Efficient Algorithm for Region-Disjoint Survivable Routing in Backbone Networks

Erika Bérczi-Kovács, Péter Gyimesi, Balázs Vass, János Tapolcai

***Abstract*—Survivable routing is crucial in backbone networks to ensure connectivity, even during failures. At network design, groups of network elements prone to potential failure events are identified. These groups are referred to as *Shared Risk Link Groups* (SRLGs), and if they are a set of links intersected by a connected region of the plane, we call them regional-SRLGs. A recent study has presented a polynomial-time algorithm for finding a *maximum number of regional-SRLG-disjoint paths* between two given nodes in a planar topology, with the paths being node-disjoint. However, existing algorithms for this problem are not practical due to their runtime and implementation complexities.**

This paper investigates a more general model, the maximum number of non-crossing, regional-SRLG-disjoint paths problem. It introduces an efficient and easily implementable algorithmic framework, leveraging an arbitrarily chosen shortest path finding subroutine for graphs with possibly negative weights. Depending on the subroutine chosen, the framework improves the previous worst-case runtime complexity, or can solve the problem w.h.p. in near-linear expected time. The proposed framework enables the first additive approximation for a more general $\mathcal{NP}$-hard version of the problem, where the objective is to find the maximum number of regional-SRLG-disjoint paths. We validate our findings through extensive simulations.

I. Introduction

For a given graph $G = (V, E)$ with undirected topology, finding disjoint paths between two nodes s, t is the central algorithmic problem for any backbone network mechanism that aims to maintain connectivity in the event of a failure. Currently, the most widely used algorithm for this is to find edge- or node-disjoint paths, which is perfect for mechanisms dealing with single-point failures. However, extensive research [1]–[10] has revealed that network failures can manifest as multi-point failures, where a significant physical region experiences simultaneous equipment outages triggered by catastrophic events such as earthquakes, hurricanes, tsunamis, tornadoes, and more. These multi-point failures are often called regional failures or *regions* for brevity. Another widely used terminology is the Shared Risk Link Group (SRLG), which is more general and can be any set of edges subject to common failures [11]–[17]. We assumed that the list of regions (or SRLGs) $\mathscr{R} \subseteq 2^E$ is also part of the input, which was already identified during the network design phase based on some historical data and exploration of network vulnerabilities. Two st-paths are $\mathscr{R}$-disjoint if there is no edge set in $\mathscr{R}$ intersecting both paths.

The planarity of the network G is also assumed in our approach, similarly to the work presented in [18]. To protect sensitive information related to the exact location of network equipment, which is crucial for military and economic reasons, we do not require knowledge of the precise positions. However, we are provided with the dual representation of the planar topology graph, denoted as $G^* = (V^*, E^*)$ and a one-to-one mapping of primal and dual edges, see Fig. 3a. In the dual representation, each face f in the primal graph $G = (V, E)$ corresponds to a node $f^* \in V^*$ in the dual graph. Similarly, each edge e that separates faces f_1 and f_2 in G corresponds to a dual edge $e^* = (f_1^*, f_2^*) \in E^*$ in G^*, and this mapping is also given. The term "region" emphasizes that these edge sets can be the intersection of E with a connected subset U of the plane, where the nodes u, v of an edge uv are considered as part of uv. This condition can be captured accurately by assuming that for each region $r \in \mathscr{R}$, the corresponding dual edges form a connected subgraph in G^*.

Even with the above assumptions, finding the maximum number of region-disjoint st-paths problem is $\mathcal{NP}$-hard [19]. This also holds for more restricted failure models, such as circular disk failures or line segment failures. To have a polynomial-time solvable problem, [18] added one last assumption that the obtained paths should be node-disjoint as well, or in other words, node failures should also be listed as regional failures. This implicitly also holds when circular disk failures are considered [20]. Both [18], [20] have presented polynomial-time algorithms to address the respective problems. While their worst-case complexity is reasonable, we argue they may not be suitable for practical applications. Both algorithms consist of two steps: firstly, searching for an appropriate starting path, and secondly, iteratively extending the solution with more region-disjoint paths. The second

Erika Bérczi-Kovács and Péter Gyimesi are with Department of Operations Research, ELTE Eötvös Loránd University, Budapest, Hungary. Erika Bérczi-Kovács is also with HUN-REN–ELTE Egerváry Research Group on Combinatorial Optimization, and Alfréd Rényi Institute of Mathematics. Contact them on erika.berczi-kovacs@ttk.elte.hu, and peti1234@student.elte.hu.

Balázs Vass and János Tapolcai are with Department of Telecommunication and Media Informatics, Faculty of Electrical Engineering and Informatics (VIK), Budapest University of Technology and Economics (BME) and ELKH-BME Information Systems Research Group. Balázs Vass is also affiliated to Faculty of Mathematics and Computer Science, Babeș-Bolyai University of Cluj Napoca, Romania. Contact them on {balazs.vass,tapolcai}@tmit.bme.hu.

This work was partly supported by Project nos. ANN_20 135606, FK_20 134604, FK 132524, K_18 128062, PD_21 137698, and K_23 146347, which have been implemented with support from the National Research, Development and Innovation Fund of Hungary. Supported by the ÚNKP-23-4-II-BME-345 and ÚNKP-23-5 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. This research has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 funding scheme. This work is supported by the János Bolyai Research Scholarship of the Hungarian Academy of Science.

step is relatively straightforward; the main theoretical and implementation challenges lie in the first step. The algorithms proposed in [18], [20] perform well only when more than two region disjoint paths exist, which in our experience, is rare in practice. The study in [20] offered an algorithm relying on the topological properties of the graph (e.g. the exact location of the nodes) of solving the first step, which was further generalized in [18] such that knowing the dual graph is sufficient. Nonetheless, the first step remains challenging to implement, and it is not surprising that it was omitted in the implementation provided with [18]. Instead, a simple heuristic approach was employed, leading to satisfactory performance for many practical instances of the problem.

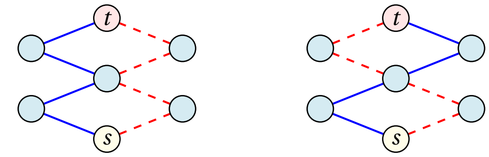
The primary contribution of this paper is to present a fundamentally different approach that bypasses the challenging first step altogether. Instead, we directly solve the problem using an auxiliary graph, the so-called regional dual graph, as depicted in Figure 3. This alternative approach offers a novel perspective and overcomes the complexities associated with the initial step of the previous algorithms. The main results of the paper are the followings:

- We generalize the problem of maximum region-disjoint st-paths, and instead of assuming disjointness of nodes, we just assume that the paths cannot cross, see Fig. 1. Our model generalizes all previous tractable ones mentioned in §VI. We give a polynomial-time algorithm for this problem. Our method is significantly different from previous approaches for similar problems, as it uses a dual technique. It is also easy to implement, since it only needs a shortest path algorithm on graphs with negative weights as a subroutine. We provide an efficient C++ implementation that can solve networks with 10 000 nodes in < 1 second.
- We prove that the optimum of the non-crossing model above gives a tight 2-additive approximation for the $\mathcal{NP}$-hard maximum region-disjoint paths problem in general (Thm. 3), which is better than the multiplicative approximation given in [19].

The paper is organized as follows: In §II, we describe the investigated problems and some necessary tools. In §III, we present our algorithm for the maximum number of region-disjoint, non-crossing st-paths problem and analyze its running time in §IV. In §V we give an additive approximation for the general case. In §VI, we summarize previous results and techniques. In §VII we provide our numerical evaluations, and finally §VIII concludes our work.

II. Problem Formulations, Main Results and Algorithm

The input of the problem is a planar graph $G = (V, E)$ with vertex set V and edge set E. Let the dual of G be denoted as G^*, which consists of vertices V^* and edges E^*. Each edge e in E corresponds to an edge in the dual graph G^*, which is denoted as e^*. An effcient way of storing such input graph is if the incident edges for every node is given clockwise order, called a rotation system [21].



(a) Non-crossing paths (b) Crossing paths

Fig. 1. Example on non-crossing and crossing paths. Edges drawn with dashed and solid lines refer to the two different paths.

We will refer to nodes of the dual graph as faces. For a subset of edges $X \subseteq E$ let X^* denote the subset of dual edges corresponding to X. For a set of edges $X \subseteq E$ let $V(X)$ denote the set of nodes incident to at least one edge in X, and let $G[X]$ denote the graph induced by X on G: $G[X] = (V(X), X)$.

With these notations, we say a subset of edges $R \subseteq E$ is a **region**, if $G^*[R^*]$ is a connected graph. In other words, the duals of the edges in a region form a connected subgraph in the dual G^* (e.g., link set $\{ta, ad, be\}$ in Fig. 3 a), depicted with dashdotted dual edges). It is easy to see that any connected disaster area in the plane can be represented by a *region*.

Further, given a set $\mathscr{R}$ of regions, two st-paths are said to be **region-disjoint**, if there is no region $R \in \mathscr{R}$ intersecting both paths (see Fig. 3). Finally, given a set $\mathscr{R}$ of regions, for a given pair of nodes $s, t \in V$, a set of regions $X \subseteq \mathscr{R}$ is a **regional st−cut** if $\cup_{R \in X} R$ is an edge set separating s and t. E.g., in Fig. 3 a), the purple-and-dashed region does form a regional st-cut with the blue-and-densely-dashed region, but does not form one with the green-and-dashdotted region. For a set of regions $\mathscr{R}$ let $\|\mathscr{R}\| := \sum_{R \in \mathscr{R}} |R|$.

A. Problem Statements and Main Results

Next, we define the two problems we are dealing with, the first one being the more general one.

Problem 1: Maximal number of region-disjoint st-paths

Input: A planar graph $G = (V, E)$, rotation system, nodes $s, t \in V$, regions $\mathscr{R} \subset 2^E$

Output: A maximum number of region-disjoint st-paths $P_1, P_2 \ldots, P_k$

Unfortunately, Problem 1 is $\mathcal{NP}$-hard [19, Thm. 6], and only a multiplicative approximation was known to its optimum [19]. In this paper, we give the first algorithmic framework that enables to efficiently compute a nearly optimal solution of the problem. We may assume that every edge is part of at least one region (otherwise, it can be contracted in the input).

Theorem 1. *Let a planar graph $G = (V, E)$, rotation system, nodes $s, t \in V$, and regions $\mathscr{R} \subset 2^E$ be given such that $\cup_{R \in \mathscr{R}} R = E$. If k^* denotes the maximum number of region-disjoint st-paths, a collection of $k^* - 2$ such paths can be found in $O\left(\log(k^*) \|\mathscr{R}\|^{\frac{3}{2}} \log(\|\mathscr{R}\|)\right)$ deterministic worst case time complexity, or with high probability in $O\left(\log(k^*) \|\mathscr{R}\| \log^9(\|\mathscr{R}\|)\right)$ expected time.*

The proof of Thm. 1 will be immediate from Thm. 2 and Thm. 3. In a nutshell, the key in our proof is that the optimum of an easily solvable special case of the above problem, when paths are non-crossing, is a lower bound on the maximum number of paths. More formally, we say two st-paths in G are **non-crossing** if after contracting their common edges there is no node where the edges of the paths are alternating (Fig. 1); k paths are non-crossing, if they are pairwise non-crossing.

Problem 2: Maximum number of region-disjoint non-crossing st-paths

---

Input: A planar graph $G=(V,E)$, rotation system, nodes $s,t\in V$, regions $\mathscr{R}\subset 2^E$

Output: A maximum number of region-disjoint, non-crossing st-paths $P_1,P_2\dots,P_k$

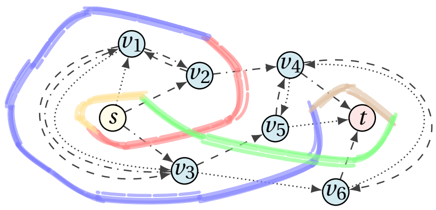


Fig. 2. Example for the tightness of Thm. 3. If regions are only the five colored lines, then $MF_{nc}=1$, $MF=MC=3$. Paths of the crossing max-flow are depicted by the dotted, dashed, and dashdotted arcs, respectively. By adding all node failures (except from s and t), MF becomes 1.

As presented throughout this paper, Problem 2 is efficiently solvable using a simply implementable algorithmic framework.

Theorem 2. *Given a planar graph $G=(V,E)$, rotation system, nodes $s,t\in V$, and regions $\mathscr{R}\subset 2^E$ such that $\cup_{R\in\mathscr{R}}R=E$, a maximum number of k^* non-crossing region-disjoint st-paths can be found in $O\left(\log(k^*)\|\mathscr{R}\|^{\frac{3}{2}}\log(\|\mathscr{R}\|)\right)$ deterministic worst case time complexity, or with high probability in $O\left(\log(k^*)\|\mathscr{R}\|\log^9(\|\mathscr{R}\|)\right)$ expected time.*

The main parts of our algorithmic framework are described in §II-C. Its details and the proof of correctness are presented in §III. Finally, the runtime complexity is analyzed in §IV.

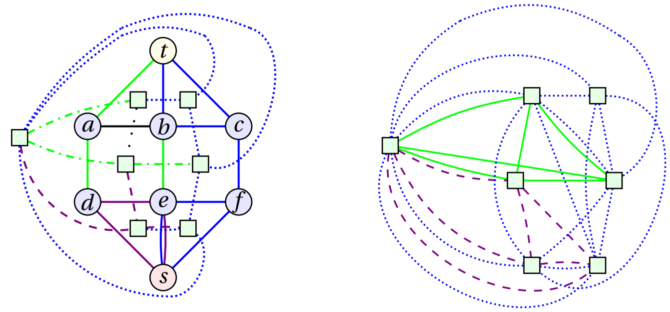
For a maximal number of region-disjoint st-paths problem the corresponding min-cut problem can be solved in polynomial time [19]. Next, we present a theorem comparing these optimum values.

Theorem 3. *Let a maximal number of region-disjoint st-paths problem instance and its corresponding minimum regional st-cut problem be given, and let MF and MC denote their optimal values, respectively. Moreover, let MF_{nc} denote the optimal value of the non-crossing version of the problem. Then $MC-2\le MF_{nc}\le MF\le MC$.*

The proof of the theorem can be found in §V. The example on Fig. 2 show that the theorem is tight in the sense that both $MF-MF_{nc}$ and $MC-MF$ can be 2 (and it is easy to give an example where $MF_{nc}=MC$).

B. Regional dual graph

The algorithm we will describe for Problem 2 works on an auxiliary directed graph, which we will call **regional dual** of G, and denote by $D^*_{\mathscr{R}}$. Nodes of $D^*_{\mathscr{R}}$ are faces in V^*, and the arcs are derived from $\mathscr{R}$: for every region R we add a complete directed graph on $V(R^*)$ to $A^*_{\mathscr{R}}$. Note that on Fig. 3b, we draw an undirected version of $D^*_{\mathscr{R}}$, omitting the arrowheads on the arcs, and for each arc pair u^*v^* -v^*u^* drawing only a single edge u^*v^*. Every arc u^*v^* belongs to a region R and we say that an oriented path in $G^*[R^*]$ is **representing** arc $u^*v^*\in A^*_{\mathscr{R}}$ if the path is completely in R^*. Note that the regional dual is not necessarily planar and there can be parallel arcs.



(a) Graph G with its dual G^*, edge colors/styles refer to regions $R\in\mathscr{R}$

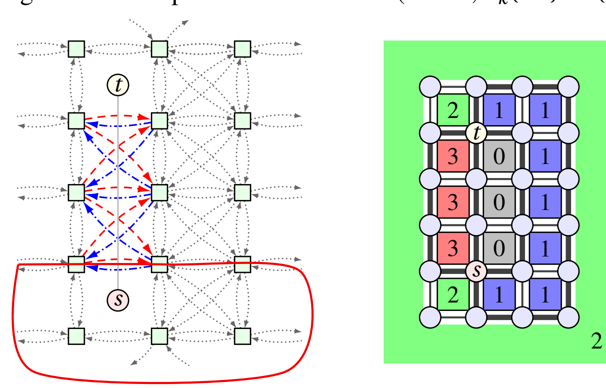
(b) Regional dual $D^*_{\mathscr{R}}$corresponding to graph G and set of regions $\mathscr{R}$

Fig. 3. Graph G, its dual G^* and regional dual $D^*_{\mathscr{R}}$, respectively. Edge colors refer to regions in $\mathscr{R}$. Path s,d,a,t is region disjoint with path s,f,c,b,t, but it is not with s,f,e,b,t, since links ad and eb are part of the same region (depicted with dashdotted dual edges).

C. Overview of the algorithm

The main idea of the algorithm is that the existence of k region-disjoint non-crossing st-paths is equivalent to the non-existence of a negative cycle in $D^*_{\mathscr{R}}$ with respect to properly chosen arc weights c_k (i.e. c_k is **conservative**). Oversimplified, the vague description of c_k is the following. First, we fix a directed st-path P. Then if an arc a of $D^*_{\mathscr{R}}$ does not cross P, we set $c_k(a)=1$, if it crosses P from left to right $c_k(a)$ will be $1-k$, and finally, in case of a right-to-left crossing, $c_k(a)$ is set to $1+k$. A formal definition of weights c_k will be provided in §III-B.

We will see in the next section that if c_k is conservative, we get a feasible potential $\pi:V^*\to\mathbb{Z}$ (that is, $c_k(uv)+\pi(u)-$



(a) Regional dual $D^*_{\mathscr{R}}$. Cost c_k of black-and-dotted, red-and-dashed, and blue-and-dashdotted arcs is 1, $1-k$, and $1+k$, resp. For $c_{k\ge5}$, the red closed arc encodes a negative cycle.

(b) Topology G, with the regions being exactly the nodes $v\in V\setminus\{s,t\}$. Numbers on the faces form a feasible potential for $c_{k=4}$.

Fig. 4. Example topology G being a 4×6 node grid lattice graph, with the regions being exactly the nodes $v\in V\setminus\{s,t\}$. The st-path P in G is the shortest path, through the three vertical edges.

Algorithm 1: Algorithm for finding the maximum number of region-disjoint, non-crossing st-paths

Input: Planar graph $G=(V,E)$, rotation system, nodes $s,t\in V$, regions $\mathscr{R}\subset 2^E$
Output: Region-disjoint, non-crossing st-paths $P_1,P_2\dots,P_k$ and witness for non-existence of $k+1$ paths.
1 binary search on k (check existence of k paths with Alg. 2)
2 $\implies k^*$ optimum
3 $c_{k^*}\implies\pi\implies$ paths $P_1,\dots,P_k$
`// k region disjoint non-crossing paths`
4 $c_{k^*+1}\implies C^*$ `// Witness of non-existence`
5 **return** $P_1,\dots,P_k$ *and* C^*

$\pi(v)\ge 0$ for all $uv\in A^*_{\mathscr{R}}$), then create a corresponding arc set F which describes the required paths $P_1,\dots,P_k$. Intuitively, the boundaries between the $\bmod k$ classes of faces of G according to π determine k non-crossing $\mathscr{R}$-disjoint paths (as depicted on Fig. 4b).

If c_k is not conservative, we consider a negative cycle C' in $D^*_{\mathscr{R}}$ (as the red closed arc shows on Fig. 4a), which gives a witness for the non-existence of k paths, and then move on to the next k.

The maximum k for which weighting c_k is conservative (and a number of k non-crossing st-paths exist) can be found via binary search (see Alg. 1).

III. Finding k non-crossing region-disjoint paths

The existence of k region-disjoint non-crossing st-paths can be reduced to checking the conservativity of weightings in two steps. First, we show that some dual walks in G^* with some special properties are witnesses for the non-existence of k required paths (see next subsection, Lemma 4).

Second, with a proper weighting on $D^*_{\mathscr{R}}$ (to be introduced in §III-B), these special dual walks in G^* can be reformulated as negative cycles in $D^*_{\mathscr{R}}$.

A. Witness for the non-existence of region-disjoint, non-crossing st-paths

In order to give a witness we need to define some notions on the dual graph (also used in [18]).

First, we introduce the *winding number*. Let P be an st-path and C^* a closed oriented walk in G^*. Let $w_{lr}(C^*)$ and $w_{rl}(C^*)$ denote the number of times C^* intersects P from left to right and from right to left, respectively. Then the **winding number** of the walk is $w(C^*)=|w_{lr}(C^*)-w_{rl}(C^*)|$. Note that $w(C^*)$ does not depend on the choice of P.

In some proofs we need a similar notion for dual paths as follows. Let P be an st-path and Q^* an orientation of a path in G^*. Let $w^P(Q^*)$ denote the number of times path Q^* intersects path P from left to right minus the number of times it intersects right to left.

Let C^* be a closed walk in G^*. Partition $C_1^*,C_2^*,\dots,C_l^*$ is a **region-cover** of C^* with l regions if each C_i^* is a subpath of C^* and each C_i^* is a subset of an R_i^* for a region $R_i\in\mathscr{R}$. The **region-length** of C^*, denoted by $l(C^*)$ is the minimum l such that there is a region-cover of C^* with l regions. In [18] it was shown that $\lfloor l(C^*)/w(C^*)\rfloor$ is an upper bound for the maximum number of node- and region-disjoint paths problem (if the optimum value is at least 2). Here we show that the same argument carries over to non-crossing paths.

Lemma 4. *Let a maximum number of region-disjoint non-crossing paths problem instance be given with optimal value $k\ge 2$, and let C^* be a closed walk in the dual graph with $w(C^*)>0$. Then $\left\lfloor \frac{l(C^*)}{w(C^*)}\right\rfloor\ge k$.*

Proof: (See Fig. 2 with C^* of red-blue-brown-green-yellow regions: $\frac{l(C^*)}{w(C^*)}=\frac{5}{3}\ge MF_{nc}$.) Let $P_1,\dots,P_k$ be non-crossing, region-disjoint st-paths, and let $C_1^*,\dots,C_l^*$ be a region-cover of C^* with $l=l(C^*)$. We may assume that $w_{lr}(C^*)>w_{rl}(C^*)$. Since each st-path is intersected by C^* at least $w(C^*)$ times, every path P_i also intersects C^* at least $w(C^*)$ times.

Claim 5. *If $k\ge 2$, then $|w^{P_j}(C_i^*)|\le 1$ for $1\le i\le l$ and $1\le j\le k$.*

Proof: Assume indirectly that $w^{P_j}(C_i^*)\ge 2$. Then for any planar embedding of G edges $C_i^*\cup P_j$ would contain a curve in the plane separating s and t, contradicting the existence of another non-crossing path region-disjoint from P_j. ■

From the claim we get that if $k\ge 2$, each path P_i intersects at least $w(C^*)$ *distinct* subpaths C_j^*, which gives $kw(C^*)\le l(C^*)$, that is $\lfloor l(C^*)/w(C^*)\rfloor\ge k$ indeed. ■

In §V in Thm. 19 we will show that this bound is sharp.

B. Reduction to conservative weightings

In this subsection we show that with properly chosen arc weights c_k the existence of k region-disjoint non-crossing st-paths is equivalent to the conservativity of c_k on $D^*_{\mathscr{R}}$. We may assume that there is no region separating s and t (otherwise the problem is trivial). In order to define weights on the arcs of $D^*_{\mathscr{R}}$, let P be an arbitrary fixed st-path in G. For every arc $u^*v^*\in A^*_{\mathscr{R}}$ we consider a representing path $P_{u^*v^*}$ in the dual region $G^*[R^*]$ with the orientation from u^* to v^*. Let $w^P(u^*v^*):=w^P(P_{u^*v^*})$. From the following claim we get that this value is well-defined.

Claim 6. *Let u^*v^* be an arc in the regional dual graph, belonging to region R, and Q_1^* and Q_2^* two paths in R^* from u^* to v^*. If R does not separate s and t, then $w^P(Q_1^*)=w^P(Q_2^*)$ for any st-path P.*

Proof: Assume indirectly that $w^P(Q_1^*)\ne w^P(Q_2^*)$. Then the concatenation of Q_1^* and the reverse of Q_2^* would give a closed dual walk C^* with non-zero $w^P(C^*)$. Such walks contain an st-cut so region R would be separating s and t, contradicting the assumption. ■

For a positive integer k, cost function c_k is the following: $c_k(u^*v^*)=1-w^P(u^*v^*)\cdot k$.

The key of our algorithm is the following theorem.

Theorem 7. *Cost function c_k is conservative on $D^*_{\mathscr{R}}$ if and only if there are k region-disjoint, non-crossing st-paths in G.*

Proof: We will prove the theorem via two lemmas corresponding to the 'if' and 'only is' parts of the equivalence in the theorem. First we show that a negative cycle with respect to c_k is a witness for the non-existence of the required paths.

Lemma 8. *If c_k is not conservative, then there are no k region-disjoint, non-crossing st-paths in G.*

Proof: We will find a closed walk C^* in the dual of G with $\frac{l(C^*)}{w(C^*)} < k$, which proves the lemma by Lemma 4. If c_k is not conservative, then there is a negative cost cycle $C' = f_1, f_2, \ldots, f_l, f_1$ in $D^*_{\mathscr{R}}$. Each arc $f_i f_{i+1}$ has a representing path Q_i from f_i to f_{i+1} in G^* (where $f_{l+1} = f_1$). Then $Q_1, Q_2, \ldots, Q_l$ give a closed dual walk C^*. Since subpaths Q_i form a regional cover of C^*, we get that $l \geq l(C^*)$. We have $0 > c_k(C') = l - k \cdot \sum_{i=1}^{l} w^P(Q_i) = l - k \cdot w^P(C^*) \geq l(C^*) - k \cdot w(C^*)$, which gives a closed dual walk with $\frac{l(C^*)}{w(C^*)} < k$ indeed. ∎

Next we turn to the second part and show that if c_k is conservative, then the required paths exist.

Lemma 9. *If c_k is conservative on $D^*_{\mathscr{R}}$, then there are k region-disjoint non-crossing st-paths in G.*

Proof: Let $\pi : V^* \to \mathscr{R}$ be a feasible potential for c_k, that is, $\pi(v^*) - \pi(u^*) \leq c_k(u^* v^*)$ for every arc in $D^*_{\mathscr{R}}$ (such a potential exists from the classic characterization of conservative weightings). The idea of the proof, in a nutshell, is to consider those edges of G where π changes by 1 (or by $k \pm 1$ on P). These edges turn out to have a nice structure and give the required paths (Fig. 4). For each node $x \neq \{s,t\}$ we define an oriented subset F_x of edges incident to x.

First, we define F_x for nodes not on P. We assumed every edge is part of at least one region, so π values on faces around x are 'smooth' in the sense that neighboring faces differ by at most 1. If for neighboring faces u and v we have $\pi(v) - \pi(u) = 1$, then we consider their common edge xy and add to F_x its anti-clockwise orientation with respect to uv (see Fig. 5a).

Second let $x \neq \{s,t\}$ be a node on P. In order to get a 'smooth' potential around x, we translate π by k on some faces neighboring x the following way. Let e and f be the edges on P preceding and following x, respectively, and let l_e, l_f and r_e, r_f denote the faces on the left and right of e and f according to the orientation on path P from s to t. We denote by L the set of faces clockwise to l_e until l_f around x, and decrease π by k on every face in L. The resulting potential around x is denoted by π_x. Since π is a feasible potential and $c_k(l_e r_e) = -k+1$ and $c_k(r_e l_e) = k+1$, we get that $\pi(l_e) - k - 1 \leq \pi(r_e) \leq \pi(l_e) - k + 1$ (and similarly for f). So after the translation the π_x values of neighbouring faces differ by at most 1 around x, and we can create F_x using π_x the same way as we did for nodes not on P.

Let $F := \cup_{x \in V \setminus \{s,t\}} F_x$. Note that this definition of F is consistent in the sense that arc $uv \in F_v$ if and only if $uv \in F_u$ ($u, v \neq s, t$). We call an arc $xy \in F$ an **$(i, i+1)$-type arc** if $\pi(u) \equiv i \mod k$, where u is the face on the left of xy in G. (Thus $\pi(v) \equiv i+1 \mod k$ for face v on the right of xy in G.)

Claim 10. *Graph spanned by arcs F is Eulerian on $V \setminus \{s,t\}$ in the directed sense. Moreover, at every node $v \in V \setminus \{s,t\}$ the incoming and outgoing arcs in F_v can be partitioned into pairs such that: 1) pairs have the same type, and 2) pairs are non-crossing.*

Proof: Let us consider the ordered set N of neighbouring faces of v in G_k in a clockwise order: $N = u_1, u_2, \ldots, u_l, u_{l+1}$, where $u_{l+1} = u_1$. Since π (or π_v if $v \in P$) on neighbouring faces can differ by at most 1, the number of indices i such that $\pi(u_i) + 1 = \pi(u_{i+1})$ equals the number of indices i for which $\pi(u_i) - 1 = \pi(u_{i+1})$ $(1 \leq i \leq l)$, which shows that graph spanned by F is Eulerian.

Now we define the arc pairs for a node v. Assume $v \notin P$ (for a node v on P the same argument holds with π_v). If π is constant on neighboring faces, then there are no arcs in F incident to v. Otherwise, let Π denote the maximum value of π on faces incident to v, and let $u_i, \ldots, u_{i+j}$ be a maximal subset of consecutive faces of this value: $\Pi = \pi(u_i) = \ldots = \pi(u_{i+j})$ and $\Pi - 1 = \pi(u_{i-1}) = \pi(u_{i+j+1})$, where $u_x = u_y$ if $x \equiv y \mod l$. Then $\pi(u_{i-1}) = \pi(u_i) - 1$ and $\pi(u_{i+j+1}) = \pi(u_{i+j}) - 1$, so they have an incoming and an outgoing corresponding arc in F with the same type. We pair them at v, and by contracting faces $u_{i-1}, u_i, \ldots, u_{i+j}, u_{i+j+1}$ in N we can continue this process until all pairs are formed (see Fig. 5b). ∎

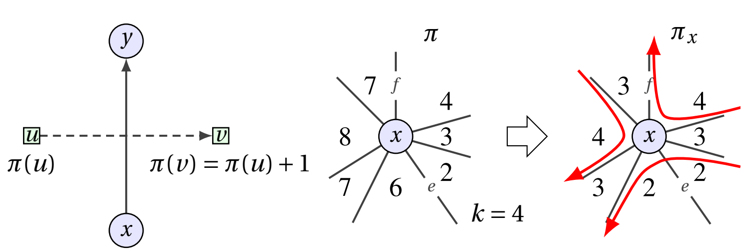
Claim 11. *There are k non-crossing st-paths $P_1, \ldots, P_k$ in F formed by the pairing and each path has a unique type.*

Proof: Pairs created in Claim 10 partition F into non-crossing directed cycles and non-crossing st-paths such that arcs within a cycle or path have the same type. Let $\rho_F(v)$ and $\delta_F(v)$ denote the in- and out-degree of a node v in F. Nodes s and t both have one incident edge on P, where π changes by k or $k \pm 1$, so $\delta_F(s) - \rho_F(s) = k$, and similarly $\rho_F(t) - \delta_F(t) = k$. Hence there are k non-crossing st-paths $P_1, \ldots, P_k$ created, and each path has a unique type. ∎

Claim 12. *Let $R \in \mathscr{R}$ be a region. Then arcs in $F \cap R$ have the same type modulo k.*

Proof: First consider the case when $R \cap P = \emptyset$. Since there is an arc of weight 1 in $D^*_{\mathscr{R}}$ connecting any two nodes in $V(R^*)$, it is easy to see that π values on R can differ by at most one and so there can be at most one type of arc in F. Second assume $R \cap P \neq \emptyset$. Then $R \cap P$ can be partitioned into node-disjoint sub-paths of $P: R_1, \ldots, R_l$. Each sub-path R_i forms a cut in $G^*[R^*]$, and these cuts are non-crossing, so these cuts partition faces in $V(R^*)$ into ordered sets $U_1, \ldots, U_{l+1}$ such that face-sets U_i and U_{i+1} have common border R_i (for $i = 1..l$), see Fig. 6. We reduce this case to the first by translating π on each U_i by a constant to get a 'smooth' potential. Let $\Delta_i := w^P(Q_i^*)$, where Q_i^* is a path in R^* from a face in U_1 to a face in U_i. We add $\Delta_i k$ to π on each set U_i. Then the resulting potential π' differs by at most one on $V(R^*)$. Moreover, for every node $x \in V(R) \setminus \{s,t\}$ potential π_x is a translation of π' by a constant on faces in $V(R^*)$ neighbouring x. Thus the edges in R with different π'-valued neighbouring faces are exactly $R \cap F$. Since π' differs by at most one on $V(R^*)$, we can apply the same argument as in the first case. ∎

In Claim 11 we showed that each type class modulo k



(a) Creating F (b) Pairing edges

Fig. 5. Illustrations for Lemma 9

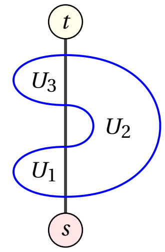


Fig. 6. Partition of $V(R^*)$ with respect to P

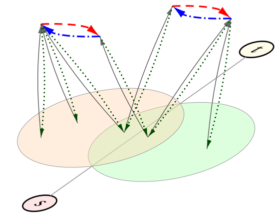


Fig. 7. Illustration for D_0. There are two circular regions. Solid, and dotted arcs cost 1, and 0, while red-and-dashed, and blue-and-dashotted arcs cost $-k$ and k, respectively.

belongs to a path P_i, we may assume that path P_i has type $(i, i+1)$. From Claim 12 we get that a region can intersect at most one type of arcs in F, so it can intersect at most one path P_i, which proves this lemma. ∎

From Lemmas 8, and 9, we get the proof of Thm. 7. ∎

IV. Running time analysis of the algorithm

In this section we give a detailed running time analysis of Alg. 1. First observe that the running time of building up k paths from a feasible potential on $D^*_{\mathscr{R}}$ is negligible: if for a certain k weighting c_k is conservative on $D^*_{\mathscr{R}}$ and a feasible potential π is given, arc set F can be created in $O(|V|)$ time. Then both the pairings of arcs in F around all nodes in $V \setminus \{s,t\}$ and the creation of k required paths can be done in $O(|V|)$ time also. Thus, the bottleneck of the algorithm is the decision of the conservativity of c_k on $D^*_{\mathscr{R}}$ for a given k. In the following subsection we show how the regional dual graph $D^*_{\mathscr{R}}$ can be substituted by another directed graph to get a better running time. Then in §IV-B we analyze some subroutine options for the decision of conservativity of c_k on $D^*_{\mathscr{R}}$.

A. A smaller representation of $D^*_{\mathscr{R}}$

We have seen in Thm. 7 that directed graph $D^*_{\mathscr{R}}$ and weighting c_k capture enough information to decide the existence of k regional-SRLG-disjoint st-paths in G. The number of arcs $|A^*_{\mathscr{R}}| = O(\sum_{R\in\mathscr{R}} |R|^2)$. In this subsection we show that the set of arcs can be substituted by a collection of subgraphs with a total number of $O(\sum_{R\in\mathscr{R}} |R|)$ arcs, giving a better running time (see Alg. 2).

We build a new auxiliary graph D_0 and define arc weights c^0_k such that c_k is conservative on $D^*_{\mathscr{R}}$ if and only if c^0_k is conservative on D_0. We start from the empty graph on V^*, and for each region $R \in \mathscr{R}$ instead of the complete directed graph on $V(R^*)$ we add the following subgraph to D_0: we consider again the partition $U_1,\dots,U_l$ of $V(R^*)$ as in Claim 12 and for each U_i we add a node u^R_i to V^* and arcs $u^R_i u^R_{i+1}$ and $u^R_{i+1} u^R_i$ $(1 \le i \le l)$. If set U_i is on the left (or right) of separating subpath R_i, we set $c^0_k(u^R_i u^R_{i+1}) := -k$ and $c^0_k(u^R_{i+1} u^R_i) := k$ (or $c^0_k(u^R_i u^R_{i+1}) := k$ and $c^0_k(u^R_{i+1} u^R_i) := -k$). For every set U_i and every node $v \in U_i$ we add arcs $v u^R_i$ and $u_i v$ with weights 1 and 0, respectively (see Fig. 7 for illustration).

Claim 13. *Weighting c^0_k is conservative on D_0 if and only if c_k is conservative on $D^*_{\mathscr{R}}$. The number of arcs and nodes in D_0 are both $O(\|\mathscr{R}\|)$.*

Proof: It is easy to check that for every region R and for each arc $u^* v^* \in A^*_{\mathscr{R}}$ belonging to R there is a corresponding path in the subgraph created for R with the same weight. Moreover, given a feasible potential π^0 on D_0, its projection onto V^* gives a feasible potential on $D^*_{\mathscr{R}}$ and similarly a negative cycle C^0 in D_0 corresponds to a negative cycle C' in $D^*_{\mathscr{R}}$. For a region R $O(|R|)$ nodes and arcs are created. ∎

B. Algorithm options for finding a feasible potential

In this subsection, we investigate some algorithms that are suitable for computing the feasible potential π, or proving that no such potential exists. Particularly, we will take advantage of the following fact.

Proposition 14. *Weighting c_k on $D^*_{\mathscr{R}} = (V^*, A^*_{\mathscr{R}})$ is conservative if and only if for any fixed node v^* of $D^*_{\mathscr{R}} = (V^*, A^*_{\mathscr{R}})$ by setting $\pi(w^*) := d_{c_k}(v^*, w^*)$ for each $w^* \in V^*$, we get a feasible potential π.*

In line with this proposition, in all the following cases, we check the conservativity of the weighting of $D^*_{\mathscr{R}} = (V^*, A^*_{\mathscr{R}})$, but instead of $D^*_{\mathscr{R}}$ we will use auxiliary directed graph D_0 described in §IV-A. We compute a feasible potential by using the distances of the nodes of D_0 from any fixed node, the only difference will be the exact algorithm that is plugged in to provide these information. All the subroutines we propose below either calculate the distances from a given node if the weighting is conservative or return a negative cycle if it is not.

Algorithm 2: Algorithm for checking the existence of $k \ge 2$ region-disjoint, non-crossing st-paths

Input: Planar graph $G = (V, E)$, rotation system, nodes $s, t \in V$, regions $\mathscr{R} \subset 2^E$, $k \ge 2$: # of paths
Output: Region-disjoint, non-crossing st-paths $P_1, P_2 \dots, P_k$ or dual walk C^* witness of non-existence.

fix st-path P
create D_0; create c^0_k
check c^0_k conservative: $\Longrightarrow C^0$ negative cycle or π^0 feasible potential
if *c^0_k conservative* **then**
  $\pi^0 \Longrightarrow \pi \Longrightarrow F \Longrightarrow P_1,\dots,P_k$
  return $P_1,\dots,P_k$ `// k region disjoint non-crossing paths`
else
  $C^0 \Longrightarrow C^*$ *in G*
  return C^* `// Witness of non-existence`

1) Bellman-Ford and SPFA: Perhaps the most well-known algorithm for computing the shortest path lengths from a single source vertex to all of the other vertices in a weighted digraph is the Bellman-Ford (BF) algorithm that has a complexity of $O(nm)$ on a graph with n nodes and m arcs [22]. In our case, for D_0, this means a complexity of $O(\|\mathscr{R}\|^2)$ by Claim 13. For the simulations, we have implemented a heuristic speedup, the so-called Shortest Path Faster Algorithm (SPFA) [23], that has a same worst-case time complexity as the BF, but there is anecdotic evidence suggesting an average runtime somewhere around being linear in the number of network links (for D_0, this would mean a typical runtime in the order of $\|\mathscr{R}\|$). Our simulation results (§VII) are in line with this expected performance. While the SPFA, in the worst case, is not faster than the classic BF, the next algorithm reduces this complexity.

2) A worst-case faster algorithm: Having a graph with n nodes and m arcs, and integer weights on the arcs of absolute value at most W, [24] claims the following.

Theorem 15 (Theorem 2.2. of [24])**.** *The single-source shortest path problem on a directed graph with arbitrary integral arc lengths can be solved in $O(\sqrt{n}\cdot m\log(nW))$ time and $O(m)$ space.*

Applied to our problem with D_0, this means:

Corollary 16. *Given a maximum number of region-disjoint non-crossing st-paths problem instance and integer $k \geq 2$, the existence of k required st-paths can be decided in time $O(\|\mathscr{R}\|^{\frac{3}{2}}\log(\|\mathscr{R}\|))$.*

Proof: The proof is immediate from Thm. 15, Proposition 14, Claim 13 and that the maximal absolute value of a weight on the links is $O(|V|^2)$. ■

3) A near-linear time randomized algorithm: The following result grants a near-linear runtime for our framework.

Theorem 17 (Theorem 1.1. of [25])**.** *There exists a randomized (Las Vegas) algorithm that takes $O(m\log^8(n)\log(W))$ time with high probability (and in expectation) for an m-edge input graph G_{in} and source s_{in}. It either returns a shortest path tree from s_{in} or returns a negative-weight cycle.*

By the same observations as in Cor. 16, we get the following.

Corollary 18. *Given a maximum number of region-disjoint non-crossing st-paths problem instance and integer $k \geq 2$, the existence of k required st-paths can be decided in time $O(\|\mathscr{R}\|\log^9(\|\mathscr{R}\|))$ with high probability (and in expectation).*

The running time complexities in Thm. 2 follow from Cor. 16, Cor. 18 and the observation that the optimum k^* is found via binary search, giving a multiplication of $\log(k^*)$ to the above runtimes.

Comparison with previous running time: The most efficient polynomial-time algorithm was given for the node- and region-disjoint special case of the problem [18]. The running time of their solution is $O(|V|^2\mu(\log(k)+\rho\log(d)))$, where d denotes the maximum diameter of a region in G^*, whereas μ and ρ are (typically small) parameters denoting the maximum number of regions an edge can be part of and the maximum size of a region, respectively. Note that $\|\mathscr{R}\| = O(|V|\mu)$, so our deterministic algorithm has a running time of $O(|V|^{\frac{3}{2}}\mu^{\frac{3}{2}}\log(|V|\mu))$, which is indeed faster than the one in [18].

V. A min-max theorem for non-crossing paths and an additive approximation for the general case

In this section, we mention some theoretical consequences of the correctness of the algorithm. First, we derive a min-max theorem for Problem 2.

Theorem 19. *Let k^* denote the optimum value of a maximum number of region-disjoint non-crossing st-paths problem. If $k^* \geq 2$, then it equals the minimum of $\lfloor l(C^*)/w(C^*)\rfloor$, where C^* is a closed walk in G^* with $w(C^*) > 0$. For $k^* = 1$ we can find a closed walk C^* with $\lfloor l(C^*)/w(C^*)\rfloor < 2$.*

Proof: The optimum k^* equals the maximum k such that c_k is conservative on $D^*_{\mathscr{R}}$. If $k^* \geq 2$, from Thm. 2 we get that there are k region-disjoint non-crossing st-paths and since c_{k+1} is not conservative, there is a negative cycle in $D^*_{\mathscr{R}}$ with respect to c_{k+1}, which gives a closed dual walk C^* in G^* with $\lfloor l(C^*)/w(C^*)\rfloor < k+1$. If $k^* = 1$, then c_2 is not conservative, and there is a dual walk C^* with $\lfloor \frac{l(C^*)}{w(C^*)}\rfloor < 2$. ■

We will apply the min-max theorem above to prove Thm. 3.

A. Additive approximation for Problem 1

Proof of Thm. 3: The upper bound $MF \leq MC$ is trivial. For the lower bound let MF_{nc} denote the optimal value of the corresponding path packing problem with the non-crossing constraint and let C^* be a closed walk as described in Thm. 19.

Claim 20. *There exists a regional cut $X \subseteq \mathscr{R}$ such that $|X| \leq \lfloor l(C^*)/w(C^*)\rfloor + 2$.* ■

The proof is analogous to that of a similar result for node- and region-disjoint st-paths in [18, Thm. 7]. Clearly, $MF_{nc} \leq MF$ and from Claim 20 $MC \leq \lfloor l(C^*)/w(C^*)\rfloor + 2 \leq MF_{nc} + 2$. By merging the inequalities, we get the lower bound on MF: $MC - 2 \leq MF_{nc} \leq MF \leq MC$. ■

VI. Previous work

The maximum number of region-disjoint paths problem and some of its special cases have been studied by numerous papers. The results range from $\mathcal{NP}$-hardness, heuristics, and general (M)ILP formulations to polynomial time solutions to some special cases. The related papers can be divided into two branches. One branch concerns the theoretical preludes of region-disjoint routing problems. The other branch is focused mainly on computing SRLG-disjoint paths in communication networks. In the following, we summarize the main results of these papers.

A. Theoretical preludes

Maximum number of (crossing) region-disjoint paths: Seminal work [19] investigates scenarios when a planar graph is given with a fixed embedding, and each edge set in $\mathscr{R}$ is the intersection of the graph with a subset of the plane that is homeomorphic to an open disc (called as 'holes' in [19]). It gives a high-degree polynomial-time algorithm for the minimum regional st-cut, even for the directed and weighted problem version. As for the corresponding maximum number of $\mathscr{R}$-disjoint st-paths problem, it shows to be $\mathscr{NP}$-hard. Finally, [19] also proves that the minimum number of separating regions is at most twice the maximum number of $\mathscr{R}$-disjoint st-paths plus two.

d-separate paths: [26] considers generalizations of disjoint paths problems, where paths are required to be 'far' from each other. Here distance is measured by the number of edges in a shortest path connecting the paths (apart from their endpoints). If this length is at least $d+1$, the paths are called d-separate. Note that by choosing for each node or for each edge the set of edges at a distance at most d (neighboring edges are at a distance 0), we can define undirected d-separated paths as a special case of region-disjointness, since such edge sets form a connected subgraph in the dual graph. [26] gives a min-max formula for the existence of k d-separated st-directed paths in planar graphs. Their dual problem is not purely combinatorial because it minimizes a value on a set of certain appropriate curves in the plane.

B. Survivable routing in communication networks

Papers [1], [27] consider a network protection problem when geographic failures modeled as circular disks may occur. In their model, a region is a set of edges that can be the intersection of the planar graph with a circular disk of a given radius (apart from a protective zone around s and t). They give a polynomial-time algorithm for the minimum regional st-cut version of the problem and conjecture that the maximum number of region-disjoint paths and the size of the minimum cut differ by at most one in this case.

Later, [20], [28] proved this conjecture. These papers adapted the method of [26], [29] for circular disk failures, and gave a polynomial-time algorithm for the problem, as well as a min-max formula. They also used a proper curve in the plane for the characterization of the maximum number of region-disjoint st-paths.

The problem was generalized from circular disk failures to regions in [18], [30], so only assume that each edge set in R is connected in the dual of the graph and all node failures are part of an SRLG. They do not use the embedding of the graph in the plane, only the clockwise order of incident edges for every node (a rotation system) is part of the input. They give a polynomial-time algorithm for this problem by generalizing the method of [29] and [20] for planar rotation systems. Also, they prove that the size of a minimum cut and the maximum number of region-disjoint st-paths differ by at most two in this general model, and this inequality is sharp. Their min-max formula uses closed walks in the dual graph instead of curves in the plane.

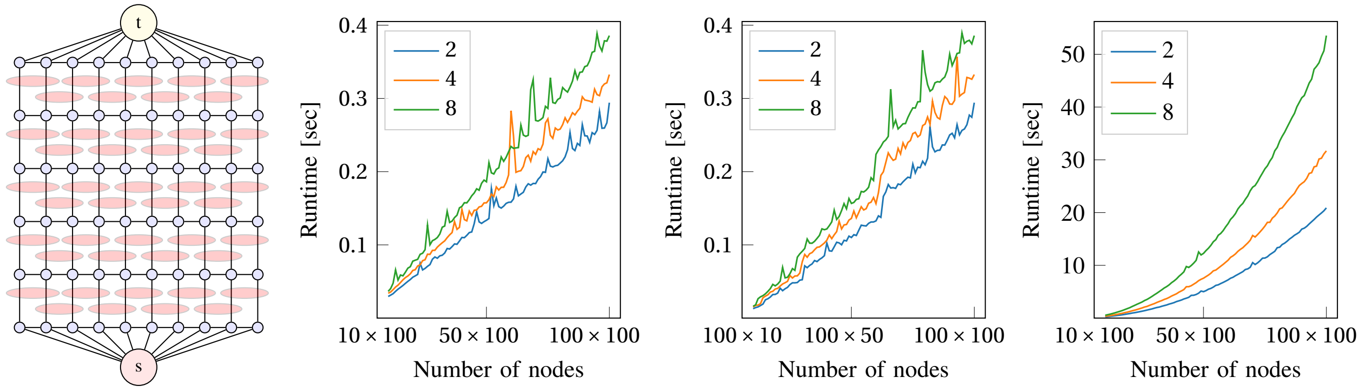
Further works in the field of region-disjoint routing: The first paper to prove the $\mathscr{NP}$-completeness of finding two SRLG-disjoint (region-disjoint) paths was [31]. The result was achieved by showing the $\mathscr{NP}$-hardness of the so-called fiber-span-disjoint paths problem, which is a special case of the SRLG-disjoint paths problem. As it turns out, SRLG-disjoint routing is $\mathscr{NP}$-complete even if the links of each SRLG S are incident to a single node v_S [32]–[34]. Some polynomially solvable subcases of this problem are also presented in [32], [33]. An ILP solution for the SRLG-disjoint routing problem is given in [35]. To solve, or at least approximate the weighted version of the SRLG-disjoint paths problem some papers use ILP (integer linear program) or MILP (mixed ILP) formulations [36]–[38]. Based on a probabilistic SRLG model, [39] aims to find diverse routes with minimum joint failure probability via an integer non-linear program (INLP). Heuristics were also investigated [40], [41], unfortunately, with issues like possibly non-polynomial runtime or possibly arising forwarding loops when the disaster strikes.

VII. Numerical Evalution

In this section, numerical results are presented to demonstrate the effectiveness of our algorithm on different real physical networks. The algorithm was developed using C++, and to facilitate reproducibility, we have uploaded our implementation of the algorithm and the input data to a publicly accessible repository (see §VIII). To measure the runtime performance, we conducted the experiments on a standard laptop equipped with a 2.8 GHz CPU and 8 GB of RAM. We employed the SPFA algorithm to calculate the potential, as it is the simplest approach and still demonstrated a satisfactory level of performance. We investigate two aspects: first, whether the runtime of the algorithm is in line with the theoretical bounds; second, we compare the algorithm with the previous state-of-the-art method in terms of runtime and path length.

A. Runtime analysis

To measure the algorithm's runtime increase concerning the input size, we have generated numerous grid graphs, as shown in Fig. 8a. Such a series of grid graphs contain various numbers of rows and columns and feature uniform-sized regions composed of 2, 4, or 8 adjacent vertical edges. In Fig. 8b, we show the results for the series of graphs with a gradually increasing number of rows in a 100×10 grid graph until it reached a 100×100 grid graph, resulting in a total of 273 problem instances. Thus the total size of regions $\|\mathscr{R}\|$ is a linear function of the number of nodes, and we expect a nearly linear running time. In this experiment, the number of paths remains the same, but their length increases as the graph has more rows. The number of region disjoint paths depends on the size of the regions: 50 paths for size 2, 25 for size 4, and 12 for size 8. The average runtime exhibits linear growth with the size of the graph, as expected. For the largest graph with 10002 nodes and 20000 edges, the runtime was 0.4 seconds. We repeated the aforementioned process, but this time, we generated a sequence of graphs with gradually increasing the number of columns of a 10×100 grid graph, see Fig. 8c. As a consequence, the number of region-disjoint paths increased with the network size. This resulted in slightly



(a) A 6×10 grid graph with regions drawn in red, each consisting of a size of 2. (b) Grid graphs with 100 columns and varying numbers of rows of the Proposed algorithm. Region sizes: 2, 4, or 8. (c) Grid graphs with 100 rows and varying numbers of columns of the Proposed algorithm. Region sizes: 2, 4, or 8. (d) Grid graphs with 100 columns and varying numbers of rows for Dervish [18] algorithm. Region sizes: 2, 4, or 8.

Fig. 8. The runtime of the algorithm solving grid graphs of different sizes.

steeper curves; nevertheless, the algorithm still demonstrated convincing performance in this scalability test. In the overall slope of the runtime increment in function of the number of nodes, we can observe a stepwise increase, which is attributed to the nature of the binary search for path numbers, and for the fact that more columns result in more region-disjoint paths.

B. Comparison to previous algorithms

Next, we compare the proposed algorithm with its state-of-the-art counterparts [18], [20]. In our comparison, we refer to the algorithm in [18] as *Dervish*. As mentioned in the introduction, both algorithms have two main phases: the first phase finds a proper path P with some special properties, then the second phase applies a greedy iterative path search method starting from P. The greedy part is simple and relatively fast; however, finding a proper path in the first phase could be time-consuming. The simulations in [18] suggested using a fast heuristic for finding P, which may fail to find a proper first path to start with. Note that the second phase is also a bit slower than our approach. The main difference is that their algorithm processes each region multiple (possibly $O(|V|)$) times, whereas our approach processes each region only once (for creating D_0). Furthermore, for proving maximality, the algorithm of [18] needs to perform $O(|V^*|)$ depth first searches, indicating a worst-case runtime complexity not better than $O(|V|^2)$. Fig. 8d shows the runtime of the Dervish algorithm for grid graphs with varying numbers of rows.

Table I presents the results obtained on the *real-world networks* and corresponding regional failures used in [18]. We specifically selected problem instances where the fast heuristic in the first phase successfully found a proper P. The first and second columns display the runtimes of the two algorithms. Our implementation of Dervish is in C++ and uses the proposed algorithm instead of a heuristic in the first phase. The hop length of the paths obtained by the proposed algorithms is shown in the fifth column. The average hop length of all the 208 problem instances is 21.2 for the proposed algorithm, compared to 18.5 for Dervish. However, after applying a simple heuristic approach to reduce path lengths [18], [20], both algorithms achieve similar lengths. The post-processing step took a maximum of 14ms to complete.

VIII. Conclusion

In this paper, we propose an efficient algorithm for finding the maximum region-disjoint st-paths. While the general maximum path problem is known to be $\mathcal{NP}$-hard, there are theoretical results for polynomial algorithms for special cases when the network topology is planar. This is the first paper to suggest an efficient and relatively easy-to-implement algorithm for this problem. Our approach works on all planar graphs, where each set of failed links to protect corresponds to a connected geographical region, and the resulting paths must be non-crossing. Our algorithm encompasses and improves upon previous models in the field.

The key innovation of our approach is the use of an auxiliary graph called the regional dual graph. This reduces the problem of finding a single-source shortest path in a weighted directed graph, where the links can have negative weights. We implemented the algorithm in C++, and we managed to solve problem instances with 10 000 nodes within seconds. This is the first highly scalable solution for the problem, demonstrated by both theoretical runtime analysis and our measurements. The authors have provided public access to their code and data at https://github.com/jtapolcai/regionSRLGdisjointPaths.

TABLE I
Backbone network topologies used in the simulations [18]. For the details of the topologies, refer to [42].

Network name	$\lvert V \rvert$	Runtime [ms] Proposed	Runtime [ms] Dervish	Avg. path hops Proposed	Avg. path hops Dervish	Shortened hops Proposed	Shortened hops Dervish
Pan-EU	16	0.2	0.6	5.8	5.8	4.5	4.5
EU optic	22	0.4	1.5	7.0	7.1	3.7	3.7
US optic	24	0.3	1.2	7.6	7.6	3.1	3.1
EU (Nobel)	28	0.4	1.3	9.6	9.6	5.7	5.7
N.-American	39	0.6	2.7	12.6	12.6	5.3	5.3
US (NFSNet)	79	0.9	4.8	18.4	18.3	13.4	13.3
ATT-L1	162	1.8	16.4	11.7	11.7	10.9	10.9
US (Fibre)	170	2.2	5.2	39.7	9.6	6.1	6.5
US (Sprint-Phys)	264	2.2	21.9	45.7	45.7	21.9	21.8
US (Att-Phys)	383	4.8	68.2	60.8	60.8	26.2	25.4

REFERENCES

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] A. Pašić, R. G. ao Silva, F. Mogyorósi, B. Vass, T. Gomes, P. Babarczi, P. Revisnyei, J. Tapolcai, and J. Rak, "eFRADIR: An Enhanced FRAmework for DIsaster Resilience," *IEEE Access*, vol. 9, pp. 13 125–13 148, 2021. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9319646

[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. 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.

[17] L. Liu, B. Li, B. Qi, X. Ye, Y. Sun, S. Tian, C. Zhu, and P. Xi, "Optimization of communication capacity for load control considering shared risk link group in source-grid-load system," *International Journal of Electrical Power & Energy Systems*, vol. 122, p. 106166, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061519316230

[18] 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.

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

[20] 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.

[21] J. L. Gross and S. R. Alpert, "The topological theory of current graphs," *Journal of Combinatorial Theory, Series B*, vol. 17, no. 3, pp. 218–233, 1974. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0095895674900288

[22] R. Bellman, "On a routing problem," *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[23] F. Duan, "A faster algorithm for shortest path-spfa," *Journal of Southwest Jiaotong University*, vol. 29, no. 6, pp. 207–212, 1994.

[24] H. N. Gabow and R. E. Tarjan, "Faster scaling algorithms for network problems," *SIAM Journal on Computing*, vol. 18, no. 5, pp. 1013–1036, 1989.

[25] A. Bernstein, D. Nanongkai, and C. Wulff-Nilsen, "Negative-weight single-source shortest paths in near-linear time," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, 2022, pp. 600–611.

[26] C. McDiarmid, B. Reed, A. Schrijver, and B. Shepherd, "Non-interfering network flows," in *Algorithm Theory—SWAT'92: Third Scandinavian Workshop on Algorithm Theory Helsinki, Finland, July 8–10, 1992 Proceedings 3*. Springer, 1992, pp. 245–257.

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

[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] B. Vass, E. Bérczi-Kovács, A. Barabás, Z. L. Hajdú, and J. Tapolcai, "A whirling dervish: Polynomial-time algorithm for the regional srlg-disjoint paths problem," *IEEE/ACM Transactions on Networking*, 2023.

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

[32] 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.

[33] ——, "Finding disjoint paths in networks with star shared risk link groups," *Theoretical Computer Science*, vol. 579, pp. 74–87, 2015.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] R. Girão-Silva, B. Nedic, M. Gunkel, and T. Gomes, "Shared Risk Link Group disjointness and geodiverse routing: A trade-off between benefit and practical effort," *Networks*, vol. 75, no. 4, pp. 374–391, 2020.

[39] 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.

[40] 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.

[41] 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.

[42] 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.