NYILATKOZAT

Név: Barabás Ábel

ELTE Természettudományi Kar, szak: Matematika

NEPTUN azonosító: CJWWYL

Szakdolgozat címe:
Polynomial-Time Algorithm for the Regional SRLG-disjoint Paths Problem

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2022.05.07

Barabás Ábel

a hallgató aláírása

EÖTVÖS LORÁND UNIVERSITY

FACULTY OF SCIENCE

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

BSC THESIS

Ábel Barabás
Mathematics BSc

Supervisors:

Erika Bérczi-Kovács
Department of Operations Research

Balázs Vass
BME, Department of Telecommunications and Media Informatics

Budapest, 2022

Contents

Acknowledgments

I would like to thank my supervisors, Erika Bérczi-Kovács and Balázs Vass for their help and guidance in the past two years, János Tapolcai and Zsombor Hajdú for their work on the paper, and my family for their support.

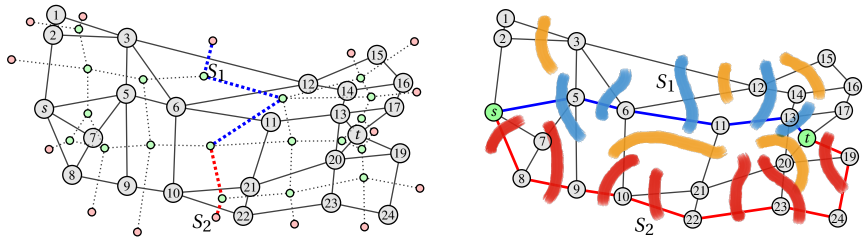
Chapter 1

Introduction

Routing is the process of choosing a path that connects two nodes of a network. To protect against the failure of network elements we can choose backup paths such that if the primary path fails there is still a route connecting the two nodes. The current best practice is to utilize network flow algorithms, such as Suurballe's algorithm [1], to efficiently compute link or node disjoint paths in the network topology graph. However, several papers studied [2–10] that the 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 links in a network subject to potential failure events is called Shared Risk Link Group (SRLG), identified during network planning [11–15].

Given an arbitrary topology graph and a list of SRLGs, finding two SRLG-disjoint paths is NP-complete [16, 17]. There are special cases where a polynomial algorithm is known, such as for circular disk failures in a planar graph provided by Kobayashi-Ostuki in [18]. In this thesis we show a slightly more general model, where the graph is planar and for each SRLG the set of its edges forms a connected subgraph in the dual. Later we relax this requirement a bit such that this class of SRLGs covers essentially all cases in practice, since if the disaster area is connected in the topological sense then the corresponding SRLG fulfills the required property. Furthermore, unlike in [18], the proposed algorithm does not require the exact embedding of the graph, which is useful for practical applications, since the exact coordinates of the network elements are often sensitive information. Instead, we only require knowing the dual of the graph.

The thesis is organised as follows: Sec. II provides the problem formulation and a simple upper bound on the number of SRLG-disjoint paths. Sec. III describes the proposed algorithm. Sec. IV gives a max-min theorem for the regional SRLG-disjoint paths problem. Sec. V describes a polynomial algorithm for finding a minimal SRLG-cut. Sec. VI heuristically shortens the paths, deals with a more general class of SRLGs, gives a tighter min-max theorem in a special case and deals with non-planar input graphs. Sec. VII overviews the related works. Sec. VIII presents our simulation results. Finally Sec. IX concludes the report.



(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 ($\mathscr{S}_{region}$) 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.

Maximum Regional SRLG-disjoint Paths Problem (MRSDP)
Input: a planar graph $G = (V, E)$, one of its duals $G^* = (V^*, E^*)$, a bijection between the edges and their duals, two distinct nodes s and t, and a set $\mathscr{S} \subseteq 2^{|E|}$ of dual-connected SRLGs with $\mathscr{S}_V \subseteq \mathscr{S}$.

Find: maximum cardinality set of pairwise $\mathscr{S}$-disjoint s-t paths.

(c) MRSDP Problem definition. MAX-FLOW denotes the value of an optimal solution of this problem.

Figure 1.1: 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.

1.1 Results

This thesis is based on a TDK report, which is based on our accepted paper [1] [19] to IEEE INFOCOM 2022.

The main contributions of the paper and the report are the following:

1. We provide a broad definition of 'regional SRLGs' where the SRLG-disjoint routing can be efficiently solved. It is important to note that the list of SRLGs have to contain single node failures as well otherwise the problem is NP-complete [20].

2. We provide an efficient polynomial-time algorithm for the SRLG-disjoint routing and the minimal SRLG-cut problems. Our work is a generalization of ideas presented in [18] and [21].

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.

[1] https://bit.ly/31p63b9

Chapter 2

Problem Formulation and Upper Bounds

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 let $G^* = (V^*, E^*)$ be the dual of the planar graph G, see Fig. 1.1a.

When it does not confuse, we identify the faces of G with their duals in $G^*(V^*, E^*)$. In other words $G^*(V^*, E^*)$ is composed of a *face* set V^* and a *dual-edge* set E^*. In what follows, a link is sometimes called an edge. Based on G and G^*, a consistent clockwise order of the links incident to each node $v \in V$ can be easily calculated. Let $\mathscr{S}_{region} \subseteq 2^{|E|}$ be a set of link sets representing a set of *regional SRLGs*. Protecting single network element failures (link or node failures) is the current best practice (e.g., Huawei [22, Sec. 4.5.4], Alcatel-Lucent [23, pp. 46-50], Cisco Systems [24, Chpt. 19], Juniper [25, Chpt. 3], Infinera [26]). Thus for simplicity, we assume the set of SRLGs contains all the single link and node failures. It ensures the obtained SRLG-disjoint paths are node-disjoint s-t paths. Let E_v denote the set of links in G incident to a node v and let $\mathscr{S}_V$ represent the set of SRLGs modeling the node failures, i.e.,

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

Let $\mathscr{S}$ denote the set of all SRLGs: $\mathscr{S} = \mathscr{S}_{region} \cup \mathscr{S}_V$. Let ρ denote the maximum size of an SRLG: $\rho := \max\{|S| \big| S \in \mathscr{S}\}$, and let μ denote the maximum number of SRLGs that contain the same edge: $\mu = \max\{|T| : T \subset \mathscr{S}, |\cap_{S \in T} S| > 0\}$.

We say that two paths are **($\mathscr{S}$-)disjoint** if there is no SRLG $S \in \mathscr{S}$ intersecting both of them[1]. We may omit $\mathscr{S}$ from the notation when the SRLG set is clear from the context. When searching for $k = 2$ $\mathscr{S}$-disjoint paths P_1 and P_2, for algorithmic reasons, we will replace the constraint of disjointness with demanding the paths being clockwise ($\mathscr{S}$-)disjoint (exact definition in Sec. 3) from one another. Intuitively, P_1 being 'far enough' from P_2 in clockwise direction around s, combined with P_2 being also 'far enough' from P_1 in clockwise direction around s ensures that P_1 and P_2 are $\mathscr{S}$-disjoint.

For a link set $S \subseteq E$, let S^* be the set of duals of links of S. For an SRLG $S \in \mathscr{S}$, let $V^*(S) := \{f \in V^* | \text{there is a dual-edge } \{f, f'\} \in S^* \text{ for some } f'\}$. Let d denote the maximal diameter of the dual of an SRLG: $d := \max\{diam(S^*) | S \in \mathscr{S}\}$. We call a set of links $S \subseteq E$ **dual connected**, if the edge-induced subgraph of S^* is connected.

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

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 ($\mathcal{S}_V \subseteq \mathcal{S}$).*

Fig. 1.1c shows the problem definition the report focuses on. Let MAX-FLOW denote the optimal value of this problem, see Fig. 1.1b as an illustration.

We note that, in Sec. 6.2, we further relax Property 1 to deal with SRLG-s with their dual-connected components being (primal-)connected. As shown in the discussion, MAX-FLOWand MIN-CUTcan be calculated in the same complexity for this broader SRLG family as for the dual-connected SRLGs.

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

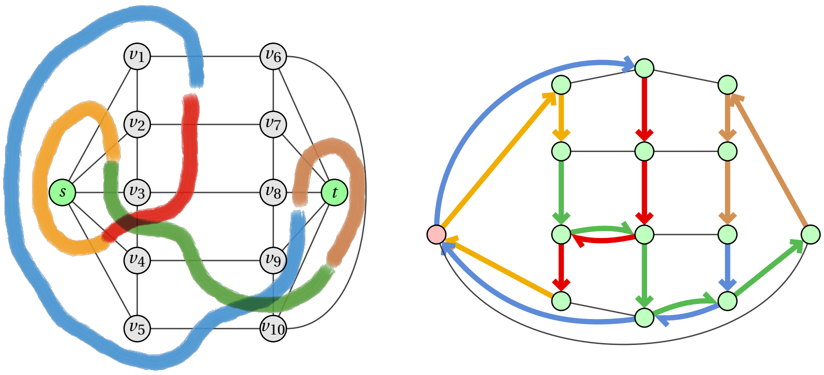
First, we define a trivial upper bound on MAX-FLOW using the analogy of max-flow min-cut theorems for network flows. A set of SRLGs from $\mathcal{S}$ that disconnect s from t is called an **SRLG cut** in this report, see SRLG S_1 and S_2 on Fig. 1.1b as an illustration. It is easy to see that the size of an SRLG cut is an upper bound for MAX-FLOW. It is because two disjoint paths cannot traverse any of these SRLGs simultaneously by definition. Note that the above holds for all SRLG cuts. Let MIN-CUT denote the minimum cardinality subset of $\mathcal{S}$ that disconnect s from t. Fig. 2.1a 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 4). In the rest of 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. 6). A **walk** is a finite sequence of edges which joins a sequence of vertices. 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, and consider the edges of P being oriented towards t. Let us consider a one-way orientation of the dual-edges of closed dual walk C. Let $w_1(C) = \{\#e_d \in C | e_d$ crosses an $e_p \in P$ from left to right$\}$. Similarly, $w_2(C) := \{\#e_d \in C | e_d$ crosses an $e_p \in P$ from right to left$\}$. Lastly, we define $w(C) := |w_1(C) - w_2(C)|$. E.g., the (colored) dual walk on Fig. 2.1b separates s and t three times. Note that $w(C)$ is indifferent to the choice of P and orientation of C.

Let $C = \{C_1, ..., C_k\}$ be a partition of the dual-edges of a closed walk in the dual-graph 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. 2.1b 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 1. *For any instance of the MRSDP problem, if MAX-FLOW ≥ 2, then*

$$\text{MAX-FLOW} \leq \min\left\{ \left\lfloor \frac{l(C)}{w(C)} \right\rfloor \middle| C \text{ is a closed dual walk, } w(C) \geq 1 \right\}. \tag{2.1}$$



(a) The network topology and the SRLGs ($\mathscr{S}_{region}$) are drawn with brush of unique color. No two SRLGs separate s and t, thus MIN-CUT = 3.

(b) The dual graph with a closed dual walk C such that $l(C) = 5$, $w(C) = 3$, and hence $l(C)/w(C) < 2$. This, by Lemma 1, means that MAX-FLOW = 1.

Figure 2.1: 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.

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 C_j is contained by the dual of an SRLG $S_j \in \mathscr{S}$. By measuring $w(C)$ at P_i, we can observe that since the paths are vertex disjoint (by Property 2), each C_j adds at most 1 to the value of $w(C)$, which can happen when it starts and ends on different sides of P_i, respectively. This means that each P_i has to intersect at least $w(C)$ sub-walks C_j. Since two disjoint paths cannot cross C at the same C_j, we have $l(C) \geq k \cdot w(C)$. The proof follows. □

Chapter 3

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

In this section we show that Lemma 1 can be extended into exact min-max theorem for MAX-FLOW, and Eq. (2.1) 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. We suppose any s-t path P is oriented from s to t.

3.1 Induction step

In what follows we show the equality in (2.1) 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 1.

We may assume that the first edges of $P_1, \dots, P_{k-1}$ occur in this order clockwise 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 3). 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}. In order to get a precise algorithm, in the following we define nearness and clockwise separation.

First, we give the definition of clockwise separation.

We say two s-t paths P_1 and P_2 are **crossing** if, after contracting their common edges, there is a node $v \in V \setminus \{s, t\}$ contained by both paths such that the links of the paths incident to v are alternating according to their incidence to P_1 and P_2. We note

Algorithm 1: Search for one more SRLG-disjoint path

Input: MRSDP problem input, $P_1,\ldots,P_{k-1}$ 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 $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k-1$

$P_0 := P_{k-1}$

for $l = k,\ldots,k+|V^*|$ **do**

  $P_l := P_{\text{nearest}}(P_{l-1}, P_{l-k})$ (see Alg. 2)

  if P_l, P_{l-k+1} *are disjoint* **then**

    return $P_{l-k+1},\ldots,P_{l-1},P_l$

return *a closed dual walk C in G^* with* $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k-1$

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.

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 for every link $e \in P$ if the dual edge e^* is in Q^*, then it crosses P from left to right. 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 those links of S which have duals reachable on a directed dual path Q^* starting at a neighboring face on the right of a link in $P \cap S$ such that Q^* is clockwise to P within the subgraph induced by S^* (see Fig. 3.1).

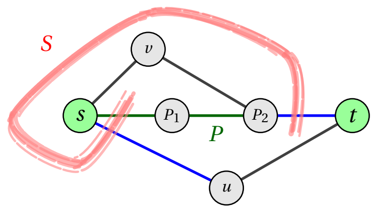


Figure 3.1: Clockwise part $\{su, P_2t\}$ of SRLG $S = \{su, sP_1, P_2t\}$ with respect to path $P = s, P_1, P_2, t$

For two s-t paths P_1 and P_2 without crossings, a pair (P_1, P_2) is **clockwise ($\mathscr{S}$-)disjoint** if for any SRLG S in $\mathscr{S}$ 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 pair (P_1, P_2) and (P_2, P_1) are clockwise disjoint.

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. 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).

3.2 Algorithm 2

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. 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 bijection between the s-t paths of G not crossing Q_2 and the s-t paths of 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 the left and right copies of the link (see Fig. 3.2). 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.

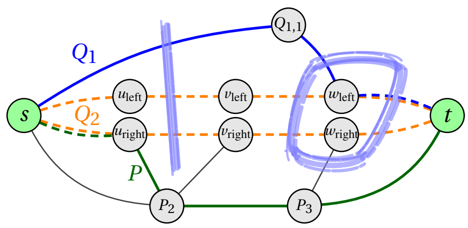


Figure 3.2: s-t path P that is minimum with respect to $<_{Q_2}$, clockwise-disjoint to Q_1 and does not cross Q_2

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

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

b) 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 disjoint. Since by induction P_{l-1} and P_{l-k+1} are disjoint, they are also clockwise 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

Algorithm 2: Nearest clockwise SRLG-disjoint path

```
Input: Planar graph G(V,E), SRLG set 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}).
return the s-t path in DFS-TREE
```

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 $e \in P_l, f \in P_{l-k+2}$. 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 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 $S^*_{\text{clw}}(P_l)$. 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. □

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 3. *For $i = N, \dots, k$, we can compute links $e_i \in E$, faces $f_i \in V^*$, SRLGs $S_i \in \mathscr{S}$, 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 $S_N \in \mathscr{S}$ such that a dual path Q^* in $S^*_{N_{\text{clw}}}(P_{N-1})$ connects the dual of a link $e_{N-1} \in P_{N-1}$ and e'_N, see also Fig. 7.1. 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 $i = N, \dots, k$ as prescribed in the statement of the claim. □

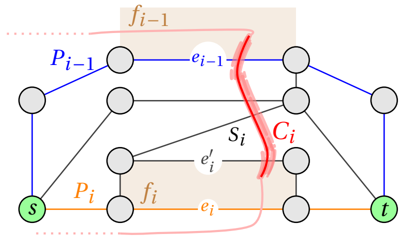


Figure 3.3: Illustration for Claim 3

For $i = N, \ldots, k$, let e_i, f_i, S_i, and C_i be as described in Claim 3. 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 3).

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 $\lfloor {}^{l(C)}/_{w(C)} \rfloor = k - 1$.

3.3 Base cases

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 $\mathscr{S}$-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 $\mathscr{S}_0, \mathscr{S}_1, \mathscr{S}_2, \ldots, \mathscr{S}_d = \mathscr{S}$ such a way that if two s-t-paths P, R are $\mathscr{S}_i$-disjoint then (P, P) is clockwise $\mathscr{S}_{i+1}$-disjoint, generalizing the inductive idea applied in [21].

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 at most i distance away from p^* in the subgraph of G^* induced by S^*. 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 $\mathscr{S}_0^* := \mathscr{S}_V^*$ and $\mathscr{S}_i^* := \mathscr{S}_V^* \cup \{S_{2^{i-1}}^*(p^*) |\ S \in \mathscr{S}, p^* \in V(S^*)\}$ ($i = 1 \ldots \lceil \log_2(d) \rceil$). Note that $\mathscr{S}_{\lceil \log_2(d) \rceil} = \mathscr{S}$.

Lemma 4. *Suppose that P and R are s-t paths that are* $\mathscr{S}_{i-1}$-disjoint. *Then the pair (P, P) is clockwise $\mathscr{S}_i$-disjoint.*

Proof. Assume indirectly that (P, P) is not clockwise disjoint. Then there is an SRLG $S_i \in \mathscr{S}_i$ 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 5. *Path Q^* is a shortest path from e^* to f^* in S_i^*.*

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. □

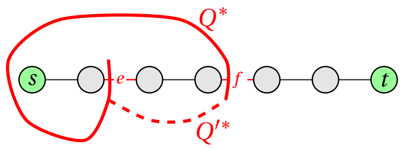


Figure 3.4: Illustration for Claim 5

By Claim 5 path Q^* can be chosen shortest, that is, we may assume it has at most 2^i edges. Since paths P and R are $\mathscr{S}_{i-1}$-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 $\mathscr{S}_{i-1}$, contradicting the assumption that (P, R) are $\mathscr{S}_{i-1}$-disjoint. □

Menger's Theorem [27] characterizes the maximum number of node-disjoint (that is, $\mathscr{S}_0 = \mathscr{S}_V$-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''. Our algorithm for finding an s-t path P such that (P, P) is clockwise $\mathscr{S}$-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 $\mathscr{S}_i$. If the algorithm finds two $\mathscr{S}_i$-disjoint s-t paths P_i' and P_i'', then 1) if $i \geq \lceil \log(d) \rceil$, we return with the $\mathscr{S}$-disjoint paths P_i' and P_i'', or else, 2) we go to the first step with path P_i' and SRLG set $\mathscr{S}_{i+1}$. In the other case, the algorithm finds a closed dual walk C as in Theorem 6 with $\mathscr{S}_i$, then we stop the process. Since for every $S \in \mathscr{S}_i$ $(1 \leq i \leq \lceil \log(d) \rceil)$ there is an SRLG $S' \in \mathscr{S}$ with $S \subseteq S'$, for this closed dual walk C we have $\lfloor l(C)/w(C) \rfloor \leq 1$ for $\mathscr{S}$, too. Note that if $\lfloor l(C)/w(C) \rfloor = 0$ we can subdivide some C_i to get a partition with $\lfloor l(C_i)/w(C_i) \rfloor = 1$.

3.4 Complexity Analysis

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

Theorem 6. *For any instance of the MRSDP problem, we can find a maximum number of $k = $ MAX-FLOW SRLG disjoint paths in $O\left(n^2 \left(k + \log(d)\rho\right)\rho\mu + |\mathscr{S}|\log(d)\rho^2\right)$, and* *we determine closed dual walk C in G^*, for which $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k$. For MAX-FLOW ≥ 2 we also have*

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

Proof. First we analyze Algorithm 2. The algorithm has two sections. The second is a DFS, which runs in $O(n)$. The first section runs in $O(n\rho\mu)$, since we go over each edge of Q_1 ($O(n)$), and then every SRLG which contains each edge ($O(\mu)$), and then compute $S_{\mathrm{clw}}(Q_1)$ in $O(\rho)$. The overall complexity of Algorithm 2 is O($n\rho\mu$).

In Algorithm 1, we call Algorithm 2 at most $|V^*|+1 = O(n)$ times, so the complexity of Algorithm 1 is $O(n^2\rho\mu)$.

In the base case, when we calculate the first s-t path, which is clockwise-disjoint from itself, first we determine SRLG sets $\mathscr{S}_i$ and then call Algorithm 1 $log(d)$ times.

There are $O(\rho)$ nodes in an SRLG, so $|\mathscr{S}_i|$ is $O(|\mathscr{S}|\rho)$, which means $\sum_{i\in\{1,\ldots,d\}} |\mathscr{S}_i|$ is $O(|\mathscr{S}|d\rho)$. We only have to construct $\mathscr{S}_i$ if i is a power of two smaller than d. This means that we can construct the truncated SRLG sets $\mathscr{S}_i$ in $O(|\mathscr{S}|\log(d)\rho^2)$ time. For an SRLG set $\mathscr{S}_i$ the maximum number of SRLGs that have a common edge can be larger than μ. Since for each SRLG $S \in \mathscr{S}$ we create $O(\rho)$ new SRLGs when we create $\mathscr{S}_i$, this number is $O(\mu\rho)$. So calling Algorithm 1 $\log(d)$ times takes $O(\log(d)n^2\rho^2\mu)$ time.

When two disjoint s-t paths are given, we execute algorithm k = MAX-FLOW times, which gives a running time of $O(kn^2\rho\mu)$ for this part.

So the total complexity of finding the maximum number of pairwise disjoint paths is $O(|\mathscr{S}|\log(d)\rho^2 + \log(d)n^2\rho^2\mu + kn^2\rho\mu)$.

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 3) in $O(n)$ time. □

Chapter 4

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

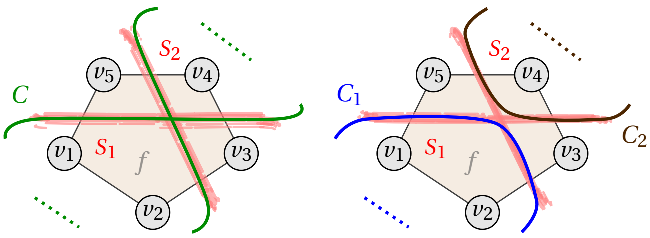
By using Theorem 6, we prove the following.

Theorem 7. *For any instance of the MRSDP problem,*

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

Proof. Since MAX-FLOW ≤ MIN-CUT is obvious, we prove MIN-CUT ≤ MAX-FLOW + 2. By Theorem 6, we can take a closed dual walk C such that $\lfloor l(C)/w(C) \rfloor = \text{MAX-FLOW}$. 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 $\mathscr{S}$ that disconnect s and t).

If $w(C) \geq 2$, 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. 6.1. 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 $w_2(C_1) + w_2(C_2) = w_2(C)$. Furthermore, $l(C_1) + l(C_2) \leq l(C) + 2$. By repeating the



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

Figure 4.1: 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 $l(C_1) + l(C_2) \leq l(C) + 2$.

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 $\lfloor l(C)/w(C) \rfloor + 2$. $\square$

Chapter 5

MIN-CUT in Polynomial Time

By Theorem 6, we can compute s-t paths $P_1,\dots,P_k$, that are pairwise separated, where $k :=$ MAX-FLOW. Furthermore, by Theorem 6, we can obtain an SRLG cut (that is, a collection of SRLGs disconnecting s and t) of size at most $k+2$. Since MAX-FLOW ≤ MIN-CUT, our remaining task is to find an SRLG cut of size k if one exists, or of size $k+1$ if one exists, similarly to the min-cut algorithm in [18].

Since the case of $k = 1$ is easy, in what follows, we suppose $k \geq 2$. We may suppose $P_1,\dots,P_k$ do not cross each other, and the first edges of $P_1,\dots,P_k$ occur in this order clockwise at s.

The duals of the edges of P_i and P_{i-1} ($i = 1,2,...,k$ and $P_0 = P_k$) contain a cut in the dual of G since the concatenation of the two paths is a cycle. The dual graph is partitioned into two parts by this cut and let R_i be the subgraph of G which is the dual of the part in G^*, that does not contain the rest of the paths if $k > 2$. If $k = 2$, then let R_1 and R_2 be the duals of the two parts. Intuitively, R_i is the region encircled by P_i and P_{i-1}. For $i = 1,\dots,k$, let $V_i^* \subseteq V^*$ denote the set of faces of G in R_i, where $P_0 := P_k$.

Lemma 8. *The dual edge set of an SRLG cut $S \subset \mathscr{S}$ of size k contains a closed dual walk separating s and t such that the dual walk can be partitioned into at most k paths, each covered by an SRLG in the cut.*

Proof. The set of edges contained in an SRLG in S contains an $s-t$-cut, which means there is a closed walk C in the dual graph using the edges of the SRLGs of S. If it has two non-consecutive edges from the same SRLG, there is a dual path P connecting them in that SRLG. The union of C and this P contains two closed dual walks such that both of them has P as its' subwalk. One of these two closed dual walks will have a winding number of at least one. Repeating this procedure for that closed dual walk we can construct a closed dual walk with $l(C') \leq k$. □

Lemma 9. *If MIN-CUT = MAX-FLOW then we can compute MIN-CUT in $O(\frac{n|\mathscr{S}|\rho^2}{k})$.*

Proof. If a cut of size k exists then by Lemma 8 we have a closed dual walk C with $w(C) \geq 1$ and $l(C) \leq k$. Since the existence of such a curve means there exists a cut of size $l(C)$ and $k =$ MAX-FLOW ≤ MIN-CUT, an SRLG cut of size k exists if and only if there exists a closed dual walk C with $w(C) = 1$ and $l(C) = k$. If $C = \{C_1,...,C_{l(C)}\}$, where each C_i is an SRLG-path, then each P_i intersects C_j for exactly one j since the paths are pairwise separated.

To check the existence of such a curve, we construct a directed graph $D(V^*, A)$, where, with $V^*_{k+1} := V^*_1$,

$$A := \big\{(f_i, f_{i+1})|\ \ i \in \{1,\dots,k\},\ \ f_i \in V^*_i,\ \ f_{i+1} \in V^*_{i+1}, \\ \exists S_i \in \mathscr{S} : \{f_i, f_{i-1}\} \in V^*(S^*_i),\ \ S_i \cap P_i \neq \emptyset\big\}.$$

Then, finding an SRLG cut of size k is equivalent to finding a directed cycle of length k in D. Since every directed cycle has a length of at least k, this can be done by finding the shortest directed cycle through every node in D by calling a BFS. Actually it is enough to do this for faces contained in R_1 since every directed cycle passes through R_i for every $i \in \{1,2,...,k\}$, so the running time is $O(\frac{nm}{k})$, where m is the number of edges in D and $m = O(|\mathscr{S}|\rho^2)$, since for each SRLG S there are $O(\rho^2)$ pairs of faces connected in the dual of S. □

It remains to decide whether there exists an SRLG cut of size $k+1$.

Lemma 10. *If MIN-CUT = MAX-FLOW+1 then we can compute MIN-CUT in* $O(n|\mathscr{S}|\rho^2)$.

Proof. If MIN-CUT = $k+1$ then $l(C) = k+1$ for the closed dual walk C described in Lemma 8. This means that if $C = \{C_1,...,C_{k+1}\}$ (a partition of the edges describe din the definition of l(C)), for exactly one i the dual path C_i connects two faces contained in R_j for some j.

That is, we argue that there is an SRLG cut of size $k+1$ exactly if there exists two faces $f_i, f_{i'}$ dual connected by an SRLG in $\mathscr{S}$ such that they are contained in R_j for some j and the length of the shortest path connecting them in D is k (note that it is at least k).

To find a cycle like this, first we find for every face the faces in V^*_i that are dual-connected to them in $V^*(S)$ for an SRLG $S \in \mathscr{S}$ and $i \in \{1,2,\dots,k\}$, by finding the connected components of the duals of each SRLG. For a face f let the set of such faces be $F_{\mathscr{S}}(f)$ (note that this set depends on $P_1, P_2, \dots, P_k$). This can be done in $O(|\mathscr{S}|\rho)$. Next for every face f we check if there exists a path of length k in D connecting f to a face in $F_{\mathscr{S}}(f)$. This can be done in $O(nm)$ by running a BFS from every face.

□

The algorithm is summarized in Algorithm 3. As a consequence of Lemma 9, Lemma 10 and Theorem 6 we get the following theorem:

Theorem 11. *Given a set of $\mathscr{S}$-separate paths of maximum cardinality, an optimal solution of the GMCRS can be computed in polynomial time. More precisely, in* $O(n|\mathscr{S}|\rho^2)$.

Algorithm 3: Find MIN-CUT

Input: Planar graph $G(V,E)$, SRLG set $\mathscr{S}$, pairwise $\mathscr{S}$-separate s-t paths P_1, $P_2,\ldots,P_k$, such that $k = \text{MAX-FLOW}$

Output: An SRLG-cut of minimal size

1 Using Algorithm 1 compute a closed dual walk C in G^* with $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k$ and the corresponding SRLG-cut

2 Construct the directed graph $D = (V^*, A)$ defined in Lemma 9

3 **for** $f \in V_1^*$ **do**

4   by BFS in D find the shortest directed cycle C containing f

5   **if** $|C| = k$ **then**

6   **return** *the SRLG-cut corresponding to C*

7 **for** $S \in \mathscr{S}$ **do**

8   compute the connected components of S

9 **for** $f \in V^*$ **do**

    compute $F_{\mathscr{S}}(f)$

10 **for** $f \in V^*$ **do**

11   by BFS find the shortest path P from f to a face in $F_{\mathscr{S}}(f)$

12   **if** $|P| = k$ **then**

13   **return** *the SRLG-cut corresponding to P*

14 **return** *the SRLG-cut computed in line 1*

Chapter 6

Discussion

6.1 Heuristics improving the performance of the algorithm

6.1.1 A heuristic approach to reduce path lengths

After the completion of Alg. 1, similarly to [28], 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 iteration.

6.1.2 Additional exit criteria

Similarly to [28], if $P_l = P_{l-k}$ holds for $k-1$ 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 $k = \text{MAX-FLOW} + 1$.

6.2 Dealing with primal-dual connected SRLG-s

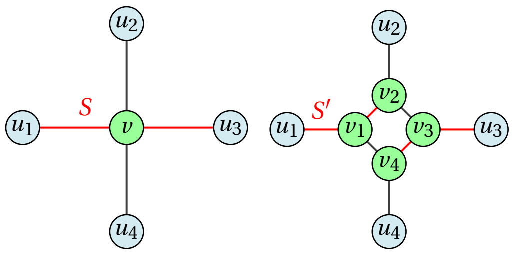
We say a **primal-dual walk** is a sequence of edges such that consecutive edges are adjacent either in the primal or the dual graph. We call an SRLG S primal-dual connected if for every pair of edges in S, there is a primal-dual walk connecting them. Let $\mathscr{S}$ be a set of primal-dual connected SRLGs (including the node-SRLGs). We can find the maximum number of $\mathscr{S}$-disjoint $s-t$-paths if the SRLGs are primal-dual connected (which does not mean they are dual-connected) by modifying the graph: we replace every node $v \in V \backslash \{s,t\}$ with a cycle C_v such that the size of C_v is $d(v)$ and every new node of C_v is connected with exactly one original neighbor of v. Let G' be the transformed graph.

Lemma 12. *We can transform the SRLGs in $\mathscr{S}$ (let the set of transformed SRLGs be $\mathscr{S}'$) such that the transformed SRLGs are dual-connected in G' and for every set of*

$\mathscr{S}'$*-disjoint* $s-t$*-paths in* G' *there is a set of* $\mathscr{S}$*-disjoint* $s-t$*-paths in* G *of the same size.*

Proof. Let $S \in \mathscr{S}$ be a primal-dual connected, but not dual-connected SRLG. The dual-connected components of S are connected by nodes in G. Let v be such a node, and e, f be two edges in S which are in two different components and are connected by v. If we add two edges from C_v to S such that one of them is next to e and the other one is next to f the two components will be connected in the dual. If the number of components in S is l, then adding $2l$ edges to S will transform it to a dual-connected SRLG. We also add every C_v to $\mathscr{S}'$.

Two $\mathscr{S}'$-disjoint $s-t$-paths in G' can not use the same C_v for a node of G, so the paths we get by contracting every C_v will be $\mathscr{S}$-disjoint. □



(a) original node v (b) C_v

Figure 6.1: The edges of S (u_1v and vu_3, colored red) are primal connected in G at node v. In the transformed graph we replace v with cycle $C_v = \{v_1, v_2, v_3, v_4\}$ and add edges v_1v_2 and v_3v_4 to S'.

If we solve MAX-FLOW$'$ in G' for $\mathscr{S}'$, then we get a solution for MAX-FLOW in G. Since for a maximal set of $\mathscr{S}$-disjoint paths in G, there are corresponding solutions in G', MAX-FLOW$'$ = MAX-FLOW.

The complexity stays the same because the maximum size of an SRLG in $\mathscr{S}'$ is $\max(O(\delta), \rho)$, where δ is the max degree in G, but ρ is at least $O(\delta)$ because of $\mathscr{S}_V$.

6.3 Additional natural constraint and tighter min-max theorem

The following Property 3 is not demanded for the SRLG set $\mathscr{S}$, 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* $S_1, S_2 \in \mathscr{S}$ *are crossing in a face* $f \in V^*$*. Then, there exists an SRLG* $S_3 \in \mathscr{S}$ *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.*

If this property holds, the uncrossing procedure used in Theorem 7 can be done in a way such that $l(C) \le l(C_1) + l(C_2) + 1$. This means that in this case, MIN-CUT $\le$ MAX-FLOW + 1. Thus, we can state:

Corollary 13. *For any instance of the MRSDP problem, where Property 3 holds,*

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

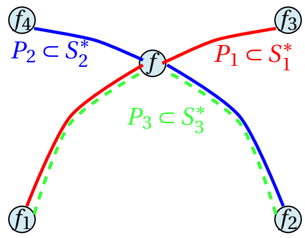


Figure 6.2: Illustration for Property 3. Note that the nodes are vertices of the dual graph.

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 induced by all the circular discs of the same radius.

6.4 Dealing with non-planar graphs

This report 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 of x 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 $O(\log n)$, 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.

Chapter 7

Related Works

7.1 Theoretical preludes

Papers [29] and [21] 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 primal-connected SRLGs.

Based on the former, and motivated by [30], [18] and [28] 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, [16] 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.

[31] corrects [32], 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.

[33] offers an ILP solution for the SRLG-disjoint routing problem. Some papers, like [34, 35] 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, [36] 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 [37, 38], unfortunately, with issues ranging from possibly non-polynomial runtime to possibly arising forwarding loops in the presence of disasters.

7.3 Capacitated and minimum cost SRLG-flows

Let u be a positive capacity vector on $\mathscr{S}$. A vector x on the set of $s-t$-paths is u-capacitated if $\forall S \in \mathscr{S} : \sum x(P) : P$ intersects $S \leq u(S)$, we also call a solution to the problem an u-capacitated $s-t$-flow.

[39, Theorem 3.1] states a theorem without a proof, so here is a reconstruction of the proof in the case of $u = \mathbf{1}$.

It follows from the following theorem, that if we can solve the fractional version of the problem, then we can find an integral solution.

Theorem 14. *For a planar graph G, and an $\mathbf{1}$-capacitated $s-t$-flow with value μ, there exists an integral $\mathbf{1}$-capacitated $s-t$-flow with value $\lfloor\mu\rfloor$*

Proof. The proof of this theorem has two sections. First we show that a flow can be chosen such that no $s-t$-paths assigned positive weight are crossing. In the second part we assign an integral weight to $s-t$-paths assigned positive weight in the fractional solution such that the statement of the theorem holds.

Let P_1 and P_2 be two $s-t$-paths that cross each other at node v. The weights of the paths are $0 < x_1 \le x_2$. Delete P_1, decrease the weight of P_2 by x_1 and add two new paths with weight x_1: $P_1[s,v] + P_2[v,t]$ and $P_2[s,v] + P_1[v,t]$.

For each SRLG, the sum of the weights of the paths intersecting it in the modified solution will not exceed the capacity, otherwise either P_1 or P_2 is not SRLG-self disjoint.

We can apply the operation described above, such that after finite number of steps, the procedure ends and the solution is as required.

For a node v let $f(v) = \sum x_P x_Q$ where P and Q are $s-t$ paths that cross each other at v or there is a node u in $P[s,v] \cap Q[s,v]$ such that $P[u,v]$ consists of the same edges as $Q[u,v]$ and they enter u in a different order than they exit at v.

After an uncrossing operation at node v, $f(w)$ does not change for $w \ne v$, and strictly decreases for v which means that the procedure is finite.

For the second part let the paths be $P_1, \ldots, P_k$ such that if $i < j$ then $P_i <_{P_1} P_j$. Let x' be the characteristic vector of the integral solution and let $x'(P_1) = 1$. Let $j_1 = 1$ and let j_{i+1} be the smallest index such that P_{j_i} and $P_{j_{i+1}}$ are SRLG-disjoint. Let $x'(P_{j_{i+1}}) = 1$. Since the paths are ordered, if there is an SRLG intersecting P_i and P_j for some $i < j$ it intersects P_l for either every $i < l < j$ or every $l < i$ and $j < l$. This means that there is an SRLG S_{j_i} such that it intersects P_l for every $j_i \le l < j_{i+1}$, which means that $\sum_{l=j_i}^{j_{i+1}-1} x(P_l) \le 1$.

This means that there is an SRLG S_{j_i} such that it intersects P_l for every $j_i \le l < j_{i+1}$, which means that $\sum_{l=j_i}^{j_{i+1}-1} x(P_l) \le 1$.

$P_{j_{\lfloor\mu\rfloor}}$ will be SRLG-disjoint from P_{j_1} otherwise $x\mathbf{1} < \mu$ since $S_{j_1}, \ldots, S_{j_{\lfloor\mu\rfloor}}$ intersects every path assigned positive weight in x so $x\mathbf{1} \le \sum_{i=1}^{\lfloor\mu\rfloor} x(P) : P$ intersects $S_i < \lfloor\mu\rfloor$ because P_1 would be covered by S_1 and $S_{\lfloor\mu\rfloor}$. This means that $P_{j_1}, \ldots, P_{j_{\lfloor\mu\rfloor}}$ are $\lfloor\mu\rfloor$ pairwise SRLG-disjoint $s-t$-paths. □

We do not know how to find a fractional u-capacitated flow problem in general. [39] gives a method for solving the problem in a special case where the number of paths passing 'near' an edge is bounded. Specifically, the d-environment of an edge e is the set of those edges and nodes for which there exists a path P with at most d elements of $V \cup E$ which connects them and e and P is not incident to s or t. The 1-environment of an edge e is the edge and its end nodes, the 2-environment is the 1-environment and the edges incident to e.

In the case when d is even we construct SRLGS corresponding to the d-environments of every edge. When d is odd then we contruct the SRLGs corresponding to the d-environments of nodes.

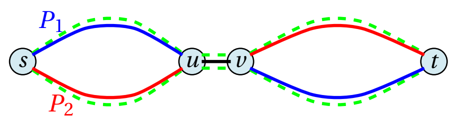


Figure 7.1: Illustration for the uncrossing procedure. The dashed paths are the new added ones.

Given a capacity vector on the d-environments of the edges, finding a u-capacitated vector of maximum size is NP-complete if $d > 2$. If $d = 2$ then we can construct an LP problem to find x so we can find the integral solution too. Let c be a cost-vector on the edges of d. Given that there exists an u-capacitated vector such that $\sum x(p) \geq k$, our goal is to find a minimum cost u-capacitated vector of the same size. [39, Theorem 3.3] states a theorem similar to Theorem 14 without proof. It states the if we have an u-capacitated flow with size k of minimal cost, then we can find an integral solution with the same cost (here the SRLGs are the 2-environments of edges). The first half of the proof of Theorem 14 works even if we have a cost function, but we have not proved the rest. We do not know if the minimum cost capacitated version of the problem is NP-complete.

Chapter 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 implemented in Python version 3.8 using various libraries. Our implementation of the algorithm and the input data used for evaluation is available at a public repository[1]. 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 8.1 for the details) and analyzed the results for various known lists of SRLGs (Table 8.2). 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.

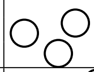
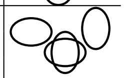
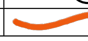
For 1) we have set radius to $r = 50, 100, 200, 300$km and used the algorithm in [41] to generate the SRLGs that over every possible epicenter for the circular disk. For 2), first, we have transformed the node coordinates by multiplying the vertical coordinates

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

Table 8.1: Backbone network topologies used in the simulations [40]. 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	$\|\mathscr{S}_{region}\|$
						avg. over all SRLGs of Table 8.2				
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

Table 8.2: 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 $|\mathscr{S}_{region}|$. All the values in the table are averages over the networks shown in Table 8.1.

SRLG name	d_{min}	d_{avg}	d	ρ_{min}	ρ_{avg}	ρ	$\|\mathscr{S}_{region}\|$	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	

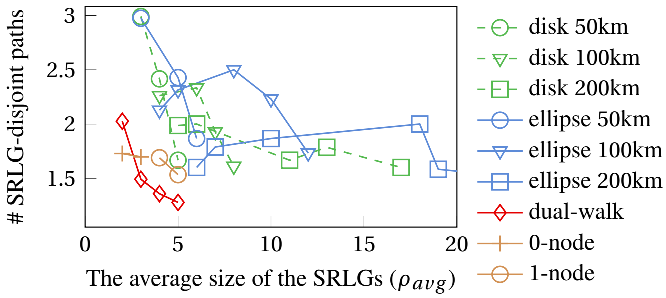
(the latitude values) by 0.5 and run the algorithm in [41] to generate the SRLGs. After transforming back the coordinates, we have SRLGs covered by an ellipse where the minor axis is 2 times longer than the major axis. We perform a second round of generating SRLGs but multiply the horizontal coordinates (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 [42]. For 4), we generated SRLGs as random walks in the dual graph with $\rho = 2, 3, 4, 5$ dual edges and the number of 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

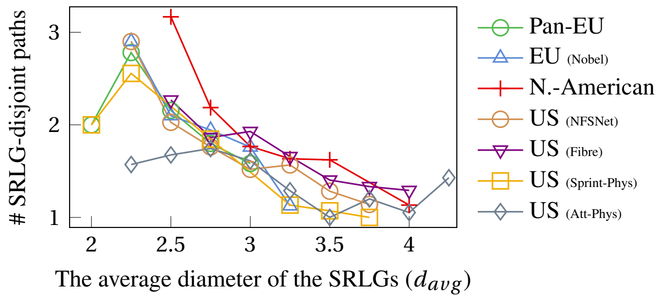
In this section, we investigate the correlation of the number of SRLG-disjoint paths with respect to the size of the SRLGs. We expect that having larger SRLGs results in less number of SRLG-disjoint paths. Fig. 8.1 shows two charts where the vertical axis is the number of SRLG-disjoint paths; and the horizontal axis is the size of SRLG in terms of the number of edges (Fig. 8.1a) and the diameter (Fig. 8.1b) of the SRLGs. On Fig. 8.1a we draw different curve for each type of SRLG of Table 8.2 and on Fig. 8.1a we draw different curve for each network of Table 8.1. 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. 8.1b we can observe that larger networks have larger SRLGs as well (it can be also seen on Table 8.1). 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. 8.2 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



(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 8.1: The number of SRLG-disjoint paths compared to the size of the SRLGs.

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. 8.3 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.

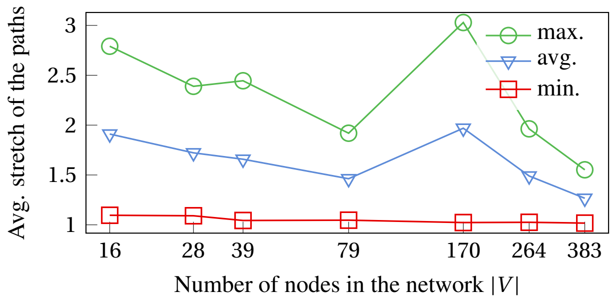


Figure 8.2: Average stretch of SRLG-disjoint paths for each network

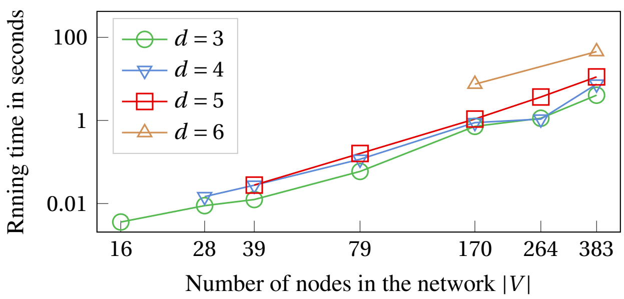


Figure 8.3: The runtime for each network

Chapter 9

Conclusions

Finding SRLG-disjoint paths in a network between a given pair of nodes is an essential task in 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 a 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, even though 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 [21] 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 report aims to close this gap, and generalize the algorithm for a broader range of SRLGs that covers all cases in practice (the primal-connected components of SRLG-s must be connected in the dual graph), 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 one of the resulting SRLG-disjoint paths is almost as short as the absolute shortest path through simulations.

Bibliography

[1] J. W. Suurballe, “Disjoint paths in a network,” *Networks*, vol. 4, pp. 125–145, 1974.

[2] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, “Assessing the vulnerability of the fiber infrastructure to disasters,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1610–1623, 2011.

[3] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo, “Elastic optical networking: A new dawn for the optical layer?” *IEEE Commun. Mag.*, vol. 50, no. 2, pp. s12–s20, 2012.

[4] M. F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee, “Design of disaster-resilient optical datacenter networks,” *J. Lightw. Technol.*, vol. 30, no. 16, pp. 2563–2573, 2012.

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

[6] F. Dikbiyik, M. Tornatore, and B. Mukherjee, “Minimizing the risk from disaster failures in optical backbone networks,” *J. Lightw. Technol.*, vol. 32, no. 18, pp. 3175–3183, 2014.

[7] I. B. B. Harter, D. Schupke, M. Hoffmann, G. Carle *et al.*, “Network virtualization for disaster resilience of cloud services,” *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 88–95, 2014.

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

[9] B. Mukherjee, M. Habib, and F. Dikbiyik, “Network adaptability from disaster disruptions and cascading failures,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 230–238, 2014.

[10] 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.*, vol. 52, no. 10, pp. 52–58, 2014.

[11] D. Zhou and S. Subramaniam, “Survivability in optical networks,” *IEEE network*, vol. 14, no. 6, pp. 16–23, 2000.

[12] O. Crochat, J.-Y. Le Boudec, and O. Gerstel, “Protection interoperability for WDM optical networks,” *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 384–395, 2000.

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

[14] S. Yang, S. Trajanovski, and F. Kuipers, “Availability-based path selection and network vulnerability assessment,” *Wiley Networks*, vol. 66, no. 4, pp. 306–319, 2015.

[15] 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.

[16] J.-Q. Hu, “Diverse routing in optical mesh networks,” *IEEE Trans. Communications*, vol. 51, pp. 489–494, 2003.

[17] 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*, vol. 4, no. 1, pp. 91–106, January/February 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*, April 2014, pp. 1635–1643.

[19] B. Vass, E. Bérczi-Kovács, A. Barabás, Z. L. Hajdú, and J. Tapolcai, “Polynomial-time algorithm for the regional SRLG-disjoint paths problem,” in *Proc. IEEE INFOCOM*, London, United Kingdom, May 2022.

[20] D. Bienstock, “Some generalized max-flow min-cut problems in the plane,” *Mathematics of Operations Research*, vol. 16, no. 2, pp. 310–333, 1991.

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

[22] “Huawei NE20E-S Universal Service Router, feature description - MPLS,” 2018, v800R010C10SPC500.

[23] “Alcatel-Lucent 7705 service aggregation router OS, MPLS guide,” 2013, release 6.0.R4.

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

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

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

[27] K. Menger, “Zur allgemeinen kurventheorie,” *Fundamenta Mathematicae*, vol. 10, no. 1, pp. 96–115, 1927.

[28] 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*, vol. 248, no. 2, pp. 396–403, 2016.

[29] C. Mcdiarmid, B. Reed, A. Schrijver, and B. Shepherd, "Induced circuits in planar graphs," *Journal of Combinatorial Theory, Series B*, vol. 60, no. 2, pp. 169 – 176, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0095895684710112

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

[31] 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)*. IEEE, 2013, pp. 163–170.

[32] 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.

[33] 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)*, 2011, pp. 180–187.

[34] 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*, vol. 23, no. 4, pp. 1067–1103, 2015.

[35] 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.

[36] H.-W. Lee, E. Modiano, and K. Lee, "Diverse routing in networks with probabilistic failures," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1895–1907, 2010.

[37] 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)*, 2018, pp. 622–633.

[38] 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*. IEEE, 2014, pp. 1617–1625.

[39] C. McDiarmid, B. Reed, A. Schrijver, and B. Shepherd, "Non-interfering network flows," in *Scandinavian Workshop on Algorithm Theory*. Springer, 1992, pp. 245–257.

[40] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "SNDlib 1.0: survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[41] 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*, vol. 28, no. 6, pp. 2421–2434, 2020.

[42] 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.