SRLG-Diverse Routing with the Star Property

Jean-Claude Bermond, David Coudert, Gianlorenzo D'Angelo, Fatima Zahra Moataz

SRLG-Diverse Routing with the Star Property

Jean-Claude Bermond*, David Coudert*, Gianlorenzo D'Angelo*, Fatima Zahra Moataz*
*Mascotte Project – INRIA, I3S, CNRS, Univ. Nice Sophia– Sophia Antipolis, France
Email:{jean-claude.bermond, david.coudert, gianlorenzo.d_angelo, fatima-zahra.moataz}@inria.fr

***Abstract* — The notion of Shared Risk Link Groups (SRLG) has been introduced to capture survivability issues where some links of a network fail simultaneously. In this context, the *diverse routing* problem is to find a set of pairwise SRLG-disjoint paths between a given pair of end nodes of the network. This problem has been proved NP-complete in general and some polynomial instances have been characterized.**

In this paper, we investigate the diverse routing problem in networks where the SRLGs are localized and satisfy the *star property* [1]. This property states that a link may be subject to several SRLGs, but all links subject to a given SRLG are incident to a common node. We first provide counterexamples to the polynomial time algorithm proposed in [1] for computing a pair of SRLG-disjoint paths in networks with SRLGs satisfying the star property, and then prove that this problem is in fact NP-complete. Then, we devise polynomial time algorithms for practically relevant subcases, in particular when the number of SRLGs is constant, the maximum degree of the vertices is at most 4, and when the network is a directed acyclic graph.

Finally we consider the problem of finding the maximum number of SRLG-disjoint paths in networks with SRLGs satisfying the star property. We prove that such problem is NP-hard and hard to approximate. Then, we provide exact and approximation algorithms for relevant subcases.

I. Introduction

To ensure reliable communications, many protection schemes have been proposed. One of the most used, called *dedicated path protection*, consists in computing for each demand both a working and a protection path. A general requirement is that these paths have to be diversely routed, so that at least one of them can survive a single failure in the network. This method works well in a single link failure scenario, as it consists in finding two edge-disjoint paths between a pair of nodes. This is a well-known problem in graph theory for which there exist efficient polynomial time algorithms. However, the problem of finding two diversely routed paths between a pair of nodes becomes much more difficult in case of multiple correlated link failures that can be captured by the notion of *SRLG (Shared Risk Link Group)*. In fact, an SRLG is a set of network links that fail simultaneously when a given event (risk) occurs. The scope of this concept is very broad. It can correspond, for instance, to a set of fiber links of an optical backbone network that are physically buried at the same location and therefore could be cut simultaneously (i.e. backhoe or JCB fade). It can also represent links that are located in the same seismic area, or radio links in access and backhaul networks subject to localized environmental conditions affecting signal transmission, or traffic jam propagation in road networks. Note that a link can be affected by more than one risk. In practice, the failures are often localized and common SRLGs are SRLGs verifying the *star property* [1] (coincident SRLGs in [2]). Under this property, all links of a given SRLG share an endpoint. Such failure scenarios can correspond to risks arising in router nodes like card failures or to the cut of a conduit containing links issued from a node (see Figure 1).

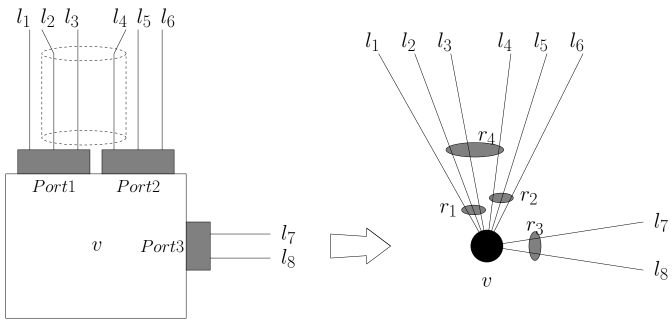


Fig. 1. Example of localized risks: link l_4 shares risk r_2, corresponding to Port 2 failure, with l_5 and l_6, and shares risk r_4, corresponding to a conduit cut, with links l_2 and l_3.

Related work

In the context of SRLG, basic network connectivity problems have been proven much more difficult to address than their counterparts for single failures. For instance, the problem of finding a "SRLG-shortest" st-path that is a path from node s to node t having the minimum number of risks has been proven NP-hard and hard to approximate in general (see [3]). However, the problem can be solved in polynomial time in two generic practical cases corresponding to localized failures: when all risks verify the star property [4] and when risks are of *span 1* (i.e. when a link is affected by at most one risk and links sharing a given risk form a connected component [3]).

The *diverse routing* problem in presence of SRLGs consists in finding two SRLG-disjoint paths between a pair of vertices (i.e. paths having no risk in common). It has been proven NP-complete in general [5]–[8] and many heuristics have been proposed. The problem is polynomial in some specific cases of localized failures: when SRLGs have span 1 [3], and in a specific case of SRLGs having the star property [9] in which a link can be affected by at most two risks and two risks affecting the same link form stars at different nodes (this result also follows from results of [3]).

Our results

We study the diverse routing problem when SRLGs have the star property and there are no restrictions on the number of risks per link. This case has been studied in [1] in which the authors claim that the diverse routing problem with the star property can be solved in polynomial time. Unfortunately their algorithm is not correct; indeed we exhibit, in Section II of our paper, counterexamples for which their algorithm concludes to the non existence of two SRLG-disjoint paths although two such paths exist. We give the notation used in the paper in Section III. In Section IV, we prove that the problem is in fact NP-complete (again, contradicting the supposed polynomiality of the algorithm of [1], unless $P = NP$). On the positive side, we show, in Section V, that the diverse routing problem can be solved in polynomial time in particular subcases which are relevant in practice. Namely, we solve the problem when the number of SRLGs is bounded by a constant, when the maximum degree is at most 4 or when the input network is a directed acyclic graph. Finally, we consider the problem of finding the maximum number of SRLG-disjoint paths. This problem has been shown to be NP-hard in [7]. We prove that it is also NP-hard under the star property, that it is hard to approximate and we give polynomial time algorithms for the above relevant subcases.

II. Counterexamples to the algorithm of Luo & Wang

Luo and Wang have published in [1] an algorithm to find a pair of SRLG-disjoint paths with minimum total cost from a source s to a destination t in graphs with SRLGs satisfying the star property. The algorithm is an adaptation of the Bhandari's algorithm [10] with the use of a modified version of the Bellman-Ford algorithm to find the second path. At first, the algorithm finds the shortest path P_a between s and t, then it reverses the edges of P_a and assigns to them the negative values of their costs. Virtual nodes and edges are added afterwards to deal with some specific cases. The second path P_b is then found with the use of the modified Bellman-Ford where SRLG-specific routing information is kept, particularly in the nodes of P_a. In case P_a and P_b share common segments, the procedure of deletion and exchange of the Bhandari's algorithm is used.

The algorithm is incorrect for two reasons:

The first weakness is that the first and last edges of P_a should be contained necessarily in the pair of paths returned by the algorithm: if no edge incident to s (resp. t) is SRLG-disjoint from the first (resp. last) edge of P_a, the algorithm will ignore the existence of 2 SRLG-disjoint paths even if they exist.

The counterexample in Figure 2 illustrates the first weakness: risk r is shared between edges $\{s, v_0\}$ and $\{s, v_1\}$ and risk $r' \neq r$ is shared between edges $\{s, v_0\}$ and $\{s, v_2\}$. The shortest path is $P_a = \{s, v_0, t\}$. Let us apply the algorithm. As it is explained in [1] page 451 lines 9–10, the initialization part guarantees that the first edge of P_b is risk-disjoint from P_a. Therefore in Figure 2, the algorithm does not find any edge to start and hence terminates concluding that there are no two SRLG-disjoint paths. However two SRLG-disjoint paths clearly exist, namely they are $P_1 = \{s, v_1, w_1, t\}$ and $P_2 = \{s, v_2, w_2, t\}$. A similar counterexample exists, with the risks at destination vertex, putting risk r_1 shared between edges $\{v_0, t\}$ and $\{w_1, t\}$ and risk r_1' shared between edges $\{v_0, t\}$ and $\{w_2, t\}$. This weakness can be overcome by running the algorithm for all possible pairs (u, v) such that u is incident to s and v is incident to t.

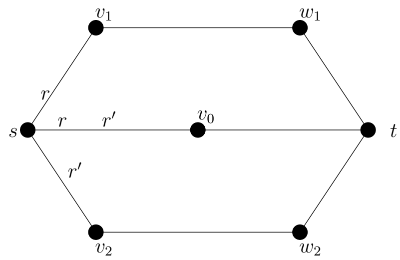


Fig. 2. Example 1.

The second weakness is that the algorithm only checks SRLG-disjointness around nodes of P_a and never checks other nodes. It assumes implicitly that the only nodes that can be shared by the two SRLG-disjoint paths are nodes belonging to P_a and ignores the existence of any other possibilities. Any attempt to modify the algorithm to face this weakness results in exponential time algorithms.

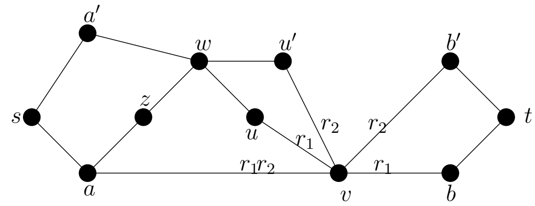


Fig. 3. Example 2

Figure 3, illustrating the second weakness, shows a counterexample to the algorithm in [1] that furthermore can give to the reader a flavor of the difficulty of the problem. In this figure, we have 2 specific risks r_1 and $r_2 \neq r_1$ forming a star in v. As vertex v is a cut-vertex any path should contain v. In this case, and to ensure the SRLG-disjointedness, one path should use the subpath u, v, b and the other one should use the subpath u', v, b'. We have two SRLG-disjoint paths $P_1 = \{s, a, z, w, u, v, b, t\}$ and $P_2 = \{s, a', w, u', v, b', t\}$. The algorithm of [1] uses the shortest path $P_a = \{s, a, v, b, t\}$ and then performs a backwards phase which never finds w again. Then the algorithm stops missing the fact that there exist two SRLG-disjoint paths.

III. Notations

We model the network as an undirected graph $G = (V, E)$, where the vertices in V represent the nodes and the edges in E represent the links. We associate a color to each SRLG. Let us denote by $\mathcal{C}$ the set of all the colors. Then a network with SRLGs is modeled by a *multi-colored graph* that is a triple

$mG = (V, E, \mathcal{C})$, where (V, E) is a graph and $\mathcal{C}$ is a set of colors assigned to E.

We denote by $E(c)$ the set of edges having color $c \in \mathcal{C}$, by $\mathcal{C}(e)$ the set of colors associated with edge $e \in E$, by CPE $= \max_{e \in E} |\mathcal{C}(e)|$ the *maximum number of colors per edge*, and by EPC $= \max_{c \in \mathcal{C}} |E(c)|$ the *maximum number of edges having the same color*. Given a vertex v, $\Gamma(v)$ denotes the set of neighbors of v and $d(v) = |\Gamma(v)|$ its *degree*. A color is *incident* to v if it is assigned to an edge incident to v. The *colored degree* of v, denoted by $d_c(v)$, is the number of colors incident to v. The *maximum degree* and the *maximum colored degree* of a graph are denoted by Δ and Δ_c, respectively.

We can now model the star property defined in the introduction as follows. A color $c \in \mathcal{C}$ is called a *star color* if all edges of $E(c)$ are incident to the same vertex. A multi-colored graph has the *star property* if it has only star colors.

Given a multi-colored graph mG and two vertices s and t, an st-path is an alternating sequence of vertices and edges, beginning with s and ending with t, in which each edge is incident to the vertex immediately preceding it and to the vertex immediately following it. A path is denoted by the sequence of vertices or by the sequence of vertices and edges. We say that two paths P_1 and P_2 are *color-disjoint* if $(\cup_{e \in P_1} \mathcal{C}(e)) \cap (\cup_{e \in P_2} \mathcal{C}(e)) = \emptyset$.

The diverse routing problem defined in the introduction consists then in finding k color-disjoint paths and can be formulated formally as follows:

Problem 1 (k-Diverse Colored st-Paths, k-DCP): Given a multi-colored graph mG and two vertices s and t, are there k color-disjoint st-paths from s to t?

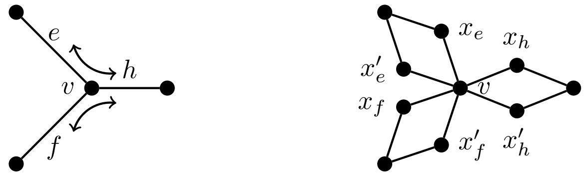
IV. NP-COMPLETENESS

In this section we will prove that the k-DCP problem is NP-complete, by using a reduction to the problem described below of finding a T-compatible path (or a path avoiding forbidden transitions), which was proved NP-complete in [11].

Let $G = (V, E)$ be an undirected graph. A transition in $v \in V$ is a pair of edges incident to v. To each vertex v we associate a set $T(v)$ of admissible (or allowed) transitions in v. We call transition system the set $T = \{T(v) \mid v \in V\}$. Let $G = (V, E)$ be a graph and T a transition system. A path $P = \{v_0, e_1, v_1, \ldots, e_k, v_k\}$ in G, with $v_i \in V$, $e_i \in E$, is said to be T-compatible if, for every $1 \leq i \leq k-1$, the pair of edges $\{e_i, e_{i+1}\}$ is an admissible transition , i.e. $\{e_i, e_{i+1}\} \in T(v_i)$. The transition system is represented in [11] by a graph also denoted as $T(v)$ where the vertices represent the edges of G incident to v and two vertices in $T(v)$ are joined if the edges they represent in G form an admissible transition in v. We now can define the T-Compatible path problem.

Problem 2 (T-Compatible path, T-CP): Given a graph $G = (V, E)$, two vertices s and t in V, and a transition system T, does G contain a T-compatible path from s to t?

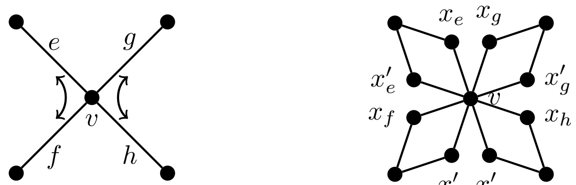
It has been proved in [11] that problem T-CP is NP-complete in the strong sense and it remains NP-complete for the family $\mathcal{G}_4$ of simple graphs where vertices s and t have



(a) v in G, $T(v) = \{\{e, h\}, \{f, h\}\}$ (b) v in H

Edge	Colors
$\{v, x_e\}$	$C_{ef}, C_{ef'}$
$\{v, x'_e\}$	$C_{e'f}, C_{e'f'}$
$\{v, x_f\}$	$C_{ef}, C_{e'f}$
$\{v, x'_f\}$	$C_{ef'}, C_{e'f'}$
$\{v, x_h\}$	no colors
$\{v, x'_h\}$	no colors

Fig. 4. Color assignment for vertices with degree 3.



(a) v in G, $T(v) = \{\{e, f\}, \{g, h\}\}$ (b) v in H

Edge	Colors
$\{v, x_e\}$	$C_{eg}, C_{eg'}, C_{eh}, C_{eh'}$
$\{v, x'_e\}$	$C_{e'g}, C_{e'g'}, C_{e'h}, C_{e'h'}$
$\{v, x_f\}$	$C_{fg}, C_{fg'}, C_{fh}, C_{fh'}$
$\{v, x'_f\}$	$C_{f'g}, C_{f'g'}, C_{f'h}, C_{f'h'}$
$\{v, x_g\}$	$C_{eg}, C_{e'g}, C_{fg}, C_{f'g}$
$\{v, x'_g\}$	$C_{eg'}, C_{e'g'}, C_{fg'}, C_{f'g'}$
$\{v, x_h\}$	$C_{eh}, C_{e'h}, C_{fh}, C_{f'h}$
$\{v, x'_h\}$	$C_{eh'}, C_{e'h'}, C_{fh'}, C_{f'h'}$

Fig. 5. Color assignment for vertices with degree 4.

degree 3 and any other vertex has degree 3 or 4 and a set of transitions $T(v)$ such that

- If $d(v) = 3$, $T(v)$ consists of two pairs of edges $\{e, h\}$ and $\{f, h\}$ where e, f and h are the 3 edges incident to v;
- If $d(v) = 4$, $T(v)$ consists of two pairs of distinct edges $\{e, f\}$ and $\{g, h\}$ where e, f, g and h are the 4 edges incident to v.

Theorem 1: The k-DCP problem is NP-complete for any fixed constant $k \geq 2$, even if all the following properties hold:

- the star property;
- the maximum degree Δ is fixed with $\Delta \geq 6 + k$;
- CPE, EPC and Δ_c are fixed with either [CPE ≥ 4, EPC ≥ 2, and $\Delta_c \geq 14+k$] or [CPE ≥ 2, EPC ≥ 4 and $\Delta_c \geq 2+k$].

Proof: We first prove the statement for $k = 2$ and then extend it for any fixed $k \geq 3$.

It is easy to see that the problem is in NP.

Given an instance G, s, t, T with G in the family $\mathcal{G}_4$ of the T-CP problem, we define an instance of 2-DCP as follows. We associate to G a multi-colored graph $H = (V_H, E_H, \mathcal{C})$ where:

- For every edge $e = \{u, v\}$ in G, we associate in H two paths of length 2: $\{u, x_e, v\}$ and $\{u, x'_e, v\}$. H has then $|V(G)| + 2|E(G)|$ vertices and $4|E(G)|$ edges.
- We assign the colors to edges incident to a vertex v in H as follows:
  No colors are assigned to the edges incident to t.
  For each pair of edges e and f incident to s in G, such that $e \neq f$, we will use 4 colors $C_{ef}, C_{ef'}, C_{e'f}$ and $C_{e'f'}$. We assign colors C_{ef} and $C_{ef'}$ to the edge $\{s, x_e\}$; colors $C_{e'f}$ and $C_{e'f'}$ to the edge $\{s, x'_e\}$, colors C_{ef} and $C_{e'f}$ to the edge $\{s, x_f\}$ and colors $C_{ef'}$ and $C_{e'f'}$ to the edge $\{s, x'_f\}$.
  For each $v \neq s, t$, and for each pair of edges e and f incident to v in G such that $e \neq f$ and $\{e, f\}$ is not an admissible transition (i.e. $\{e, f\} \notin T(v)$), we assign colors C_{ef} and $C_{ef'}$ (resp. $C_{e'f}$ and $C_{e'f'}$) to the edge $\{v, x_e\}$ ($\{v, x'_e\}$, resp.), and colors C_{ef} and $C_{e'f}$ (resp. $C_{ef'}$ and $C_{e'f'}$) to the edge $\{v, x_f\}$ (resp. $\{v, x'_f\}$). As each vertex has either degree 3 or 4, two cases can occur:
  (i) If $d(v) = 3$, let e, f and h be the 3 edges incident to v and let $T(v) = \{\{e, h\}, \{f, h\}\}$, then edge $\{v, x_e\}$ receives the 2 colors C_{ef} and $C_{ef'}$, edge $\{v, x_h\}$ receives no colors and so on (see Figure 4 for the complete list of colors assigned to all the edges).
  (ii) If $d(v) = 4$, let e, f, g and h be the 4 edges incident to v and $T(v) = \{\{e, f\}, \{g, h\}\}$. Then edge $\{v, x_e\}$ receives the 4 colors $C_{eg}, C_{eg'}, C_{eh}, C_{eh'}$. Similarly, edge $\{v, x_f\}$ receives the 4 colors $C_{fg}, C_{fg'}, C_{eh}$ and $C_{fh'}$ (see Figure 5 for the complete list of colors assigned to all the edges).

The transformation is polynomial time computable and the star property holds. Moreover, note that each edge has at most 4 colors, each color is associated with two edges, the degree of each vertex is at most 8 and the color degree is at most 16. It follows that CPE $= 4$, EPC $= 2$, $\Delta = 8$, and $\Delta_g = 16$.

To prove the theorem, we will use the following properties.

Property 1: Given an edge e incident to s in G, the edge $\{s, x_e\}$ in H shares a color with all the other edges incident to s but $\{s, x'_e\}$. Said otherwise, the only pair of edges incident to s having no color in common are of the form $\{\{s, x_e\}, \{s, x'_e\}\}$ for some e.

Property 2: If $v \neq s, t$, $d(v) = 3$ and $T(v) = \{\{e, h\}, \{f, h\}\}$, then two edges incident to v share a color if and only if one is of the form $\{v, x_e\}$ or $\{v, x'_e\}$ and the other is of the form $\{v, x_f\}$ or $\{v, x'_f\}$.

Property 3: If $v \neq s, t$, $d(v) = 4$ and $T(v) = \{\{e, f\}, \{g, h\}\}$, then two edges incident to v share a color if and only if one is of the form $\{v, x_e\}$, $\{v, x'_e\}$, $\{v, x_f\}$ or $\{v, x'_f\}$ and the other is of the form $\{v, x_g\}$, $\{v, x'_g\}$, $\{v, x_h\}$ or $\{v, x'_h\}$.

We first show that if there exists a T-compatible path in G, then there exist two color-disjoint paths in H. Let $P = \{s, e_1, v_1, \ldots, e_p, v_p, e_{p+1}, t\}$ be a T-compatible path from s to t in G. Then $Q = \{s, x_{e_1}, v_1, \ldots, x_{e_p}, v_p, x_{e_{p+1}}, t\}$ and $Q' = \{s, x'_{e_1}, v_1, \ldots, x'_{e_p}, v_p, x'_{e_{p+1}}, t\}$ are two color-disjoint paths in H. In fact, any edge $\{v, x_e\}$ has no color in common with $\{v, x'_e\}$ by Properties 1, 2, or 3.

Conversely, we now show that if there exist two color-disjoint paths in H, then there exists a T-compatible path in G. Let $Q = \{s, x_1, v_1, \ldots, x_p, v_p, x_{p+1}, t\}$ and $Q' = \{s, y_1, u_1, \ldots, y_{p'}, u_{p'}, y_{p'+1}, t\}$ be two color-disjoint paths in H. We prove by induction on $i \in \{1, \ldots, p+1\}$, that $\{x_i, y_i\} = \{x_{e_i}, x'_{e_i}\}$, $v_i = u_i$ and $p = p'$.

For $i = 1$, by Property 1, $\{s, x_1\}$ and $\{s, y_1\}$ have no color in common only if $\{x_1, y_1\} = \{x_e, x'_e\}$ for an edge e incident to s and then $v_1 = u_1$.

Let us suppose that it is true until $i = l$ and prove it for $i = l + 1$. In $v_l = u_l$, let the two edges used by Q and Q' be $\{x_{e_l}, v_l\}$ and $\{x'_{e_l}, v_l\}$.

If $d(v_l) = 3$, we distinguish two cases:

(i) e_l belongs to only one edge of $T(v_l)$ say $\{e_l, h_l\}$ and the paths Q and Q' being color-disjoint can only use the edges $\{v_l, x_{h_l}\}$ and $\{v_l, x'_{h_l}\}$.

(ii) e_l belongs to two edges in $T(v_l)$, $\{e_l, f_l\}$ and $\{e_l, h_l\}$. If one path uses the edge $\{v_l, x_{h_l}\}$ ($\{v_l, x'_{h_l}\}$, resp.) the other path cannot use the edge $\{v_l, x_{f_l}\}$ or $\{v_l, x'_{f_l}\}$ by Property 2, it has then to use edge $\{v_l, x'_{h_l}\}$ ($\{v_l, x_{h_l}\}$, resp.).

If $d(v_l) = 4$, by Property 3, the only possibility as Q and Q' are color-disjoint is that they use the edges $\{v_l, x_{e_{l+1}}\}$ and $\{v_l, x'_{e_{l+1}}\}$ where $\{e_l, e_{l+1}\} \in T(v_l)$ and so the statement is true for $i = l + 1$.

It follows that the path $P = \{s, e_1, v_1, \ldots, e_p, v_p, e_{p+1}, t\}$ satisfies $\{e_i, e_{i+1}\} \in T(v_i)$ for every $i \in \{1, p\}$ and then it is T-compatible.

To show that the problem remains NP-complete even for fixed CPE ≥ 2, EPC ≥ 4 and $\Delta_c \geq 4$, it is enough to modify the above transformation by using a different color assignment (details in [12]). To extend the proof to the case of $k \geq 3$, it is enough to add $k - 2$ paths of length 2 from s to t $P_i = \{s, w_i, t\}$, $i = 3, 4, \ldots k$, with a new color assigned to each edge $\{s, w_i\}$. These paths are pairwise color-disjoint and also color-disjoint from the two paths in the above transformation. Moreover, this assignment does not change CPE and EPC and increases Δ and Δ_c by $k - 2$. ■

V. Polynomial cases

In this section we give polynomial time algorithms for k-DCP for some important special cases.

A. Bounded number of colors

In this section, we give an algorithm to find k color-disjoint paths in the special case where the number $|\mathcal{C}|$ of colors in the network is bounded by a constant, i.e. $|\mathcal{C}| = O(1)$. We observe that such an algorithm works for every graph topology and even if the star property does not hold.

We will reduce our problem to the k-Set Packing problem.

Problem 3 (k-Set Packing): Given a set X and a collection $\mathcal{S}$ of subsets of X, is there a collection of disjoint sets $\mathcal{S}' \subseteq \mathcal{S}$ such that $|\mathcal{S}'| = k$?

A subset $S \subseteq \mathcal{C}$ of colors will be called *realizable*, if the subgraph G_S induced by the edges whose colors are all in S (i.e. edges e such that $\mathcal{C}(e) \subseteq S$) contains a path from s to t. Note that such a path uses only colors of S.

The idea of the algorithm is to enumerate all the realizable subsets of $\mathcal{C}$ and then find k disjoint realizable subsets by using an exact algorithm for the k-Set Packing. As the size of $\mathcal{C}$ is constant, the computational time required by such algorithm is polynomial.

The details of the algorithm along with its correctness and complexity are given in the next theorem.

Theorem 2: If $|\mathcal{C}| = O(1)$, there exists an algorithm that solves k-DCP with $O(|V| + |E|)$ computational time.

Proof: Let $X = \mathcal{C}$ and $\mathcal{S}$ be the family of realizable subsets of colors. Then there exist a collection of k disjoint sets $\mathcal{S}' \subseteq \mathcal{S}$ if and only if there exist k color disjoint paths from s to t. Indeed, to each subset S' of $\mathcal{S}'$ is associated a path using uniquely colors of S' (as S' is realizable) and two disjoint subsets correspond to two color disjoint paths.

Determining if a subset of colors is realizable takes $O(|V| + |E|)$ computational time. Furthermore, it is known that there exist polynomial algorithms to solve the k-Set Packing problem when the size of X is bounded. For example the exact algorithm in [13] has complexity $O(|\mathcal{S}|2^{|X|}|X|^{O(1)})$ and so, as $|X| = |\mathcal{C}| = O(1)$ and $|\mathcal{S}| \leq 2^{|\mathcal{C}|} = O(1)$, it requires $O(1)$ time. Therefore, finding k color disjoint paths requires $O(|V| + |E|)$ overall time. ∎

B. Bounded degree

In this section, we give algorithms for finding k color-disjoint paths when $\Delta < 4$ and for finding 2 color-disjoint paths when $\Delta = 4$. First, note that the maximum number of color-disjoint paths in a graph is upper bounded by Δ.

If $\Delta \leq 2$ the problem is trivial as the graph is either a path or a cycle. In the first case, it always exists only one path from s and t. In the second case, the only vertices where the two possible paths can share colors are s and t and hence it is enough to check if the two edges incident to s (and t) are color-disjoint.

If $\Delta \geq 3$, observe that if two different paths share an internal vertex of degree 3, they necessarily share also an edge and hence all the colors of that edge and so they cannot be color-disjoint. Furthermore, if two paths are color-disjoint the colors of their first edges should be different and also the colors of their last edges should be different.

When $\Delta = 3$, there are 3 color-disjoint paths if and only if G has 3 vertex-disjoint paths between s and t (i.e. G is 3-connected) and the 3 first edges of these paths have different colors and also the 3 last edges. That can be checked in $O(|V| + |E|)$ time: constant time for checking the color disjointness of the 3 first (last) edges, and $O(|V| + |E|)$ time for testing is G is 3-connected (see [14]).

For $k = 2$ and $\Delta = 3$ or 4 we have the following theorem:

Algorithm 1: Solving 2-DCP when $\Delta = 3, 4$.

1 **foreach** *admissible graph* $G(s_i, s_j, t_{i'}, t_{j'})$ **do**
2   **if** *there exist 2 vertex-disjoint paths form s to t in* $G(s_i, s_j, t_{i'}, t_{j'})$ **then**
3   There exist two color-disjoint paths from s to t in G;
4   **else**
5   **if** *all the cut-vertices in* $G(s_i, s_j, t_{i'}, t_{j'})$ *have degree 4* **then**
6   **if** *for each cut vertex v, e and f denote the edges of the connected component of* $G(s_i, s_j, t_{i'}, t_{j'}) - v$ *containing s and e' and f' the edges of the connected component containing t, we have* $\mathcal{C}(e) \cap \mathcal{C}(f) = \emptyset$, $\mathcal{C}(e') \cap \mathcal{C}(f') = \emptyset$ *and either* $[\mathcal{C}(e) \cap \mathcal{C}(e') = \emptyset$ *and* $\mathcal{C}(f) \cap \mathcal{C}(f') = \emptyset]$ *or* $[\mathcal{C}(e) \cap \mathcal{C}(f') = \emptyset$ *and* $\mathcal{C}(f) \cap \mathcal{C}(e') = \emptyset]$ **then**
7   There exist two color-disjoint paths from s to t in G;
8 No 2 color-disjoint paths from s to t exist in G;

Theorem 3: Algorithm 1 solves 2-DCP in graphs with the star property and $\Delta = 3$ or 4 in time $O(|V| + |E|)$.

Proof: We say that a pair $\{s_i, s_j\}$ of neighbors of s in G is admissible, if they have different colors, i.e. $\mathcal{C}((s, s_i)) \cap \mathcal{C}((s, s_j)) = \emptyset$ and similarly we say that $\{t_{i'}, t_{j'}\}$ is an admissible pair of neighbors of t if they have different colors. Then we consider the admissible graph $G(s_i, s_j, t_{i'}, t_{j'})$ obtained from G by deleting the edges $\{s, s_\ell\}$ with $\ell \neq i, j$ and the edges $\{t_{\ell'}, t\}$ with $\ell' \neq i', j'$.

The correctness of Algorithm 1 follows from the observations given above.

By definition, if there exist two vertex disjoint paths from s to t in $G(s_i, s_j, t_{i'}, t_{j'})$ the first edges and last edges of such paths have different colors and so we conclude that there are 2 color-disjoint paths (lines $2, 3$).

Otherwise if there exists a cut vertex v of degree 3, we cannot have color disjoint paths containing this vertex. That is in particular the case when $\Delta = 3$. Let $\Delta = 4$; if in v one path uses edge e and e' (resp e and f'), the other path uses necessarily f and f' (resp f and e') and the conditions on colors are necessary and sufficient for the color disjointness of the paths at v (the center of the colors used in v). We have at most 6 admissible pairs of neighbors of s (resp. t) and so at most 36 graphs to consider. For each graph we have to check if it is 2-connected (that can be done in $O(|V| + |E|)$ time) and if it is not 2-connected to verify coloring conditions at each cut vertex that can be done in constant time for a given vertex and so with an $O(|V|)$ complexity. ∎

Note that Algorithm 1 cannot be extended neither to find 3 or 4 color-disjoint paths on a graphs with $\Delta = 4$ nor to the case of $\Delta \geq 5$.

C. Directed acyclic graphs

The Directed Acyclic Graphs (DAGs) being the simplest non trivial case of directed graphs, we give, in this section, an algorithm for finding k color-disjoint paths in a DAG with the star property. Note that we are using a model with a directed graph obtained from the graph model by assigning an orientation to each arc. As each color is a star color we can associate to each color c its center v defined as the common vertex to all arcs with color c. If the color has only one occurrence we choose arbitrarily as associated center one of the end vertices of the arc containing this color. We will say that the color c is centered in v.

The algorithm given in the proof of the next theorem uses ideas of [15], in particular that of layered digraph and a construction similar to that used to find a polynomial time algorithm for disjoint paths with forbidden pairs (Theorem 6 of [15]).

Definition 1 (Layered digraph): A directed graph $G = (V, E)$ is layered if there is a layering function $l : V \rightarrow [0, 1, \ldots, (|V| - 1)]$ such that for every arc $(u, v) \in E$, $l(v) = l(u) + 1$. We say that vertex u is in layer $l(u)$ and arc (u, v) is in layer $l(u)$. Layered directed graphs are acyclic.

Note that the computational time required by the algorithm is $O(|V||E|^{2k})$ which is polynomial only when k is a fixed constant. However, in the next section we will show that in DAGs it is not possible to find an algorithm requiring a computational time polynomial in k. In detail, it is not possible to solve k-DCP in DAGs with the star property in less then $(|V| + |E|)^{O(k)}$ time.

Theorem 4: There exists an algorithm that solves k-DCP in a DAG with the star property in time $O(|V||E|^{2k})$.

Proof: Let D be a multicolored DAG and s and t be two given vertices. As we want to find dipaths from s to t, we can delete the vertices not on a dipath from s to t and so we suppose in what follows that D is this reduced DAG. Now s is the unique vertex with no predecessor and t the unique vertex with no successor. We first associate to a multicolored DAG D a multicolored layered DAG LD as follows. We denote by $\Gamma^-(v)$ the set of vertices preceding v, i.e; vertices u such that $(u, v) \in E$. We topologically order the vertices of the DAG D, with the ordering function $l : V \rightarrow \mathbb{N}$ as follows:

$$l(v) = \begin{cases} 0 & \text{when } v = s, \\ 1 + \max_{u \in \Gamma^-(v)} l(u) & \text{otherwise.} \end{cases}$$

In such an ordering t has the maximum level as there is a dipath from any vertex to t in the reduced digraph. In the graph D of Figure 6(a), we have $l(u_1) = 1, l(u_2) = 2, l(u_3) = 3, l(u_4) = l(u_5) = l(u_6) = 4, l(t) = 5$.

Now we replace every arc (u, v), such that $l(v) > l(u) + 1$, with a directed path P_{uv} from u to v of length $l(v) - l(u)$. We assign to the first arc of the dipath P_{uv} the colors of the arc (u, v) centered in u and to the last arc of the dipath P_{uv} the colors of the arc (u, v) centered in v. Doing so we have obtained from D a layered digraph LD, such that there exist k color disjoint paths in D from s to t if and only if there exist k color-disjoint paths in LD from s to t. In Figure 6(b), we indicate the layered graph LD obtained from the digraph D of Figure 6(a). We have given a name to each arc with a lower index indicating the level of the arc; we also indicate in () the colors attributed to each arc. For example the arc (s, u_3) which had colors c_1, c_4 has been replaced by a path with 3 arcs: e_0 at level 0 which gets the color c_1 (centered at s), e_1 and e_2 at level 2 which gets the color c_4 (centered at u_3).

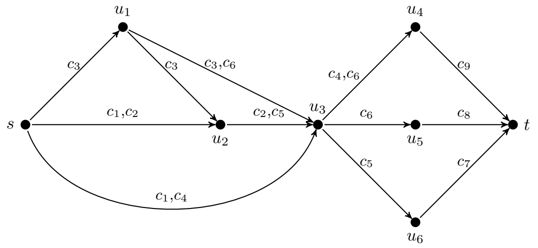
Therefore, in what follows we consider a layered digraph D with two specific vertices s and t. To D we will associate a digraph H with two specific vertices s and t, such that there exist k color disjoint dipaths in D from s to t if and only if there exists in H a dipath from s to t. For the ease of reading we first give the transformation for $k = 2$.

H has as vertices s, t, and all the pairs of arcs $\{e_i, f_i\}$ having the same level i, $0 \leq i \leq l(t) - 1$, and no color in common. Now we join, by an arc in H, s to all the vertices (pairs) $\{e_0, f_0\}$. Similarly we join a vertex $\{e_{l(t)-1}, f_{l(t)-1}\}$ in H by an arc to t. Finally, for $0 \leq i \leq l(t) - 2$, we join in H a vertex $\{e_i, f_i\}$ to a vertex $\{e_{i+1}, f_{i+1}\}$ if in G we have the following properties:

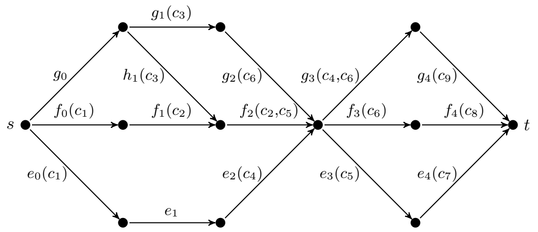
- the terminal vertex of e_i (resp. f_i) is the initial vertex of e_{i+1} (resp. f_{i+1}) and
- the set of colors of $\mathcal{C}(e_i) \cup \mathcal{C}(e_{i+1})$ centered at their common vertex u_i is disjoint from the set of colors of $\mathcal{C}(f_i) \cup \mathcal{C}(f_{i+1})$ centered at their common vertex v_i.

Note that if $u_i \neq v_i$ the second property is always satisfied as the colors are star colors. Figure 6(c) indicates the graph H obtained from LD of Figure 6(b). For example, consider the pair of arcs $\{e_2, f_2\}$, which have different colors. Consider two paths ending in u_3 with e_2 and f_2; the path ending with e_2 cannot continue with e_3 which has a color in common with f_2; similarly the path ending in f_2 cannot continue with g_3 so we can have at u_3 either subpaths (e_2, f_3) and (f_2, e_3) or (e_2, g_3) and (f_2, e_3). In the graph H it means that the pair $\{e_2, f_2\}$ is joined to the pairs $\{e_3, f_3\}$ and $\{e_3, g_3\}$.

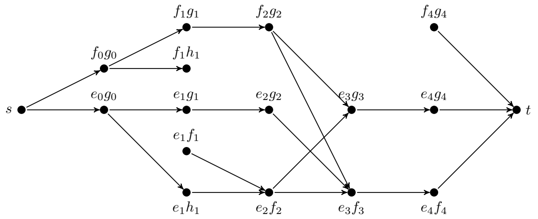
The transformation is done in polynomial time. Now the existence in D of two disjoint colored dipaths $P = (s, e_0, u_0, e_1, \ldots, e_{l(t)-2}, u_{l(t)-1}, e_{l(t)-1}, t)$ and $Q = (s, f_0, v_0, f_1, \ldots, f_{l(t)-2}, v_{l(t)-1}, f_{l(t)-1}, t)$ implies the existence in H of a dipath from s to t namely $PQ = (s, \{e_0, f_0\}, \{e_1, f_1\}, \ldots, \{e_{l(t)-1}, f_{l(t)-1}\}, t)$ and vice versa. Indeed s is joined in H to $\{e_0, f_0\}$ if and only if e_0 and f_0 have no color in common; similarly $\{e_{l(t)-1}, f_{l(t)-1}\}$ is joined to t if and only if $e_{l(t)-1}$ and $f_{l(t)-1}$ have no color in common. Furthermore $\{e_i, f_i\}$ is joined in H to $\{e_{i+1}, f_{i+1}\}$ if and only if P and Q are dipaths, i.e. the terminal vertex of e_i (resp. f_i) is the initial vertex of e_{i+1} (resp. f_{i+1}) and the the second condition on colors is satisfied. In the example of Figure 6(c), H has many dipaths from s to t. For example to the dipath $P = (s, \{e_0, g_0\}, \{e_1, h_1\}, \{e_2, f_2\}, \{e_3, g_3\}, \{e_4, g_4\}, t)$ are associated in LD the two color-disjoint dipaths $P_1 = (s, e_0, e_1, e_2, g_3, g_4, t)$ and $P_2 = (s, g_0, h_1, f_2, f_3, f_4, t)$, and in D the two dipaths s, u_3, u_4, t and s, u_1, u_2, u_3, u_6, t.



(a) Example of muticolored DAG D



(b) LD obtained after the layering of D



(c) The graph H associated to LD

Fig. 6. Transformations for the DAG.

The transformation can be generalized to any $k \geq 2$ but instead of using as vertices of H the pairs of arcs at the same level, we use the k-element subsets of arcs at the same level.

Deleting the vertices and arcs not on a path from s and t can be done in $O(|V| + |E|)$ time. Finding the levels can be done in $O(|V| + |E|)$ time. The transformation to a layered DAG can be done in $O(|V||E|)$ time. Note that each arc is replaced by a path containing at most one arc of each level and so the number of arcs at a given level is at most $|E|$. The graph H obtained in the transformation has therefore at most $l(t)|E|^k$ vertices and $l(t)|E|^{2k}$ edges and so we get the complexity of the theorem as $l(t) \leq |V|$ ■

Remark 1: The previous algorithm can be adapted to find a minimum cost pair of color-disjoint paths by applying the following modifications. Let us consider a weight function on the arcs of a graph D. We attribute the original weight of the arc(u, v) to the first arc of the path replacing it in LD. Then in H we attribute to the edge joining s to $\{e_0, f_0\}$ the sum of the weights of e_0 and f_0, and to the edge joining $\{e_i, f_i\}$ to $\{e_{i+1}, f_{i+1}\}$ the sum of the weights of $\{e_{i+1}$ and $f_{i+1}\}$. With these modifications, the shortest path in H corresponds to the optimal pair of color-disjoint paths in D.

Remark 2: We can also use the previous algorithm to find a pair of color-disjoint paths with the minimum total number of colors by applying the following modifications. In H, we assign to the edge joining $\{e_i, f_i\}$ to $\{e_{i+1}, f_{i+1}\}$ a weight equal to $|\mathcal{C}((e_i) \cup \mathcal{C}(e_{i+1})) \cup \mathcal{C}((f_i) \cup \mathcal{C}(f_{i+1}))|$. The shortest path in the weighted graph H will correspond then to the pair of color-disjoint paths with the minimum number of colors.

VI. Maximum number of color-disjoint paths

We aim at finding the maximum number of color-disjoint paths, formally:

Problem 4 (Max Diverse Colored st-Paths, `MDCP`*):* Given a multi-colored graph mG and two vertices s and t, find the maximum number of color-disjoint st-paths.

The corresponding decision problem is the following.

Problem 5 (Diverse Colored st-Paths, `DCP`*):* Given a multi-colored graph mG, two vertices s and t, and an integer k, are there k color-disjoint st-paths?

We show that `DCP` is NP-complete in the strong sense and that `MDCP` is hard to approximate within some bound, even with some restrictions on CPE and EPC. The proofs for this section are omitted and can be found in [12].

Theorem 5: `DCP` is NP-complete in the strong sense, even in DAGs, if the star property holds and CPE and EPC are fixed with CPE ≥ 3 and EPC ≥ 3 or CPE ≥ 6 and EPC ≥ 2.

In the next theorem we give an approximation factor preserving reduction from Maximum Set Packing (`MSP`).

Definition 2 (Maximum Set Packing, `MSP`*):* Given a set X and a collection $\mathcal{S}$ of subsets of X, find the maximum cardinality set packing, i.e., a collection of disjoint sets $\mathcal{S}' \subseteq \mathcal{S}$ such that $|\mathcal{S}'|$ is maximized.

Theorem 6: `MDCP` is as hard to approximate as `MSP`, with $|V| = |\mathcal{S}|$, even in DAGs and if the star property holds.

It has been proven (see [16]–[18]) that problem `MSP` is not approximable within $O(|\mathcal{S}|)$. Moreover, if the cardinality of all sets in $\mathcal{S}$ is upper bounded by a constant $s \geq 3$ or the number of occurrences in $\mathcal{S}$ of any element is upper-bounded by a constant $B \geq 2$, then the problem is APX-complete.

Corollary 1: Unless $P = NP$, `MDCP` cannot be approximate within $O(|V|)$ even if EPC is fixed, EPC ≥ 2. Moreover, it is APX-hard if CPE is fixed, CPE ≥ 3. These inapproximability results hold even in DAGs with the star property.

The next results are expressed in terms of *parameterized complexity* [19]. Recall that a problem is *fixed parameter tractable*, and so is *in FPT*, if it can be solved in time $O(f(k) \cdot |V|^{O(1)})$ where the function f is polynomial and depends only on the input parameter k. A problem is $W[1]$-hard if a problem of the class $W[1]$ (e.g., deciding if the graph contains a clique of size k) can be reduced to it in FPT-time.

Theorem 7: `MDCP` is $W[1]$-hard where the parameter is the number of color-disjoint paths, even with the star property.

Corollary 2: `MDCP` is not in FPT, unless $FPT = W[1]$.

As $|V|$ is an upper bound to any optimal solution of `MDCP`, Corollary 1 implies that, if CPE is unbounded, we cannot find an approximation factor better than $|V|/c$, where c is a constant. This corresponds to find c color-disjoint st-paths, where c is a given constant. In other words, the only way to cope with this problem (in a general graph with the star property) is to find a *fixed* number of color-disjoint paths.

Graph	Δ	**EPC**	**CPE**	k-DCP	MDCP
Undirected	≥ 8	≥ 2	≥ 4	Strongly NP-hard for $\Delta \geq 6 + k$	Not approximable within $O(\lvert V\rvert)$
		≥ 4	≥ 2		
Directed	≤ 3	any	any	Solvable in $O(\lvert V\rvert + \lvert E\rvert)$ time	Optimum in $O(\lvert V\rvert + \lvert E\rvert)$ time
	$= 4$	any	any	Solvable in $O(\lvert V\rvert + \lvert E\rvert)$ time for $k = 2$	2-approximation in $O(\lvert V\rvert + \lvert E\rvert)$ time
$\lvert\mathcal{C}\rvert = O(1)$, even without star	any	any	any	Solvable in $O(\lvert V\rvert + \lvert E\rvert)$ time	Optimum in $O((\lvert V\rvert + \lvert E\rvert) \log \Delta)$ time
DAG	unbounded	≥ 3	≥ 3	Solvable in $O(\lvert V\rvert\lvert E\rvert^{2k})$ time	Strongly NP-hard
		≥ 2	≥ 6		
		≥ 2	unbounded		Not approximable within $O(\lvert V\rvert)$
		unbounded	3		APX-hard
		unbounded	unbounded		$W[1]$-hard

TABLE I
SUMMARY OF COMPLEXITY RESULTS.

However, Theorem 7 implies that no FPT algorithms can be devised, unless $FPT = W[1]$; so it cannot exist an algorithm which finds k color-disjoint paths in $O(2^k \cdot poly(|V| + |E|))$ time. Moreover, Theorem 1 shows that even finding a fixed number $k \geq 2$ of color-disjoint paths is NP-complete. This implies that it is impossible to devise an algorithm which finds k color-disjoint paths in $O((|V| + |E|)^{O(poly(k))})$ time. Note also that the transformation used in Theorems 5–7 makes use of simple graphs that can be easily turned into DAGs. This implies that no algorithm can solve k-DCP in DAGs in less then $(|V| + |E|)^{O(k)}$ time and then the algorithm in Section V-C is basically the best possible w.r.t this lower bound.

The algorithm in Section V-A can be used to find an exact algorithm for MDCP when $|\mathcal{C}| = O(1)$. In fact, it is enough to search for the maximum k for which such algorithm returns k color-disjoint paths. As the maximum number of color-disjoint paths is upper bounded by Δ, by using a binary search such an approach requires to apply at most $\log \Delta$ times the algorithm of Section V-A. The following corollary follows.

Corollary 3: There exists an algorithm that solves MDCP if $|\mathcal{C}| = O(1)$ in time $O((|V| + |E|) \log \Delta)$.

We can use the algorithms for bounded degree presented in Section V-B for solving MDCP when $\Delta \leq 4$. We get a polynomial algorithm for $\Delta \leq 3$ and a 2-approximation algorithm for $\Delta = 4$ as the maximum number of color-disjoint paths is upper bounded by $\Delta = 4$. The following corollaries follow.

Corollary 4: If $\Delta \leq 3$, there exists an algorithm that solves MDCP in a graph, with the star property, in time $O(|V| + |E|)$.

Corollary 5: If $\Delta = 4$, there exists a 2-approximation algorithm for MDCP in a graph, with the star property, in time $O(|V| + |E|)$.

VII. CONCLUSION

The results, presented in this paper, give an almost complete characterization of the problem of finding SRLG-disjoint paths in networks with SRLGs satisfying the star property. We summarize them all in Table I.

REFERENCES

[1] X. Luo and B. Wang, "Diverse routing in WDM optical networks with shared risk link group (SRLG) failures," in *Proc. DRCN*. IEEE, Oct. 2005, pp. 445–452.

[2] J. Doucette and W. Grover, "Shared-risk logical span groups in span-restorable optical networks: Analysis and capacity planning model," *Photonic Network Communications*, vol. 9, no. 1, pp. 35–53, 2005.

[3] D. Coudert, P. Datta, S. Perennes, H. Rivano, and M.-E. Voge, "Shared risk resource group: Complexity and approximability issues," *Parallel Processing Letters*, vol. 17, no. 2, pp. 169–184, Jun. 2007.

[4] D. Coudert, S. Perennes, H. Rivano, and M.-E. Voge, "Shared risk resource groups and colored graph: Polynomial and FPT optimisation problems," Submitted.

[5] J. Hu, "Diverse routing in mesh optical networks," *IEEE Transactions on Communications*, vol. 51, pp. 489–494, Mar. 2003.

[6] G. Ellinas, E. Bouillet, R. Ramamurthy, J.-F. Labourdette, S. Chauduri, and K. Bala, "Routing and restoration architectures in mesh optical networks," *Optical Networks Magazine*, pp. 91–106, Jan. 2003.

[7] G. Li, C. Kalmanek, and R. Doverspike, "Fiber span failure protection in mesh optical networks," *Optical Networks Magazine*, pp. 21–31, May 2002.

[8] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: a new approach to the design of WDM-based networks," *IEEE Journal of Selected Areas in Communication*, vol. 20, no. 4, pp. 800–809, May 2002.

[9] P. Datta and A. Somani, "Graph transformation approaches for diverse routing in shared risk resource group (SRRG) failures," *Computer Networks*, vol. 52, no. 12, pp. 2381–2394, Aug. 2008.

[10] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Axademic Publishers, 1998.

[11] S. Szeider, "Finding paths in graphs avoiding forbidden transitions," *Discrete Applied Mathematics*, vol. 126, no. 2–3, pp. 261–273, 2003.

[12] J.-C. Bermond, D. Coudert, G. D'Angelo, and F. Z. Moataz, "Diverse Routing with the star property," INRIA, Research Report RR-8071, Sep. 2012. [Online]. Available: http://hal.inria.fr/hal-00733869

[13] A. Björklund, T. Husfeldt, and M. Koivisto, "Set partitioning via inclusion-exclusion," *SIAM J. Comput.*, vol. 39, no. 2, pp. 546–563, 2009.

[14] J. E. Hopcroft and R. E. Tarjan, "Dividing a graph into triconnected components," *SIAM J. Comput.*, vol. 2, no. 3, pp. 135–158, 1973.

[15] V. T. Chakaravarthy, "New results on the computability and complexity of points–to analysis," in *ACM SIGPLAN-SIGACT symposium on principles of programming languages (POPL)*. ACM, 2003, pp. 115–125.

[16] G. Ausiello, A. D'Atri, and M. Protasi, "Structure preserving reductions among convex optimization problems." *J. Comput. Syst. Sci.*, pp. 136–153, 1980.

[17] P. Berman and T. Fujito, "On approximation properties of the independent set problem for degree 3 graphs," in *4th International Workshop on Algorithms and Data Structures (WADS)*, ser. Lecture Notes in Computer Science. Springer, 1995, vol. 955, pp. 449–460.

[18] V. Kann, "Maximum bounded 3-dimensional matching is max snp-complete," *Inf. Process. Lett.*, vol. 37, no. 1, pp. 27–35, Jan. 1991.

[19] R. G. Downey and M. R. Fellows, *Parameterized Complexity*. Springer, 1999.