DateLine: Efficient Algorithm for Computing Region Disjoint Paths 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. During network design, groups of network elements prone to potential failure events are identified. These groups are referred to as *Shared Risk Link Groups* (SRLGs). When these SRLGs consist of a set of links intersected by a connected region of the plane, they are termed 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, where the paths are 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 in two aspects. First, instead of node-disjointness, we search for non-crossing regional-SRLG-disjoint paths. Second, we show how the algorithm can be extended to solve problems in directed networks. 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 either improves the previous worst-case runtime complexity or can solve the problem with high probability (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.

Index Terms—**Survivable routing, disaster resilience, network failures, shared risk link groups, SRLG, polynomial algorithms**

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. Contact them on erika.berczi-kovacs@ttk.elte.hu, and peti1234@student.elte.hu.

Balázs Vass and János Tapolcai are with Department of Telecommunications and Artificial Intelligence, Faculty of Electrical Engineering and Informatics (VIK), Budapest University of Technology and Economics (BME) and HUN-REN-BME Information Systems Research Group. Balázs Vass' main affiliation is the Faculty of Mathematics and Computer Science, Babeş-Bolyai University of Cluj Napoca, Romania. Contact them on balazs.vass@ubbcluj.ro, tapolcai@tmit.bme.hu.

This work was partly supported by Project nos. ANN_20 135606, FK_20 134604, FK 132524, PD_21 137698, and K_23 146347, which have been implemented with support from the National Research, Development and Innovation Fund of Hungary. 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. This study has received funding from the European Union's Horizon Europe research and innovation programme. *(Corresponding author: Balázs Vass.)*

Co-funded by the European Union

An earlier version of this paper appeared at IEEE INFOCOM '24 main track [1]. Parts of this work are based on [2], appeared in IEEE INFOCOM WKSHPS '24.

I. Introduction

The calculation of disjoint routes is key to surviving network failures. Using network flow algorithms, such as the Suurballe algorithm [3], we can efficiently compute node- and edge-disjoint paths. However, node- and edge-disjointness are usually not sufficient because network elements often fail together. These groups of elements that can fail together are identified during the network design phase and are called Shared Risk Link Groups (SRLGs) [4]–[11]. Algorithmically, finding SRLG-disjoint paths is $\mathcal{NP}$-Complete in general [12], [13].

Recently, significant progress has been made in resolving the above apparent contradiction. Many studies [14]–[23] have confirmed that severe outages in networks often result from catastrophes such as earthquakes, hurricanes, tsunamis, and tornadoes. These are called regional SRLGs and consist of network equipment in a well-defined physical area. Furthermore, the topology of backbone networks is usually planar because the links traverse long distances, and installing a cross-connect at intersections of optical fibers is economically beneficial. With the constraints that the topology graph is planar and the SRLGs have some planar property, node and SRLG disjoint path finding becomes efficiently solvable [24]. The main issue, which has relegated this to a theoretical result, is that the algorithms are very complex and lacked reliable implementations. This paper makes significant strides in this area by presenting a simple and easily implementable algorithm that quickly finds regional-SRLG-disjoint paths in planar graphs.

To be precise, the input is a planar undirected graph $G = (V, E)$ representing the network topology. Note that most real backbone topologies provided by public repositories [25], [26] are planar. In line with this, many previous works in this area also assumed the planarity of the input network G [24], [27]–[33] and others. Our techniques can be extended to topologies with a few edge crossings in some cases (see §VII-B). However, dealing meticulously with non-planar networks would exceed the limits of this paper.

Operators often dislike planarity because they do not want to reveal the exact coordinates of their network equipment, which is crucial for military and economic reasons. However, our algorithms only require the dual graph and its relation to the network topology. Specifically, 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 provided.

We assumed that the list of regions (or SRLGs) is also part of the input, given as a list of edge sets $\mathscr{R} \subseteq 2^E$. This list of regions is identified during the network design phase based on historical data and exploration of network vulnerabilities. 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^*. Here, two st-paths are $\mathscr{R}$-disjoint if there is no edge set in $\mathscr{R}$ intersecting both paths.

Even with the above assumptions, finding the maximum number of region-disjoint st-paths is $\mathscr{NP}$-hard [27]. To have a polynomial-time solvable problem, [33] added one last assumption that the obtained paths should be node-disjoint as well. This assumption holds for typical applications, for example when circular disk failures are considered [31].

Both [31] and [33] 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: first, searching for an appropriate starting path, and second, iteratively extending the solution with more region-disjoint paths. The second step is relatively straightforward to implement; the main implementation challenges lie in the first step. The algorithms proposed in [31], [33] perform well only when more than two region disjoint paths exist. The study in [31] offered an algorithm relying on the topological properties of the graph (e.g., the exact location of the nodes) for solving the first step, which was further generalized in [33] 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 [33]. 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. In fact, both steps of the previous algorithms have a slower running time than our approach. The main results of the paper are the following:

- We consider the problem of maximum region-disjoint st-paths, and instead of assuming disjointness of nodes, we just assume that the paths are *non-crossing*, see Fig. 1. Our model generalizes all previous tractable ones mentioned in §VIII. 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 10000 nodes in < 1 second.
- We prove that the optimum of the non-crossing model above gives a tight 2-additive approximation for the $\mathscr{NP}$-hard maximum region-disjoint paths problem in general (Thm. 3), which is better than the multiplicative approximation given in [27].
- We generalize our theoretical findings to directed graphs, as presented in §VI.

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. The generalization of our theoretical findings to directed graphs is presented in §VI. §VII discusses some further issues of our approach. In §VIII, we summarize previous results and techniques. In §IX, we provide our numerical evaluations, and finally, §X concludes our work.

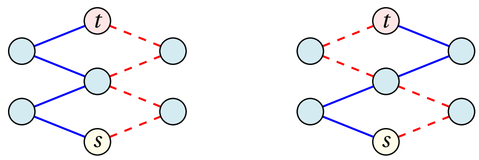
II. Problem Formulations, Main Results and Algorithm

The input of the problem is a planar graph $G = (V, E)$ with vertex set V and edge set E. An efficient way of storing such an input graph is if the incident edges for every node are given in clockwise order, called a rotation system [34]. 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^*.

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 dash dotted 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 $\mathscr{R} \subseteq 2^E$ of regions and a pair $s, t \in V$ of nodes, two st-paths are said to be **region-disjoint**, if there is no region $R \in \mathscr{R}$ intersecting both paths (see Fig. 3). A set of st-paths is region-disjoint if they are pairwise region disjoint. Finally, 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-dash-dotted region. For a set of regions $\mathscr{R}$ let $\|\mathscr{R}\| := \sum_{R \in \mathscr{R}} |R|$.



(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.

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: Maximum 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 $P_1,P_2\dots,P_k$

Unfortunately, Problem 1 is $\mathscr{NP}$-hard [27, Thm. 6], and only a multiplicative approximation was known to its optimum [27]. 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 $P_1,P_2\dots,P_k$

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.*

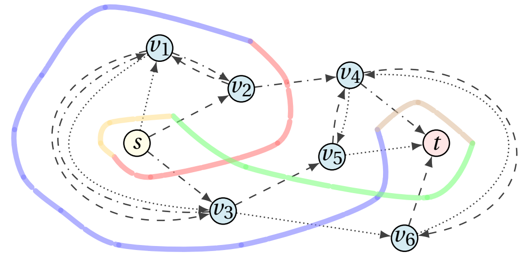


Fig. 2. Example for the tightness of Thm. 3. If regions are only the five colored lines, then $MF_{nc}=1$, $MF=MC=3$. Paths of the crossing max-flow are depicted by the dotted, dashed, and dashdotted arcs, respectively. By adding all node failures (except from s and t), MF becomes 1.

Remark. In fact, the time complexity of our algorithm can be expressed as $O(\log(k^*)A)$, where A is the complexity of an arbitrary algorithm (e.g., the Bellman-Ford [35]) suitable for computing a feasible potential (or finding a negative cycle) on an auxiliary directed graph with $O(\|\mathscr{R}\|)$ nodes and edges and $|V|^2$ maximum absolute value of edge weights. In essence, the complexity of our algorithm solely depends on the complexity $O(A)$ of the chosen subroutine (see §IV-B).

The main parts of our algorithmic framework are described in §II-C. Its details and the proof of correctness are presented in §III. The runtime complexity is analyzed in §IV. Dealing with issues related to the generalization for directed graphs is outlined in §VI.

For a maximum number of region-disjoint st-paths problem the corresponding min-cut problem can be solved in polynomial time [27]. Next, we present a theorem comparing these optimum values.

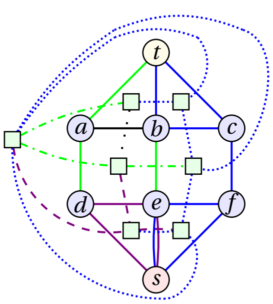
Theorem 3. *Let a maximum number of region-disjoint st-paths problem instance and its corresponding minimum regional st-cut problem be given, and let MF and MC denote their optimal values, respectively. Moreover, let MF_{nc} denote the optimal value of the non-crossing version of the problem. Then $MC-2\le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

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 $MF_{nc}=MC$).

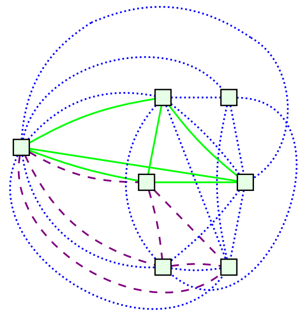
Remark. The above Theorems can be restated as they are on directed network topologies (instead of undirected ones), see §VI.

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 $D^*_{\mathscr{R}}=(V^*,A^*_{\mathscr{R}})$. Nodes of $D^*_{\mathscr{R}}$ are faces in V^*, and the arcs are derived from $\mathscr{R}$: for every region R we



(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).

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.

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[1].

We say a function on the faces $\pi : V^* \to \mathbb{Z}$ is a **feasible potential** with regards to c_k, if $c_k(uv) + \pi(u) - \pi(v) \geq 0$ for all $uv \in A^*_{\mathscr{R}}$. We will see in the next section that if c_k is conservative, we can compute a feasible potential. Then, a corresponding arc set F can be created, which describes the required paths $P_1, \ldots, P_k$. Intuitively, the boundaries between the $\mod k$ 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.

[1]The following fake application of our approach may help to form an intuition about its workings. Suppose s and t are the South and North Poles, respectively, and that we want to create k time zones on the earth, respecting some criteria. Each face of G represents a territory that should not be split by a time zone border. A face pair f_1, f_2 neighboring the same region (i.e., $(f_1, f_2) \in A^*_{\mathscr{R}}$) means that the resulting time zones of f_1 and f_2 may differ by at most 1 mod k. The above-mentioned path P corresponds to the International Date Line: while traveling around the globe and accumulating 'hours' (± 1), occasionally, we have to add/subtract a 'day' (i.e., $\pm k$), to remain consistent. Note that each resulting (time) zone may cross the 'Date Line', i.e. path P.

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 $P_1, P_2 \ldots, P_k$ and witness for non-existence of $k+1$ paths.

```
binary search on k (check existence of k paths with Alg. 2)
    ⟹ k* optimum
c_{k*} ⟹ π ⟹ paths P_1,...,P_k
// k region disjoint non-crossing paths
c_{k*+1} ⟹ C* // Witness of non-existence
return P_1,...,P_k and C*
```

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

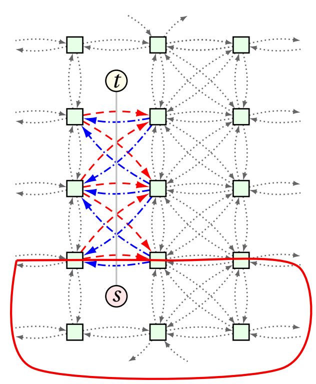
The existence of k region-disjoint non-crossing st-paths can be reduced to checking the conservativity of weightings in two steps. First, we show that some dual walks in G^* with some special properties are witnesses for the non-existence of k required paths (see next subsection, Lemma 4).

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 $D^*_{\mathscr{R}}$.

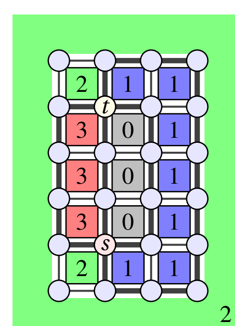
A. Witness for the non-existence of region-disjoint, non-crossing st-paths

In order to give a witness we need to define some notions on the dual graph (also used in [33]).

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.



(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 $c_{k\geq 5}$, 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.

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 [33] 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 $k \geq 2$, and let C^* be a closed walk in the dual graph with $w(C^*) > 0$. Then $\left\lfloor \frac{l(C^*)}{w(C^*)} \right\rfloor \geq k$.*

Proof: 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 $k \geq 2$, then $|w^{P_j}(C_i^*)| \leq 1$ for $1 \leq i \leq l$ and $1 \leq j \leq k$.*

Proof: Assume indirectly that $w^{P_j}(C_i^*) \geq 2$. Then for any planar embedding of G with respect to the given rotation system, 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 $k \geq 2$, each path P_i intersects at least $w(C^*)$ *distinct* subpaths C_j^*, which gives $kw(C^*) \leq l(C^*)$, that is $\lfloor l(C^*)/w(C^*) \rfloor \geq k$ indeed. (See Fig. 2 with C^* of red-blue-brown-green-yellow regions: $\frac{l(C^*)}{w(C^*)} = \frac{5}{3} \geq MF_{nc}$.) ■

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 $w^P(Q_1^*) \neq w^P(Q_2^*)$. 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 if' 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 $C' = f_1, f_2, \dots, f_l, f_1$ 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 $Q_1, Q_2, \dots, Q_l$ 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. ■

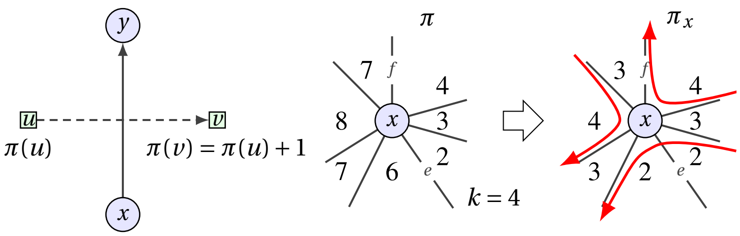
Next we turn to the second part and show that if c_k is conservative, then the required paths exist.

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).



(a) Creating F

(b) Pairing edges

Fig. 5. Illustrations for Lemma 9

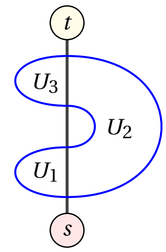


Fig. 6. Partition of $V(R^*)$ with respect to P

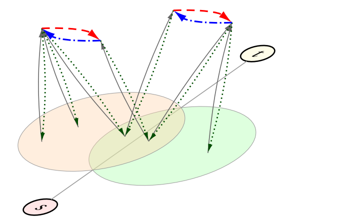


Fig. 7. Illustration for D_0. There are two circular regions. Solid, and dotted arcs cost 1, and 0, while red-and-dashed, and blue-and-dash-dotted arcs cost $-k$ and k, respectively.

So after the translation the π_x values of neighboring 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 neighboring 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 neighboring 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 $\pi(u_i) - 1 = \pi(u_{i+1})$ ($1 \le i \le l$), 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 maximum 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 $x \equiv y \mod k$. 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 $V(R^*)$ neighboring x. Thus the edges in R with different π'-valued neighboring 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 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).

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

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

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

We build a new auxiliary graph D_0 and define arc weights c_k^0 such that c_k is conservative on $D^*_{\mathscr{R}}$ if and only if c_k^0 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 $U_1, \ldots, U_l$ of $V(R^*)$ as in Claim 12 and for each U_i we add a node u_i^R to V^* and arcs $u_i^R u_{i+1}^R$ and $u_{i+1}^R u_i^R$ $(1 \leq i \leq l)$. If set U_i is on the left (or right) of separating subpath R_i, we set $c_k^0(u_i^R u_{i+1}^R) := -k$ and $c_k^0(u_{i+1}^R u_i^R) := k$ (or $c_k^0(u_i^R u_{i+1}^R) := k$ and $c_k^0(u_{i+1}^R u_i^R) := -k$). For every set U_i and every node $v \in U_i$ we add arcs $v u_i^R$ and $u_i v$ with weights 1 and 0, respectively (see Fig. 7 for illustration).

Claim 13. *Weighting c_k^0 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.

For an edge weighting c on a directed graph, let $d_c(u, v)$ denote the shortest path from u to v.

Proposition 14. *Weighting c_k on $D^*_{\mathscr{R}} = (V^*, A^*_{\mathscr{R}})$ is conservative if and only if for any fixed node $v^* \in V^*$ 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 this 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.

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 with possibly negative weights is the Bellman-Ford (BF) algorithm that has a complexity of $O(nm)$ on a graph with n nodes and m arcs [35]. 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) [36], 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 (§IX) 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, [37] claims the following.

Theorem 15 (Theorem 2.2. of [37])**.** *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 [38])**.** *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 [33]. 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 [33].

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 $\lfloor \frac{l(C^*)}{w(C^*)} \rfloor < 2$. ■

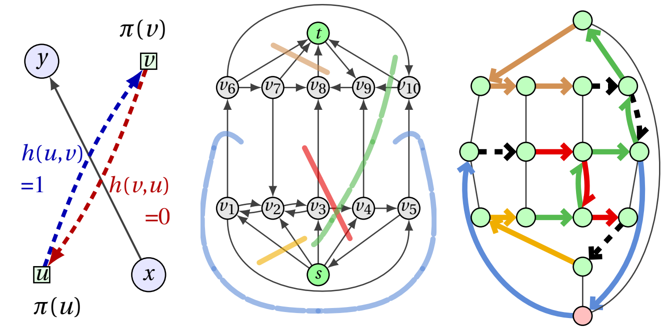
We will apply the min-max theorem above to prove Thm. 3.

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 [33, Thm. 7]. Clearly, $MF_{nc} \leq MF$ and from Claim 20 $MC \leq \lfloor l(C^*)/w(C^*) \rfloor + 2 \leq MF_{nc} + 2$.



(a) Dual and reverse arcs (b) Directed topology (c) Directed witness

Fig. 8. Generalization to directed graphs. Fig. 8a shows a dual arc uv of a directed arc xy with $h(uv) = 1$ and its reversed arc vu with $h(vu) = 0$. On Fig. 8b is drawn a directed network topology, and the regions ($\mathscr{R}$) have unique colors. Fig. 8c depicts the (undirected) dual graph with a closed (directed) dual walk C such that $l(C) = 5$, $w(C) = 3$, and hence $l(C)/w(C) < 2$; note that the black-and-dashed arcs count as 0 in $l(C)$.

By merging the inequalities, we get the lower bound on MF: $MC - 2 \leq MF_{nc} \leq MF \leq MC$. ■

We note that, for regions derived from circular failures in a given planar embedding of G, even $MC - 1 \leq MF$ holds [31]. Paper [24] mentions a broader class of regions for which this tighter relation of $MC - 1 \leq MF$ is true.

VI. Generalization to directed graphs

In the problem formulation we assumed the graph G is undirected, but similar techniques work for the directed case with minor modifications as we sketch below. For a directed planar graph $G = (V, A)$ with a given rotation system, the dual of an arc $xy \in A$ is an arc uv crossing xy from left to right (see Fig. 8a).

In the directed case we also assume that R^* is weakly connected in $V^*[R^*]$ for every region R (a directed graph is weakly connected if it is connected in the undirected sense). Similarly to the undirected case, the regional dual graph $D^*_{\mathscr{R}} = (V^*, A^*_{\mathscr{R}})$ is a graph on the set of faces V^*, and for each region R the complete directed graph on $V^*[R^*]$ is in $D^*_{\mathscr{R}}$. These type of arcs are called *normal arcs*. In addition, the reverse of each arc in A^* is added to $D^*_{\mathscr{R}}$, which are called *reversed arcs*.

Let $h : A^*_{\mathscr{R}} \to \{0, 1\}$ be a function which is 1 and 0 on normal and reversed arcs, respectively. Then cost function c_k is defined as $c_k(u^*v^*) = h(u^*v^*) - w^P(u^*v^*) \cdot k$ (where w^P is determined with respect to an arbitrary st-path P). This way, arc $yx \notin$ cannot be part of F_x, only $xy \in A$ (see Figure 8a). Here, the right orientation is guaranteed by reversed arcs: for a reversed arc vu crossing P λ times we have $c_k(vu) = \lambda \cdot k$, which ensures for a c_k-feasible potential that $\pi(v) - \pi(u) \leq \lambda \cdot k$. After translation of π to π_x by $\lambda \cdot k$ (see Figure 5b), we get $\pi_x(v) - \pi_x(u) = (\pi(v) - \lambda \cdot k) - \pi(u) \leq 0$.

It can be shown analogously to the undirected case that c_k is conservative if and only if there are k region-disjoint directed st-paths in G, and the same running times apply for the directed case as in Theorem 1.

Given a normal arc uv, a representing path in G^* is a uv-path P_u^v in $V^*[R^*]$ in the undirected sense, thus it may contain arcs of G^* oriented in the opposite direction as P_u^v. To define a witness for the non-existence of k region-disjoint directed paths in G^*, we consider closed walks of G^* in the undirected sense (where some arcs have the same orientation as the walk, and some arcs are opposite). The region-length $l(C^*)$ of such a closed walk C^* is the minimum number of regions that cover those arcs which have the same orientation as C^*. The witness for non-existence is a closed walk C^* such that $w_{lr}(C^*) - w_{rl}(C^*) > 0$ and $\left\lfloor \frac{l(C^*)}{w(C^*)} \right\rfloor < k$. Such a witness can be derived from a negative c_k-weight directed cycle C in $D^*_{\mathscr{R}}$, where a normal arc corresponds to its representing path in G^* and a reversed arc vu is replaced by arc $uv \in A^*$. Then the region-length of the derived closed walk in G^* has region-length at most the number of normal arcs in C, which gives the witness above indeed. The corresponding min-max theorem can be formulated analogously to Theorem 19 with this witness.

For illustrations, see Fig. 8b and Fig. 8c, respectively. The two figures explain a setting where the input topology is a directed graph, and, in the directed sense, there are no two non-crossing $\mathscr{R}$-disjoint paths; however, the min-cut equals 3.

VII. Discussion

A. Searching for a fixed number of paths

If one wants to compute a fixed number of k $\mathscr{R}$-disjoint paths (e.g., $k = 2$ for 1+1), only Alg. 2 has to be called (once). Compared to the basic version (Alg. 1), this eliminates a factor of $\log k^*$ from the complexity, where k^* is the maximum number of $\mathscr{R}$-disjoint paths.

B. Non-planar network topologies

This paper assumed the network topology to be planar, enabling the efficient calculation of the maximum number of region-disjoint paths. Naturally rises the question if the problem can be solved efficiently if there is a strictly positive number x of link crossings in an embedding of the network in the plane. [24, §VI/C] presents a natural setting in which a very heuristic recycling of a polynomial algorithm solving a planar problem instance yields a polynomial algorithm if x is $O(\text{poly}(\log n))$. However, it is possible the circumstances depicted in [24, §VI/C] do not hold at some specific locations of the network. A comprehensive study of such cases would exceed the limits of current paper.

In this regard, we state the following exploratory chain of thoughts. Suppose we have a network embedded in the plane with a small number of x link crossings, and that each disaster forms a connected area in the topological sense. In this setting, let MF^x, MF^x_{nc}, and MC^x denote the maximum number of non-crossing and (possibly) crossing region-disjoint paths, and the size of a minimum cut, respectively. Then, by removing a number of x edges at crossings, we get instances of Problem 1/Problem 2. In the process, the values of MF^x and MF^x_{nc} decreased at most by x. Thus, using Alg. 1, at least $MF^x_{nc} - x$ number of region-disjoint paths can be efficiently computed. Further, based on Thm. 3, we can state that $MC^x - x - 2 \le MF^x_{nc} \le MF^x \le MC^x$.

C. Incorrectness of the natural greedy algorithm

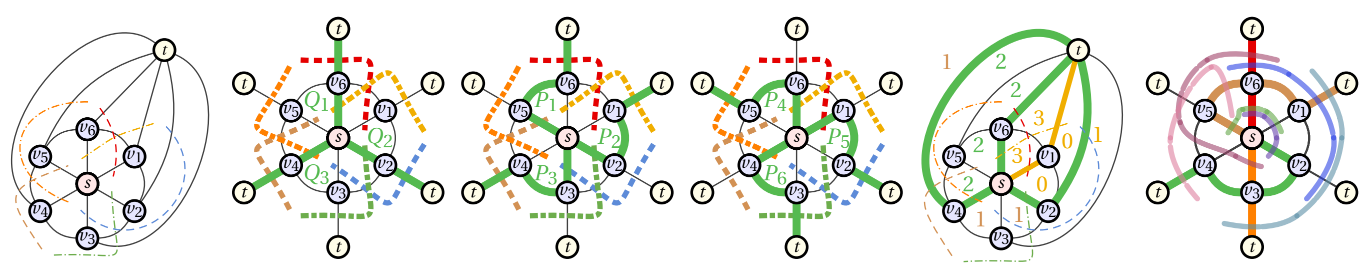
By a careful reevaluation of the theoretical bases, current paper drastically simplifies the algorithmic machinery of [24] (that was, broadly speaking, a simplification of [31], cf. §VIII-A). Oversimplified, paper [24] operated with an almost entirely greedy algorithm: given a set of $k-1$ $\mathscr{R}$-disjoint paths $P_1, \ldots, P_{k-1}$, it generates a series of new paths $P_k, P_{k+1}, \ldots, P_l, \ldots$ with the rule that a new path P_l should be

1) the closest path to P_{l-1} that is clockwise $\mathscr{R}$-disjoint from P_{l-1}, and
2) non-strictly clockwise to P_{l-k}.

Note that for rule 2), for $l = k$, we consider $P_{l-k=0} := P_{k-1}$. Intuitively speaking, while rule 1) feels natural, at first glance, it is not clear whether rule 2) is inevitably needed to make the algorithm correct. We call the algorithm neglecting rule 2) and operating with only rule 1) as *Greedy*. A consequence of the next Theorem to be presented is that Greedy is not necessarily suitable for computing routing for 1+1 path protection.

Theorem 21. *The Greedy algorithm does not necessarily find an optimal solution for Problem 2 even if $MF_{nc} = 2$.*

Proof: A counterexample is depicted in Fig. 9f. We note that t is drawn in multiple copies, for easier visualization. There are six disaster regions. Starting from an unfortunate path s, v_6, t, the Greedy does not find 2 region-disjoint paths. ■



(a) $G = (V, E)$ and $\mathscr{R}$ (b) Paths Q_1, Q_2, Q_3 (c) Paths P_1, P_2, P_3 (d) Paths P_4, P_5, P_6 (e) Paths from a potential (f) Greedy failing $k = 2$

Fig. 9. Example for the incorrectness of Greedy. Subfigures a) and e) show the input graph, in the rest of the subfigures t is drawn in multiple copies. b) shows three region-disjoint paths, c) and d) combined show a cycle of paths the Greedy generates, in which no 3 consecutive is region-disjoint. Finally, f) shows an example where there exist region-disjoint path pairs (e.g., s, v_1, t and s, v_4, t), but, starting from an unfortunate path, the Greedy does not find such a pair: the red-orange-brown-green is a cycle of non-region-disjoint paths; this means Greedy is not necessarily suitable for computing routing for 1+1 protection.

D. Visualization of both Greedy based approaches and our proposed algorithm

To provide a visual comparison between the Greedy, the algorithm of [24] (which we refer to as the *Dervish* hereafter), and the algorithm proposed by this paper, the rest of the subfigures of Fig. 9 depict an example where $MF_{nc} = 3$. The input graph is drawn in Fig. 9a, while in the rest of the subfigures, t is drawn in multiple copies. There are six (disaster) regions (depicted in Fig. 9b-Fig. 9d), in dashed-and-red, and dash-dotted-and-blue lines, respectively. While Fig. 9b shows a lucky run of the Greedy, Fig. 9c-Fig. 9d show an infinite cycle of paths generated by the algorithm, among which there are no three that are region-disjoint.

In the meantime, on Fig. 9c, when searching for the third $\mathscr{R}$-disjoint path, starting from path pair P_1, P_2, the Dervish generates the following paths: $P_3^D = \{s, v_3, v_4, t\}$, $P_4^D = \{s, v_6, t\}$, $P_5^D = \{s, v_2, t\}$, $P_6^D = \{s, v_4, t\}$. Since P_4^D, P_5^D, and P_6^D are pairwise node- and $\mathscr{R}$-disjoint, the Dervish returns with them. Intuitively, the key difference compared to Greedy is that the 4^{th} path is not let to have links locally anti-clockwise from P_1. Finally, Fig. 9e depicts the paths and potential our algorithm yields for $P = \{s, v_1, t\}$ (drawn yellow), and the potential values of faces are the $c_{k=3}$-distances from face $t - v_1 - v_2$.

VIII. Related work

A. Theoretical preludes

Maximum number of (crossing) region-disjoint paths: Seminal work [27] 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 [27]). 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, [27] 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: [28] considers generalizations of disjoint paths problems in directed graphs, 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 (in the undirected sense). 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 directed d-separated paths as a special case of region-disjointness in directed graphs, since such edge sets form a strongly connected subgraph in the dual graph. [28] 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, but their work is the starting point for the approach in [31].

Maximum flows on planar graphs: Dual graph-based techniques similar to our approach are used for maximum flows in planar graphs [39], where it is shown that the existence of a certain valued feasible flow is equivalent to the conservativity of a parametric edge weighting on the dual graph. Our algorithm extends this dual view for the region disjoint paths problem. However, there are several differences compared to the planar maximum flow that need to be handled. First, the regional dual graph is not necessarily planar, so running times of planar maximum flow algorithms do not carry over for this problem. In addition, region-disjointness of the given paths relies on the property that the arcs derived from the feasible potential can be partitioned into paths of the same type (Lemma 9).

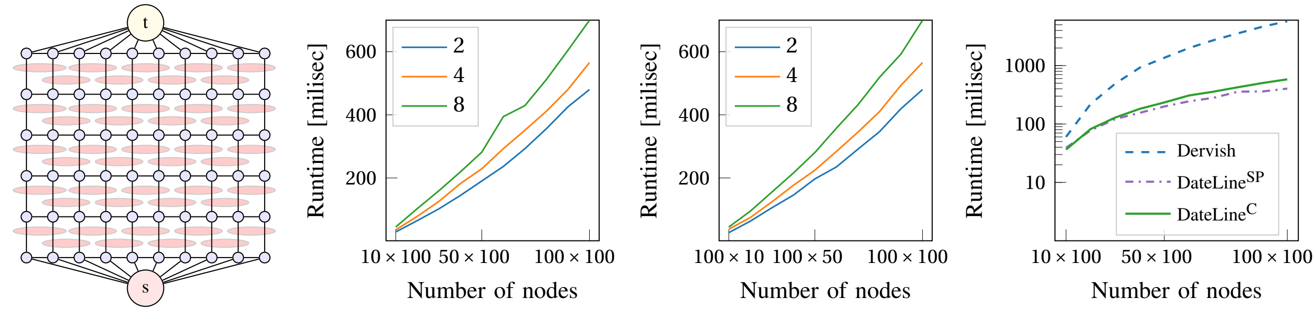
B. Survivable routing in communication networks

[40], [41] show that SRLGs can be enumerated effectively for several geographic failure shapes, such as circle, ellipse, rectangle, square, or equilateral triangle. Papers [14], [29] 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). [29] gives 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, [31], [32] proved this conjecture. These papers adapted the method of [28], [42] 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 [24], [33], 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 [42] and [31]. 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 [12]. 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 [43]–[45]. Some polynomially solvable subcases of this problem are also presented



(a) A 6 × 10 grid graph with regions drawn in red, each consisting of a size of 2.

(b) Runtime of DateLine^C on grids with 100 cols and varying numbers of rows. Region sizes: 2, 4, or 8.

(c) Runtime of DateLine^C on grids with 100 rows and varying numbers of columns. Region sizes: 2, 4, or 8.

(d) Grid graphs with 100 columns and varying numbers of rows for various algorithm. Region sizes: 2, 4, or 8.

Fig. 10. The runtime of the algorithm solving grid graphs of different sizes.

in [43], [44]. An ILP solution for the SRLG-disjoint routing problem is given in [46]. 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 [47]–[49]. Based on a probabilistic SRLG model, [50] aims to find diverse routes with minimum joint failure probability via an integer non-linear program (INLP). Heuristics were also investigated [51], [52], unfortunately, with issues like possibly non-polynomial runtime or possibly arising forwarding loops when the disaster strikes.

IX. Numerical Evaluation

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[2]. To measure the runtime performance, we conducted the experiments on a 2015 Macbook Pro equipped with a 2.2 GHz CPU and 16 GB 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. 10a. 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. 10b, we investigate a series of graphs with each having 100 columns, and their number of rows varying between 10 and 100. 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 less than a second. We repeated the aforementioned process, but this time with gradually increasing the number of columns of a 10×100 grid graph, see Fig. 10c and got very similar results.

B. Comparison to previous algorithms

Next, we compare the proposed algorithm with its state-of-the-art counterpart called *Dervish* [33], because it gave better running times than previous approaches both in theory and practice. As mentioned in the introduction, this algorithm has two main phases: the first phase finds a proper path P with some special properties, then the second phase applies a nearly-greedy iterative path search method starting from P. The nearly-greedy part is simple and relatively fast; however, finding a proper path in the first phase could be time-consuming. The simulations in [33] suggested using a fast heuristic for finding P, which may fail to find a proper first path to start with[3]. Therefore, in our implementation of *Dervish*, we have replaced the first step with the DateLine algorithm for $k = 2$ that finds the first path to start with. This could significantly decrease the runtime of *Dervish* compared to [33]. Dervish is still much slower than the proposed algorithms, because, during the nearly-greedy iterative path search method, it processes each region multiple (possibly $O(|V|)$) times, whereas the proposed approach processes each region only once (for creating D_0). Furthermore, for proving maximality, the algorithm of [33] needs to perform $O(|V^*|)$ depth first searches, indicating a worst-case runtime complexity not better than $O(|V|^2)$. Also, we have compared two versions of the proposed DateLine algorithm, the *DateLine^{SP}* [1] where $\pi(v)$ is the length of the shortest uv-path for an arbitrary face-node u of maximum degree, and the *DateLine^C* where π is the so-called *canonical* feasible potential, where $\pi(v)$ is the length of the shortest path ending at v. Fig. 10d shows the runtime of

[2] https://github.com/jtapolcai/regionSRLGdisjointPaths

[3] In our experiments 6% percent of the problem instances, this fast heuristic approach failed to find a proper first path to start with.

TABLE I

PATH LENGTHS OF REAL-WORLD BACKBONE NETWORK TOPOLOGIES. FOR PATH LENGTHS, X% MEANS THE PATHS ARE ON AVERAGE X% LONGER COMPARED TO THE SHORTEST PATH. FOR EXAMPLE, 0% MEANS WE HAVE THE SHORTEST PATH.

Network			$s-t$ pairs		Without shortening			After the path shortening heuristic								
			# disj. paths		Avg. relative path length			Avg. relative path length			The shortest path			The longest path		
name	$\|V\|$	$\|E\|$	SRLG	node	DateLine^{SP}	DateLine^{C}	Dervish	DateLine^{SP}	DateLine^{C}	Dervish	DateLine^{SP}	DateLine^{C}	Dervish	DateLine^{SP}	DateLine^{C}	Dervish
Pan-EU	16	22	1.61	2.00	44%	26%	60%	21%	26%	26%	15%	0%	0%	26%	51%	51%
EU optic	22	45	2.22	2.68	111%	87%	82%	29%	29%	30%	2%	1%	2%	58%	59%	60%
US optic	24	42	1.97	2.77	104%	85%	93%	36%	36%	35%	3%	0%	0%	73%	75%	74%
EU (Nobel)	28	41	1.86	2.00	148%	148%	17%	17%	17%	17%	0%	0%	0%	33%	33%	33%
N.-American	39	61	1.99	2.53	149%	122%	106%	44%	41%	43%	4%	1%	2%	86%	84%	88%
US (NFSNet)	79	108	2.24	3.00	318%	227%	89%	68%	68%	68%	0%	0%	0%	120%	120%	120%
% ATT-L1	162	244	2.75	3.00	283%	206%	65%	44%	44%	46%	0%	0%	0%	114%	114%	119%
US (Fibre)	170	229	1.76	2.06	223%	150%	231%	49%	46%	51%	11%	4%	9%	87%	88%	93%
US (Sprint-Phys)	264	312	1.88	2.00	394%	64%	59%	33%	26%	42%	3%	2%	6%	62%	51%	78%
US (Att-Phys)	383	483	1.81	2.19	319%	185%	192%	57%	48%	51%	13%	2%	7%	102%	96%	97%

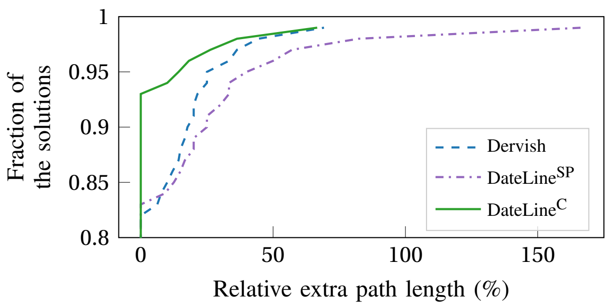


Fig. 11. The Cumulative Distribution Function (CDF) of the length of the shortest path in the solutions.

the Dervish algorithm for grid graphs with different numbers of rows. Our implementation of Dervish is still slower than DateLine^{SP} or DateLine^{C}. Overall, both versions of DateLine have remarkable runtime, they can find regional-SRLG disjoint paths on graphs with 10 000 nodes within a second.

C. Path lengths

Table I presents the results obtained on the *real-world networks* and corresponding regional failures used in [33]. For the details of the topologies, see [26], and for the details of the regional failures, see [33]. We have generated 12413 different problem instances for each network, and different regional-SRLGs, and source and destination node pairs. We have filtered out the SRLGs that form a cut between s and t to always have a feasible solution. Column 4 of the table shows the average number of paths we obtained (denoted by k, and it is the optimal value), which is slightly less than the number of node disjoint paths (Column 5).

The hop lengths of the paths obtained by the proposed algorithms are shown in the remaining columns for DateLine^{SP}, DateLine^{C}, and Dervish. We have evaluated the paths in four aspects: the path length obtained by the algorithms (Cols. 6-8), the path length after shortening them (Cols. 9-11) using a simple heuristic as follows [24], [31]–[33]. Iteratively takes each path among the k, and fix $k-1$ other paths, and we compute a shortest s-t path that is $\mathscr{R}$-disjoint from these. As the total length of the paths decreases after each successful shortening, the heuristic terminates after a finite number of iterations. The runtimes of the algorithms are omitted from the table because the real-world networks are relatively small compared to large lattice graphs, with runtimes ranging from a few milliseconds to a few tens of milliseconds.

We have also evaluated the lengths of the shortest (Cols. 12-14) and longest paths (Cols. 15-17) in the solution after applying the shortening heuristic. The values are normalized compared to the shortest path, and we present them in %. For example, 44% for DateLine^{SP} on Pan-EU network means the obtained paths in the solution of DateLine^{SP} were 44% longer than the shortest path between the same source and destination nodes.

As we expected the paths obtained by the algorithms are relatively long, as the primary goal of the algorithm is to find the maximal number of regional-SRLG disjoint paths. DateLine^{SP} is the worst in this metric as adapts a special path search approach, which results in rather winding roads. DateLine^{C} can improve in this aspect, by selecting a more proper canonical path. Nevertheless, the shortening heuristic can significantly reduce the path lengths. Note that, we have no theoretical guarantee on the length of the paths. We have also evaluated the length of the shortest path among the obtained regional-SRLG disjoint paths. It is very close to the absolute shortest path between the same source and destination nodes. The gap is surprisingly small for DateLine^{C}, where it was $0-2$% on average for all the graphs and source-destination pairs. This is evaluated in Fig. 11, where the Cumulative Distribution Function of the relative path length is shown for each algorithm for the 12413 different problem instances we have examined. In 93% of the problem instances, one of the paths was actually the shortest path for the DateLine^{C} algorithm. In the remaining instances, one of the obtained paths was still relatively short, being at most 50% longer than the shortest path. We have also examined the longest paths among the regional-SRLG disjoint paths. Our expectation was that a shorter first path will result in longer second and third paths. However, surprisingly the longest paths of DateLine^{C} are not longer (and often moderately shorter) than that of DateLine^{SP} and Dervish.

X. CONCLUSION

In this paper, we propose an efficient algorithm for finding the maximum number of region-disjoint non-crossing st-paths, when the input network topology is a planar graph (without crossings links), each SRLG is a connected area of the plane (called region or regional-SRLG). This is the first paper to give an efficient and easy-to-implement algorithm for this problem, making it suitable for practical use. Furthermore, on the theoretical side 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 to finding a feasible potential on the dual nodes, which can be done by a single-source shortest path search 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 a second. This is the first highly scalable solution for the problem, demonstrated by both theoretical runtime analysis and our measurements.

REFERENCES

[1] E. Bérczi-Kovács, P. Gyimesi, B. Vass, and J. Tapolcai, "Efficient algorithm for Region-Disjoint survivable routing in backbone networks," in *IEEE INFOCOM 2024*, Vancouver, Canada, 2024.

[2] B. Vass, E. Bérczi-Kovács, P. Gyimesi, and J. Tapolcai, "Efficient computing of disaster-disjoint paths: Greedy and beyond," in *2024 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Vancouver, Canada, May 2024.

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

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

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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

[24] 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.

[25] "The internet topology zoo." [Online]. Available: http://www.topology-zoo.org/dataset.html

[26] 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.

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

[28] 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.

[29] 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.

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

[31] 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.

[32] 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.

[33] 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.

[34] 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

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

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

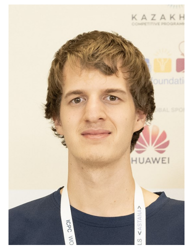
[37] H. N. Gabow and R. E. Tarjan, "Faster scaling algorithms for network problems," *SIAM Journal on Computing*, vol. 18, no. 5, pp. 1013–1036, 1989.

[38] 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.

[39] J. Erickson, "Maximum flows and parametric shortest paths in planar graphs," in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '10. USA: Society for Industrial and Applied Mathematics, 2010, p. 794–804.

[40] S. Trajanovski, F. Kuipers, P. Van Mieghem *et al.*, "Finding critical regions in a network," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2013, pp. 223–228.
[41] S. Trajanovski, F. A. Kuipers, A. Ilić, J. Crowcroft, and P. Van Mieghem, "Finding critical regions and region-disjoint paths in a network," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 908–921, 2015.
[42] 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
[43] 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.
[44] ——, "Finding disjoint paths in networks with star shared risk link groups," *Theoretical Computer Science*, vol. 579, pp. 74–87, 2015.
[45] 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.
[46] 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.
[47] 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.
[48] 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.
[49] 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.
[50] 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.
[51] 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.
[52] 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.

Erika R. Bérczi-Kovács received the M.Sc. Degree in Mathematics and the Ph.D. degree in Applied Mathematics from the Eötvös Loránd University (ELTE), Budapest, in 2007 and 2015, respectively. She is currently a senior lecturer at the Department of Operations Research, ELTE, and she is with the MTA-ELTE Egerváry Research Group on Combinatorial Optimization. Her research interests are combinatorial optimization and operations research. In 2020 she was a recipient of the András Gács Prize for exceptional teaching.



Péter Gyimesi received his BSc degree and currently studies Mathematics at Eötvös Loránd University (ELTE), Budapest, with a focus on applications. He won the first prize at the university-wide Scientific Students Associations Conference in 2024. Qualified with his team to the International Collegiate Programming Contest (ICPC) World Finals for the first time from Hungary. Former IOI silver medalist.

Balázs Vass received his MSc degree in applied mathematics at ELTE, Budapest in 2016. He finished his PhD in informatics in 2022 on the Budapest University of Technology and Economics (BME). His research interests include networking, survivability, combinatorial optimization, and graph theory. He was an invited speaker of COST RECODIS Training School on Design of Disaster-resilient Communication Networks '19. He is a TPC member of IEEE INFOCOM '23, '24, and '25.

János Tapolcai received the M.Sc. degree in technical informatics and the Ph.D. degree in computer science from the Budapest University of Technology and Economics (BME), Budapest, in 2000 and 2005, respectively, and the D.Sc. degree in engineering science from the Hungarian Academy of Sciences (MTA) in 2013. He is currently a Full Professor with the High-Speed Networks Laboratory, Department of Telecommunications and Artificial Intelligence, BME. He has authored over 200 scientific publications. He is a winner of the MTA Lendület Program and the Google Faculty Award in 2012, Microsoft Azure Research Award in 2018. He is a TPC member of leading conferences, e.g. IEEE INFOCOM 2012-, and the general chair of ACM SIGCOMM 2018.