (possibly end node) point of that link. For example the SRLGs S_1 and S_2 shown on Fig. 1b correspond to the dual-edges colored red and blue on Fig. 1a that are connected in the dual graph.

The main contributions of this paper are the following:

1. We provide a broad definition of 'regional SRLG,' where the regional SRLG-disjoint routing can be efficiently solved. For this, we define a pure combinatorial

routing problem input, which contains a planar network topology and the corresponding dual graph. We show that this input is sufficient for efficient routing computations, and no other information on the geometry of the physical topology is needed. We have a very flexible definition of regional failure, where we assume the SRLGs mapped to the dual-edges of the planar graph are connected. It is important to note that SRLGs must contain single node failures as well, otherwise the problem is NP-hard [7].

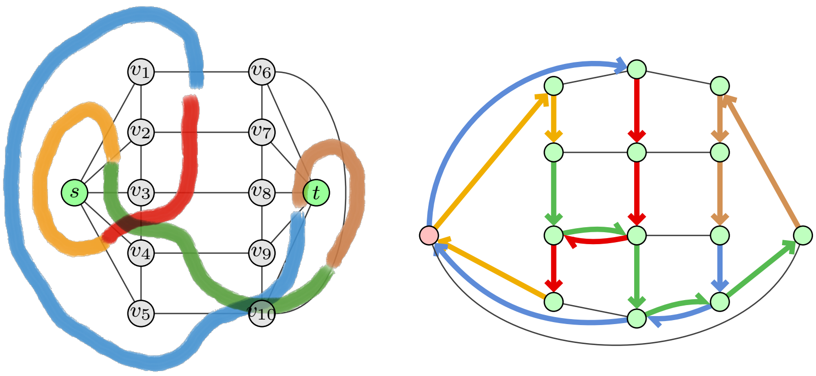
2. We provide an efficient polynomial-time SRLG-disjoint routing algorithm for the regional SRLG model defined above and planar network topology. Note that the SRLG-disjoint routing is NP-Complete in general [17, 11]. Our work heavily relies on the mathematical techniques used in [18] and [22]. The algorithm in [18] can be extended to solve the problem for circular disk failures, or in general for SRLGs that meet a complicated Property, see the conclusions of [18]. Unfortunately, Property of [18] strongly restricts the usability of their algorithm for a more general SRLG model. Motivated by the above, we have generalized their ideas into self-content graph-theoretical arguments that cope with a generalized SRLG model that contains all types of known failure models. We have adopted the max-min theorem for the regional SRLG-disjoint paths problem. In the special case of the circular disk failure model, the complexity of our algorithm is an improvement on those presented in [18, 31, 27, 28], respectively.

3. Through extensive simulation, we have shown that the corresponding routing problem scales well. We have observed that, after post-processing to shorten the resulting SRLG-disjoint paths, the shortest among them is just 4% longer than the absolute shortest path. Selecting it as the working path, the increase in the delay is negligible, while the other SRLG-disjoint paths can be the backup paths.

The paper is organized as follows. Sec. II provides the problem formulation and presents a max-min theorem for the regional SRLG-disjoint paths problem. Sec. III gives a simple upper bound on the number of SRLG-disjoint paths. Sec. IV describes the proposed algorithm. Sec. V gives a lower bound on the number of SRLG-disjoint paths. Sec. VI heuristically shortens the paths and deals with non-planar input graphs. Sec. VI overviews the related works. Sec. VII presents our simulation results. Finally Sec. VIII concludes the paper.

2 Problem Formulation and Main Results

Let $G = (V, E)$ be a planar network topology graph with a *node* set V, a *link* set E, and two distinct nodes $s, t \in V$. We do not know any geometric embedding of G, instead we only know the order of incident links at every node in the embedding. Note that from this information the dual graph $G^* = (V^*, E^*)$ can be easily calculated. When it does not confuse, we identify the faces of G with their dual nodes in $G^* = (V^*, E^*)$. In other words $G^* = (V^*, E^*)$ is composed of a *face* set V^* and a *dual-edge* set E^*, see Fig. 1a. In what follows, a link is sometimes called an edge.



(a) The network topology and the SRLGs ($\mathcal{S}_{region}$) are drawn with brush of unique color.

(b) The dual graph with a closed dual walk C such that $l(C) = 5$, $w(C) = 3$, and hence ${}^{l(C)}/_{w(C)} < 2$.

Figure 2: A graph, where the MIN-CUT = 3, but there is no two SRLG-disjoint paths between s and t, meaning MAX-FLOW = MIN-CUT − 2.

Let $\mathcal{S}_{region} \subseteq 2^{|E|}$ be a set of link sets representing a set of *regional SRLGs*. We assume the set of SRLGs also contains all the single node failures, which ensures the obtained SRLG-disjoint paths to be node-disjoint. Let E_v denote the set of links in G incident to a node v and let $\mathcal{S}_V$ represent the set of SRLGs modeling the node failures, i.e.,

$$\mathcal{S}_V = \{E_v | v \in V \setminus \{s, t\}\} \ .$$

Let $\mathcal{S}$ denote the set of all SRLGs: $\mathcal{S} = \mathcal{S}_{region} \cup \mathcal{S}_V$. Let ρ denote the maximum size of a regional SRLG: $\rho := \max\{|S| \big| S \in \mathcal{S}\}$, and let μ denote the maximum number of SRLGs that contain the same edge: $\mu = \max\{|T| : T \subset \mathcal{S}, |\cap_{S \in T} S| > 0\}$. We say that two paths are **($\mathcal{S}$-)disjoint** or SRLG disjoint if there is no SRLG $S \in \mathcal{S}$ intersecting both of them[1]. We may omit $\mathcal{S}$ from the notation when the SRLG set is clear from the context.

Formally, for a link set $X \subseteq E$, let X^* be the set of duals of links of X. For an SRLG $S \in \mathcal{S}$, let $V^*(S^*) := \{f \in V^* | \text{there is a dual-edge } \{f, f'\} \in S^* \text{ for some } f'\}$. We denote by d the maximal diameter of the dual of an SRLG: $d := \max\{\, diam(S^*) | S \in \mathcal{S}_{region}\}$, where $diam(S^*) = \max_{f,f' \in V^*(S^*)} \min\{$ edge lengths of f-f' paths in $S^*\}$. We call a set of links $S \subseteq E$ **dual connected**, if the edge-induced subgraph of S^* is connected in G^*. For example, each $E_v \in \mathcal{S}_V$ is clearly dual connected. We demand $\mathcal{S}$ to fulfill the following property:

Property 1. Each set $S \in \mathcal{S}$ is dual connected.

Recall we have a second property:

Property 2. All node failures are listed apart from s and t ($\mathcal{S}_V \subseteq \mathcal{S}$).

Our main goal in this paper is to find the maximum number of $\mathcal{S}$-disjoint $s - t$ paths in planar graphs and SRLG sets with properties 1, 2, which we call **Maximum**

[1]In the related literature, 'disjointness' is sometimes called 'separatedness'.

Regional SRLG-disjoint Paths Problem (MRSDP). See Figure 1 for the exact problem definition. Let MAX-FLOW denote the optimal value of the problem. First, we give a trivial upper bound on MAX-FLOW using the analogy of max-flow min-cut theorems for network flows. A set of SRLGs from $\mathcal{S}$ that disconnect s from t is called an **SRLG cut** in this paper, see SRLG S_1 and S_2 on Fig. 1b as an illustration. It is easy to see that the size of an SRLG cut is an upper bound for MAX-FLOW, because two disjoint paths cannot traverse any of these SRLGs simultaneously by definition. Let MIN-CUT denote the minimum size of an SRLG cut. Fig. 2a shows an example graph where the MAX-FLOW = 1, while MIN-CUT = 3. Later, we will show that the gap between the MAX-FLOW and MIN-CUT is at most 2 (see Section 5).

Theorem 2.1. *For any instance of the MRSDP problem and its corresponding MAX-FLOW and MIN-CUT values we have*

$$\textit{MAX-FLOW} \leq \textit{MIN-CUT} \leq \textit{MAX-FLOW} + 2.$$

Although MIN-CUT does not give a sharp upper bound for MAX-FLOW, a min-max characterization can be given, which is one of the main results of this paper. In order to state this sharp upper bound, we need a more complex structure than a cut. Here we intuitively present the necessary notions, which are precisely defined in Sec. 3. A **walk** is a finite sequence of edges which joins a sequence of vertices. For a closed walk C in the dual graph G^*, the length $l(C)$ is the minimum number of times one has to "switch SRLG" to go around C, while the winding number $w(C)$ of C is the number of times that C separates s and t. Our main results are the following.

Theorem 2.2. *For any instance of the MRSDP problem, we can find a maximum number of $k = \textit{MAX-FLOW}$ SRLG disjoint paths in $O\left(n^2\mu\left(\log k + \rho \log d\right)\right)$, and we determine closed dual walk C in G^*, for which $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k$. For $\textit{MAX-FLOW} \geq 2$ we also have*

$$\textit{MAX-FLOW} = \min\left\{ \left\lfloor \frac{l(C)}{w(C)} \right\rfloor \middle| C \textit{ closed dual walk, } w(C) \geq 1 \right\}.$$

3 Upper Bounds on the Number of Maximum Regional SRLG-disjoint Paths

In this section, we will provide another upper bound for MAX-FLOW by generalizing the approach of [18]. This upper bound will turn out to be tight (cf. Thm. 2.2). Let C be a closed walk in G^*. We define the **winding number** $w(C)$ of C as the number of times that C separates s and t. More precisely, let us fix an s-t path P in G, and consider the edges of P being oriented towards t. Let us consider a one-way orientation of the dual-edges of dual walk C. Let $w_1(C) = \{\#e_d \in C | e_d \text{ crosses an } e_p \in P \text{ from left to right}\}$. Similarly, $w_2(C) := \{\#e_d \in C | e_d \text{ crosses an } e_p \in P \text{ from right to left}\}$. Lastly, we define $w(C) := |w_1(C) - w_2(C)|$. E.g., the (colored) dual walk on Fig. 2b separates s

and t three times. Note that if C is a closed walk, then $w(C)$ is indifferent to the choice of P and orientation of C.

Now we define $l(C)$ for a closed dual walk C. Let $C = \{C_1, ..., C_k\}$ be a partition of the dual-edges such that each C_i consists of consecutive edges of C, and there exists an SRLG $S_i \in \mathcal{S}$ such that S_i^* contains C_i. Let $l(C)$ be the minimal number for which there exists such a partition. For example, to cover the dual walk on Fig. 2b we need at least 5 SRLGs. We note that $l(C) \leq |V^*|$ will hold for the closed dual walks constructed in our proofs.

By using these notations, we can give an upper bound for MAX-FLOW as follows.

Lemma 3.1. *Consider an instance of the MRSDP problem. If MAX-FLOW* ≥ 2, *then*

$$MAX\text{-}FLOW \leq \min\left\{ \left\lfloor \tfrac{l(C)}{w(C)} \right\rfloor \middle| C \text{ closed dual walk, } w(C) \geq 1 \right\}. \tag{1}$$

Proof. Suppose we have s-t paths $P_1, \ldots, P_{k\geq 2}$ that are pairwise disjoint and let $C = \{C_1, ..., C_{l(C)}\}$ be a closed dual-walk such that each subwalk C_j is contained by the dual of an SRLG $S_j \in \mathcal{S}$. We show that each P_i has to intersect at least $w(C)$ subwalks C_j. Observe that each C_j adds at most 1 to the value of $w(C)$: $w(C_j) := |w_1(C_j) - w_2(C_j)| \leq 1$, since paths P_i are vertex disjoint (by Property 2). Two disjoint paths cannot cross C at the same C_j, so we have $l(C) \geq k \cdot w(C)$. □

4 Polynomial Time Algorithm to Find a Maximum Number of Regional SRLG-Disjoint Paths

In this section we show that Lemma 3.1 can be extended into exact min-max theorem for MAX-FLOW, and Eq. (1) holds with equality. If MAX-FLOW = 1, we give a closed dual walk C with $l(C)/w(C) < 2$. Our proof generalizes ideas in [18], which shows a geometric min-max theorem for the special case of the MRSDP problem, where the disaster regions are circular disks.

The algorithm has two main parts: the base case (4.3) and the inductive part (4.1). The inductive part decides whether there exist k $\mathcal{S}$-disjoint paths, assuming that $k-1$ such paths are given as starting paths.

When searching for $k = 2$ $\mathcal{S}$-disjoint paths P_1 and P_2, for algorithmic reasons, the starting path needs to be 'clockwise far enough' from itself. We use the term clockwise $\mathcal{S}$-disjointness to capture the intuition precisely (see definition below). The goal of the base case is to decide whether there exists a path that is clockwise $\mathcal{S}$-disjoint from itself.

First we introduce the notion of crossings. We say two s-t paths P_1 and P_2 are **crossing** if, after contracting their common edges, there is a subpath P' contained by both paths such that the links entering/leaving P' in P_1 and P_2 are alternating according to their incidence to P_1 and P_2. We note that with this definition, two non-crossing paths may have common edges, intuitively, the only restriction for them is not to change their clockwise order along the way from s to t.

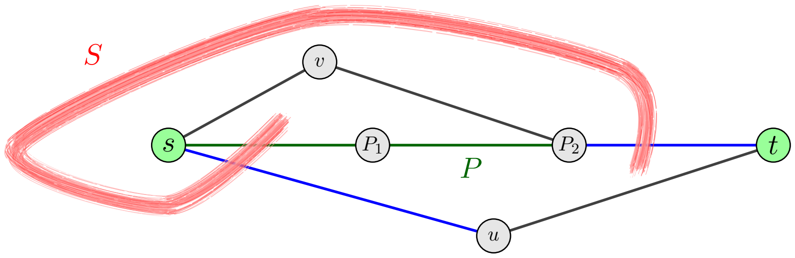


Figure 3: Clockwise part $\{su, P_2t\}$ of SRLG $S = \{su, sP_1, P_2t\}$ with respect to path $P = s, P_1, P_2, t$

Now we turn to the definition of clockwise $\mathcal{S}$-disjointness. For an s-t path P in G and a directed dual path Q^* in G^* we say that Q^* is **clockwise to** P if it does not cross P from right to left, that is, $w_2(C^*) = 0$. For an s-t path P and an intersecting SRLG S we define $S_{\text{clw}}(P)$ the **clockwise part of** S **with respect to** P as the subset of those links in $S \setminus (S \cap P)$ for which the corresponding dual edge is reachable from $(S \cap P)^*$ on a path clockwise to P. (see Fig. 3).

For two s-t paths P_1 and P_2 without crossings, an ordered pair (P_1, P_2) is **clockwise (**$\mathcal{S}$**-)disjoint** if for any SRLG S in $\mathcal{S}$ intersecting P_1, $S_{\text{clw}}(P_1)$ does not intersect P_2. Obviously, paths P_1 and P_2 are disjoint exactly if both pairs (P_1, P_2) and (P_2, P_1) are clockwise disjoint.

4.1 Induction step

In what follows we show the equality in (1) for MAX-FLOW ≥ 2. First, we assume that for some $k \geq 2$ we have $k-1$ pairwise disjoint s-t paths $P_1, \dots P_{k-1}$ (when $k = 2$ we assume that P_1 is clockwise disjoint from itself). We will give an algorithm for finding either k pairwise disjoint s-t paths or a closed dual walk C with $\lfloor l(C)/w(C) \rfloor = k-1$ (see Algorithm 1). Then applying the algorithm repeatedly for $k = 2, \dots,$ MAX-FLOW, we get an inductive proof of the equality in Lemma 3.1.

We may assume that the first edges of $P_1, \dots, P_{k-1}$ occur in this clockwise order at s. We continue this series of paths by generating new s-t paths $P_k, P_{k+1}, \dots$. At each step, a new path P_l is generated and if $P_{l-k+1}, \dots, P_l$ are pairwise disjoint, we stop. Otherwise we generate a new path again. If we do not find k pairwise disjoint paths after $|V^*| + 1$ path generations, then the algorithm stops and we can determine a closed dual walk C with $\lfloor l(C)/w(C) \rfloor = k-1$ (see Claim 4.2). Our algorithm is described in Algorithm 1.

When generating a new path P_l we use previous paths P_{l-1} and P_{l-k}. Intuitively, P_l is the path clockwise 'nearest' to P_{l-k} among those that are clockwise-disjoint from P_{l-1}.

Now we give the precise definition of 'nearness' by describing an ordering of the paths. The clockwise order of the links incident to a node v gives a cyclic ordering of those links. For a fixed link e incident to v this cyclic ordering induces a complete ordering $<_{v,e}$ of the links incident to v: for links e_1, e_2 incident to v we say that $e_1 <_{v,e} e_2$ if e_1 is earlier than e_2 in the clockwise order starting from e. Given an $s - t$

Algorithm 1: Search for one more SRLG-disjoint path

Input: MRSDP problem input, $P_1, \dots, P_{k-1}$ pairwise disjoint s-t paths if $k \geq 3$ or an s-t path P_1 that is clockwise disjoint from itself if $k = 2$.

Output: k pairwise disjoint s-t paths or a closed dual walk C in G^* with $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k - 1$

1 $P_0 := P_{k-1}$
2 **for** $l = k, \dots, k + |V^*|$ **do**
3 $\quad P_l := P_{\text{nearest}}(P_{l-1}, P_{l-k})$ (see Alg. 2)
4 $\quad$ **if** P_l, P_{l-k+1} *are* $\mathcal{S}$*-disjoint* **then**
5 $\quad\quad$ **return** $P_{l-k+1}, \dots, P_{l-1}, P_l$
6 **return** *a closed dual walk* C *in* G^* *with* $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k - 1$

path P, these orderings induce an ordering $<_P$ on the set of s-t paths the following way. Let P_1 and P_2 be s-t paths and let v denote the first node where they enter on the same link (say e) but continue on different links, say e_1 and e_2 (if $v = s$, let e be the first link of P). We say that $P_1 <_P P_2$ if $e_1 <_{v,e} e_2$.

Now we are ready to give a precise definition of P_l: it is an s-t path that is clockwise disjoint from P_{l-1}, does not cross P_{l-k} and within these constraints minimum with respect to $<_{P_{l-k}}$ (see Algorithm 2).

4.2 Computing the next nearest clockwise SRLG-disjoint path

In Algorithm 2 we have two non crossing paths Q_1, Q_2 as input such that Q_1 is clockwise disjoint from itself. We determine a path P that is clockwise-disjoint to Q_1, does not cross Q_2 and within these constraints minimum for $<_{Q_2}$. Note that by calling the algorithm with $Q_1 = P_{l-1}$ and $Q_2 = P_{l-k}$ we get the required path P_l in Algorithm 1.

Algorithm 2 uses DFS on a proper auxiliary graph G' and explores the nodes in clockwise order to find the optimal path. In order to avoid path P to cross Q_2, we modify G. We duplicate path Q_2 by 'cutting' it into two along its route, creating a left and a right copy of Q_2: instead of each internal node v on Q_2 we add two nodes v_{left} and v_{right} to G, and for each internal link $uv \in Q_2$ we add two links $u_{\text{left}}v_{\text{left}}$ and $u_{\text{right}}v_{\text{right}}$. For a link uv incident to a node $v \in Q_2$ but not on Q_2 we create the link $v_{\text{left}}u$ if uv is on the left side of Q_2 and we create $v_{\text{right}}u$ if the link is on the right side of Q_2. Similarly we add two copies of links of the form vu with v on Q_2 but u not on Q_2. The first and last links (say sv and ut) have two copies: sv_{left}, sv_{right} and $u_{\text{left}}t$, $u_{\text{right}}t$, respectively. Let G_{Q_2} denote the resulting graph. Note that G_{Q_2} is also planar, and there is a bijection between the s-t paths of G not crossing Q_2 and the s-t paths of G_{Q_2} (apart from Q_2, which has two copies in G_{Q_2}).

Clockwise separation to Q_1 can be guaranteed by deleting the clockwise part of all SRLG-s intersecting Q_1 (see line 3). If a link e to be deleted is in Q_2, we delete both

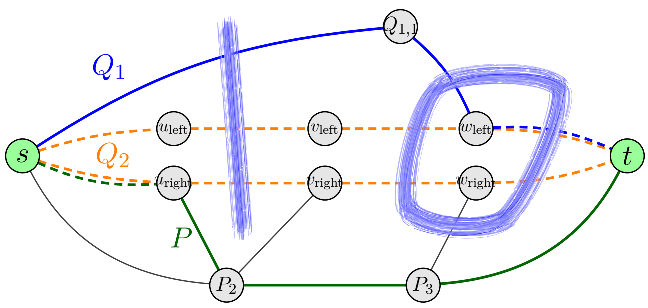


Figure 4: s-t path P that is minimum with respect to $<_{Q_2}$, clockwise-disjoint to Q_1 and does not cross Q_2. (Usually, we call Alg. 2 with $P = P_l$, $Q_1 = P_{l-1}$ and $Q_2 = P_{l-k}$)

Algorithm 2: Nearest clockwise SRLG-disjoint path

Input: Planar graph $G(V, E)$, SRLG set $\mathcal{S}$, non crossing s-t paths Q_1, Q_2, such that (Q_1, Q_1) is clockwise disjoint

Output: An s-t path P that is clockwise-disjoint to Q_1, does not cross Q_2, and is minimum with respect to $<_{Q_2}$

```
G' := G_{Q_2}
for (v_1, v_2) ∈ E(Q_1) do
    for S ∈ 𝒮 : (v_1, v_2) ∈ S do
        E' := E' \ S_clw(Q_1)
DFS-TREE:= DFS tree on E' rooted at s, exploring nodes in clockwise order (see <_{v,e}).
Starting link of DFS: sq_right, where sq ∈ Q_2.
return the s-t path in DFS-TREE
```

the left and right copies of the link (see Fig. 4). The resulting graph is G'. Then an optimal path with respect to $<_{Q_2}$ can be easily determined by a DFS if we fix the order of node exploration according to the clockwise order of the links. Since Q_1 does not cross Q_2 and is clockwise disjoint from itself, Q_1 is in G'. Hence t is reachable from s in G' and the DFS finds an s-t path indeed.

Now we show by induction that the last $k-1$ paths in the series behave similarly to the input paths.

Claim 4.1. *1. Paths $P_{l-k+2}, \ldots, P_l$ are pairwise $\mathcal{S}$-disjoint and in this clockwise order at s if $k \geq 3$.*

2. Path P_l is clockwise disjoint from itself if $k = 2$.

Proof. First, we prove part a). It is enough to show that the paths are in this clockwise order at s and that P_l and P_{l-k+2} are $\mathcal{S}$-disjoint. Since by induction P_{l-1} and P_{l-k+1} are $\mathcal{S}$-disjoint, they are also clockwise $\mathcal{S}$-disjoint and P_{l-k+1} does not cross P_{l-k}. We know that P_l is minimum with respect to $<_{P_{l-k}}$ among such paths, hence $P_l \leq_{P_{l-k}} P_{l-k+1}$, which shows the clockwise order of the paths. All we have to show is that P_l is clockwise $\mathcal{S}$-disjoint to P_{l-k+2}. Assume indirectly that there is an SRLG S such that there is a dual path $Q^* \subseteq S_{\text{clw}}(P_l)$ connecting dual edges e^*, f^* such that $e \in P_l, f \in P_{l-k+2}$.

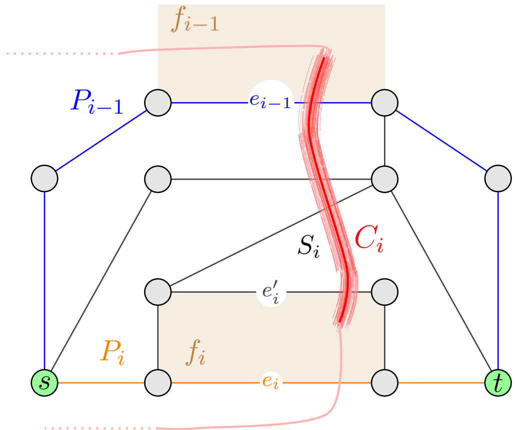


Figure 5: Illustration for Claim 4.2

Since path P_{l-k+1} is between P_l and P_{l-k+2} in the clockwise order, this dual path would have a dual edge h^* such that $h \in P_{l-k+1}$ contradicting that P_{l-k+1} and P_{l-k+2} are clockwise $\mathcal{S}$-disjoint.

Now we similarly prove the second part of the claim. Assume indirectly that P_l is not clockwise disjoint and there are (not necessarily different) dual edges e^*, f^* such that there is a dual path connecting e^* to f^* in $S^*_{\mathrm{clw}}(P_l)$. Then this dual path would have a dual edge h^* where $h \in P_{l-1}$, contradicting that P_{l-1} and P_l are clockwise disjoint. $\square$

If we find pairwise disjoint paths $P_{l-k+1}, \dots, P_{l-1}, P_l$ in line 5 of Algorithm 1, then we are done. In what follows, we give a procedure for finding a closed dual walk C with ${}^{l(C)}/_{w(C)} < k$ (line 6) when such paths do not appear while $l = k, k+1, \dots, k+|V^*|$. Let $N := k + |V^*|$.

Claim 4.2. *For $i = N, \dots, k$, we can compute links $e_i \in E$, faces $f_i \in V^*$, SRLGs $S_i \in \mathcal{S}$, and paths $C_i \subseteq S^*_i$ such that*

- *e_i is part of $P_i \setminus P_{i-k}$,*
- *f_i is the face left to e_i (as we walk on P_i from s to t)*
- *C_i is a dual path connecting f_{i-1} to f_i starting with e^*_{i-1} and then going in $S^*_{i\ clw}(P_{i-1})$.*

Proof. By the assumption, (P_{N-k}, P_{N-k+1}) is clockwise disjoint, but (P_N, P_{N-k+1}) is not clockwise disjoint, and hence there exists a link $e_N \in P_N \setminus P_{N-k}$ (intuitively, P_{N-k} is not close to P_{N-k+1}, but there is a link $e_N \in P_{N-k+1}$ close to P_N). Let the face left to e_N be f_N. By replacing e_N with other the links of f_N we get an s-t path that is smaller with respect to $<_{P_{N-k}}$. Thus there is a link e'_N neighboring f_N which is not in E' when the DFS in Algorithm 2 is started. So there is an SRLG $S_N \in \mathcal{S}$ such that a dual path Q^* in $S^*_{N_{\mathrm{clw}}}(P_{N-1})$ connects the dual of a link $e_{N-1} \in P_{N-1}$ and e'_N, see also Fig. 5. Since P_{N-1}, P_{N-k}, P_N do not cross and follow each other in this clockwise order, path P_{N-k} intersects Q. Thus $e_{N-1} \notin P_{N-k-1}$, otherwise pair (P_{N-k-1}, P_{N-k}) would not be clockwise disjoint. By repeating the same argument, we can find e_i, f_i, S_i and C_i for $i = N, \ldots, k$ as prescribed in the statement of the claim. $\square$

For $i = N, \ldots, k$, let e_i, f_i, S_i, and C_i be as described in Claim 4.2. By pigeonhole principle, $f_i = f_j$ for some $k \le i \le j \le N$. Let C be the closed dual walk yielding from the concatenation of $C_{i+1}, \ldots, C_j$. We will show that C satisfies ${}^{l(C)}/_{w(C)} < k$, which is equivalent to $u := \lfloor {}^{(j-i)}/_{k} \rfloor < w(C)$, because $l(C) = j - i$. If $u = 0$, then the inequality is trivial. Otherwise, e_j is strictly to the right of P_{j-k} (by Claim 4.2).

By line 3 of Alg. 1, $P_{j-(l+1)k}$ is to the left of P_{j-lk} for all $l = 1, \ldots, u$. Based on this, we can see that $C_{j-(l+1)k+1} \cdot \ldots \cdot C_{j-lk}$ makes at least one turn clockwise. Concentrating now on path P_N, we can see that we have an extra right-to-left crossing of the path at the last edge of C_{i+1}, that hitherto was not considered, which means $w(C_i \cdot \ldots \cdot C_j) = w(C) \ge u + 1$.

By the above procedure, we can find a closed dual walk C with ${}^{l(C)}/_{w(C)} < k$ in line 6 of Algorithm 1. Since the input of the Algorithm was a number of $k-1$ SRLG-disjoint paths, we also have $k - 1 \le {}^{l(C)}/_{w(C)}$, thus $\lfloor {}^{l(C)}/_{w(C)} \rfloor = k - 1$.

4.3 Base cases: finding a self-SRLG-disjoint path

What remains is to deal with the base cases ($k = 1, 2$) of the induction. It is trivial to decide whether there is an s-t path in G, so we may assume that such a path exists. Also, we may assume that there is no SRLG separating s and t. We have seen that Algorithm 1 can be run for an s-t path P if (P, P) is clockwise disjoint, by choosing $P_1 = P_2 = P$ as input ($k = 2$). For such an input the algorithm either finds a closed dual walk C in G^* with $\lfloor {}^{l(C)}/_{w(C)} \rfloor = 1$ or finds two $\mathcal{S}$-disjoint s-t paths. So our aim is to find an s-t path P such that (P, P) is clockwise disjoint or if no such path exists to find a closed dual walk C in G^* with $\lfloor {}^{l(C)}/_{w(C)} \rfloor < 2$ proving that MAX-FLOW is 1.

In order to find the path above, we will repeatedly use Algorithm 1 for $k = 2$ with an expanding series of SRLG sets. The key is to define SRLG sets $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{\lceil \log_2(d) \rceil} = \mathcal{S}$ such a way that if two s-t-paths P, R are $\mathcal{S}_i$-disjoint then (P, P) is clockwise $\mathcal{S}_{i+1}$-disjoint, generalizing the inductive idea applied in [22].

For an SRLG S, a face $p^* \in V(S^*)$ and a positive integer i let $S^*_i(p^*)$ be the set of dual edges that are part of a path in S^* that has at most i edges and starts from p^*. It is easy to see that in the subgraph induced by $S^*_i(p^*)$, there is a path of length at most $2i$ between any two nodes. Let $\mathcal{S}^*_0 := \mathcal{S}^*_V$ and $\mathcal{S}^*_i := \mathcal{S}^*_V \cup \{S^*_{2^{i-1}}(p^*) |\ S \in \mathcal{S}_{region}, p^* \in V(S^*)\}$ ($i = 1 \ldots \lceil \log_2(d) \rceil$). Note that $\mathcal{S}_{\lceil \log_2(d) \rceil} = \mathcal{S}$.

Lemma 4.3. *Suppose that P and R are s-t paths that are $\mathcal{S}_{i-1}$-*disjoint*. Then the pair (P, P) is clockwise $\mathcal{S}_i$-disjoint.*

Proof. Assume indirectly that (P, P) is not clockwise disjoint. Then there is an SRLG $S_i \in \mathcal{S}_i$ and link $e \in S_i \cap P$ such that a dual path in $S^*_{i\,\mathrm{clw}}(P)$ connects clockwise the right node of e^* to the dual of a link $f \in P \cap S_i$. Let Q^* denote this dual path extended with dual edge e^*. We assume Q^* is of minimum length.

Claim 4.4. *Path Q^* is a shortest path from e^* to f^* in S^*_i.*

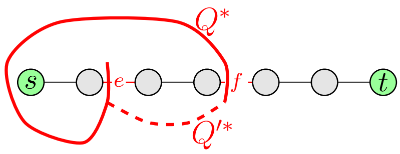


Figure 6: Illustration for Claim 4.4

Proof. If there were a shorter dual-path Q'^* from e^* to f^*, it could only cross P from right to left. Together with the reverse of Q^*, they would form a dual walk separating s and t, which is a contradiction because we assumed that there is no separating SRLG. $\square$

By Claim 4.4 path Q^* can be chosen shortest, that is, we may assume it has at most 2^i edges. Since paths P and R are $\mathcal{S}_{i-1}$-disjoint, they are link-disjoint. Hence path R intersects $Q \subseteq S_i$ at a link $h \neq e, f$. If $i = 1$, $|Q| \leq 2$ hence there is no such link and the claim follows. If $i \geq 2$, assume that there is such a link h. Dual edge h^* subdivides path Q^* into two shorter paths, which are also shortest paths. Observe that at least one of them has length at most 2^{i-1} and thus covered by an SRLG in $\mathcal{S}_{i-1}$, contradicting the assumption that (P, R) are $\mathcal{S}_{i-1}$-disjoint. $\square$

Menger's Theorem [24] characterizes the maximum number of node-disjoint (that is, $\mathcal{S}_0 = \mathcal{S}_V$-disjoint) s-t paths, which we can find in polynomial time. Since we assumed that there is no SRLG separating s and t thus, there is no separating node either. Hence there are two node-disjoint s-t paths P_0' and P_0'' (that can be found e.g., via Suurballe's algorithm [37]). Our algorithm for finding an s-t path P such that (P, P) is clockwise $\mathcal{S}$-disjoint is the repetition of the following steps, starting with $i = 1$. First we call Algorithm 1 with $k = 2$ for $P_1 = P_2 = P_{i-1}'$ and SRLG set $\mathcal{S}_i$. If the algorithm finds two $\mathcal{S}_i$-disjoint s-t paths P_i' and P_i'', then 1) if $\mathcal{S}_i = \mathcal{S}$ that is $i = \lceil \log(d) \rceil$, we return with the $\mathcal{S}$-disjoint paths P_i' and P_i'', or else, 2) we go to the first step with path P_i' and SRLG set $\mathcal{S}_{i+1}$. In the other case, the algorithm finds a closed dual walk C as in Theorem 2.2 with $\mathcal{S}_i$, then we stop the process. Since for every $S \in \mathcal{S}_i$ $(1 \leq i \leq \lceil \log(d) \rceil)$ there is an SRLG $S' \in \mathcal{S}$ with $S \subseteq S'$, for this closed dual walk C we have $\lfloor {}^{l(C)}/_{w(C)} \rfloor \leq 1$ for $\mathcal{S}$, too.

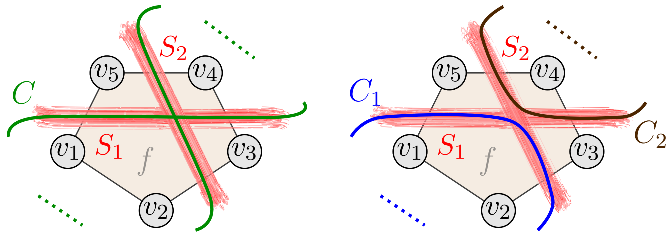
4.4 Complexity Analysis

We have just built an algorithm solving the MRSDP problem. Now we turn to its complexity:

Proof of Thm. 2.2. Suppose for now that we have $l - 1$ $\mathcal{S}$-disjoint s-t paths $(2 \leq l \leq$ MAX-FLOW), and we are searching for one more of them.

First we analyze Algorithm 2. The algorithm has two sections. The second is a DFS, which runs in $O(n)$.

Turning to the first section, we can observe that, for every set of $l-1$ consecutive s-t paths $P_i, P_{i+1}, \ldots, P_{i+l-2}$ generated by Algorithm 2 it holds that an SRLG S intersects



(a) Closed dual walk C (b) Uncrossed walks C_1 and C_2

Figure 7: Closed dual walk C crosses itself along the dual-edges of SRLGs S_1 and S_2 at a face f. The dual-edges of C can be reordered such that it results in two closed dual walks C_1 and C_2, both using the edges of both S_1 and S_2, switching between S_1 and S_2 at f, meaning $l(C_1) + l(C_2) \leq l(C) + 2$.

at most one of them. We argue that the total complexity of $l - 1$ consecutive first sections is $O(n\mu)$, based on the following. Creating graphs G_{Q_2} and their related SRLG sets runs in $O(n\mu)$, since all SRLGs have $O(n\mu)$ edges in total. Then, determining and deleting from graphs G_{Q_2} the parts of SRLGs clockwise to paths Q_1 also runs in the proposed total complexity.

In Algorithm 1, we call Algorithm 2 at most $|V^*| + 1 = O(n)$ times, so, for fixed l, the complexity of Algorithm 1 is $O(n^2 \frac{\mu}{l})$. This gives a running time of $O(n^2 \mu \log k)$ for finding $k =$ MAX-FLOW paths, if we have a single self-$\mathcal{S}$-disjoint path to start with.

In the base case, when we calculate the first self-SRLG-disjoint s-t path, after calculating two node-disjoint s-t paths (that can be done in $O(n^2)$ [37]), for each $i \in \{1, \ldots, \lceil \log d \rceil\}$, first we determine SRLG sets $\mathcal{S}_i$ and then call Algorithm 1. For a fixed i, $\mathcal{S}_i$ can be constructed in $O(n\mu\rho)$, since there are $O(n)$ edges, each part of at most μ SRLGs $S \in \mathcal{S}$, each S having $O(\rho)$ nodes. With $\mathcal{S}_i$, Algorithm 1 runs in $O(n^2\mu\rho)$. Thus, the runtime of the base case is $O(n^2\mu\rho \log d)$.

We can conclude that $k =$ MAX-FLOW $\mathcal{S}$-disjoint paths can be found in polynomial time, in $O\left(n^2\mu\left(\log k + \rho \log d\right)\right)$. Computing the dual-walk at the end of the algorithm can be done in $O(n^2)$ if while executing Algorithm 2 we store for each link visited in the DFS a link of P_{l-1} and an SRLG, that contains them both (if there is any). This way we can find e_i, f_i and C_i (described in Claim 4.2) in $O(n)$ time.

□

5 Lower Bound on the Maximum Number of Regional SRLG-disjoint Paths

By using Theorem 2.2, we prove Theorem 2.1.

Proof. Since MAX-FLOW $\leq$ MIN-CUT is obvious, we need to prove MIN-CUT $\leq$ $\leq$ MAX-FLOW $+ 2$. By Theorem 2.2, we can take a closed dual walk C such that $\lfloor l(C)/w(C) \rfloor =$ MAX-FLOW. Hence it suffices to find an SRLG cut of size $\lfloor l(c)/w(C) \rfloor + 2$ (i.e., a set of $\lfloor l(c)/w(C) \rfloor + 2$ SRLGs in $\mathcal{S}$ that disconnect s and t).

If $w(C) \geq 2$, then C crosses itself at a face. Similarly to the technique in [18] we can decompose C into two closed dual walks C_1 and C_2 by the uncrossing procedure described in Fig. 7. We claim that $w(C_1) + w(C_2) = w(C)$, since the orientation of the dual-edges in C_1 and C_2 can be chosen to be the same as it is in C, inducing both $w_1(C_1) + w_1(C_2) = w_1(C)$ and $w_2(C_1) + w_2(C_2) = w_2(C)$ for any $s-t$ path P. Furthermore, $l(C_1) + l(C_2) \leq l(C) + 2$. By repeating the uncrossing procedure, we have closed dual walks $C_1, C_2, \ldots, C_{w(C)}$ such that $w(C_i) = 1$ for each i, and $\sum_i l(C_i) \leq l(C) + 2 \cdot (w(C) - 1)$. Since we have

$$\min_i l(C_i) \leq \left\lfloor \frac{1}{w(C)} \sum_i l(C_i) \right\rfloor \leq \left\lfloor \frac{l(C)-2}{w(C)} \right\rfloor + 2 \leq \left\lfloor \frac{l(C)}{w(C)} \right\rfloor + 2,$$

there exists a closed dual walk C_i such that $w(C_i) = 1$ and $l(C_i) \leq \lfloor l(C)/w(C) \rfloor + 2$. This shows the existence of an SRLG cut of size at most $\lfloor l(C)/w(C) \rfloor + 2$, and this can be found in linear time. □

6 Discussion

6.1 Heuristics improving the performance of the algorithm

6.1.1 Additional exit criteria

Similarly to [31], if $P_l = P_{l-k}$ holds for $k-1$ consecutive iterations (in line 3 of Algorithm 1), then we can stop, since this means that $[P_l, \ldots, P_{l-k+1}]$ will remain the same set of paths for the rest of the iterations. Note that since the set consists only of $k-1$ paths instead of k this can only happen, when $k = \text{MAX-FLOW} + 1$.

6.1.2 A heuristic approach to reduce path lengths

After the completion of Alg. 1, similarly to [31], a heuristic shortening of the $k =$ MAX-FLOW disjoint paths can be applied as follows. In each iteration, we fix $k-1$ paths, and we compute a shortest s-t path that is SRLG-disjoint from these. The algorithm stops when there are no $k-1$ paths for which a shorter disjoint s-t path exists as the current k^{th} path. As the total length of the paths decreases after each successful shortening, the heuristic terminates after a finite number of iterations.

6.2 Additional natural constraint and tighter min-max theorem

The following Property 3 is not demanded for the SRLG set $\mathcal{S}$, but if it fulfills it, a stronger max flow- min cut theorem can be stated:

Property 3. Suppose that two paths P_1 and P_2 in the duals of SRLGs $S_1, S_2 \in \mathcal{S}$ are crossing in a face $f \in V^*$. Then, there exists an SRLG $S_3 \in \mathcal{S}$ such that S_3^* involves f, some end-faces f_1 and f_2, and $f - f_1$ and $f - f_2$ sub-paths of P_1, and P_2, respectively.

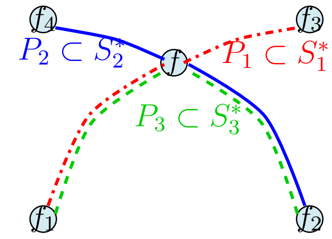


Figure 8: Illustration for Property 3. Nodes are vertices of the dual graph.

If this property holds, the uncrossing procedure used in Theorem 2.1 can be done in such a way that $l(C) \leq l(C_1) + l(C_2) + 1$. This means that in this case, MIN-CUT $\leq$ MAX-FLOW $+ 1$. Thus, we can state:

Corollary 6.1. *For any instance of the MRSDP problem, where Properties 1 and 3 hold,*

$$\textit{MAX-FLOW} \leq \textit{MIN-CUT} \leq \textit{MAX-FLOW} + 1.$$

We note that Property 3 holds in many natural settings, including the model of [18], where the geographical embedding of the network is known and SRLGs are induced by all the circular discs of the same radius.

6.3 Dealing with non-planar graphs

This paper assumed the network topology to be planar, which enabled the design of a polynomial algorithm for calculating a maximal number of regional SRLG-disjoint paths. Naturally rises the question if the problem can be solved efficiently if there are a strictly positive number x of link crossings in any embedding of the network in the plane. We believe the answer is affirmative. To argue, in the following, we present a very heuristic approach as follows. We assume that for any crossing link pairs e, f there is an SRLG S containing e and f. This means that there are no s-t paths P_1 and P_2 containing e and f, respectively. We also ban every single path to use both crossing edges. Then, the MAX-FLOW in $G \setminus \{e\}$ or in $G \setminus \{f\}$ will be a maximal solution in the original graph too. It is easy to see that in the presence of x non-overlapping link crossings, we can find the MAX-FLOW via solving 2^x planar problem instances, where we delete one edge of each crossing. If x is $O(\text{poly}(\log n))$, this means a runtime polynomial in n. A more elaborated study on calculating a maximal number of regional SRLG-disjoint s-t paths in a network with some link crossings will be part of a future work.

7 Related Works

7.1 Theoretical preludes

Papers [23] and [22] provided polynomial algorithms and min-max theorems to find a maximal number of interiorly d-hop disjoint paths (i.e., no walk of length d is connecting any pair of these paths) in planar graphs, for $d = 1$, and $d \geq 1$, respectively. The condition of interiorly d-hop disjointness can be rephrased as interiorly SRLG-disjointness for a special class of primal-connected SRLGs.

Based on the former, and motivated by [28], [18] and [31] designed a tight min-max theorem and faster polynomial algorithms for finding a maximal number of circular disk-disjoint paths in geometric graphs without link crossings. The disk-disjointness can be rephrased as SRLG-disjointness for a special class of dual-connected SRLGs.

7.2 Prior works related to SRLG-disjoint routing

To the best of our knowledge, [17] was the first to prove that the problem of finding two SRLG-disjoint paths is NP-complete via showing the NP-hardness of one of its special cases, the so-called fiber-span-disjoint paths problem.

[6] corrects [21], and shows that the SRLG-disjoint routing is NP-complete even if the links of each SRLG S are incident to a single node v_S. It also presents some polynomially solvable subcases of this special problem.

[44] offers an ILP solution for the SRLG-disjoint routing problem. Some papers, like [13, 9] rely at least partly on ILP/MILP formulations, i.e., on (mixed) integer linear programs to solve or approximate the weighted version of the SRLG-disjoint paths problem.

Under a probabilistic SRLG model,[19] aims finding diverse routes with minimum joint failure probability via an integer non-linear program (INLP).

Due to the complexity of the problem family, heuristics are also investigated [43, 45], unfortunately, with issues ranging from possibly non-polynomial runtime to possibly arising forwarding loops in the presence of disasters.

8 Simulation Results

In this section, we present numerical results to demonstrate the performance of the proposed algorithms on some realistic physical networks. The algorithms were

Table 1: Backbone network topologies used in the simulations [30]. The *diam* is the physical length of the longest shortest path, *cable* is the total physical length of the cables, k^* is the average number of node disjoint paths between the node-pairs.

Network name	$\|V\|$	$\|E\|$	diam. [km]	cable [km]	k^*	d_{avg}	d	ρ_{avg}	ρ	$\|\mathcal{S}_{region}\|$
						avg. over all SRLGs of Table 2				
Pan-EU	16	22	1713	6321	2.72	2.70	3.00	4.27	5.39	9.56
EU (Nobel)	28	41	3314	16864	2.69	2.78	3.50	4.05	5.61	23.22
N.-American	39	61	5121	32796	2.89	3.07	3.89	4.03	5.39	31.00
US (NFSNet)	79	108	5502	37071	2.85	2.89	3.67	3.99	6.22	63.00
US (Fibre)	170	230	5695	41530	2.42	3.20	4.83	7.18	14.61	107.00
US (Sprint-Phys)	264	313	5539	40595	2.00	2.88	4.11	6.65	13.39	156.94
US (Att-Phys)	383	488	5617	58866	2.46	3.29	5.00	9.06	18.78	234.11

implemented in Python version 3.8 using various libraries. Our implementation of the algorithm and the input data used for evaluation is uploaded to a public repository[2]. Runtimes were measured on a commodity laptop with a CPU at 2.8 GHz and 8 GB of RAM. We investigate various aspects of system performance, e.g., how the list of SRLGs or the network parameters impacts the number of SRLG-disjoint paths, their length, and runtime.

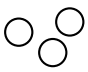
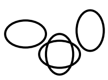
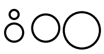
For the performance evaluation of the algorithms, we selected seven topologies (see Table 1 for the details) and analyzed the results for various known lists of SRLGs (Table 2). We have adopted four approaches to generate SRLGs:

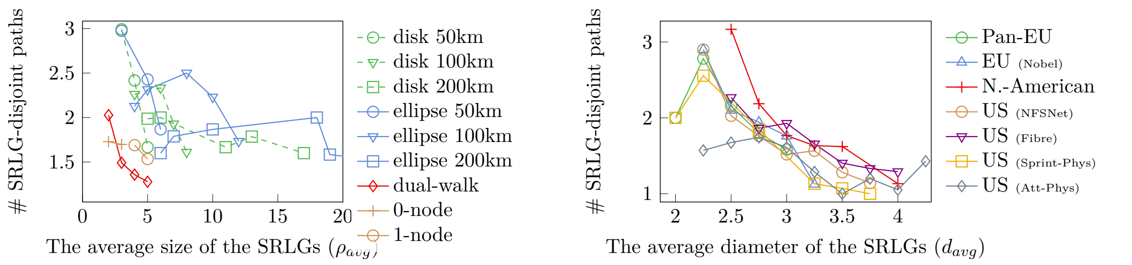
1. circular disk failures of a given radius like in [18],
2. ellipse disk failures of a given radius,
3. circular disks with $k = 0, 1$ nodes in their interior and
4. random walks in the dual graph.

For 1) we have set radius to $r = 50, 100, 200, 300$km and used the algorithm in [39] to generate the SRLGs that over every possible epicenter for the circular disk. For 2), first, we have transformed the node coordinates by multiplying the vertical coordinates (the latitude values) by 0.5 and run the algorithm in [39] to generate the SRLGs. After transforming back the coordinates, we have SRLGs covered by an ellipse where the minor axis is 2 times longer than the major axis. We perform a second round of generating SRLGs but multiply the horizontal coordinates (the longitude values) by 0.5. For 3) we select SRLGs that can be covered with a circular disk having $k = 0, 1$ nodes in its interior. This will result in a circular disk with different radii, and the generation is based on the Delaunay graphs, see [41]. For 4), we generated SRLGs as random walks in the dual graph with $\rho = 2, 3, 4, 5$ dual edges and the number of

[2] `https://github.com/hajduzs/regsrlg`

Table 2: The list of SRLGs used in the simulation. The minimal, average, and maximal diameter of the dual of an SRLG is denoted by d_{min}, d_{avg} and d, respectively. The minimal, average and maximal size of an SRLG is denoted by ρ_{min}, ρ_{avg} and ρ. The number of SRLGs is $|\mathcal{S}_{region}|$. All the values in the table are averages over the networks shown in Table 1.

SRLG name	d_{min}	d_{avg}	d	ρ_{min}	ρ_{avg}	ρ	$\lvert\mathcal{S}_{region}\rvert$	illustration
disk 50km	1.43	2.27	3.57	2.00	3.41	7.86	103.71	
disk 100km	1.71	2.71	4.00	2.71	5.25	11.14	96.71	
disk 200km	1.43	3.08	4.29	2.57	8.88	18.00	117.00	
ellipse 50km	1.43	2.30	3.71	2.00	3.64	8.14	102.71	
ellipse 100km	1.71	2.79	4.00	2.86	5.90	11.71	99.14	
ellipse 200km	1.57	3.18	4.57	2.57	10.55	21.29	115.57	
0-node	1.43	2.34	3.86	1.14	2.18	4.57	122.43	
1-node	1.71	2.68	4.14	2.29	4.05	7.00	145.86	
dual-walk	2.59	3.17	3.84	3.50	3.50	3.50	57.25	



(a) Num. of SRLG-disjoint paths vs. avg. number of edges in the SRLGs

(b) Number of SRLG-disjoint paths vs. average diameter of the SRLGs

Figure 9: The number of SRLG-disjoint paths compared to the size of the SRLGs.

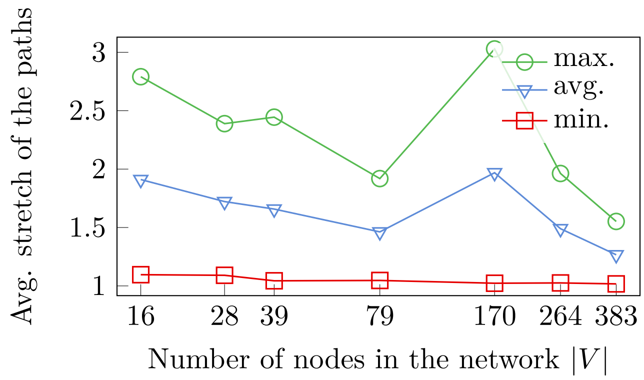


Figure 10: Average stretch of SRLG-disjoint paths for each network

SRLGs is $\lfloor {}^{|E|}/_{\rho} \rfloor$. Finally, for a given s and t, the SRLGs that form an s-t cut are omitted.

8.1 Larger SRLGs lead to less number of SRLG-disjoint paths

In this section, we investigate the correlation of the number of SRLG-disjoint paths with respect to the size of the SRLGs. We expect that having larger SRLGs results in less number of SRLG-disjoint paths. Fig. 9 shows two charts where the vertical axis is the number of SRLG-disjoint paths; and the horizontal axis is the size of SRLG in terms of the number of edges (Fig. 9a) and the diameter (Fig. 9b) of the SRLGs. On Fig. 9a we draw different curve for each type of SRLG of Table 2 and on Fig. 9a we draw different curve for each network of Table 1. We can observe that 0-node, 1-node, and dual-walk SRLGs are smaller than the methods where SRLGs have fixed physical sizes (disk and ellipse). The backbone network is denser in heavily populated areas (e.g., east and west coast in the USA). On Fig. 9b we can observe that larger networks have larger SRLGs as well (it can be also seen on Table 1). We can also observe that for larger networks, the impact of the size of the SRLG decreases.

8.2 Increase in the path lengths

We have also investigated the length of the paths. Fig. 10 shows the stretch, i.e., the length of the path divided by the shortest path, where the lengths are the physical length of the paths. The figure shows the length of the shortest paths among the k SRLG-disjoint paths. We can observe that it is just 1%-10% longer than the shortest path. It is essential in network resiliency because only one of the paths is set up, called the working path, while the others are the backup paths set up only in case of failure. It also shows that the longest paths among the k SRLG-disjoint paths have stretch 2-3. As expected, for networks with more nodes and links, the difference is smaller. The chart also shows the average stretch over all the k SRLG-disjoint paths. Note that, on average, there were 2.05 SRLG-disjoint paths in our evaluation.

8.3 Running time

We have also measured the running time of the proposed algorithm. Fig. 11 shows the running times for networks of different sizes. The horizontal axis shows the number of nodes in the network on a logarithmic scale. We have sorted the running times depending on the maximal diameter of the SRLGs that was $d = 3, 4, 5, 6$ to illustrate that the algorithm runs in a moderately longer time for larger SRLGs. In general, we observe a scalable performance with a quadratic increase in the runtime with respect to the number of nodes.

9 Conclusions

Finding SRLG-disjoint paths in a network between a given pair of nodes is essential to network resiliency. The problem, in general, was known to be computationally complex; thus, heuristic algorithms (mostly Integer Linear Programming) were used. It was observed that heuristic algorithms perform well in most cases; however, they cannot provide the performance guarantee required in operational networks. Therefore, the best practice remained to degrade the requirements in the Service Level Agreements to protect the network against single (or dual) link/node failures. It eventually leads to networks being very reliable except during natural disasters (e.g., earthquakes, flooding, hurricanes), where multiple pieces of equipment in a small area fail within a short time, called regional failures.

On the other hand, although several NP-hard problems can be efficiently solved for planar graphs, the (almost) planarity of backbone network topologies has not yet been exploited in previous approaches. In the last decades, most of the related algorithmic tools were already available in geometric topology to close this gap [22] and precisely identify the properties SRLGs must meet to have fast algorithms for finding SRLG-disjoint paths. An important step was on this road in 2014 by Kobayashi-Otsuki [18], giving a polynomial-time algorithm for one particular type of SRLGs (circular disk failures of a given radius). This paper aims to close this gap, and generalize the algorithm for a broader range of SRLGs that covers all cases in practice (the edges in

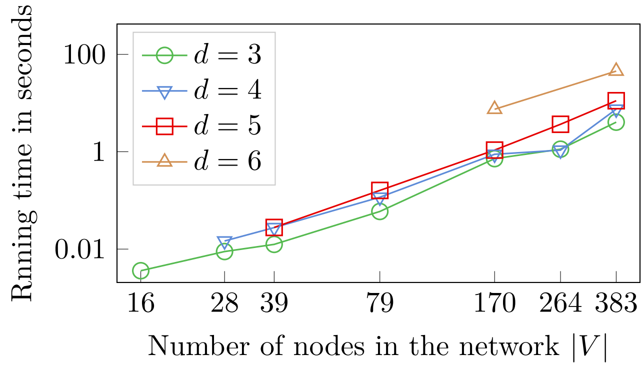


Figure 11: The runtime for each network

the dual graph must be connected), show that the algorithm is very efficient by proving that the runtime of the algorithm is $O(n^2)$ roughly (with additional, in most cases small parameters). Furthermore, we give a pure combinatorial algorithm description that does not utilize the exact geographical embedding of the network. We provide a Python implementation and show that, on average, one of the resulting SRLG-disjoint paths is almost as short as the absolute shortest path through simulations.

Acknowledgement

The authors would like to express their sincere gratitude to Yusuke Kobayashi for his support in the early stage of this research.

This research was partially supported by the National Research, Development and Innovation Fund of Hungary (grant No. 124171, 128062, 134604, 135606, and FK 132524). The research reported in this paper was supported by the BME Artificial Intelligence TKP2020 IE grant of NKFIH Hungary (BME IE-MI-SC TKP2020). Research supported in part by National Research, Development and Innovation Office - Grant No. NKFI-115288. Supported by the ÚNKP-22-4-II-BME-248 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. Application Domain Specific Highly Reliable IT Solutions" project has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

[1] Shaping Europe's digital future, Actors in the broadband value chain. European Commission, `https://ec.europa.eu/digital-single-market/en/actors-broadband-value-chain`. Accessed: 2019.

[2] Alcatel-Lucent 7705 service aggregation router OS, MPLS guide, 2013. Release 6.0.R4.

[3] Cisco WAE Design 7.1.2 User Guide, Chapter 19 : LSP Disjoint Path Optimization, 2018.

[4] Huawei NE20E-S Universal Service Router, feature description - MPLS, 2018. V800R010C10SPC500.

[5] Juniper Networks Junos® OS MPLS Applications User Guide, Chapter 3 MPLS Traffic, Shared Risk Link Groups for MPLS, 2020.

[6] J.-C. Bermond, D. Coudert, G. D'Angelo, and F. Z. Moataz. SRLG-diverse routing with the star property. In *Design of Reliable Communication Networks (DRCN)*, pages 163–170. IEEE, 2013.

[7] D. Bienstock. Some generalized max-flow min-cut problems in the plane. *Mathematics of Operations Research*, 16(2):310–333, 1991.

[8] O. Crochat, J.-Y. Le Boudec, and O. Gerstel. Protection interoperability for WDM optical networks. *IEEE/ACM Trans. Netw.*, 8(3):384–395, 2000.

[9] A. de Sousa, D. Santos, and P. Monteiro. Determination of the minimum cost pair of D-geodiverse paths. In *DRCN 2017 - 13th International Conference on the Design of Reliable Communication Networks*, Munich, Germany, March 8-10 2017.

[10] F. Dikbiyik, M. Tornatore, and B. Mukherjee. Minimizing the risk from disaster failures in optical backbone networks. *J. Lightw. Technol.*, 32(18):3175–3183, 2014.

[11] G. Ellinas, E. Bouillet, R. Ramamurthy, J.-F. Labourdette, S. Chaudhuri, and K. Bala. Routing and restoration architectures in mesh optical networks. *Optical Networks Magazine*, 4(1):91–106, January/February 2003.

[12] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo. Elastic optical networking: A new dawn for the optical layer? *IEEE Commun. Mag.*, 50(2):s12–s20, 2012.

[13] T. Gomes, M. Soares, J. Craveirinha, P. Melo, L. Jorge, V. Mirones, and A. Brízido. Two heuristics for calculating a shared risk link group disjoint set of paths of min-sum cost. *Journal of Network and Systems Management*, 23(4):1067–1103, 2015.

[14] M. F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee. Design of disaster-resilient optical datacenter networks. *J. Lightw. Technol.*, 30(16):2563–2573, 2012.

[15] I. B. B. Harter, D. Schupke, M. Hoffmann, G. Carle, et al. Network virtualization for disaster resilience of cloud services. *IEEE Commun. Mag.*, 52(12):88–95, 2014.

[16] J. Heidemann, L. Quan, and Y. Pradkin. *A preliminary analysis of network outages during hurricane Sandy.* University of Southern California, Information Sciences Institute, 2012.

[17] J.-Q. Hu. Diverse routing in optical mesh networks. *IEEE Trans. Communications*, 51:489–494, 2003.

[18] Y. Kobayashi and K. Otsuki. Max-flow min-cut theorem and faster algorithms in a circular disk failure model. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 1635–1643, April 2014.

[19] H.-W. Lee, E. Modiano, and K. Lee. Diverse routing in networks with probabilistic failures. *IEEE/ACM Trans. Netw.*, 18(6):1895–1907, 2010.

[20] X. Long, D. Tipper, and T. Gomes. Measuring the survivability of networks to geographic correlated failures. *Optical Switching and Networking*, 14:117–133, 2014.

[21] X. Luo and B. Wang. Diverse routing in WDM optical networks with shared risk link group (SLRG) failures. In *Proceedings of 5th International Workshop on the Design of Reliable Communication Networks (DRCN 2005)*, Oct. 16-19 2005.

[22] C. MacDiarmid, B. Reed, and L. Schrijver. Non-interfering dipaths in planar digraphs. Jan. 1991.

[23] C. Mcdiarmid, B. Reed, A. Schrijver, and B. Shepherd. Induced circuits in planar graphs. *Journal of Combinatorial Theory, Series B*, 60(2):169 – 176, 1994.

[24] K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[25] B. Mukherjee, M. Habib, and F. Dikbiyik. Network adaptability from disaster disruptions and cascading failures. *IEEE Commun. Mag.*, 52(5):230–238, 2014.

[26] Y. Nemoto and K. Hamaguchi. Resilient ICT research based on lessons learned from the Great East Japan Earthquake. *IEEE Commun. Mag.*, 52(3):38–43, 2014.

[27] S. Neumayer, A. Efrat, and E. Modiano. Geographic max-flow and min-cut under a circular disk failure model. In *IEEE INFOCOM*, pages 2736–2740, 2012.

[28] S. Neumayer, A. Efrat, and E. Modiano. Geographic max-flow and min-cut under a circular disk failure model. *Computer Networks*, 77:117–127, 2015.

[29] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano. Assessing the vulnerability of the fiber infrastructure to disasters. *IEEE/ACM Trans. Netw.*, 19(6):1610–1623, 2011.

[30] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski. SNDlib 1.0: survivable network design library. *Networks*, 55(3):276–286, 2010.

[31] K. Otsuki, Y. Kobayashi, and K. Murota. Improved max-flow min-cut algorithms in a circular disk failure model with application to a road network. *European Journal of Operational Research*, 248(2):396–403, 2016.

[32] C. S. Ou and B. Mukherjee. *Survivable Optical WDM Networks*. Springer Science & Business Media, 2005.

[33] K. Pithewan, M. Yaaseen, R. P. Ramamoorthy, and M. Misra. Disjoint path computation for arbitrary directed graph, 2016. US Patent 9,253,032.

[34] M. Rahnamay-Naeini, J. E. Pezoa, G. Azar, N. Ghani, and M. M. Hayat. Modeling stochastic correlated failures and their effects on network reliability. In *IEEE Int. Conf. on Comp. Comm. and Networks (ICCCN)*, pages 1–6, 2011.

[35] L. Shen, X. Yang, and B. Ramamurthy. Shared risk link group (SRLG)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks. *IEEE/ACM Transactions on networking*, 13(4):918–931, 2005.

[36] R. Souza Couto, S. Secci, M. Mitre Campista, K. Costa, and L. Maciel. Network design requirements for disaster resilience in IaaS clouds. *IEEE Commun. Mag.*, 52(10):52–58, 2014.

[37] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4:125–145, 1974.

[38] J. Tapolcai, L. Rónyai, B. Vass, and L. Gyimóthi. List of shared risk link groups representing regional failures with limited size. In *IEEE INFOCOM*, Atlanta, USA, May 2017.

[39] J. Tapolcai, L. Rónyai, B. Vass, and L. Gyimóthi. Fast Enumeration of Regional Link Failures Caused by Disasters With Limited Size. *IEEE/ACM Transactions on Networking*, 28(6):2421–2434, 2020.

[40] B. Vass, L. Németh, and J. Tapolcai. The Earth is nearly flat: Precise and approximate algorithms for detecting vulnerable regions of networks in plane and on sphere. *Networks, Wiley*, 2020.

[41] B. Vass, J. Tapolcai, and E. Bérczi-Kovács. Enumerating maximal shared risk link groups of circular disk failures hitting k nodes. *IEEE Transactions on Networking*, 2021.

[42] B. Vass, J. Tapolcai, Z. Heszberger, J. Bíró, D. Hay, F. A. Kuipers, J. Oostenbrink, A. Valentini, and L. Rónyai. Probabilistic shared risk link groups modelling correlated resource failures caused by disasters. *IEEE Journal on Selected Areas in Communications (JSAC) - issue on Latest Advances in Optical Networks for 5G Communications and Beyond*, 2021.

[43] K. Xie, H. Tao, X. Wang, G. Xie, J. Wen, J. Cao, and Z. Qin. Divide and conquer for fast SRLG disjoint routing. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 622–633, 2018.

[44] D. Xu, G. Li, B. Ramamurthy, A. Chiu, D. Wang, and R. Doverspike. SRLG-diverse routing of multiple circuits in a heterogeneous optical transport network. In *8th International Workshop on the Design of Reliable Communication Networks (DRCN 2011)*, pages 180–187, 2011.

[45] B. Yang, J. Liu, S. Shenker, J. Li, and K. Zheng. Keep forwarding: Towards k-link failure resilient routing. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1617–1625. IEEE, 2014.

[46] S. Yang, S. Trajanovski, and F. Kuipers. Availability-based path selection and network vulnerability assessment. *Wiley Networks*, 66(4):306–319, 2015.

[47] D. Zhou and S. Subramaniam. Survivability in optical networks. *IEEE network*, 14(6):16–23, 2000.