Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics

Survivable Optical Network Design with Unambiguous Shared Risk Link Group Failure Localization

Péter Babarczi

PhD Dissertation

Advisor:

Dr. János Tapolcai

High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

External advisor:

Dr. Pin-Han Ho

Department of Electrical and Computer Engineering
University of Waterloo, ON, Canada

Budapest, Hungary
2011.

Abstract

The ever increasing thirst for bandwidth and the strict reliability and timing requirements of applications requires new network resilience methods in the fault management of optical backbone networks. It is particularly critical when an all-optical backbone is in place owing to its high data rate along each fiber and transparency in the data plane. In order to ensure an infrastructure of providing services requiring high Quality of Service (QoS), precise modeling of core optical mesh networks is crucial. For this purpose, the Shared Risk Link Group (SRLG) concept is introduced for modeling physical and geographical dependency among seemingly unrelated link failures.

There have been numerous studies showing various benefits of (1+1) dedicated protection, which is the widely deployed technology in current optical backbone. However, all of the methods seek the solution of the routing problem in a pre-defined form, which is rather inefficient to find the most flexible routing structure to the QoS needs of the customers. In the first part of this thesis, we introduce a mathematical model which generalizes all previously reported dedicated protection schemes, called Generalized Dedicated Protection (GDP). The routing solutions required to be resilient and robust against all failures in a list of SRLGs defined by the network operator for each QoS class based on operational premises. Based on the equipments available at the network nodes various aspects of the problem is investigated. To further generalize the GDP problem, the applicability of network coding is investigated, and shown to be efficient in practical scenarios. Owing to the mathematical operations required, network coding necessarily incurs some additional cost. However, the complexity of GDP with network coding is polynomial-time, thus, makes the proposed method for on-line routing possible.

In the second part of this thesis, as the key point of rapid optical layer restoration, unambiguous failure localization using supervisory lightpaths in all-optical networks is discussed. Using the most flexible structures (called monitoring-trails), the M-trail Allocation Problem (MAP) is introduced in order to minimize the signaling complexity of failure localization. Sufficient conditions on code assignment for multi-link SRLGs are presented, which can be used in various research fields as the basis of algorithm design. The algorithms of the thesis covers various important application scenarios in the fault-management of all-optical networks, which were not addressed efficiently in the literature previously, i.e. unambiguously localizing SRLGs with heterogeneous number of links contained, including the node failure scenario. The impact of the results is demonstrated with publications in the most prestigious journals in our research field.

Kivonat

Az alkalmazások növekvő sávszélesség igényének, valamint a magas megbízhatósági és szigorú időzítési követelményeinek kielégítésére új hibamenedzsment eljárások kidolgozására van szükség optikai gerinchálózatokban. A helyzet tisztán optikai hálózatok esetén még kritikusabb a hatalmas adatsebesség és az adat sík átlátszósága miatt. Annak érdekében, hogy a hálózati infrastruktúra alkalmas legyen magas szolgáltatás minőségi (QoS) követelmények teljesítésére, a meghibásodások pontos modellezése elengedhetetlen. Ilyenkor a hálózatok modellezésekor a közös kockázatú csoportok (SRLG) alkalmazására van szükség a hibamenedzsment tervezésekor, mely a látszólag független linkek fizikai és földrajzi összefüggőségeit is figyelembe veszi.

A hozzárendelt (1+1) védelmi algoritmusok előnyeiknek köszönhetően a legelterjedtebb gerinchálózati védelmi megoldásokká váltak a gyakorlatban. A jelenlegi módszerek hátránya viszont, hogy előre megadják a védelmi megoldás formáját, mely lehetetlenné teszi a felhasználó QoS igényeihez leginkább illeszkedő struktúra kiválasztását. A disszertáció első részében egy általános matematika modellt (általános hozzárendelt védelem, GDP) vezetek be, amely általánosítja az irodalomban ajánlott hozzárendelt védelmi módszereket. Az útvonalválasztási feladat megoldásai ellenállóak és robusztusak a QoS osztályhoz rendelt valamennyi, a hálózat operátor által kialakított SRLG listában található hibák ellen. A hálózat csomópontjaiban rendelkezésre álló eszközöknek megfelelően a feladat több megközelítését is megvizsgálom. Tovább általánosítva a GDP problémát megmutatom, hogy a hálózati kódolás (network coding) hatékonyan alkalmazható a feladatra. Annak ellenére, hogy a hálózati kódolás extra költséget visz a rendszerbe, az útvonalválasztási feladat polinomiális futási idejének köszönhetően on-line útvonalválasztás esetén is kiválóan alkalmazható.

Tézisem második részében az optikai helyreállítás megvalósításához szükséges egyértelmű hibalokalizáció feladatát vizsgálom meg felügyeleti fényutak segítségével átlátszó optikai hálózatokban. A legáltalánosabb struktúrájú fényutakat használva (m-trail) bevezetem az m-trail tervezési feladatot (MAP) a hibalokalizációra felhasznált jelzési költségek minimalizálására. Elégséges feltételeket adok többszörös linkhiba kezelésére, melyek segítséget nyújtanak heurisztikus algoritmusok tervezéséhez. A tézisben bevezetett algoritmusok több olyan fontos alkalmazási környezetet fednek le, melyekre korábban nem létezett hatékony eljárás az irodalomban, például jelentősen eltérő számú linket tartalmazó SRLG-k egyértelmű lokalizálására, magába foglalva a csomóponti hibák lokalizációját is. A mutatott eredmények hasznosságát jól jelzik a tudományterület legfontosabb folyóirataiban megjelent publikációim is.

Acknowledgments

I would like to thank to my supervisor, János Tapolcai, whose encouragement, guidance and support were indispensable to becoming a researcher in the field of telecommunications. His mathematical supervision, help and support were essential in understanding the research methodologies during my work.

I would also like to thank to Pin-Han Ho for his care and advices which made my research more adequate and useful. I am really grateful for the time I could spend at the University of Waterloo, Ontario, Canada with his guidance. Those months enabled me to have a wider look on the research techniques. It is my pleasure to cooperate with him.

My work was done in the research cooperation framework between Ericsson and the High-Speed Networks Laboratory (HSNLab) at the Budapest University of Technology and Economics. I am grateful to Róbert Szabó and Tamás Henk for their continuous support.

I would like to thank to all my co-authors, particularly to Tibor Cinkler and Bin Wu, whose view of the field always helped me to choose the best journals and conferences for our papers. Special thanks to my colleges and friends at the department, especially to my former roommates at IL106B and my fellow students Péter Soproni, and László Gyarmati.

I am heartily thankful to my mother, Mária Veszelka, who made sacrifices to support my studies and enabled me to learn and work untroubled. Her love and patience always helped me a lot, without her support my dreams would never have come true. I would also like to thank to János Pogány his support and advices. Further, I would like to thank to all of my relatives.

I wish to thank to all of my teachers at Fazekas Mihály Secondary School, especially to Erzsébet Müllner for her continuous support, and my mathematics teacher András Hraskó, for helping me develop a love for this wonderful field of science. Finally, I wish to thank to everyone who supported me in any respect during the completion of my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the recent years instead of the first mile (i.e. the origin infrastructure of a web application) and – owing to the rapid bandwidth increase – the last mile (i.e. access networks) the middle mile (core optical networks) introduce the main bottleneck and reliability problems in the networks [52]. There are several examples worldwide which had severe effects on the service availability of optical backbone networks in the last decade. Cable cuts may cause outages and makes a large number of end users offline from Australia [13] through the US [15], Europe and the Middle East [86] to Asia [21] [50]. For example, during the Baltimore tunnel fire in 2001 [15], the fire melted away the fiber along the tunnel, leading to a large number of correlated failures. Another case, when an undersea cable was cut during the Taiwan earthquake in 2006 [50], disrupting most communications out of Taiwan. From a financial point of view, the compensation claims for the Optus network failure [13] in 2008 could run into tens of millions of dollars, because a contractor laying pipe for a water grid accidentally cut the network's main fiber optic cable. The network outage affected more than a million subscribers.

Reliable communication network design serves as an important issue for service providers among the rapidly changing and emerging technologies. It is particularly critical when an all-optical backbone is in place due to its high data rate along each fiber and transparency in the data plane. The transparency - lack of Optiacal-to-Electronic-to-Optical (O/E/O) conversion at the intermediate nodes - enables very high data rates exceeding 10 or even 40 Gbps on each wavelength. In Wavelength Division Multiplexing (WDM) networks each optical fiber carries a large number of wavelength channels, thus a short transport level interruption may lead to an enormous loss of application data. Furthermore, there has been an increasing interest in providing high data-rate services such as video-conferencing or multimedia internet access recently. The rapidly increasing thirst for bandwidth and the spread of multicast technology provide new challenges for engineers. The persistent change of the underlying technology (e.g. WDM networks, wavelength conversion capability, dynamically switched multi-layer networks) always requires new methodologies. However, the main design goals and *Quality of Service (QoS)* requirements of the network are permanent: low capital expenditure (CAPEX) and operational expenditure (OPEX), throughput efficiency, and survivability.

Faults possibly cause the disruption of a connection if the users' data is carried only along one path (often referred to as *active or working path*) in the network, which might not be sufficient to fulfill the required QoS parameters defined in the *Service Level Agreement (SLA)* contracted between the service provider and customers. *Survivability* - the capability of a network to recover ongoing connections disrupted by a failure of a network component - has emerged to be the most important aspect in designing the control and management planes for next-generation networks [64]. In circuit switched and virtual circuit switched mesh networks, like the extensively deployed wavelength-division multiplexing networks, one of the key quantifiable properties of survivability is the connection (or end-to-end) availability provided by the network to the connection during its lifetime.

Availability refers to the probability of a reparable system to be found in the operational state at some time t in the future. End-to-end connection availability refers to the case when the source and destination nodes are connected by at least one path of operating edges and nodes, given that the connection was established at time $t = 0$ [78]. The availability of a network element is calculated from the average time elapsed between two subsequent failures of the same network element (called Mean Time Between Failures, MTBF) and from the average time needed to repair the given link (called Mean Time To Repair, MTTR) [89]. We have seen that in optical networks cable cuts have severe effects and are quite frequent, in fact, these are the main cause of the disruption of the connections [60, 91]. For the long-distance links the operator has cable-cut recordings, and they know how many cable cuts they can expect in a year approximately. Typical MTBF values for optical links range between 50 and 200 days per 1000 km of cable, while for an optical node is about 10^{-5} – 10^{-6} [90]. As a result, the link availability values in optical environment are about $A_{edge} = 0.999$, while nodes are more reliable, they have about $A_{node} = 0.99999$ availability.

Providing optical backbone network services high connection availability is essential for service providers, as they gain more profit from higher rates on reliable transfer. In the (optical) SLA, the operator declares the minimal service conditions able to carry the customer's data in the network for a given charge. The customer states his/her required bandwidth - if it is known - and chooses one of the QoS service classes offered by the provider. At the service provider, each service class corresponds to a given list of failure patterns $\mathcal{F}$, against which the connection have to be resilient to fulfill the availability declared in the SLA, or have to be rapidly localized in order to fulfill timing requirements of optical layer restoration. If the SLA is violated by the service provider, millions of dollars have to be paid for the customers [13]. Thus, precise modeling of the network and choosing the proper failure management techniques is critical in backbone networks. Therefore, the most important failure management tasks in reliable optical networks can be categorized in the following three phases [61]:

(i) protecting the connections against all failures in $\mathcal{F}$ (pre-designed *protection*),

(ii) fast and precise localization of the failed element(s) (i.e. detect all $f \in \mathcal{F}$ unambiguously),

(iii) restoring the disrupted connections (dynamic *restoration*).

The dissertation deals with the first two issues of optical failure management, namely with dedicated protection approaches and unambiguous failure localization using supervisory lightpaths. First, in Chapter 3 I introduce a novel general mathematical model for dedicated protection to support **phase (i)** of fault management in opaque and transparent optical networks. I proved that the complexity of the problem is NP-complete, thus, an Integer Linear Program (ILP) and a fast, yet efficient heuristic is introduced to solve the routing problem. I proved that with the application of network coding the problem is polynomial-time tractable. Second, in Chapter 4 I introduce the theoretical principles as well as practical algorithms for unambiguous failure localization with supervisory lightpaths to support **phase (ii)**. I prove that the complexity of unambiguous failure localization is NP-complete, thus, Integer Linear Programs are introduced. Necessary and sufficient conditions are formulated as the basis of unambiguous code assignment. Based on the sufficient conditions, fast, yet efficient heuristic approaches are introduced for failure localization, including node failures. Unambiguous and rapid failure localization serves as the foundation of rapid restoration in **phase (iii)** in survivable all-optical network design. Finally, the dissertation is summarized and the applicability of the proposed methods is discussed in Chapter 5.

Chapter 2

Survivable Optical Network Design [B1]

2.1 Evolution of Technologies: A Survivability Perspective

In the case of statically configured networks, the network was provisioned, configured, maintained and supervised through the management plane via a centralized management system. Such networks were mainly designed in a point-to-point manner, and the signal was converted to the electronic domain at each node. As a second step networks were designed in a shape of ring. In these synchronous digital hierarchy / synchronous optical networks (SDH / SONET) networks survivability mechanisms like automatic protection switching (APS) between redundant links in a point-to-point manner or SONET self-healing rings (SHR) in a ring topology were implemented. Later, due to the limited connectivity and reliability potentials, networks were deployed in the shape of a mesh in the backbone and metro networks. Mesh topologies offer high connectivity which greatly improves network reliability and design flexibility. On the other hand, because of the greater number of routing and design decisions [58] it leads to a bunch of complex problems like *signaling between the nodes* or the availability calculation of a connection. In opto-electronical cross-connects (EXCs) the optical signal is first converted to electrical signal then electrical space-switching is performed and finally it is converted back to optical domain again to any wavelength. By using EXCs in the network the total transparency of bit rates and signal formats is lost (called *opaque networks*).

In order to improve the transmission potentials of the fiber optic cables on the same cable topology, the wavelength division multiplexing and dense wavelength division multiplexing (DWDM) technology was introduced, offering tremendous amount of bandwidth by simultaneously transmit data of multiple connections on non-overlapped wavelength channels on a single fiber. The bandwidth of a fiber link can be divided into tens (or hundreds) of non-overlapped wavelength channels (i.e., frequency channels) and each cable contains many (e.g. 20 or more) fibers. Thus, the WDM technology is expected to play a significant role in next-generation networks.

As the technology evolved and optical cross-connects (OXCs), optical add / drop multiplexers (OADMs) and photonic switches were introduced the optical signal of a WDM channel could be switched from an

input port to an output port without any optoelectronic conversion (called *transparent or all-optical networks*). Thus, the costly and time consuming operation of electronic processing was eliminated at the intermediate nodes. In DWDM networks OXCs are used to switch individual wavelengths optically and establish lightpaths between nonadjacent nodes. A *lightpath* is an optical path established between two nodes of the network, carrying only optical signals. Assuming wavelength granularity, two lightpaths can use the same links if and only if they use different wavelengths. In these high capacity networks there was an even growing need for dynamically change the optical layer connectivity within milliseconds, i.e. a whole lightpaths can be deployed and released with user initiated signals within milliseconds in a distributed manner. Thus, the control plane (CP) was introduced in the networks, which communicates with signals to perform dynamic behavior with the other layers through well-defined interfaces (Automatically Switched Optical Network, ASON) [69].

In order to improve the coarse granularity of SDH / SONET networks, ITU-T has defined Virtual Concatenation (VCat) [40] and Link Capacity Adjustment Scheme (LCAS) [41] which together with the Generic Framing Procedure (GFP) [42] form the Next Generation SDH / SONET technology (ngSDH / SONET). The ngSDH / SONET networks are capable to perform inverse multiplexing, thus, owing to the finer granularities provided the demand flow can be split into multiple flows [20]. Later, the horizontal diversification of the network was started [19]. Thus, multi-layer networks emerged, from which the most promising architecture at this time is the Automatically Switched Transport Network / Generalized Multi-Protocol Label Switching (ASTN / GMPLS) [12]. In ASTN / GMPLS networks the communicating entities could be connected on fiber, waveband, wavelength, TDM frame or packed level granularity. In such dynamic networks, the connection requests are handled independently, they are arriving and getting served sequentially, without any knowledge of future incoming requests.

In dynamic networks it is important to develop a suite of inter-operable strategies that can, in real-time, find working path and protection resources upon the current load with efficient resource utilization. However, the trade-off has to be considered in optical backbone design between the network cost and operational complexity. Optical cross-connects may or may not be equipped with wavelength converters, i.e. devices that transform data streams coming in at one specific wavelength into an outgoing data stream at another specific wavelength. The price of an optical wavelength converter is high, thus, most of the all-optical networks have been built without any wavelength converter. In these networks lightpaths have to be routed along the same wavelength on each link it traverses, leading to a complex routing problem. However, with the application of wavelength converters minimal cost routes can be found rapidly.

The *protection methods* introduced in the dissertation were designed for optical layer protection in WDM networks; however, most of the protection methods can also be implemented at a number of layers including IP, Multiprotocol Label Switching (MPLS), Asynchronous Transfer Mode (ATM), SDH / SONET, ngSDH / SONET, ASON, and ASTN / GMPLS [14, 48]. Although each layer could have its own recovery schemes, they all show a rather similar succession of phases, that is, the recovery cycle [90]. The very first, and thus, one of the most crucial step of the recovery cycle is fault detection

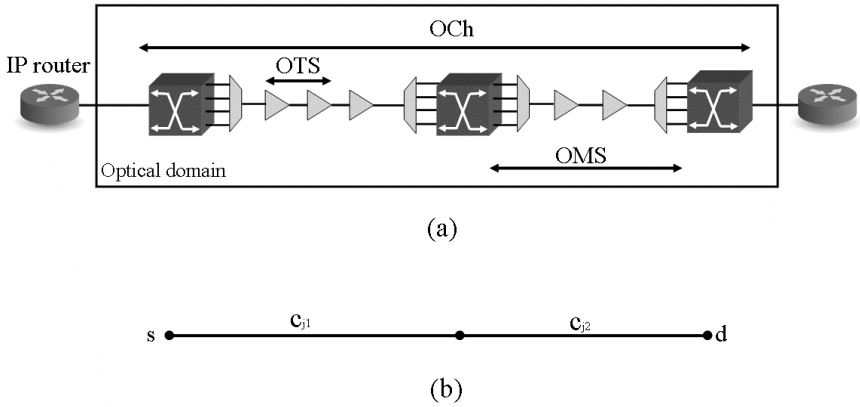


Figure 2.1: Optical Channel (OCh), Optical Multiplex Section (OMS) and Optical Transmission Sections (OTS) and the corresponding graph representation

(and in order to get the actual state of the network, *exact failure localization*), which is essential for rapid failure restoration. In multi-layer networks the inter-working between the layers is a challenging issue. In the case the failure is reparable at the optical layer in milliseconds, a survivable network should not allow the upper layers to take their own recovery action as it could lead to an unacceptable long interruption in the service. The first solution for this problem is applying a hold-off timer, i.e. the recovery action is delayed at the upper layers to allow the lower layers to repair the failure. The second approach use a recovery token signal, that is, the layer which owns the signal is responsible to recover from the failure. In the lower layers, e.g. WDM layer operating with high capacity and carrying aggregated traffic it is essential to keep the recovery time as short as possible. The ideal recovery time is considered to be *less than 50 ms in next generation optical networks* [34]. In this scenario, the interruption perceived by higher layers can be managed in a graceful manner.

2.2 Notions and Graph Representations in the Realm of Optical Networks

The aim of this section is to introduce the $G = (V, E)$ graph representations built up on optical networks, which serves as the input of the resilient routing and failure localization algorithms presented in the dissertation. Optical networks architecturally have two layers: the physical layer and the optical layer. The *physical layer* consists of fibers and *Optical Cross-Connects (OXCs)*, while the *optical layer* consists of optical links (lightpaths) and the corresponding nodes from the physical layer where lightpaths terminate. In contrast to static configured networks, assuming dynamically switched networks (e.g. supported by the GMPLS control plane), lightpaths can be established within milliseconds between arbitrary pairs of nodes in the network. Thus, we introduce the graph representation of the physical layer (OXCs and fiber links), where an arbitrary path could be an optical link in the optical layer.

Figure 2.1 presents an example of the graph representation $G = (V, E)$ with a set of links E and nodes V for an optical network. The nodes of the graph represent Optical Cross-Connects or Optical

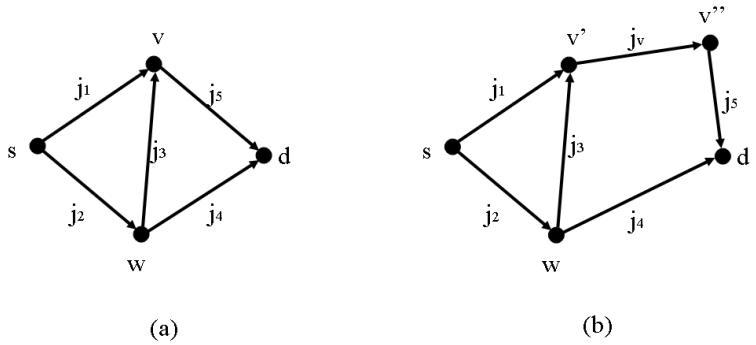


Figure 2.2: Example network with cost function $c_{j_1} = c_{j_4} = 3; c_{j_2} = c_{j_3} = c_{j_5} = 1$ and the auxiliary graph applying the node splitting technique on node v

Add-Drop Multiplexers (OADMs), where connection demands can enter and leave the network. In most of the approaches, the network nodes are assumed to be fully reliable (have an availability equal to one). An undirected edge (representing bi-directional fiber links between adjacent nodes) of the graph corresponds to an Optical Multiplex Section (OMS) of the network between two OXCs, and the cost function c_j on edge j corresponds to the cost of allocating a unit of demand flow (i.e. wavelength) on that particular edge. Cost function c_j may represent the length of the link (the number of optical amplifiers (or Optical Transmission Sections (OTS)), or signal quality degradation on long links. The connection between the first and last OXC in the optical domain is called the Optical Channel (OCh). An OCh is represented as a path in the graph.

In practical applications, often more network features are required. First of all, for certain communication network models, instead of bi-directional fiber links, we may need to consider directed links (arcs) and similarly, directed demands, and directed or bi-directed link capacities. Furthermore, in some practical applications the assumption of fully reliable nodes is not an appropriate model. Thus, node failures may have to be considered in addition to link failures. Node failures can be simulated by link failures in an auxiliary graph, where the node splitting technique in [65] is applied. First, each undirected edge is replaced by a pair of anti-parallel arcs. Secondly, every node v is split into two nodes v' and v'' connected by an arc $v' \to v''$. Each incoming edge of v is then directed to v', while each outgoing edge of v is directed from v'', as shown in Fig. 2.2(b).

In dynamically switched networks, connection requests arrive one after the other without any knowledge of future arrivals. In such a scenario the general goal is to develop a suite of inter-operable strategies that has a superb overall performance with low blocking probability, short average, and maximal waiting time of establishing connections, and low network utilization. In the working path selection stage Dijkstra's shortest path finding algorithm [24] is the most commonly applied method, which uses the cost function on the edges of the underlying graph. Thus, setting the cost function on the edges properly could be used for routing the traffic on those parts of the network where sufficient resources are available, while avoiding network components the free capacities of which are scarce. Applying this

method (called *Traffic Engineering (TE)*) can lead to lower blocking probability and can increase the overall network performance. A very common idea in TE is to use load balancing functions, which set the weights on the links (c_j) according to the topology and traffic characteristics such that a good overall performance can be expected using capacity-efficient routing algorithms for each connection request.

Finally, each edge has a capacity function corresponding to the available bandwidth on the given link (e.g. the free wavelength channels). The total capacity of link $j \in E$ of the graph $G = (V, E)$ could be categorized into the following three types:

Working capacity (denoted as q_j) which is the link capacity already taken by some working paths, and cannot be taken used until the corresponding working paths are torn down,

Spare capacity (denoted as v_j), which is the link capacity reserved by some backup paths,

Free capacity (denoted as k_j), which is the unreserved link capacity that can be reserved as either working or spare capacity, or reserve for supervisory lightpaths.

Given the *traffic demand* $\mathcal{D} = (s, d, b)$ (or sometimes $\mathcal{D} = (s, d, b, t_a, t_d)$ with t_a arrival and t_d departure times) in the single link failure scenario the task is to find a working path and a *protection path* between the source s and destination node d with the required bandwidth b. Depending on the applied protection scheme different constraints need to be satisfied by the solution. In the case of *dedicated protection* the working and protection paths can use any links with a sufficient free capacity ($k_j \geq b$). For *shared protection*, the constraint of spare capacity sharing must be investigated upon each network link before the best protection path can be derived for a pre-calculated working path. Whether or not a link has *sharable spare capacity* for a protection path depends on the physical location of the corresponding working paths. If the corresponding working paths are link-disjoint, than the spare capacity is shareable among them, as after a failure at most a single working path fails and uses the shared protection resource. This is also known as the dependency of the protection path on its working path.

2.3 Shared Risk Link Groups

In this section the $\mathcal{F}$ list of SRLGs is introduced which have to be protected (or have to be localized unambiguously) to fulfill the availability (timing requirements) declared in the SLA.

Most of the restoration architectures are designed assuming statistically independent single failure cases, which is not adequate in present day networks. This simplification comes from the assumption that the probability of each physical conduit to be subject to a failure is small and thus can be regarded as independent events even under the single failure scenario. However, dual failures are the most significant effects of disruptions in a single failure resilient network. Modeling multiple failures purely at the graph representation, failure states [65] can be defined. At this representation we concentrate failures in a single layer of the network, e.g. in this example on the element failures in the physical layer. In this case, as the

input of the routing problem a list of *failure scenarios* is given, and the connection needs to be resilient against all failures in the list. For this, we introduce a set $S \subseteq 2^E$ of network states each of which corresponds to a subset of failing links. Set S is called the failure scenario. It is assumed that S contains the normal, failure-less state $\emptyset$ in which all links are operational. The set $S^* = S \setminus \emptyset$ contains the *failure states (FS)* in which at least one link fails and each FS has a probability value that the corresponding failure state occurs. The number of states is exponential in the size of the network. In optical networks, the network elements have quite high availability. Therefore, in survivable network design failure states containing more than two or three elements are not worth our attention. Thus, the number of states is reduced to be polynomial with respect to the network size. If the states are assigned with the probability corresponding to the given dependent failure scenario measured by the network operator rather than the probability calculated from the independent individual link availability values; then the failure state approach can model failure dependencies, as well. Single link failure resilience could be treated as the special case of the failure state model (i.e. each single network element in the network topology serves a failure state).

At the graph representation level, the layered structure of the WDM optical network, the topology layout (e.g. physical location of the cables, common threats for multiple fibers) is lost. However, in an accurate network model, these properties have to be considered in a resilient network design. One of the possible ways of handling dependent multiple failures in optical networks uses *Shared Risk Link Groups (SRLG)* (or Shared Risk Resource Group or Shared Risk Group) [26, 68, 79]. SRLG expresses statistical dependencies between failures, that is, a group of network elements (i.e., links, nodes, physical devices, software/protocol identities, etc, or a mix of them) possibly subject to the same risk of single failure. In practical cases an SRLG may contain several seemingly unrelated and arbitrarily selected network elements. For instance, two links belong to the same SRLG if they share the same tunnel or conduit. Based on the observations at AT&T [79] a link may belong to over 100 SRLGs, each corresponding to a separate fiber group. In [79], SRLGs are characterized by 2 parameters. Type of compromise refers to the shared risk (e.g. shared fiber cable, shared conduit, etc.). The extent of compromise expresses the length of the sharing. The mapping of links and different types of SRLGs is in general defined by network operators based on the definition of each SRLG type. Links belong to the same SRLG because they are in the same *physical hierarchy*, which is related to the fiber topology (more generally the physical resources) of the optical network including the lightpaths built on top of this physical topology, or *logical hierarchy*, which is related to the geographical topology of the network [68].

The failures like cable cuts and OXC failures occur in the physical layer. However, in the physical hierarchy circuits are routed in the optical layer on optical links (lightpaths). Thus, an optical link failure could be affected by multiple link or node failures in the physical layer and belongs to those SRLGs. An example of possible SRLGs is defined in Figure 2.3. Since link failures in the optical layer are not mutually independent, the overall availability of a lightpath in the optical layer is lower than if assuming independent failures and leads to an inaccurate end-to-end availability value. In order to

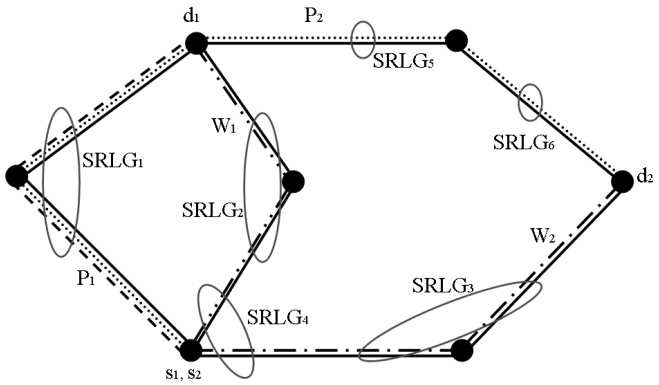


Figure 2.3: SRLGs defined on an example network; the two working paths (W_1 between source s_1 and destination d_1 and W_2 between s_2 and d_2) are link disjoint, but they are involved in a common SRLG (namely $SRLG_4$)

achieve a precise availability evaluation, these failure dependencies should be considered at the path selection stage. Considering multiple failures and dependencies among failures allow us to develop efficient routing methods and serves as the foundation of fast restoration with providing the opportunity to unambiguous failure localization. In addition, since SRLG relationships are not necessarily self-discoverable [22] and do not change dynamically, they don't need to be advertised by network elements. It can be configured in some central database and be distributed to or retrieved by the nodes. On the other hand, the information about link failure dependencies of SRLGs in the same logical hierarchy is inaccurate even at the service provider - who may have a long list of historical failure events, since they can only expect possible failures (e.g. disruptions in the same geographic region because of earthquakes, floods, etc.) in the future with the measured probability values. This makes the SRLG model hard to use in practice.

The presented SRLG model assumes that once an SRLG failure event occurs, all of its associated links fail simultaneously. However, this deterministic failure model cannot describe e.g. an event of a natural disaster, where some, but not necessarily all links in the vicinity of the disaster may be affected. There are promising ways of generalizing the notion of an SRLG to account for probabilistic link failure, called Probabilistic SRLG [51]. However, the original and widely used definition of SRLGs will be considered in the dissertation.

In contrast to single link failure resilience, when a general definition of the SRLGs is desired, a more complicated description and further elaborations are required to achieve an efficient implementation of any survivable routing algorithm for dedicated and shared protection or for backup reprovisioning [74]. This is because an SRLG could contain a wide range of number and type of network elements. These elements are mainly overlapped and/or contained by other elements; thus these routing problems are mainly NP-complete. Therefore, most of the solutions proposed for this problem are either optimal (e.g.

Integer Linear Programming (ILP) formulations) and slow, or fast, but do not give optimal solution for all problem instances (e.g. heuristic or approximation methods). Without loss of generality, we assume that a single failure event corresponding to an SRLG arrives at a time. In the case when two simultaneous failure events (corresponding to two SRLGs f_1 and f_2) need to be considered, the two failure events will be redefined as a single failure event, and the links in f_1 and f_2 will be taken as a new SRLG (i.e., f_3), which is further considered in the approaches.

At the $G = (V, E)$ model of the network, for each $f \in \mathcal{F}$ of SRLGs an auxiliary graph $G_f = (V, E_f)$ is constructed, where E_f is obtained by removing the corresponding failed links in f from E.

2.4 Operational Assumptions

Based on the observations made in the previous sections on the design methodologies in optical backbone networks, we define the operational environment for our algorithms. Note that *these assumptions do not restrict the generality of the proposed approaches and covers most of the practical scenarios*, thus, they can be generally applied in several networks with different underlying technology.

In [75], link failure independence was investigated, and it was shown that such an assumption could be dangerously inaccurate. In order to provide strict QoS requirements defined in the SLA, failure dependency among link failures have to be considered.

We used in the algorithm design for fault management of optical networks the most widespread SRLG lists $\mathcal{F}$ in the literature proposed for survivable telecommunication network design. These lists contain statistically the most probable [60] failure scenarios $f \in \mathcal{F}$, namely:

(1) $\mathcal{F}$ contains all single-link failures,

(2) $\mathcal{F}$ contains all single- and dual-link failures (*dense-SRLG scenario*),

(3) $\mathcal{F}$ contains all single-link failures and multiple-link failures adjacent to a common node (*sparse-SRLG scenario*), including node failures.

The following assumptions are made from the **data plane** of the underlying optical network:

- A wavelength channel is a single wavelength on a link, and has a single unit of bandwidth e.g. a single OC-192 channel with 10 Gbps speed by following the settings of SONET [105] networks. A lightpath is unidirectional and consist of a series of wavelength channels between the source and destination.

- We assume that all connections are bidirectional, i.e. each connection consists of two unidirectional lightpaths using the same links in the opposite direction. Thus, the network is modeled by an undirected graph $G = (V, E)$.

- Connections with required bandwidth higher than the capacity of a single wavelength channel (e.g. 2 OC-192 channels) reserves two lightpaths. The second lightpath can use the same links as the first one, or can be routed on a different route.
- This work assumes the online version of the Routing and Wavelength Assignment (RWA)/resilience problem, i.e. traffic demands arriving and getting served sequentially, without knowledge of future incoming requests, thus, each connection is protected individually.

The following assumptions are made from the **control and management plane** of the underlying optical network:

- Note that SRLG relationships are not necessarily self-discoverable and do not change dynamically, they don't need to be advertised by network elements. In survivable network design the SRLG list $\mathcal{F}$ corresponding to the QoS requirement can be retrieved by the nodes from the *central database*.
- A central fault-manager is assumed which receives alarms from all monitors in the network. The fault-manager can deactivate alarms and the resulting alarm vector is disseminated to the routing entities in the network [77].
- The central fault-manager is always reachable via the control plane or on a reliable channel. If monitoring nodes fail, a backup signaling system have to be in place (see Section 4.2.5 for details).
- Only protectable SRLGs in $\mathcal{F}$ are considered for the given $s-d$ pair, as none of the survivability methods can protect an $s-d$ cut in the network (see Section 3.1.1 for details).
- Wavelength channels can be used for failure localization purposes without carrying any useful customer data (out-of-band monitoring).

The **network nodes** are assumed to be capable of:

- The methods proposed for surviving shared risk link group failure are capable to survive node failures as well. The node failures can be simulated by link failures in an auxiliary graph as shown previously [65].
- At each node full wavelength conversion capability is assumed, i.e. no wavelength continuity required.
- There are scenarios, where the nodes are assumed to be capable of performing inverse multiplexing and/or algebraic operations on incoming signals. In these situations it is clearly stated in the name of the method (e.g. bifurcated, network coding (NC)).
- Traffic grooming is allowed on the wavelength channels, i.e. the date of multiple connections can be multiplexed on the same wavelength channel if there is some spare capacity.

Because of the application of traffic grooming and with the assumption of full wavelength conversion capable OXCs, the connections are treated as flows in this model.

Chapter 3

Dedicated Protection in Core Optical Networks

3.1 Challenging Issues in Dedicated Protection Approaches

3.1.1 Principles of Protection Survivability Architectures

The restoration of the connection for applications with high QoS requirement (e.g. remote surgery) may result in an intolerably long outage, which does not meet the QoS requirements declared in the SLA. With such applications in the network protection approaches have to be used to ensure the survivability of the connection, i.e. the connection have to **survive all failures in the SRLG list** $\mathcal{F}$ corresponding to the QoS level declared in the SLA. Thus, the resources used in the failure state of the network (*working resources*), as well as *protection (or spare) resources* (wavelengths, switches etc.) have to be reserved in advance for the connection. Spare resources are only used if failure occurred which cause the disruption of the working resources.

Spare resources can be shared (*shared protection*) [36] among the customers, i.e. spare resources can be used to provide protection to multiple working paths. In the case of single link failure resilient network design, a straightforward idea is to share the protection resources among users with disjoint working paths (a single link failure affects at most one of the working paths). However, after the failure has occurred, signaling is required between the upstream and downstream nodes of the path or the segments affected by the failure to reserve the protection resources. Thus, for the price of efficient resource utilization service recovery time is long. The complexity of shared protection lies firstly in the signaling efforts in case of a failure, and secondly, in computing the appropriate working and shared protection paths during connection setup.

In *dedicated protection* the backup resources are dedicated to a single working lightpath, thus, they can be reserved and configured at connection setup (hot stand-by) and used till the connection is torn

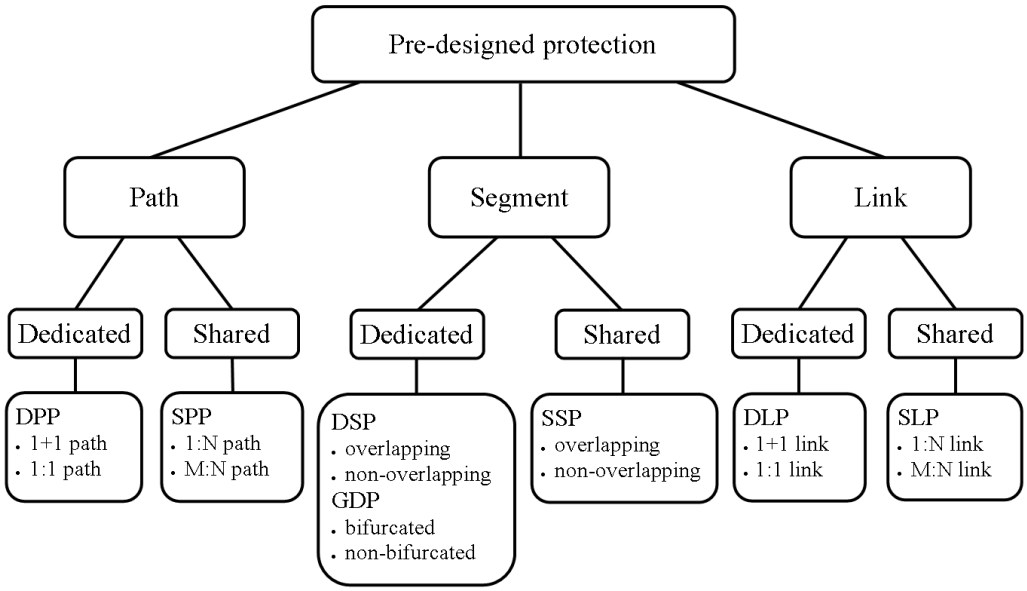


Figure 3.1: Classification of pre-designed protection schemes in optical mesh networks (method names (except GDP) can be derived from the bottom to the top, e.g. SLP corresponds to Shared Link Protection)

down. Dedicated protection is favored for its simplicity compared to shared protection. In dedicated protection the stringent timing requirements (50 ms) of optical layer restoration can be satisfied. Dedicated and shared protection schemes have their main differences in the amount of spare resources reserved for a connection, the signaling complexity, and the recovery time of the traffic after a failure occurred. The service provider's goal is to maximize the number of customers to gain more income, while minimizing the total resources allocated for a single user but still maintaining the required QoS level. Thus, *dedicated protection is the widely deployed protection approach in optical backbone networks*.

Although different protection approaches require different algorithms and different auxiliary graphs to get a working and protection path pair, finding a link-disjoint pairs of paths between two nodes (often referred to as *diverse-routing*) in the network is the basis of the previously introduced single failure resilient schemes. In the diverse-routing problem the task is to find a link-disjoint pair of paths between two nodes of the topology graph $G = (V, E)$. On the stipulation of resource availability and dependencies of the applied protection method, Suurballe's algorithm solves the diverse-routing problem in the graph with the modified cost function c_j. Suurballe's algorithm [81], first reported in the early 70's is famous for its polynomial computation complexity (originally $O(n^2 \cdot \log n)$ time) in finding optimal disjoint pairs of paths in terms of cost sum of the two paths in a directed graph. It is notable that the algorithm uses the same suite of link-state to derive the two paths. Suurballe's algorithm finds the minimal cost disjoint pair of paths among all pairs of paths in the network (if exists). Finding a disjoint working and protection pair of paths with Suurballe's algorithm also avoids the trap situation which could happen due

to greedily selecting the shortest path in the network as the working path, and as a second step a disjoint protection path is computed. For instance, a *trap situation* could occur in Figure 2.2(a) if the working path is selected with Dijkstra's algorithm ($s \rightarrow w \rightarrow v \rightarrow d$). In this situation, removing the edges from the topology (in order to get edge-disjoint working and protection paths) result in an $s - d$ cut, thus, the connection is blocked as there is no disjoint pair of paths providing the required availability level. However, if Suurballe's algorithm is used for finding disjoint pairs of paths, it will return $s \rightarrow w \rightarrow d$ and $s \rightarrow v \rightarrow d$, and the connection can be established.

Note that the computational complexity of the diverse routing problem mainly depends on the wavelength conversion capability of the OXCs. If the nodes are capable of converting the wavelengths, the problem of finding a minimum cost edge and (except for the source and destination nodes of the connection) node disjoint working and protection paths is polynomial time solvable with Suurballe's algorithm. On the other hand, if the OXCs are unable to convert the wavelength i.e. the wavelength continuity constraint [17] holds along the lightpath then the problem of finding two edge-disjoint lightpath in the network is NP-complete, both for the dedicated [7] and the shared case [66].

For a connection demand $\mathcal{D} = (s, d, b)$ between source node s and destination node d the SRLGs in the set could be categorized as follows [82]:

Protectable SRLG : An SRLG belongs to this type if the network still remains $s - d$ connected after the failure occurs, that is the connection can be restored. In other words, the failed elements in the SRLG do not form a cut in the network topology; in this case, the working path affected by the failure is restorable.

Cut SRLG : An SRLG belongs to this type if the source and destination nodes are in multiple isolated fragments when the network is attacked by a failure. In other words, the failed elements in the SRLG form a cut between the source and the destination node. Thus, the interruption upon the associate working paths can never be restored.

The cut SRLGs cannot be protected or restored with any survivable routing method, thus, the given SRLG list $\mathcal{F}$ always contains protectable SRLGs. We define that a *working path is involved in an SRLG* if it crosses any of the network elements belonging to that SRLG. Two working paths share the same risk of a single failure if they are involved in any common SRLG (see Figure 2.3). A working path is said to be SRLG disjoint (or diverse) with its protection path if the two paths are not involved in any common SRLG. The diverse routing problem is to find two paths between a pair of nodes in the optical layer such that no single failure in the physical layer may cause both paths to fail [39]. The problem of finding two diversely routed paths in optical networks for SRLGs is much more difficult than the traditional edge/node disjoint path problem in graph theory. For the single link failure case, finding link and node disjoint path-pair with wavelength converters is polynomial time solvable (e.g. Suurballe's algorithm). However, if an arbitrary set of links can belong to the same SRLG, then the problem of finding an SRLG disjoint path pair between a pair of nodes in the network is NP-complete. Essentially, the difficulty of

$1+1$ SRLG-diverse routing arises because the architecture allows SRLGs to be defined in arbitrary and impractical ways which intuitively forces an algorithm to enumerate (a potentially exponential number of) paths in worst-case (unless P = NP). In [26], an auxiliary graph is used, in which each SRLG type is expressed as a subgraph. Applying these representations of the SRLGs considered in the input of the routing problem, the SRLG diverse routing problem could be solved with traditional edge/node disjoint path finding algorithms in the auxiliary graph. As expected from the general definition of SRLGs and from the high computational complexity of the SRLG diverse routing problem, for some complicated types of SRLG there is no feasible graph representation. Thus, some of the routing computations are not physically feasible.

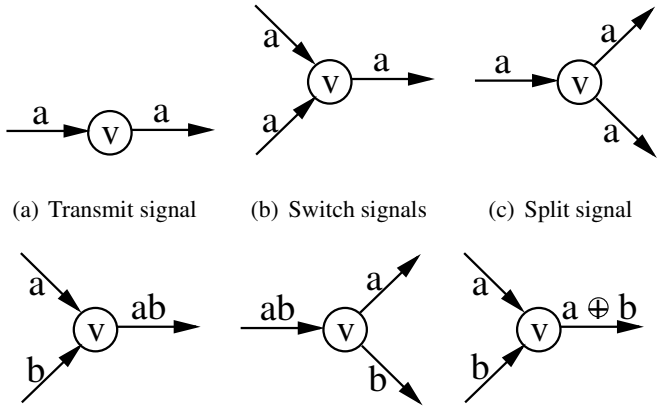
3.1.2 State-of-the-art

One of the most important targets of the Internet carriers is to meet the service requirements defined in the SLA with the subscribers in their backbones. This is particularly critical in all-optical mesh WDM networks where each lightpath may carry a huge amount of data. It has been widely noted that transferring user's data along a single active (or working) lightpath might not be sufficient to fulfill the service availability requirements in the presence of various network outages and unexpected failure events.

In the dissertation, first dedicated protection methods are investigated. In order to avoid the technical difficulties of signaling and reconfiguration of switches, we assume that all spare resources are reserved and signaled at connection setup (hot-stand-by), thus, the spare resources cannot be used to send low-priority data. We present a novel categorization of the dedicated protection problems based on the node roles in Figure 3.2, and we overview the dedicated protection approaches in the literature, which is summarized in Table 3.1. The node roles are the operations the OXCs can perform, shown in Figure 3.2 (a)-(f). The first case, when all nodes of a connection (source s, destination d and intermediate) are allowed only to transmit the signal (role (a) in Figure 3.2). This results a single path between s and d, referred to as the *working path (WP)*.

If some nodes are allowed to have roles (b) and (c) as well, we categorize these methods as follows. In the case the source node is allowed to split the signal, the destination node is allowed to switch between signals, but the intermediate nodes are restricted to transmitting the signal is the widely deployed $1+1$ or *Dedicated Path Protection (DPP)* [70]. In $1+1$ the signal is sent in parallel along two (SRLG or link)-disjoint routes from the source node to the destination node. In $1+1$ protection, an optical splitter used at the sending side, while switching takes places on the protection resources only at the destination node if it experiences the degradation of the signal quality on the working path.

Further, if any node along the working path is allowed to split the signal or switch between incoming copies of the same data, but the intermediate nodes along the protection paths are restricted to transmit the signal is called *Dedicated Segment (or Link) Protection (DSP/DLP)* [29] or partial path protection [92] [101]. Although with the application of segment protection high availability (the most



(a) Transmit signal (b) Switch signals (c) Split signal

(d) Merge signals (e) Decompose signal (f) Combine signals

Figure 3.2: Basic role of an arbitraty node v in the network regarding to the situations between the incoming and outgoing signals

important QoS parameter in circuit-switched networks) can be achieved by the connection, it is not frequently used in practice because of its high resource consumption. Figure 3.1 presents the categorization of pre-designed protection approaches, including the proposed Generalized Dedicated Protection (GDP). Although GDP is listed under segment protection, it generalizes both link protection and path protection as well (similarly to other segment protection approaches).

In networks, where the nodes are capable to perform inverse multiplexing (e.g. ngSDH / SONET and Optical Transport Networks (OTN) with VCat and LCAS) Multi-Path Routing with Protection (MPP) [20] was introduced (roles (a)-(e) in Figure 3.2 are present). The original method was proposed for single-link failure resilience, and a linear program was presented to solve the problem. Later, the method was generalized for SRLG failure protection [32] for type (3) SRLGs. However, the improved MPP routing problem is polynomial-time tractable only for SRLGs containing adjacent links.

The reserved bandwidth can be reduced by applying nodes that are capable for combining incoming signals as shown in Figure 3.2(f). Network coding capable nodes can perform basic linear operations on the data transmitted. For the sake of explanation, we assume that the addition operation in Figure 3.2(f) is over Galois Field $GF(2)$, then the combination of incoming signals is the simple *Exclusive-OR (XOR)* operation. However, in general case more complex arithmetic operations need to be performed[1].

The idea of *network coding (NC)* was first introduced in [3] for single-source multicast. It was shown that with NC the achievable multicast capacity equals the minimum of the maximal unicast flows from the source to the receivers. Later, in [53] a linear NC scheme for the same problem was introduced.

NC has been positioned as a viable solution for improving network throughput in various application scenarios. In [54] a distributed and packetized network coding implementation was introduced, where the nodes forward the random linear combinations of the received bitstreams. In order to make the packets

[1] Addition and multiplication over Galois Filed $GF(2^m)$.

decodable at the receivers, *global encoding vector* is attached in the header of each packet. Another reported NC application is in Passive Optical Networks (PONs) [10] [62]. It was shown that using NC can not only improve the downstream throughput but also reduce the end-to-end packet delay.

NC can be used for improving reliability and robustness in multi-hop wireless networks [6]. The study [37] contains an information-theoretic analysis of network management in order to improve network robustness.

In [47] robust network codes for multicast were proposed. By assuming non-zero failure probabilities for network links and a set $\mathcal{F}$ of failure patterns (or SRLGs), [47] constructed an auxiliary graph G_f for each $f \in \mathcal{F}$ that is obtained by deleting the corresponding failed links (similarly to *SRLG graphs*). Theorem 11 in [47] claims that for a linear network G and a set of single-source multicast connections $\mathcal{C}$, there exists a common static network coding solution to the network problems $\{G_f, \mathcal{C}\}$ for all $f \in \mathcal{F}$. The previous code is static (robust against network failures); i.e. only the decoding matrix at the destination node needs to be reconfigured in the presence of failures, while the intermediate node configurations remain unchanged, while full receiving rate is maintained.

In [38] random linear NC approach to solve multicast in a distributed manner was proposed. As each node selects coefficients over the Galois field randomly the computational complexity of this scheme is significantly lower than its centralized counterpart. On the other hand, with the application of random network codes the decodability can be guaranteed only with high probability. It was shown in [38] that if a multicast connection is feasible under any link failure $f \in \mathcal{F}$ than random linear network coding achieves the capacity for multicast connections, and is robust against any link failures $f \in \mathcal{F}$ with high probability.

In [43] the robust multicast NC under the same failure model as [47] was investigated. Theorem 11 of [43] states that if the transmission rate does not reduce below a given value k along any link in all the auxiliary graphs (G_f), robust linear network codes can be found in complexity of $O(|\mathcal{F}| \cdot |E| \cdot (|\mathcal{T}| \cdot k^2 + \min\{I, |\mathcal{F}| \cdot |\mathcal{T}|\} \cdot k))$, where $\mathcal{T}$ denotes the number of receivers and I denotes the maximal (in-)degree of a node.

Application of NC in core optical networks has recently emerged [9, 44, 63], which in general aim to minimize the capacity consumption for a matrix of traffic demands. With shared $M : N$ protection (Fig. 3.1), N working connections are protected by a common pool of M protection paths, where the protection resources are used only after a failure occurred in the network. Such a concept was generalized to $1 + N$ protection by ensuring the spare resources *hot stand-by* similarly to $1 + 1$ protection, provided with the capability of performing linear combination operation on the input symbols of the N working paths at the source OXC. In [44] the protection resources are in a shape of cycle, while in [63] it is in a shape of a Steiner-tree. In the latter case, it was proved as NP-complete in finding the optimal solution. In $1 + N$ protection proposed by Kamal, et al. network coding is allowed at the source and destination node of the connection, but the intermediate nodes are restricted to transmit the signal. In [44] network coding was combined with the concept of p-cycles, where the connections terminating on the same p-cycle

Table 3.1: Taxonomy of Dedicated Protection Approaches

Nodes	Roles	WP	$1+1$	DLP	DSP	**IGDP**	MPP	**BGDP**	$1+N$	**GDP-NC**
source and dest.	(a)	x	x	x	x	x	x	x	x	x
	(b)(c)		x	x	x	x	x	x	x	x
	(d)(e)						x	x		x
	(f)								x	x
working path	(a)	x	x	x	x	x	x	x	x	x
	(b)(c)			x	x	x	x	x		x
	(d)(e)							x		x
	(f)									x
protection resources	(a)		x	x	x	x	x	x	x	x
	(b)(c)					x	x	x	x	x
	(d)(e)							x		x
	(f)								x	x
Optimization method		Dijk.	Suurballe			ILP	LP	ILP	ILP	LP
References		[24]	[29] [80]			[C3]	[20]	[J3]	[44]	[C6]

were protected by sending a linear transformation of the transmitted data along the p-cycle in optical domain. In [63] some nodes along the protection routes perform network coding instead of the source and destination node of the connection. In this case, for a given set of connections a Steiner-tree is built to protect single failures, and network coding is performed along some specific nodes in the protection tree.

Although $1+N$ protection has all the merits of dedicated protection approaches while keeping the capacity consumption low, it requires the topologies with $1+N$-connectivity, which serves as a stringent constraint on its applicability. Note that it is not present in most of current networks [57] for $N \geq 3$. In [9], a *virtual layer* is defined, in which network coding is applied to protect F simultaneous failures along the disjoint paths between the source and destination node. Although the F failures in the virtual layer correspond to a single failure event in the physical layer, the high connectivity requirement of $1+N$ protection was relaxed without impairing the capacity efficiency.

In opaque optical networks, NC operations can be performed at each intermediate node in the electronic domain via electronic buffering and processing. As NC requires additional hardware, in [46] an evolutionary approach for NC was developed, where coding is performed at as few nodes as possible. The work in [87] investigated NC in WDM optical networks where O/E/O equipment is required for wavelength conversion. A method for minimizing the number of wavelengths which have to be coded or converted was introduced. In [59], dedicated protection of multicast trees was investigated, and various architectures for all-optical circuits capable of performing the operations required for network coding

were introduced.

All of the previously introduced methods in the literature defines the node roles in Fig 3.2 before the exact protection structure is known. However, this approach cannot be fit well into the protection design of the corresponding QoS parameter. On the other hand, the proposed Generalized Dedicated Protection approach in the dissertation can better explore the design space of optical protection, as the node roles are decided just after the resilient protection subgraph is known (i.e. as the result of the optimization problem). As the online version of the routing / resilience problem is assumed, i.e. traffic demands arriving and getting served sequentially, without knowledge of future incoming requests, thus, each connection is protected individually.

3.1.3 Problems Targeted in the Dissertation

All the previously reported schemes are dealing with implementation issues (e.g. find a disjoint pair of paths between two nodes) and decide the node roles in Fig. 3.2 in advance. However, the customers are not interested in the details how their connections are protected. Thus, a more flexible protection structure is required, which can better explore the design space of the routing problem provided by the topology and the network equipments available at the network nodes. Obviously, a generalized framework for dedicated protection that can incorporate all the various assumptions and design premises have never been reported yet. In Section 3.2 *I introduce a novel general mathematical model for dedicated protection*, which seeks for a minimal cost feasible solution based on the required QoS level (or equivalently $\mathcal{F}$) and leaves implementation questions to the optimization problem, i.e. the node roles are decided based on the result of the solution. *I proved that the complexity of the problem is NP-complete, thus, an Integer Linear Program and a fast, yet efficient heuristic is introduced to solve the routing problem. I proved that with the application of network coding the problem is polynomial-time tractable.*

Based on the optical equipments available at the network nodes, three different GDP problem can be formulated. First, if technology supports only connections with coarse granularity (e.g. SDH/SONET networks without inverse multiplexing), a feasible GDP solution is sought in the form of non-bifurcated flows (node roles (a)-(c) in Fig. 3.2 are allowed in the solution) is called **Integer (or non-bifurcated) GDP (IGDP)**. Second, if the technology for bifurcating network flows (i.e. inverse multiplexing is possible like in ngSDH/SONET networks and OTNs with VCat and LCAS) is available at the nodes then roles (a)-(e) are allowed in the solution, called **Bifurcated GDP (BGDP)**. We will demonstrate the IGDP and BGDP are highly related problems. Finally, as the most general case, called **GDP with Network Coding (GDP-NC)** is defined, where architectures proposed in [59] are present at the nodes to perform the operations required for network coding, i.e. roles (a)-(f) in Fig. 3.2 are allowed in the optimization problem.

3.2 The Generalized Dedicated Protection (GDP) Approach

This section provides the GDP problem formulation for dynamic survivable routing. The notations are summarized in Table 3.2.

Let $\mathcal{I} = \{G, \mathcal{D}, \mathcal{F}\}$ denote an instance of the proposed GDP problem.

- In the dynamic traffic scenario (e.g. supported by the GMPLS control plane) each connection is routed promptly after the request arrives in the ingress network entity at time t at which the actual topology is denoted by $G = (V, E)$. On each edge $e \in E$ a non-negative cost function ($c : E \to R^+$) is defined, and the free capacity ($k : E \to R^+$) is given. The free capacity is the function of the connections built up and torn down until t.

- A traffic demand $\mathcal{D} = (s, d, b)$ contains the source s and destination node d, and the bandwidth requirement ($b \in \mathbb{N}$) in OC-192 units.

- Finally, the set of failure patterns (or SRLGs), denoted as $\mathcal{F}$, is given. For each SRLG, a failure subgraph is created: $\forall f \in \mathcal{F} : G_f = (V, E_f)$, where E_f is obtained by removing the edges in f from E. We assume that each SRLG in $\mathcal{F}$ is *protectable*, i.e. each G_f is $s - d$ connected.

Let the set $\mathcal{X}_\mathcal{I}$ contain a set of feasible solutions $y_\mathcal{I} = \{H, \mathcal{R}\}$ for the problem instance $\mathcal{I}$. Each solution is actually:

- a subgraph $H = (V, E)$, where the capacity along each edge is set to $\forall e \in E : b_e$, which is the flow value in the solution. The maximum flow in $H = (V, E)$ and $\forall H_f = (V, E_f)$ with capacity values b_e between s and d is $\geq b$, where E_f is obtained by removing the failed edges in f from H.

- The other part of a GDP solution is the configuration map $\mathcal{R}$, which gives the cross-connections in the switching matrix and grooming settings of the OXCs in H.

To further explain, in a feasible solution $y_\mathcal{I} \in \mathcal{X}_\mathcal{I}$ the connection is operating on b bandwidth in the failure-less state of the network, and it is *resilient against all failures* $f \in \mathcal{F}$ defined by the network operator. Furthermore, each OXC is configured accordingly to the solution at connection setup (switching matrix, merging signals, etc.) and this configuration is *robust against all failures in* $f \in \mathcal{F}$; i.e. it remains unchanged even after a failure occurred. On the other hand, in a robust solution changes which leaves the switching matrix unchanged, or nodes can perform locally in a distributed manner are allowed similarly to other dedicated protection methods, e.g. switching between the incoming data streams based on the quality of the incoming signals similarly to $1+1$, or in the network coding based approach change the decoding matrix at the destination node based on the failure pattern f. We are rather interested in *resilient and robust (R&R)* solutions because they require a constant number of messages in the control plane after a failure occurs, which is essential to maintain scalability, simplicity, and rapid restoration time of $1 + 1$ protection.

Table 3.2: Notation list for the Generalized Dedicated Protection (GDP) Problem

Notations	Description
$G = (V, E)$	the undirected (or directed) graph representation of the topology with nodes V and edge set E
c_e	cost function defined on edge $e \in E$
k_e	free capacity along link $e \in E$ (in OC-192)
$\mathcal{D} = (s, d, b)$	source, destination and requested bandwidth, respectively, of the dynamically arrived traffic demand
$\mathcal{F}$	shared risk link groups or failure patterns against the connection needs to be resilient
$G_f = (V, E_f)$	SRLG graph obtained by removing the failed edges in $f \in \mathcal{F}$ from G
$\mathcal{I} = \{G, \mathcal{D}, \mathcal{F}\}$	instance of the routing problem
$\mathcal{X}_\mathcal{I}$	the set of feasible resource reservations $y_\mathcal{I}$ for the instance $\mathcal{I}$
$y_\mathcal{I} = \{H, \mathcal{R}\}$	a feasible *R&R* solution containing the graph H and the configuration map $\mathcal{R}$
b_e	the reserved capacity along link $e \in E$ in the solution (in OC-192)
$b_{e,f}$	the reserved capacity along link $e \in E_f$ in the SRLG graph G_f (in OC-192)

Note that resilient non-bifurcated dedicated protection solutions (e.g. $1 + 1$, DLP, DSP) are always robust. However, methods applying bifurcated flows (e.g. MPP) often requires extensive signaling in the control plane (not robust) to restore the connection, similarly to shared protection approaches, thus, the simplicity of $1 + 1$ is lost. An example is presented in Fig. 3.6, where the existence of a feasible configuration map $\mathcal{R}$ is only guaranteed if the network nodes are able to combine linearly the incoming signals. Let us consider $H = (V, E)$ obtained in Fig. 3.6(a) which is resilient against the SRLG failures $\mathcal{F} = \{(j_1, r_1), (j_2, r_2), (j_3, j_4)\}$. One can easily check that the presented configuration $\mathcal{R}$ is robust against all failures, and only the decoding matrix at the destination node need to be updated upon failures, i.e. $x = \{H, \mathcal{R}\} \in \mathcal{X}_\mathcal{I}$. However, if network coding is not allowed, we have to configure a or b instead of $a \oplus b$ on the corresponding edges, without loss of generality b. In this case, after SRLG (j_1, r_1) failure occurs as shown in Fig. 3.6(b), there exists an $s - d$ cut in the network for the a part of the data. It is not possible to recover the data at the destination node without reconfiguring node j_3 and signaling in the control plane. Thus, such a GDP solution is resilient but not robust against some failures, i.e. $x = \{H, \mathcal{R}^H\} \notin \mathcal{X}_\mathcal{I}$ for any configuration map $\mathcal{R}^H$. The design goal of the algorithm was to keep the simplicity and rapid restoration time of $1 + 1$ protection, a non-robust z solution is not treated as a feasible GDP solution ($z \notin \mathcal{X}_\mathcal{I}$).

Each feasible solution $y_{\mathcal{I}} \in \mathcal{X}_{\mathcal{I}}$ is assigned with a cost $g(y_{\mathcal{I}})$ corresponding to the reserved bandwidth in the network as follows:

$$g(y_{\mathcal{I}}) = \sum_{\forall e \in E} c_e \cdot \frac{b_e}{b}, \tag{3.1}$$

where $b_e \leq k_e$ is the OC-192 units used along link e by the connection. For the sake of simplicity, the objective is divided with b in order to get a percentage for the used bandwidth on each link. However, it is also possible to use the objective function without dividing with b. The algorithms proposed to find an optimal solution in terms of Eq. (3.1), or an arbitrary feasible solution to the different versions of the GDP problem are summarized in Table 5.1.

3.2.1 Computational Complexity of the Bifurcated and Non-Bifurcated GDP Problem [C3, *C6*]

In general, network design problems seek to find a minimum cost subgraph of given (directed and undirected) graph, which satisfies a list of requirements. These problems are the most studied problems in the fields of combinatorial optimization. Usually, directed variants of network design problems are much harder to approximate than the undirected ones [28]. For example, in the undirected graph representation, at this time the best approximation factor for Steiner Tree was decreased from 2 to ≈ 1.55 [72]. On the other hand, assuming a directed graph, there does not exist a constant approximation factor. Thus, both directed and undirected variants of IGDP are investigated.

In order to prove the NP-completeness of the IGDP, we show a polynomial-time transformations to the (Directed) Steiner Forest problem [28, 49] using either the directed or the undirected graph representation of the underlying network. In the IGDP problem only non-bifurcated flows are acceptable as only node roles (a)-(c) are allowed, thus, the edges with $k_e < b$ can be removed from the topology $G = (V, E)$. Thus, the problem of ***finding a non-bifurcated solution for*** $\mathcal{D} = (s, d, b)$ ***is equivalent with the problem*** $\mathcal{D} = (s, d, 1)$ ***for all problem instances***. We will demonstrate that instances with $\mathcal{D} = (s, d, 1)$ determine the complexity and algorithm design of both IGDP and BGDP. Furthermore, *a resilient $H = (V, E)$ subgraph for a non-bifurcated solution is always robust*, thus, the task is to find a set of edges in the network containing (directed) $s - d$ routes in all $G_f = (V, E_f)$ subgraphs.

The Directed Steiner Forest is special case of the Directed Steiner Network (DSN) problem [16] (where the traffic demand between the ordered $V \times V$ node-pairs of the network is either 1 or 0 lightpath) which is proved to be NP-complete if the number of source-destination pairs p is part of the input, which is the case in GDP for the $|\mathcal{F}|$ number of SRLGs, i.e. it is part of the input as well.

Theorem 3.2.1 *To decide whether a solution with $\leq k$ cost exists for the IGDP problem with traffic demand $\mathcal{D} = (s, d, 1)$ in directed graphs is NP-complete.*

Proof: The IGDP problem is in NP, a solution with $\leq k$ cost is a proof.

Assume we are given an instance of the Directed Steiner Network problem, that is, a graph $G_{DSN} = (V_{DSN}, E_{DSN})$ with ordered source and target pairs $\{s_1, t_1\}, \{s_2, t_2\}, \ldots, \{s_n, t_n\}$, and corresponding connectivity demands between s_i and t_i (in the special case of Directed Steiner Forest 1 for the above pairs and 0 for the others). To construct $G = (V, E)$ we add two new nodes s and t together with edges (s, s_i) and (t_i, t) with zero cost, all other edges have unit cost, formally $G = (V_{DSN} \cup \{s, t\}, E_{DSN} \cup \{(s, s_i), (t_i, t)\})$. Unit spare capacities are assigned for all edges in the transformed problem, and the traffic demand is $\mathcal{D} = (s, d, 1)$, while the SRLG list is $f_j \in \mathcal{F} : \{(s, s_i), (t_i, t)\}$, where $\forall j = 1, 2, \ldots, n$ and $i = 1, 2, \ldots |\mathcal{F}|, i \neq j$. The transformation is realizable in polynomial time. A solution exist with $\leq k$ edges in Directed Steiner Network connecting all s_i and t_i if and only if a solution with $\leq k$ cost exists for the IGDP problem. Thus, the IGDP is NP-complete. ■

Similarly, the NP-completeness of the IGDP problem is proved via Karp-reduction to the Steiner Forest Problem [28].

Theorem 3.2.2 *To decide whether a solution with $\leq k$ cost exists for the IGDP problem with traffic demand $\mathcal{D} = (s, d, 1)$ in undirected graphs is NP-complete.*

Proof: The IGDP problem is in NP, a solution with $\leq k$ cost is a proof.

Assume we are given an instance of the Steiner Forest problem, that is, an undirected graph $G = (V, E)$, edge costs $c : E \to \mathbb{R}^+$, and disjoint r subsets $S_i \subseteq V$, is there a forest F with $\leq k$ cost such that for all i and $u, v \in S_i$, there exists a path connecting u to v in F. In this proof, the definition of Steiner Forest problem is used, in which each subset S_i contains only two elements $S_i = \{s_i, t_i\}$ [49].

The polynomial time transformation is given as follows. We add two new nodes s and t together with edges $E^+ = \{(s, s_i), (t_i, t)\}$, $i = 1, 2, \ldots r$ with $\forall e \in E^+ : c_e = 0$, all other edges have the cost of the Steiner Forest problem. Unit free capacities are assigned for all edges in the transformed problem ($\forall e \in E : k_e = 1$), and the capacity request is $b = 1$. We define $|\mathcal{F}| = r$ SRLGs, one for each source-target pair, that is $f_j \in \mathcal{F} : i = 1, 2, \ldots, r, i \neq j : \{(s, s_i), (t_i, t)\}$. The problem instance is $\mathcal{I} = \{G = (V, E \cup E^+), \mathcal{D} = \{s, t, 1\}, \mathcal{F}\}$.

As the two problem has the same cost, a solution exist with $\leq k$ cost in Steiner Forest connecting all s_i and t_i if and only if a solution with $\leq k$ cost exists for the IGDP problem between s and d. Thus, the IGDP is NP-complete. ■

We give a sufficient condition to demonstrate that there exists a non-bifurcated optimal solution for the BGDP problems if all the links have sufficient free capacity to route the connection.

Lemma 3.2.3 *The optimal solution for the BGDP problem is integer ($b_e = \{0, b\}$) if all the links have sufficient spare capacity to route the connection ($\forall e \in E : b \leq k_e$).*

Proof: Let assume that BGDP has an optimal bifurcated R&R solution with configuration map $\mathcal{R}$ with lower cost than an optimal IGDP solution. The proposed b_e/b percentages are $p_1 < p_2 < \ldots < p_n \leq 1, \forall p_i \in \mathbb{Q}$ with product $p = \prod_{i=1}^{n} p_i$. In the optimal solution each part of user's data use a minimal

cost subgraph satisfying the connectivity constraint in each $f \in \mathcal{F}$, specially a part with capacity $b \cdot p$ as well.

Let us route the parts of the data as an IGDP problem with traffic demand $\mathcal{D} = (s, d, b \cdot p)$ one after the other, they should take the same subgraph (or a subgraph with $\leq$ bifurcated minimal cost) as the part $b \cdot p$ was taken in the bifurcated solution. In such a construction, all parts were routed on a minimal cost subgraph, in addition $b_e = \{0, b\}$, as *all links had sufficient spare capacity*. Thus, the optimal IGDP solution is $\leq$ than the optimal BGDP solution. It leads to a contradiction with the claim that the BGDP solution with lower cost than the IGDP solution is a feasible R&R solution. ■

Using the claim of Lemma 3.2.3, we have the following theorem about the complexity of the BGDP problem:

Theorem 3.2.4 *To decide whether a solution with $\leq k$ cost exists for the BGDP problem is NP-complete.*

Proof: The BGDP problem is in NP, a solution with $\leq k$ cost is a proof. The NP-completeness of BGDP is proved in the following two steps:

$\forall e \in E : b \leq k_e$**:** Lemma 3.2.3 shows that the optimal BGDP is non-bifurcated. We have seen that find the optimal non-bifurcated solution is NP-complete (Theorem 3.2.1 and Theorem 3.2.2).

$\exists e \in E : b > k_e$**:** Based on Lemma 3.2.3, each data part $k_{min} = \min_{\forall e \in E} k_e < b$ have to be routed on optimal non-bifurcated routes, i.e. BGDP comprise IGDP as a subproblem. As $\forall e \in E : k_e$ and b are integers, IGDP should be solved at most b times to route the connection.

Thus, the BGDP problem is NP-complete. ■

As a consequence, the problem of routing a $\mathcal{D} = (s, d, 1)$ demand optimally is the key challenge for solving both IGDP and BGDP.

3.2.2 Find Optimal Solutions for the Bifurcated and Non-Bifurcated GDP Problem [C3, *J3*]

The cost function presented in Eq. (3.1) is a joint optimization for the working and protection edges. On the other hand, the IGDP approaches presented in the dissertation works as well if the working path is calculated in advance (with a slightly modified input). The GDP use a general mathematical model and the node roles (in Fig. 3.2) are decided as the result of the optimization problem, thus, the *trap situations* – which is present in most of the dedicated protection approaches, e.g. $1 + 1$ – is avoided, even with pre-calculated working path. One can easily check, that if an arbitrary $\mathcal{I}$ GDP instance has a feasible solution with working path $\mathcal{W}_1$, than it has a feasible solution with any other working path $\mathcal{W}_2$ as well, thus, the selection of the working path does not influence the existence of a feasible solution.

In the following we discuss the requirements of the existence of a non-bifurcated GDP solution without network coding. Note, that in a non-bifurcated solution only those e links can be used ($b_e > 0$),

which has at least $b \leq k_e$ free wavelength channels. First we allow split BGDP solutions (node roles (a)-(e) in Figure 3.2 are allowed), and we investigate the existence of a non-bifurcated solution. Note, that Theorem 3.2.5 is a necessary and sufficient condition contrary to the sufficient condition presented in Lemma 3.2.3.

Theorem 3.2.5 *A minimal cost non-bifurcated solution exists for the BGDP problem* $\mathcal{I} = \{G = (V, E), \mathcal{D} = \{s, d, b\}, \mathcal{F}\}$ *without network coding if and only if there exist a minimal cost resilient subgraph* $H = (V, E)$ *for the problem* $\mathcal{I}' = \{G = (V, E), \mathcal{D} = \{s, d, 1\}, \mathcal{F}\}$*, where* $\forall e \in E, b_e = 1 : k_e \geq b$*.*

Proof: The proof is constructive, and we give a method to find a feasible solution.

Let $\mathcal{I} = \{G = (V, E), \mathcal{D} = \{s, d, b\}, \mathcal{F}\}$ be an instance of the GDP problem without network coding. As one can observe in the example in Figure 3.6, each fraction of the b OC-192 have to be not only resilient but also robust against all failures in $\mathcal{F}$. As the smallest possible fraction without traffic grooming is 1 OC-192, we could treat the problem $\mathcal{I}$ as solving b instances with unit bandwidth requirement $\mathcal{I}_i = \{G = (V, E_i), \mathcal{D} = \{s, d, 1\}, \mathcal{F}\}$ one after the other. The solution of any $\mathcal{I}_i$ is non-bifurcated, as no further division of the 1 OC-192 is possible. The final solution x for the problem instance $\mathcal{I}$ can be obtained by sum up the non-bifurcated solutions ($x_i = \{H_i, \mathcal{R}_i\} \in \mathcal{X}_{\mathcal{I}_i}$) for each of the b OC-192.

If $\exists x_1 = \{H_1 = (V, E_1), \mathcal{R}_1\} \in \mathcal{X}_{\mathcal{I}_1}$ which minimizes the cost function $g(x_1)$ in Equation (3.1) and for which $\forall e \in E_l : k_e \geq b$, then each $\mathcal{I}_i$ could use the same minimal cost subgraph in the network. Thus, there always exists a non-bifurcated solution $x \in \mathcal{X}_{\mathcal{I}}$ for the problem instance $\mathcal{I}$, which minimizes the cost function $g(x)$ while $\forall e \in E : b_e = \{0, b\}$, i.e. the flow in $H = (V, E)$ is non-bifurcated.

On the other hand, if $\forall y_1 = \{H_1, \mathcal{R}_1\} \in \mathcal{X}_{\mathcal{I}_1} : \exists e \in E_1 : k_e < b$, then some of the links have no more free wavelength channels after demands $i = 1, 2, \ldots, j < b$ are routed (i.e. $k_e = 0$). In the further iterations these edges are removed from the input topology $G = (V, E_i)$ for the problem instances $i = j + 1, j + 2, \ldots, b$. Thus, the final solution is bifurcated, as $i = j + 1, j + 2, \ldots, b$ cannot use the same subgraph $H_{i<j}$ as the previous data parts. Further, the cost $g(y)$ of the bifurcated solution $y \in \mathcal{X}_{\mathcal{I}}$ is lower than the cost $g(x)$ of any non-bifurcated solution, as $\forall i < j : g(y_i) < g(x_i)$ and $\forall j \leq i : g(y_i) \leq g(x_i)$.

■

The constructive proof of the Theorem 3.2.5 to obtain a solution is presented in the algorithm in Figure 3.3.

Note that if $\exists x_1 = \{H_1 = (V, E_1), \mathcal{R}_1\} \in \mathcal{X}_{\mathcal{I}_1}$ use only edges with $k_e \geq b$, then the optimal solution is non-bifurcated and the algorithm in Fig. 3.3 solves the IGDP problem itself in one iteration.

When a feasible non-bifurcated solution exists, it is easy to verify that the algorithm in Fig. 3.3 gives a feasible (bifurcated of non-bifurcated) solution $x = \{H = (V, E), \mathcal{R}\} \in \mathcal{X}_{\mathcal{I}}$ in finite steps. In each iteration the optimal solution for protection structure for at least 1 OC-192 is assigned. The free capacity along each edge (i.e. the number of free wavelength channels) was integer at the start and remains

Figure 3.3: Bifurcated-Flow Routing Algorithm (BFR)

Input: $\mathcal{I} = \{G = (V, E), \mathcal{D} = (s, d, b), \mathcal{F}\}$
Result: $x = \{H = (V, E), \mathcal{R}\} \in \mathcal{X}_{\mathcal{I}}$

1 **begin**
2 Initialize $\forall e \in E : b_e = 0$, $\mathcal{R}$ empty;
3 Unrouted capacity $b^u = b$;
4 **while** $b^u > 0$ **do**
5 Solve a non-bifurcated IGDP problem $\mathcal{I}_i = \{G = (V, E), \mathcal{D} = (s, d, 1), \mathcal{F}\}$;
6 $b^i := \min\{\min_{\forall e \in E, b_e^i = 1} k_e, b^u\}$;
7 Save $x_i = \{H_i = (V, E), \mathcal{R}_i\}$: $\forall e \in E, b_e^i = 1 : k_e := k_e - b^i, b_e := b_e + b^i$;
8 Add $\mathcal{R}_i$ to $\mathcal{R}$ for fraction b^i ;
9 $b^u := b^u - b^i$;
10 **end**
11 **end**

integer during the iterations. Thus, the algorithm terminates in at most b steps. Each part of the data uses a minimum cost lightpath in that iteration. Thus, the subgraphs $H_i = (V, E)$ with reserved capacity values b^i are a feasible solution. Furthermore, $\mathcal{R}$ exists, where the node roles can be derived from the set of configuration plans in each iteration. In each step i an IGDP problem was solved for data part b^i with non-bifurcated flows, thus, there exist a feasible configuration map $\mathcal{R}_i$ which is robust for each data part b^i. The configurations $\forall i : \mathcal{R}_i$ assigns the node roles (a)-(e) in the final solution $\mathcal{R}$.

Corollary 1 *Generalized dedicated protection problems without network coding comprise of at most b IGDP problem instances with $\mathcal{D} = (s, d, 1)$.*

Based on Corollary 1 the complexity of generalized dedicated protection without network coding relies in the IGDP problem in Step 5 of the algorithm in Fig. 3.3. As we mentioned that a resilient non-bifurcated solution is always robust and the configuration $\mathcal{R}$ can be derived straightforward from the node roles, we concentrate on finding a resilient subgraph $H = (V, E)$.

As we show that the IGDP is NP-complete, the well accepted approach in the literature [70] is followed, and IGDP is formulated as an Integer Linear Program (ILP) to find a minimal cost resilient subgraph $H = (V, E)$. Without loss of generality, the ILP formulation for traffic demand $\mathcal{D} = (s, d, 1)$ is presented. We assume that in the case of undirected $G = (V, E)$ each edge is replaced by two anti-parallel arcs. Furthermore, if a directed link (u, v) is in SRLG$_f$, then (v, u) will be in the same SRLG$_f$ as well.

Our goal is to minimize the following objective:

$$\min \sum_{\forall e \in E} c_e \cdot b_e.$$

The following constraints needed to formulate a IGDP solution:

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } 0 \leq b_{e,f} \leq 1, \tag{3.2}$$

$$\forall f \in \mathcal{F}, \forall i \in V\colon \sum_{\forall (i,j)\in E_f} b_{(i,j),f} - \sum_{\forall (j,i)\in E_f} b_{(j,i),f} = \begin{cases} -1 & , \text{ if } i = s \\ 1 & , \text{ if } i = d \\ 0 & , \text{ otherwise} \end{cases}, \tag{3.3}$$

$$\forall f \in \mathcal{F},\ \forall e \in E\colon b_{e,f} \leq b_e, \tag{3.4}$$

$$\forall f \in \mathcal{F},\ \forall e \in E\colon b_{e,f} \text{ are integers.} \tag{3.5}$$

The ILP formulation finds an optimal solution for IGDP problem on links with sufficient spare capacity, as links with $k_e < b$ were removed from input graph $G = (V, E)$. Based on the claim of Theorem 3.2.5, if links exist in $G = (V, E)$ with insufficient spare capacities, the R&R routing problem could have a solution with lower $g(y_{\mathcal{I}})$ minimal cost with the application of bifurcated flows (BGDP), and could be found with the algorithm in Fig. 3.3.

3.2.3 Fast Heuristic Approach for the Non-Bifurcated GDP [C1, C2, C3]

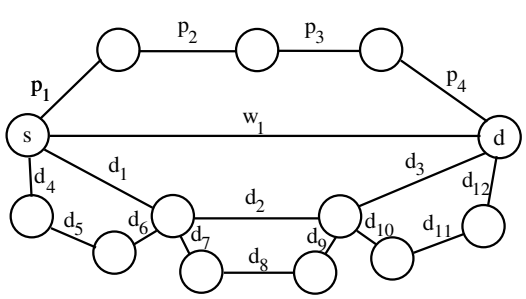
In order to find a feasible (not necessarily optimal) subgraph $H = (V, E)$ fast for the traffic demand $\mathcal{D} = (s, d, 1)$, a heuristic method was developed, called Dijkstra Heuristic. The idea of the heuristic is simple: find a shortest path $\forall f \in \mathcal{F}$ use Dijkstra's shortest path finding algorithm. In the jth iteration, the edges part of the shortest paths of SRLG graphs $G_i = (V, E_i), i = 1, 2, \ldots, j-1$ can be used with zero cost ($c_e = 0$). The final solution is obtained by add all edges to the solution found in the iterations. Detailed code of the algorithm is presented in the algorithm in Fig. 3.4.

Figure 3.4: Dijkstra Heuristic (DH)

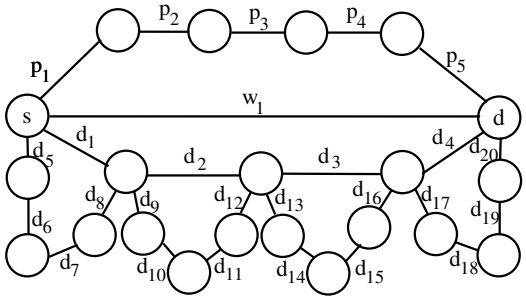
Input: $\mathcal{I} = \{G = (V, E), \mathcal{D} = (s, d, 1), \mathcal{F}\}$
Result: $x = \{H = (V, E), \mathcal{R}\} \in \mathcal{X}_{\mathcal{I}}$

1 **begin**
2 Initialize $\forall e \in E : b_e = 0$, $\mathcal{R}$ empty;
3 Initialize $\mathcal{F}' = \mathcal{F}$;
4 Initialize P empty ;
5 **while** $\mathcal{F}'$ *not empty* **do**
6 Take the next SRLG f from $\mathcal{F}'$;
7 Set $\forall e \in P : c_e = 0$ in $G_f = (V, E_f)$;
8 Find shortest path P_f in $G_f = (V, E_f)$ from s to d using Dijkstra's algorithm;
9 $\forall e \in P_f$: add edge e to P if $e \notin P$;
10 Remove f from $\mathcal{F}'$;
11 **end**
12 Set $\forall e \in H : b_e := 1$, if $e \in P$;
13 **end**

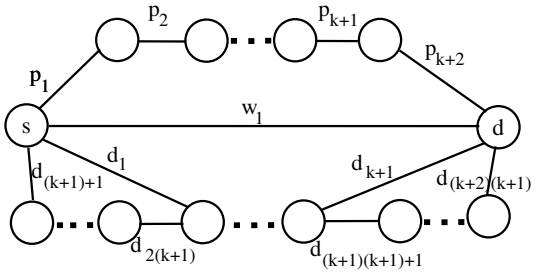
Although the heuristic has decent performance in some practical scenarios shown in Section 3.3, the solution quality depends on the order the $f \in \mathcal{F}$ failure patterns are considered in the algorithm. Even if the best (minimal cost $g(x)$ solution) SRLG order is known, the optimality of the heuristic is not guaranteed. On the other hand, as the reserved resources by the segments are self-shareable (Step 7), the



(a) For $k = 2$ the OPT is 5, DH is 13 > $k \cdot OPT$



(b) For $k = 3$ the OPT is 6, DH is 21 > $k \cdot OPT$



(c) For general k the OPT is $k + 3$, DH is $k^2 + 3k + 3$ > $k \cdot OPT$

Figure 3.5: The input graphs $G = (V, E)$ for the k-approximability counter example of the DH for $\mathcal{D} = (s, d, 1)$ and $\mathcal{F} = \{(p_i), (w_1), (w_1, d_1), (w_1, d_2), \ldots, (w_1, d_k)\}$, $\forall e \in E : c_e = 1$.

existence of the solution does not depends on the order the segments are considered. Hence, DH has the *trap avoidance property* as well, thus, only the quality of the solution depends on the order of the SRLGs are considered in the routing decision.

In the following, we show that the algorithm in Fig. 3.4 is not an approximation algorithm. An algorithm is a k-approximation for a given $k > 1$, if the cost of the solution for all inputs $\mathcal{I}$ of the problem is $\leq k \cdot OPT_I$, where OPT_I denotes the minimum cost solution for input $\mathcal{I}$ and runs in polynomial time. Computing a minimum-cost Steiner Tree is proved to be NP-hard [30] and APX-complete [8] [11]. A problem is called APX-complete, if no polynomial-time approximation scheme (PTAS) exists for it unless $P = NP$. A PTAS means that for any $\epsilon > 0$ there exists a polynomial-time $1 + \epsilon$-approximation algorithm. The Steiner Tree is a special case of the Steiner Forest problem, therefore, the Steiner Forest problem does not admit a PTAS [49]. However, the best approximation ratio for the Steiner Forest problem is 2 [2] (more precisely $2 - 1/k$ [49]). The algorithm in [2] use the primal-dual schema and achieves an approximation ratio of 2 (see also [31] for a more general algorithm and a simpler proof). As the non-bifurcated GDP problem is at least as hard as the (Directed) Steiner Forest Problem (because of the Karp-reduction), having a polynomial-time approximation algorithm within the factor of 2 is not probable.

I gave a family of graphs to demonstrate, that the solution of DH can grow above any constant k

with a special $\mathcal{F}$ SRLG list applied on the $G = (V, E)$ topology shown in Fig. 3.5 with traffic demand $\mathcal{D} = (s, d, 1)$. The optimal solution for the problem instance are the links labeled with w_1 and d_i, while the optimal solution are the w_1 and p_i links. In this construction, the solution $g(y_I)$ obtained with DH is $g(y_I) = k \cdot g(y^*{}_I) + 3$ for the corresponding k-graph, where $g(y^*{}_I)$ denotes the optimal solution.

Theorem 3.2.6 *The Dijkstra Heuristic does not k-approximate IGDP for any constant $k > 1$.*

Proof: If such a constant k exists, it means that DH gives a solution with lower cost than $k \cdot OPT$ for any input, where OPT denotes the cost of the optimal IGDP solution. To prove the lemma, an IGDP input will be created, where the heuristic solution has a greater cost than $k \cdot OPT$.

In Figure 3.5 a possible IGDP input graph $G = (V, E)$ is shown for $\mathcal{D} = (s, d, 1)$ traffic demand, while all of the edges have unit costs and unit spare capacities. The SRLG list is $\mathcal{F} = \{(p_i), (w_1), (w_1, d_1), (w_1, d_2), \ldots, (w_1, d_k)\}$.

The edges in the optimal IGDP solution are denoted with w and p in Figure 3.5, and the minimal cost is $OPT = 1 + (k + 2)$. The $g(y_{\mathcal{I}})$ cost of a solution provided by a k-approximation algorithm is $\leq k \cdot OPT = k^2 + 3k$. However, the Dijkstra Heuristic on the presented $\mathcal{I}$ input gives a minimum cost path in each SRLG auxiliary graph, thus, it gives w_1 for $f_1 = (p_i)$, the $d_1, d_2 \ldots, d_{k+1}$ route in the auxiliary graph $G_{f_2} = (V, E \setminus w_1)$. Furthermore, in the auxiliary graph $G_{f_{i+2}} = (V, E \setminus \{w_1, d_i\})$ DH finds the edges $d_{i\cdot(k+1)+1}, \ldots, d_{(i+1)\cdot(k+1)}$, $i = 1, \ldots, k$. The solution of the Dijkstra Heuristic is denoted with w and d in Figure 3.5, while the cost is $g(y_{\mathcal{I}}) = (k + 2) \cdot (k + 1) + 1 = k^2 + 3k + 3$.

Thus, the Dijkstra Heuristic gives a solution with greater cost on the input $\mathcal{I}$ than one of the family of $G = (V, E)$ k-graphs in Figure 3.5 than $k \cdot OPT$ for any constant $k > 1$. ■

Note, that these results are conforming to the observations made about the approximability of the (Directed) Steiner Forest problem, i.e. no PTAS exists for the problem. Furthermore, the best known approximation factor for (undirected) Steiner Forest is 2 [2]. As the NP-complete problems are reducible to each other, one could hope that an approximation algorithm for one problem would automatically, via the reduction, give an equally good approximation algorithm for other problems. However, NP-reductions are not strong enough to preserve all the structure of the problem. Thus, several reduction techniques can be applied (e.g. $\mathcal{L}$-reduction) to preserve a constant (not necessarily the same) approximation factor of a known approximation algorithm of another problem [45]. On the other hand, owing to the Karp-reduction of IGDP to the Steiner Forest problem, having a polynomial-time approximation algorithm within the factor of 2 is not probable. However, a 2-approximation algorithm for the IGDP problem with traffic demand $\mathcal{D} = (s, d, 1)$ is still an open question.

IGDP Dijkstra Heuristic as a Building Block of Further Protection Design

The simulation results showed that the heuristic is quite efficient in most cases, motivated by the claim of Theorem 3.2.6 the problem was further investigated for possible better solutions. As the performance of DH is highly influenced by the order of SRLGs considered in the algorithm (seen in the example in

Figure 3.5), I have investigated the performance of DH with different SRLG orders. Although the $g(y_I)$ solution is approaching $g(y^*{}_I)$, the growth in computational time required by the algorithm is not worth the gain witnessed in resource consumption. Furthermore, other heuristic approaches were investigated as a counterpart of DH (e.g. Lagrange relaxation [94] of the ILP formulation, using Suurballe's algorithm instead of Dijkstra's), but the trade-off between running time and solution quality was poor. Thus, DH is chosen as the main heuristic approach finding a feasible IGDP solution.

Furthermore, as one of the main contribution of the DH algorithm, it can be used as the building block of several heuristic and meta-heuristic approaches, for example:

- A meta-heuristic approach is proposed to find quasi-optimal solutions for IGDP [C7]. Based on the observations made on the simulation results, the proposed (**Bacterial Evolutionary Algorithm, BEA**) for problem instances containing large networks and long $\mathcal{F}$ SRLG lists seem to be efficient.
- The optimal ILP, as well as the DH method can be used for finding bifurcated solution (DH is used in Step 5 of the BFR algorithm, called (**BFR***). However, this approach is only worth if there does not exist a non-bifurcated solution, because of its heuristic nature.

3.2.4 GDP with Network Coding (GDP-NC) is Polynomial-Time Solvable [B1, *C6*]

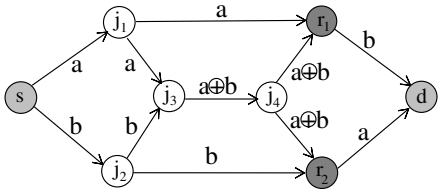
Network coding was first proposed in wireless networks and turned out to be efficient in different application environments, e.g. Delay-Tolerant Networks (DTN) [H1] and optical networks [B1]. In this section we present a Linear Program (LP) and a polynomial time coding to get the global optimum of the GDP-NC routing problem. The GDP-NC routing algorithm runs in polynomial-time for networks where the nodes are capable to perform algebraic operations on incoming signals (roles (a)-(f) in Fig. 3.2). Thus, GDP-NC owing to its computational complexity is acceptable for optimal on-line routing in networks with moderate sizes.

The process of network coding under the proposed GDP-NC scheme is divided into the following two subproblems (SBs):

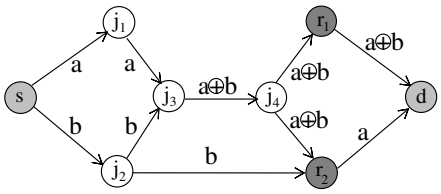
1. SB-I: find a subgraph $H = (V, E)$ which is resilient against all failures in $\mathcal{F}$ and will be utilized in network coding,
2. SB-II: find the algebraic operations $\mathcal{R}$ which the nodes must perform to realize a robust code on $H = (V, E)$.

Linear Program (LP) for SB-I

In the following a linear program (LP) is presented for SB-I where the subgraph $H = (V, E)$ is found in polynomial-time. Note that in [55] a linear program was shown for multicast demands. However, because of the different available link sets in $G_f = (V, E_f)$ it is not directly applicable for GDP-NC.



(a) In the final solution, both a and b reach the destination node.



(b) The GDP solution is resilient and robust against all failures in $f \in \mathcal{F}$, e.g. after $f = (j_1, r_1)$ SRLG failed.

Figure 3.6: A possible LP solution $H = (V, E) \in \mathcal{X}_\mathcal{I}$ for the instance $\mathcal{I} = \{G = (V, E), \mathcal{D} = \{s, d, 2\}, \mathcal{F} = \{(j_1, r_1), (j_2, r_2), (j_3, j_4)\}\}$ containing the butterfly graph with recievers r_1 and r_2. On each link $b_e = 1$, the data sent on each BU is denoted by a and b.

Our goal is to obtain a solution $x \in \mathcal{X}_\mathcal{I}$ which minimizes the cost $g(x)$ in Eq. (3.1). The following constraints are required:

$$\forall f \in \mathcal{F},\ \forall e \in E\colon 0 \leq b_{e,f} \leq \min\{b, k_e\}, \tag{3.6}$$

$$\forall f \in \mathcal{F}, \forall i \in V\colon \sum_{\forall (i,j) \in E_f} b_{(i,j),f} - \sum_{\forall (j,i) \in E_f} b_{(j,i),f} = \begin{cases} -b & , \text{ if } i = s \\ b & , \text{ if } i = d \\ 0 & , \text{ otherwise} \end{cases}, \tag{3.7}$$

$$\forall f \in \mathcal{F},\ \forall e \in E\colon b_{e,f} \leq b_e. \tag{3.8}$$

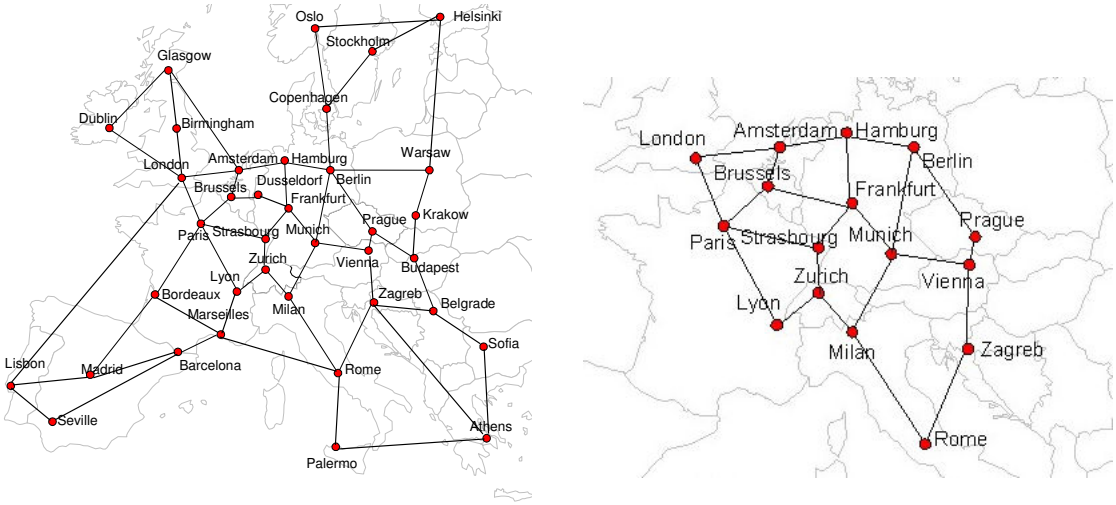
Note that in the above formulation each undirected edge is replaced by two anti-parallel arcs.

The solution $H = (V, E)$ of the LP above can minimize the consumed bandwidth while being resilient against all possible SRLG failures in $\mathcal{F}$, as b units of optical flow eventually reach the destination in each auxiliary SRLG graph. In addition to the topology, we further need to obtain a robust configuration map $\mathcal{R}$.

If the LP solution contains $0 < b_e < b$ for any edge $e \in E$ (i.e. the solution is bifurcated), the existence of a feasible configuration map $\mathcal{R}$ is only guaranteed if the network nodes are able to combine linearly the incoming signals. On the other hand, if the OXCs are unable to combine the incoming signals, it is possible that $x = \{H, \mathcal{R}^H\} \notin \mathcal{X}_\mathcal{I}$ for any configuration map $\mathcal{R}^H$. An example for the above statement is given as follows. Let us consider $H = (V, E)$ obtained in Fig. 3.6(a) which is resilient against the SRLG failures $\mathcal{F} = \{(j_1, r_1), (j_2, r_2), (j_3, j_4)\}$. One can easily check that the presented configuration $\mathcal{R}$ is robust against all failures, and only the decoding matrix at the destination node need to be updated upon failures, i.e. $x = \{H, \mathcal{R}\} \in \mathcal{X}_\mathcal{I}$.

Finding Robust Network Codes (SB-II) for the LP Solution

In this section we show a polynomial-time algorithm to realize the robust algebraic operations $\mathcal{R}$ for the LP solution $H = (V, E)$ derived in the previous subsection. The network coding operations of an



(a) COST266 37-node Large Topology (LT)

(b) COST266 16-node Core Topology (CT)

Figure 3.7: COST266 European Reference Backbone Networks [57]

example solution $H = (V, E)$ is shown in Fig. 3.6, where the well-known butterfly graph for demonstrating network coding is extended for protection purposes. Let there be three SRLGs in the network: $\mathcal{F} = \{(j_1, r_1), (j_2, r_2), (j_3, j_4)\}$. Similarly to [47], we assume that zeros received along the failed links after an SRLG failure occurs. Thus, for Subproblem 2, the results in [47] provide us the opportunity to find robust network codes in polynomial time for a resilient bifurcated solution. Either random network codes proposed in [38] or network codes presented in [43] can be used. However, here is the latter is shown only.

As $H = (V, E)$ found in SB-I is resilient against all failures in $\mathcal{F}$, it satisfies the requirements of Theorem 11 in [43] for the set of connections $\mathcal{C} = \{(s, d)\}$, which ensures that none of the SRLG failures could possibly reduce the bandwidth below $k = b$. As a corollary, it is possible to get robust network codes with the method proposed in [43] to obtain robust multicast codes for a resilient generalized dedicated protection subgraph in polynomial time. Thus, in the presence of any failure, only the destination node needs to use different decoding matrix to get the data back, as shown in Fig. 3.6(b).

We note that finding appropriate network codes in SB-II takes complexity of $O(|\mathcal{F}| \cdot |E| \cdot (|\mathcal{T}| \cdot k^2 + \min\{I, |\mathcal{F}| \cdot |\mathcal{T}|\} \cdot k))$ with the method proposed in [43]. With our notation system, a flow with a value b reaches the destination in any SRLG f, which gives $k = b$. Since the number of failure patterns $|\mathcal{F}|$ is the same, while we have a single receiver $|\mathcal{T}| = 1$, thus, the complexity of Subproblem 2 is $O(|\mathcal{F}| \cdot |E| \cdot (b^2 + \min\{I, |\mathcal{F}|\} \cdot b))$, which is nonetheless overshadowed by the complexity of finding a resilient subgraph in SB-I.

Table 3.3: The number of SRLGs in $\mathcal{F}$ for the type (3) SRLG scenarios in the COST266 networks.

Scenario	Single	Low	Medium	High
SRLG number in COST 266 LT	57	74	137	202
SRLG number in COST 266 CT	23	27	43	60

3.3 Simulation Results

3.3.1 Input Parameters

Experiments on the Cost 266 European reference networks [57] in Figure 3.7 were conducted to verify the proposed algorithms and compare them with previously reported counterparts. The larger network in Fig. 3.7(a) has 37 nodes, 57 edges, the average nodal degree of the topology is 3.08 (min. 2, max. 5), and the average edge connectivity is 2.54 (min. 2, max. 4), while the smaller core network in Fig. 3.7(b) has 16 nodes, 23 edges, the average nodal degree of the topology is 2.88 (min. 2, max. 4), and the average edge connectivity is 2.47 (min. 2, max. 4).

For the sake of simplicity, the cost of each edge in the topology is unit. In Section 3.3.2 the bandwidth consumption of the proposed methods is compared; thus in the simulations edge capacities were set high enough to possibly route all connections. In Section 3.3.3, the blocking probability is compared, thus in the simulations the number of wavelength channels was set to 32, and the intensity of the arriving traffic demands was increased from 40 to a few hundreds of Erlangs.

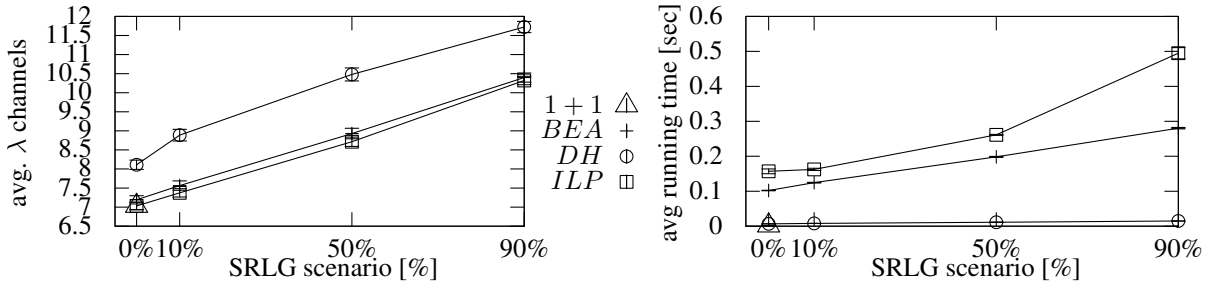
Incoming requests follow Poisson arrival process with exponentially distributed time duration with mean of 5 time units. Capacity demands of the connections are uniformly distributed between 1 and 2 OC-192. Each data was obtained by averaging the results of 5 different traffic patterns each containing 200 traffic demands between randomly chosen source and destination nodes.

We define the *density of type (3) SRLGs* (see also in Section 2.4) as the percentage of adjacent multi-link SRLGs and node-failures (i.e., all links connected to a node). In all the scenarios, all the single-link SRLGs as well as a given percentage of multiple-link SRLGs are considered. For example, if the density of SRLGs is given as $p\%$, then $p\%$ of adjacent dual-link SRLGs and $p\%$ of deg-degree node failures are selected ($deg \geq 3$) as SRLGs. If only single-link failures are considered ($p = 0$), we refer to it as *single failure scenario*. In the *low, medium and high SRLG scenarios* p *is chosen to* $10, 50$ *and* 90, respectively. The number of SRLGs in $\mathcal{F}$ is presented in Table 3.3.

With different required QoS levels, the schemes are implemented and compared in terms of the following three performance metrics:

(1) the wavelength channels used by the demands,

(2) the blocking probability in different SRLG scenarios and traffic loads,



(a) Average λ channels

(b) Average running time

Figure 3.8: The average reserved wavelength channels by 200 requests versus the SRLG scenario in the **16-node network**. Note that $1 + 1$ could protect all failures in $\mathcal{F}$ only in the single-link failure scenario (0%).

(3) running time.

Note that finding an R&R dedicated protection approach for a fair comparison with the GDP methods is a difficult task. From the enumerated schemes in Table 3.1, the $1 + 1$ path protection seems to be the best choice, as it is widely deployed, and could serve as a reference for service providers. On the other hand, the DLP and DSP approaches are always outperformed by GDP owing to the self-sharing of the protection resources among segments. The MPP is clearly not an R&R approach, while $1+N$ protection was omitted from the simulations, because it was designed for static routing (furthermore it has enormous connectivity requirement, that is not present in the investigated networks in Fig. 3.7).

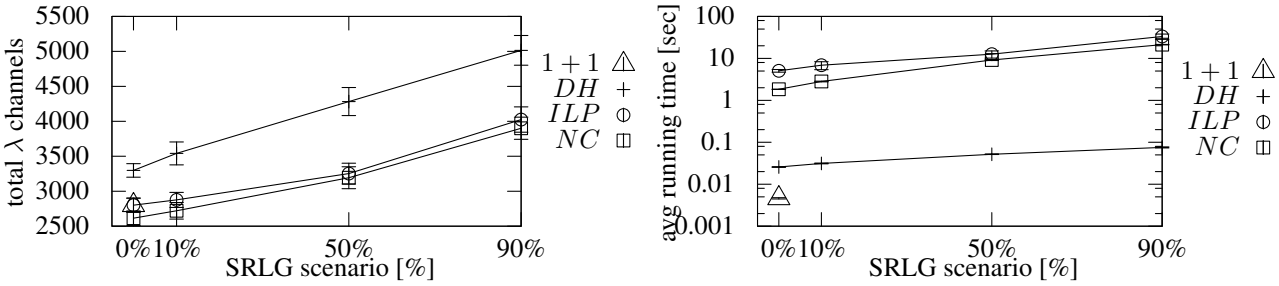
In the simulations in Section 3.3.2 $1 + 1$ finds only link-disjoint paths in order to avoid blocking because of no SRLG-disjoint path-pair exist. In Section 3.3.3 we used a two-step approach for finding SRLG-disjoint path-pair for $1 + 1$ protection. First, a working path is found as the shortest path in the network, and as a second step, an SRLG-disjoint path with the working path is given, if exists.

The approaches shown in the figures are the optimal solution of IGDP without network coding, the optimal GDP-NC solution with network coding (NC in figures), and the heuristic method DH for the non-bifurcated dedicated protection problem.

3.3.2 Bandwidth Requirement with Light Traffic Load

In Fig. 3.8 the effect of the different sparse-SRLG scenarios are investigated on the performance of the IGDP methods. As one can observe, the bacterial meta-heuristic approach BEA performs close to the optimum in all scenarios, while the running time is slightly less in all cases.

In the high SRLG scenario (90%) the advantages of IGDP are clearly demonstrated, which is due to the flexibility in choosing the appropriate node roles in the network (shown in Fig. 3.2). As the $1 + 1$ blocks almost all connection in the high SRLG scenario owing to the lack of SRLG disjoint path-pairs, it cannot be used as a reliable protection method. On the other hand, a link-disjoint (but not SRLG



(a) Total reserved wavelength channels

(b) Running time

Figure 3.9: The total reserved wavelength channels and average running time is shown versus the SRLG scenario by 200 requests in the **37-node network**. Note that $1 + 1$ could protect all failures in $\mathcal{F}$ only in the single-link failure scenario (0%).

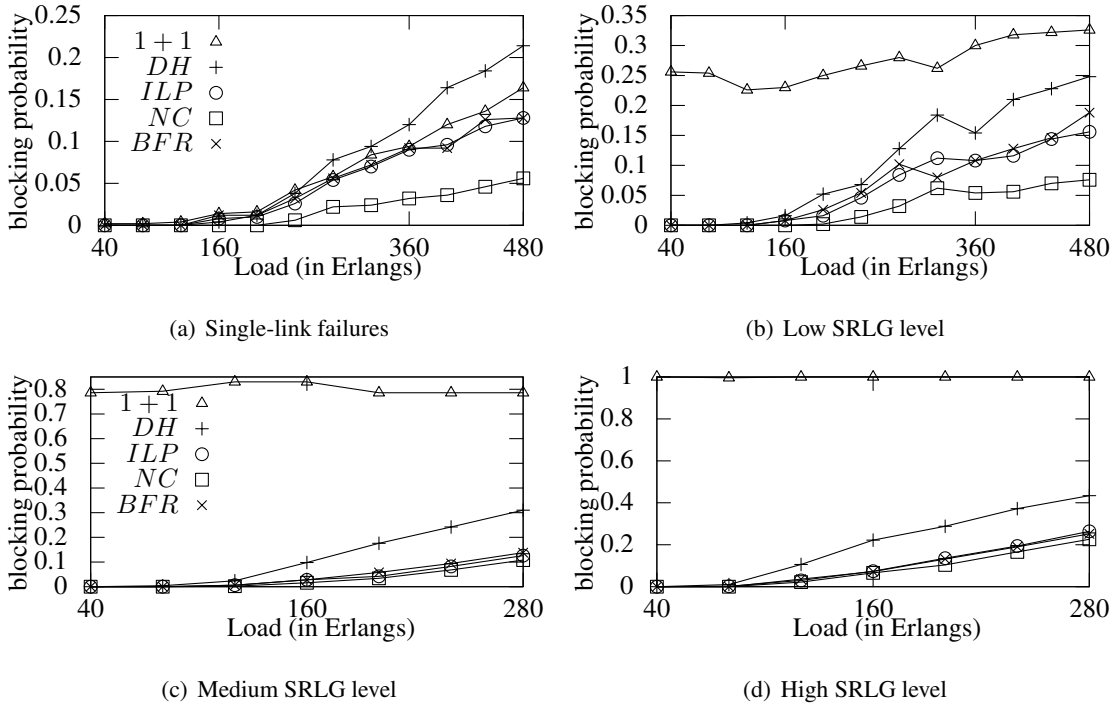
disjoint) path pair would use 7 wavelength channels, but this approach is not resilient against dual failures affecting both the working and the protection path. However, reserving 3 more λ for the connection in the IGDP approaches all SRLGs in $\mathcal{F}$ are protected and the strict QoS requirements declared in the SLA are satisfied.

The result on the total number of used wavelength channels in a non-blocking network scenario for one direction is shown in Fig. 3.9. First, we find that the number of used channels with generalized dedicated protection methods with and without network coding increases when the number of SRLGs in $\mathcal{F}$ grows, which meets our expectation.

It can be observed, that with the application of network coding in the single-link failure scenario (0%) GDP-NC uses less resources than $1 + 1$ protection. For the sake of explanation, we assume that three disjoint paths exists between s and d, each with length l in the network. In this scenario $1 + 1$ protection use total $2 \cdot l \cdot b$ bandwidth unit. On the other hand, if network coding is applied in GDP, the optimal solution is reserving $b/2$ bandwidth along each of the three paths. Such a solution is robust and resilient against single link failures in $\mathcal{F}$, furthermore, its bandwidth consumption is $3/2 \cdot l \cdot b$.

In the low, medium and high sparse-SRLG scenarios GDP-NC provides the best performance in terms of reserved bandwidth while the connection is resilient and robust against all $f \in \mathcal{F}$ (which is not in the case of $1 + 1$ protection). Also from Fig. 3.9, we find that the application of network coding can slightly outperform the non-bifurcated optimal IGDP method in bandwidth consumption and running time. However, GDP-NC is polynomial-time solvable.

Note, that not for all connections exist SRLG-disjoint path in the network, thus, $1 + 1$ protection blocks a certain amount of connections in the different scenarios for which $\exists f \in F$ is not protectable. In the case the connection would be admitted in the network without protecting each $f \in \mathcal{F}$, it would not fulfill the quality of service requirements of the service provider. For example, one of the main end-to-end service requirements of a connection is the availability. In the case of the non-bifurcated IGDP and $1 + 1$ protection we compared the availability of the solutions, evaluated by the pivotal decomposition



(a) Single-link failures

(b) Low SRLG level

(c) Medium SRLG level

(d) High SRLG level

Figure 3.10: The steady state blocking probability of 100 requests in the **37-node network**.

formula [67]. We found that with protecting all $f \in \mathcal{F}$ the 200 connections routed with the IGDP method has 0.999993 availability on average, while the minimal connection availability is 0.999971 (about 15 minutes outage per year) for the high SRLG level. On the other hand, as $1 + 1$ cannot provide SRLG-disjoint paths for all connections they perceive 0.999987 availability on average, while the minimal availability is 0.9988 (more than 10 hours outage per year), which clearly does not meet the strict QoS requirements declared in the SLA.

3.3.3 Blocking Probabilities with Heavy Traffic Load

The blocking probability of the methods was investigated with different traffic loads. The results are shown in Figure 3.10. We plotted the blocking probability of the generalized dedicated protection approaches as the increase of the traffic load in Erlangs. Note, that the blocking of $1 + 1$ is twofold: there are insufficient resources to route the connection or no SRLG-disjoint pair of paths exist. On the other hand, all other generalized dedicated protection methods in the simulations are blocking a connection because of insufficient resources. In this simulation scenario a connection is admitted iff it is resilient and robust against all failures in $\mathcal{F}$.

In order to investigate the network in a steady state, the blocking probability was measured only after 100 connections were routed.

As shown in Fig. 3.10(a) and (b), GDP with network coding has significant savings in terms of blocking if the number of SRLGs in $\mathcal{F}$ is low. As expected, GDP-NC can always yield no larger blocking

probability than the other schemes due to the fact that with network coding the network nodes can perform more operations than without coding. It is clearly observed that GDP-NC outperforms the other schemes in all the cases, especially with high traffic loads, where the saving in blocking probability is about 5% because of the application of the more complex network nodes. Further, $1 + 1$ protection is outperformed by all other generalized dedicated protection schemes due to the large diversity of node roles allowed (Fig. 3.2). Thus, the GDP methods with and without network coding can better explore the design space of the routing problem.

Fig. 3.10(c) and (d) shows the blocking probability versus the traffic load with $\mathcal{F}$ list containing 137 and 202 SRLGs, respectively. Recall that when more than 50% of the dual link SRLGs are considered in $\mathcal{F}$ (medium and high scenarios), $1 + 1$ protection blocks almost all connection because of the lack of SRLG-disjoint paths. It is observed, that the clear advantage of network coding is disappeared in these situations. In fact, with non-bifurcated flows similar blocking probability can be achieved as with a more complex network structure with network coding.

Chapter 4

Unambiguous SRLG Failure Localization

4.1 Challenging Issues in Failure Localization of All-Optical Networks

4.1.1 Principles of Failure Localization with Supervisory Lightpaths in All-Optical Networks

Unambiguous Failure Localization (UFL) for shared risk link groups serves as a critical task for achieving fast and failure-dependent traffic restoration in all-optical mesh networks. It has been considered as a very difficult job due to the transparency in the optical domain incorporated with various design requirements [56,88,103]. One of the most challenging issues is failure propagation when a failure event occurs, where an upstream link or node failure may generally trigger redundant alarms by the monitors equipped at the downstream nodes. Besides, a failure at the optical layer (such as a fiber cut) may trigger alarms in networking as well as other upper protocol layers [23]. It is reported that a single fiber-cut with 16 disrupted wavelengths could lead to hundreds of alarms in the network [56]. This not only increases management cost of the control plane, but also makes the failure localization difficult.

Without loss of generality, the device that monitors the health of a certain part of the network is called a *monitor*, which generates an alarm if it detects any status change from the monitoring result. An alarm is broadcast in the control plane via a link state protocol if any irregularity is identified, such as Open Shortest Path First (OSPF), so that remote routing entities can receive the alarm. Upon failure of an SRLG, multiple alarms could be flooded in the network. A UFL solution requires each remote routing entity (or the centralized network manager) to be able to unambiguously identify the failed SRLG by collecting the flooded alarms. *To simplify the failure management and operational complexity, it is critical to reduce the number of monitors (i.e., the flooded alarms) without sacrificing the accuracy in failure localization.*

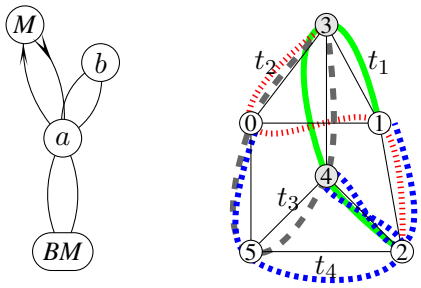
The simplest method for UFL is via *link-based* approach, where each link is closely and exclusively monitored. Obviously, the link-based approach requires the number of monitors in $O(|E|)$, where $|E|$ is the number of links in the network. Although simple, the link-based approach leaves a large space

to improve in terms of the number of monitors. Note that every monitor corresponds to not only an additional hardware cost, but also extra control overhead that the carrier has to bear. To reduce the number of monitors, people have turned to using multi-hop *supervisory lightpaths (S-LPs)* for failure localization. In the past, related studies using various monitoring structures, including monitoring-cycles (m-cycles), m-paths, and m-trails, etc., have been extensively reported [4, 5, 35, 61, 76, 93, 95, 99, 102]. A detailed comparison and descriptions can be found in [97].

The bi-directional m-trail (bm-trail) approach has been reported to be the most general supervisory lightpaths, which intends to explore the network topology diversity the best with its extreme flexibility in routing structure. A bm-trail can be a non-simple path/cycle with loop-back switching which allows a node to be traversed by multiple times and a link twice (along both directions). With bm-trails, the transmitter and receiver can be allocated at any node pair or co-allocated at a common node along the bm-trail. The receiver is equipped with a monitor which issues an alarm in the event of an unexpected and abrupt status change of the corresponding S-LP. An example is shown in Fig. 4.1(a), where the *transmitter and receiver* of the bm-trail is palced at node M, and the S-LP is $M \rightarrow a \rightarrow b \rightarrow a \rightarrow BM \rightarrow a \rightarrow M$. Note that the lightpath forms a closed trail with loopback switching at node b and BM. It will not affect the monitoring result by having different connection patterns on a set of links or different locations for the transmitter and receiver, because we only care about whether the S-LP is disrupted or not. Since a failure event could hit M or all the adjacent links of M so as to prevent it from issuing an alarm as expected, it is necessary to have a backup monitoring node BM which can monitor the status of M, such that once a failure on M is identified, BM will issue the alarm instead. Note that this is not a trivial task since BM will issue an alarm only when the failure of M is unambiguously identified by it (see Section 4.2.5 for details).

Fig. 4.1 shows an example of bm-trail solution for localizing any single link failure, where a *link code matrix* ($\underline{\underline{\mathbf{A}}}$) is shown in Fig. 4.1(c). The $\underline{\underline{\mathbf{A}}}$ matrix keeps the link code of each link (e.g., link $(3, 4)$ is assigned a link code 1010), which further defines how the four bm-trails (i.e., t_1, t_2, t_3, and t_4) should be routed in the topology to achieve UFL. Here, t_j has to traverse through all the links with the jth bit of the link code as 1 while avoiding to take any link with the jth bit of its link code as 0. By reading the status of the four bm-trails, any link failure can be unambiguously localized. For example, the unexpected darkness of t_1 and t_3 depicts the failure of $(3, 4)$.

Without loss of generality, the target of the (b)m-trail design problem is to minimize the total cost $g(y_{\mathcal{L}})$, which stands for the weighted sum of *monitoring cost* and *bandwidth cost*. The monitoring cost generally accounts for the fault management complexity in terms of the number of S-LPs. The bandwidth cost reflects the additional bandwidth consumption for monitoring, which is measured in terms of the *total cover length* of a (b)m-trail solution (i.e., the sum of the length of each (b)m-trail in the solution). Note that the *length* of a (b)m-trail is the number of links traversed by the (b)m-trail without loss of generality; and the number of (b)m-trails is the same as the length of the alarm code of each link. Formally, the target function employed in the dissertation is expressed as:



Link	t_1	t_2	t_3	t_4
(3,4)	1	0	1	0
(2,5)	0	0	0	1
(2,4)	1	0	0	1
(1,3)	1	0	0	0
(1,2)	0	1	0	1
(0,3)	0	1	1	0
(0,5)	0	0	1	1
(4,5)	0	0	1	0
(0,1)	0	1	0	0

(a) Bm-trail (b) Topology (c) Link code matrix $\underline{\underline{\mathbf{A}}}$

Figure 4.1: Fast link failure localization based on bm-trails.

$$g(y_{\mathcal{L}}) = \textit{monitoring cost} + \textit{bandwidth cost} = \gamma \cdot \#\textit{(b)m-trails} + CL, \tag{4.1}$$

where CL is the total cover length of a (b)m-trail solution (i.e. the number of used wavelength channels). The *cost ratio* γ scales the relative importance of the number of (b)m-trails in the solution. In the dissertation the monitoring cost (the number of (b)m-trails) is emphasized. Such a design is motivated by the observation that the number of (b)m-trails in the network serves as the major overhead of fault management [61, 100]. Thus, in the following context $\gamma = 1000$. The notations used in the problem formulation of unambiguous failures localization with supervisory lightpaths is summarized in Table 4.1.

In general, an effective monitoring structure allocation method must satisfy the following two requirements, either in a single step or one after the other:

(R1): Every SRLG should be uniquely coded.

(R2): Each monitoring structure must be an eligible fragment of network topology in which a lightpath can travel along from the transmitter to the receiver.

In summary, the unambiguous failure localization design problems for supporting phase (ii) of optical fault management in the dissertation aim at finding a set of supervisory lightpaths (m-trails or bm-trails) in the network. With each (b)m-trail terminated by an optical monitor, an alarm is issued as soon as the corresponding (b)m-trail becomes dark (or Loss of Light, LoL). By collecting the alarms issued by all the monitors subject to LoL, a remote routing entity (or the central network manager) should be able to unambiguously identify the failed SRLG.

Connection Between Physical Monitoring and the Logical SRLG Model

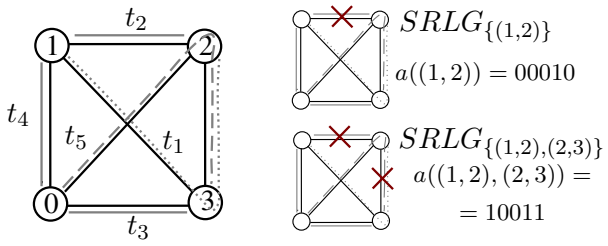
We assume at most one SRLG in the network could be in failure at a given time instant. The failure of an SRLG means all the links of the SRLG are failed. An example is shown in Fig. 4.2(a), where

Table 4.1: Notation list for the M-trail Allocation Problem (MAP)

Notations	Description
$G = (V, E)$	the undirected (or directed) graph representation of the topology with nodes V and edge set E
$\underline{a}_e^T$	link code vector of $e \in E$
$\underline{\underline{\mathbf{A}}}$	link code matrix formed from row vectors $\underline{a}_e^T, \forall e \in E$
$a(e_1, e_2, \ldots, e_d)$	function, returns the bitwise OR of link code vectors $\underline{a}_{e_i}^T, 1 \leq i \leq d$
$\underline{\underline{\mathbf{ACT}}}$	alarm code matrix, the row corresponding to $SRLG_{\{e_1,e_2,\ldots,e_d\}}$ is defined with the function $a(e_1, e_2, \ldots, e_d)$
J	number of columns in $\underline{\underline{\mathbf{A}}}$ (number of (b)m-trails in the solution)
$a_{e,j}$	j^{th} bit position in link code $\underline{a}_e^T$
$a_{\{e\},j}$	j^{th} bit position in the alarm code of $SRLG_{\{e\}}$
$\mathcal{L} = \{G, \mathcal{F}\}$	instance of the failure localization problem
$\mathcal{X}_{\mathcal{L}}$	the set of feasible monitoring solutions $y_{\mathcal{I}}$ for the instance $\mathcal{I}$
$y_{\mathcal{L}} = \{\underline{\underline{\mathbf{A}}}\}$	a feasible $\underline{\underline{\mathbf{A}}}$ link code matrix that ensures alarm code uniqueness in $\underline{\underline{\mathbf{ACT}}}$ and each column corresponds to a single trail

$SRLG_{\{(1,2)\}}$ and $SRLG_{\{(1,2),(2,3)\}}$ are two SRLGs. The failure of $SRLG_{\{(1,2),(2,3)\}}$ is a simultaneous failure of both link $(1, 2)$ and $(2, 3)$, and in this case the $SRLG_{\{(1,2)\}}$ is not taken as failed even if link $(1, 2)$ fails physically. This is a common assumption which has also taken by all the previous research [4,35] and [J2]. We assume that a set of $\mathcal{F}$ SRLGs is given in advanced according to the carrier's premise. In specific, we focus on the sparse (type (3)) SRLG scenario that the SRLGs with single link and with multiple links (mainly node failures and links adjacent to a common node) are monitored and unambiguously identified. This is motivated by the observation that geographically adjacent links are usually deployed in a common conduit for some distance, and they are subject to a common risk of being cut [18, 60]. Therefore, it is more efficient and realistic to simply monitor those SRLGs with a high likelihood of failure, instead of generally considering all the SRLGs with up to d links (type (2) SRLG scenario). Note that the number of SRLGs is a dominating factor that determines the control complexity and operational overhead of fault management in a communication carrier backbone.

Note that an SRLG is a logical entity assigned to one or a set of links corresponding to a failure event, while any monitoring technique is on physical links. Thus, the mapping between the code of each link and the code of each logical SRLG becomes a non-trivial problem. In the scenario of multi-link SRLGs with an arbitrary number of links, the alarm code of a multi-link SRLG should be the *bitwise OR* (denoted as `OR`) of the codes of all links in the SRLG. An example is given in Fig. 4.2, which shows that the failure of logical entity $SRLG_{\{(1,2),(2,3)\}}$ will trigger alarms of the (b)m-trails traversing through



(a) Topology and m-trails

$$\underline{\underline{\mathbf{A}}} = \left\{ \begin{array}{c} m-trails \\ \hline \underline{a}^T_{(0,2)} \\ \underline{a}^T_{(0,1)} \\ \underline{a}^T_{(0,3)} \\ \underline{a}^T_{(1,2)} \\ \underline{a}^T_{(1,3)} \\ \underline{a}^T_{(2,3)} \end{array} \right\} = \left\{ \begin{array}{ccccc} t_5 & t_4 & t_3 & t_2 & t_1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{array} \right\}$$

(b) Link code matrix

SRLG	Derived code $a(f_1, f_2)$	Alarm code matrix
{(0,2)}	$\underline{a}^T_{(0,2)}$	1 0 0 0 0
{(0,1)}	$\underline{a}^T_{(0,1)}$	0 1 0 0 0
{(0,3)}	$\underline{a}^T_{(0,3)}$	0 0 1 0 0
{(1,2)}	$\underline{a}^T_{(1,2)}$	0 0 0 1 0
{(1,3)}	$\underline{a}^T_{(1,3)}$	0 0 0 0 1
{(2,3)}	$\underline{a}^T_{(2,3)}$	1 0 0 0 1
{(0,2), (2,3)}	$\underline{a}^T_{(0,2)}$ OR $\underline{a}^T_{(2,3)}$	1 0 0 0 1
{(1,3), (2,3)}	$\underline{a}^T_{(1,3)}$ OR $\underline{a}^T_{(2,3)}$	1 0 0 0 1
{(1,2), (2,3)}	$\underline{a}^T_{(1,2)}$ OR $\underline{a}^T_{(2,3)}$	1 0 0 1 1
...	...	...
{(0,3), (1,3), (2,3)}	$\underline{a}^T_{(0,3)}$ OR $\underline{a}^T_{(1,3)}$ OR $\underline{a}^T_{(2,3)}$	1 0 1 0 1

(c) SRLG alarm code table (ACT)

Figure 4.2: Mapping between the physical monitoring structure and custom-defined logical SRLGs.

link $(1, 2)$, $(2, 3)$ or both. Therefore, the code for an SRLG should be a function which returns the OR of the corresponding rows $\underline{\underline{\mathbf{A}}}$ of all the links in the SRLG:

$$a(f_1, f_2, \ldots, f_d) = \underline{a}^T_{f_1} \texttt{OR}\ \underline{a}^T_{f_2} \texttt{OR}\ \ldots\ \texttt{OR}\ \underline{a}^T_{f_d}. \tag{4.2}$$

Trivially, for a single-link SRLG f_1, we have:

$$a(f_1) = \underline{a}^T_{f_1}. \tag{4.3}$$

With Eq. (4.2) and Eq. (4.3), the code uniqueness of all the SRLGs can be ensured if the OR of link codes of each SRLG is distinguished from that of all the other considered single-link and multi-link SRLGs in the network. The derived SRLG codes form the *alarm code matrix* $\underline{\underline{\mathbf{ACT}}}$.

Code Assignment (R1)

In the code assignment stage, each link is firstly assigned a unique code, such that the row vectors in $\underline{\underline{\mathbf{A}}}$ is distinguished from each other. This is a necessary condition for the considered scenario but it is only a sufficient one when only single-link failures are monitored, where any pair of link codes should be distinguishable as long as their Hamming distance is no smaller than 1. When any multi-link SRLG should be monitored, we need to additionally consider the dependency of the codes of the multi-link SRLGs upon the code of each link.

An example for SRLGs with multiple adjacent links on a 4-node full-mesh graph is provided as shown in Fig. 4.2. The matrix $\underline{\underline{\mathbf{A}}}$ and the corresponding $\underline{\underline{\mathbf{ACT}}}$ is given in Fig. 4.2(b) and Fig. 4.2(c), respectively, where each single-link SRLG is assigned with a unique code as shown in the former, while the resultant $\underline{\underline{\mathbf{ACT}}}$ by jointly considering all the single- and multi-link SRLG is shown in the latter. In theory, we need a set of codes with at least a length $J \geq 5$ bits to unambiguously distinguish totally 22 SRLGs (i.e., 6 single-link and 16 multiple adjacent link SRLGs). Thus, the $\underline{\underline{\mathbf{ACT}}}$ is a $|\mathcal{F}| \times J$ binary matrix, which is composed of two parts concatenated with each other: the upper part is the $|E|$-by-J link code matrix, and the lower part is a $|\mathcal{F}| \setminus |E|$)-by-$J$ binary matrix corresponds to the multi-link SRLGs.

For achieving UFL, each row of $\underline{\underline{\mathbf{ACT}}}$ should be unique. With overlapping multi-link SRLGs, further manipulation on each row of $\underline{\underline{\mathbf{A}}}$ is necessary; otherwise the uniqueness of the rows in $\underline{\underline{\mathbf{ACT}}}$ will not be ensured. As shown in Fig. 4.2(c), the codes for $SRLG_{\{(2,3)\}}$, $SRLG_{\{(0,2),(2,3)\}}$ and $SRLG_{\{(1,3),(2,3)\}}$ are unfortunately identical (i.e., 10001), which makes UFL not possible.

Codes with Special Property for Supporting (R1)

As we have seen, the connection between the link codes in $\underline{\underline{\mathbf{A}}}$ and the alarm codes in $\underline{\underline{\mathbf{ACT}}}$ is a non-trivial problem. While monitoring and deployment of (b)m-trails is done on the physical links (corresponding to $\underline{\underline{\mathbf{A}}}$), the code uniqueness in $\underline{\underline{\mathbf{ACT}}}$ have to be ensured (between the bitwise `OR` of the link codes). It is a straightforward idea to apply codes having the property of uniqueness between the bitwise `OR` of the codes. Fortunately, there exists such a family of codes, called *non-adaptive combinatorial group testing (CGT)*.

Formally in non-adaptive combinatorial group testing, we are given a set of n items (links of the network), at most d of which are defective (the SRLGs with up to d links), and we are interested in identifying exactly which of the n items are defective [27]. Instead of testing each item individually (i.e., link-based monitoring), we can divide the items into different subsets in combination and test each subset as a single group (i.e., the links in a (b)m-trail), in order to unambiguously identify the defective items (i.e., the failed SRLG). A test of a group is positive (the result is represented as 1) if any of the items are defective in the group, and negative (0) if none of the items is defective. With a non-adaptive CGT construction, the group tests (i.e., (b)m-trails) are determined and are launched simultaneously without any mutual interaction, for which the test outcomes (e.g. 0010110) unambiguously identify the defective

Table 4.2: Minimal CGT code length for a 100 edge network generated with the `bktrk` in [27]

Failure num (d)	2	3	4	5
Minimal code length	36	60	89	131

items (the failed SRLG). There are two code types, which identify exactly which of the n items are defective if we assume that at most d items are defective. A matrix $\underline{\underline{\mathbf{A}}}$ is *d-disjunct* if the bitwise OR of any d columns does not contain any other column (corresponding to the characteristic vectors of the codes), and $\underline{\underline{\mathbf{A}}}$ is *$\bar{d}$-separable* if the bitwise OR of up to d columns are all distinct [25].

Because of the special structure, the length of the CGT codes can grow above the number of links, which means its application is inefficient (even link-based monitoring outperforms it in terms of $g(y_{\mathcal{L}})$). Table 4.2 contains the length of the CGT codes for a 100 edge networks in the case all multiple-link failures up to d links have to be localized. For failures $d = 2, 3$ and 4 with the application of CGT codes the number of (b)m-trails can be reduced in comparison with link-based monitoring (100 (b)m-trails). However, above $d = 5$ the application of CGT codes is inefficient even for type (2) SRLG lists.

Although application of codes with special property (like combinatorial group testing codes in [35]) for type (2) SRLG lists can be reasonable, it is rather inefficient when type (3) SRLGs are under consideration. For example, if $\mathcal{F}$ contains 100 single-link SRLGs, 20 dual-link SRLGs 6 triple-link SRLGs and 1 quadruple-link SRLG, than we have to use a CGT code $d = 4$ with 89 length to localize all failure in $\mathcal{F}$. However, the information theoretic lower bound for localizing 127 failures requires codes with length of 7 bits. Thus, there is large space to improve the code design when type (3) SRLGs with extremely heterogeneous SRLGs in terms of the number of links contained are considered in the problem, addressed in the dissertation.

M-trail Formation (R2)

In the design principle followed in the algorithm design of the dissertation, after the code assignment phase in (R1) the second step is the formation of a set of m-trails that can satisfy the SRLG code uniqueness requirement. In the following paragraph we will demonstrate the idea of m-trail formation in the scenario with only single-link SRLGs.

To form the m-trail t_j in $\underline{\underline{\mathbf{A}}}$, the links with $a_{e_i,j} = 1$ in the column $\underline{a_j}$ of $\underline{\underline{\mathbf{A}}}$ should be interconnected while disjoint from any link with $a_{e_i,j} = 0$. Obviously, this may result in multiple disjoint subgraphs which cannot form a simple or non-simple path as desired. Thus, it has been proved as a viable approach to perform code swapping in order to form each m-trail one after the other. A simple rule for code swapping was successfully implemented in [85], where two link codes can be swapped for an m-trail at a bit position only if the swapping will not affect the formation of the other m-trails at the other bit position. In specific, the codes of two links in $\underline{\underline{\mathbf{A}}}$ can be swapped to form a specific m-trail t_j *locally* on the bit position $\underline{a_j}$ without affecting the formation of the other m-trails on $\underline{a_i}$ $i \neq j$). In other words, code

$\underline{a}_e^T$ can possibly be swapped with code $\underline{a}_f^T$ in the m-trail formation at the j^{th} bit position only if a_e and a_f has exactly the same code except at position j. A detailed description of the method in [85] is presented in Section 4.1.2.

With multi-link SRLGs, the m-trail formation process via code swapping will take much longer convergence time due to the dependency among the rows in $\underline{\underline{\mathbf{ACT}}}$, which in turn requires a set of completely different rules according to the specific SRLG scenario under consideration.

4.1.2 State-of-the-art

In addition to (R1) and (R2), there could be some other constraints due to specific user design premises, such as the length limitation due to the deployment of optical generators/retransmitters, the locations of monitoring nodes [5, 96], and use of working lightpaths (i.e., live connections) for failure state correlation [5, 77]. Without loss of generality, the dissertation focuses on (R1) and (R2), which are the fundamental for a (b)m-trail UFL solution. Nonetheless, we claim that our approaches in the dissertation can easily tackle any additional requirement imposed upon the proposed (b)m-trail allocation problem.

An integer linear program can be developed that satisfies both (R1) and (R2) in a single step [95, 99]. In particular, [95] is the first study that suggested to using freely routed open-loop un-directed supervisory lightpaths (in the form of m-trails) for single-link SRLG failure localization. Both studies formulated the S-LP allocation problem into ILPs, which is unfortunately subject to intolerably long computational time even in very small topologies. Thus people have turned to the design of heuristics in solving the problem. The previously reported solutions can be divided into two categories according to their design principles. The first one manipulates an accumulation mechanism such that (R2) is ensured at the beginning, while the goal of the heuristics is to satisfy (R1) [4, 5, 35]. In the second design category, (R1) is intrinsically ensured at the beginning while leaving (R2) as a goal [85].

In [35], with the help of CGT code construction the authors conducted an in-depth theoretical bound analysis on the minimum number of permissible probes required for localizing a failed SRLG with up to d arbitrary links, in which each link is assigned with multiple codes in a graph with at least $d + 1$ disjoint spanning trees. Therefore, the construction in [35] can only be applied to very densely meshed topologies. For example, the network has to be as densely meshed as $(2d + 2)$-connected in order to accommodate $d + 1$ disjoint spanning trees, which results in $(d + 1) \cdot J$ bits assigned to each link for achieving UFL of SRLGs with up to d links, where J is the CGT code length. Obviously, such a method by assigning each link $d+1$ CGT codes can well fit into theoretical analysis, but it can hardly be applied in most practical scenarios.

The studies in [4, 5] set their goal in minimizing the number of monitoring locations (MLs). For example, to localize failure of SRLG with up to 2 links (i.e., $d = 2$), all the 3- and 4-connected subgraphs should be identified, and almost each subgraph needs an ML at an arbitrarily chosen node in the subgraph. With each ML determined, graph transformation is performed such that the MLs are merged into a supernode (denoted as m), and cycles are cumulatively added into the transformed graph one by one

via Suurballe's [80] algorithm. To distinguish two SRLGs w_1 and w_2, a cycle must be disjoint from w_2 while passing m and l, where l is a link randomly selected from $w_1 \setminus w_2$. In the worst case this leads to $O(|SRLG|^2)$ of cycles to distinguish all the SRLGs, where $|SRLG|$ is the number of SRLGs considered in the network. Thus, the worst time complexity is $O(|SRLG|^2 \cdot |V|^2)$, where $|V|$ is the number of nodes in network, and the term $O(|V|^2)$ corresponds to the complexity of Suurballe's algorithm. The computation complexity becomes $O(|E|^{2d} \cdot |V|^2)$ if every multiple failure with up to d links should be localized, where $|E|$ is the number of links.

The approach taken in [85] is the first study following the second design principle, where the code uniqueness of each link (as defined in (R1)) is first guaranteed, while an algorithm was given for the formation of each monitoring structure in the context of m-trail. A superb performance was witnessed in [85] by employing random code assignment (RCA) and random code swapping (RCS) for localizing any link failure. In specific, the RCA algorithm forms the jth m-trail by randomly swapping a link code with its *bitwise code pair* at the jth position. For example, the codes "11010110" and "11000110" form a bitwise code pair at the 4-th position. Note that such the RCS algorithm in [85] can only work when single-link failures are considered and simply fails in presence of the code dependency among overlapped SRLGs, which is the most critical task to be addressed in the dissertation.

A meta-heuristic approach, called MeMoTA, were introduced in [33] to improve the performance of the RCA+RCS method in [85]. The MeMoTA approach can improve the performance of the RCA+RCS, while the computational time required by the ILPs presented for the problem is significantly reduced. However, it requires longer computation (in the order of minutes) than RCA+RCS (in the order of milliseconds).

Detailed Description of Random Code Swapping (RCS) [85]

As the RCA+RCS [85] method proposed for UFL of type (1) SRLGs plays an important role in the algorithm design of the dissertation, in this section we give a detailed description of the approach.

Fig. 4.3 shows a flowchart of the RCS algorithm. As this method was developed for single link failure localization, **A** and **ACT** are the same (called ACT). At the beginning, an initial ACT is formed by randomly assigning each link with a unique alarm code as shown in Step (1). In Step (2), the cost of the ACT is evaluated according to a given cost function. Next, a greedy cycle formed by Steps (2), (3), (4), (5), and (6) is initiated, where RCS is performed in Step (3) and (5). In every cycle, a new ACT (denoted as ACT_{new}) is generated and the corresponding cost C_{ACT_new} is evaluated in Step (2). If the cost of ACT_{new} (denoted as C_{ACT_new}) is smaller than (or equal to) that of the cost of previous ACT (denoted C_{ACT}) as in Step (2), the algorithm starts the next greedy cycle by replacing the old ACT with the new one (i.e., $ACT \leftarrow ACT_{new}$) and performing RCS as denoted as $ACT_{new} \leftarrow \Psi_{RCS}(ACT_{new})$ in Step (3). In case the new ACT has a cost larger than that of the old one, the newly derived ACT is simply disregarded, and the next greedy cycle will perform RCS based on the old ACT again. Such a greedy cycle is iteratively performed until a given number (100 in the simulation) of iterations of RCS

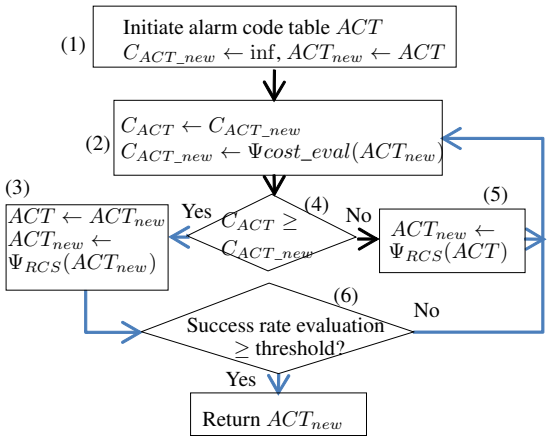
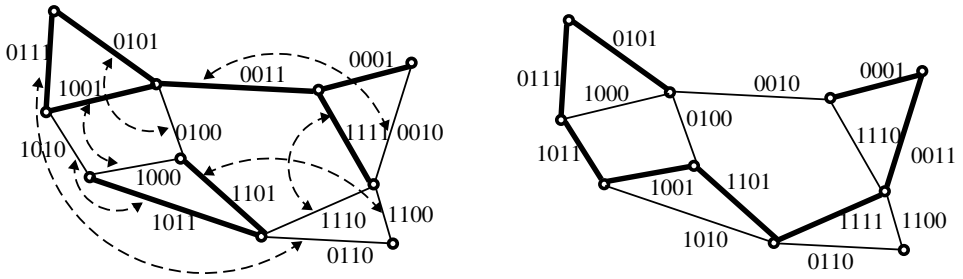


Figure 4.3: The flowchart of the RCS algorithm. [85]



(a) Link pairs on the 4^{th} bit position after code assignment

(b) 4^{th} bit position after random code swapping

Figure 4.4: The random code assignment and m-trail formation

has been done without getting a smaller cost at Step (4).

The swapping of two link codes in RCS is subject to the *strong locality constraint* (SLC), which stipulates that two links can swap their codes in formation of the j^{th} m-trail if the two codes are different only in the j^{th} bit position. By following SLC, the swapping of two link codes will not affect the m-trail formation at the other bit positions. An example is shown in Figure 4.4 on m-trail formation with random code swapping on the 4^{th} bit.

4.1.3 Problems Targeted in the Dissertation

As we have seen the simple single link failure scenario (type (1) SRLG list $\mathcal{F}$) is well investigated in the literature and various efficient solutions exist for the problem approaching the theoretical lower bound. Further, there are some work dealing with multiple failures up to d links (type (2) SRLG list $\mathcal{F}$) using codes with special property [35]. However, applying such codes with special structure when the SRLG list under consideration is heterogeneous in the number of links contained (type (3) SRLG list $\mathcal{F}$) is rather inefficient. There exist works for this scenario following the first design principle ((R2) and (R1)) [5].

However, the first design category cannot explore the design space provided by the extremely flexible structure of (b)m-trails, thus, leave a large space to improve.

The design of efficient algorithms, where the code assignment (R1) is solved first, while the monitoring structure formation (R2) is conducted in a second step for the unambiguous localization of type (3) SRLG lists $\mathcal{F}$ is investigated with the most flexible S-LPs, the (b)m-trails is highly desired. Thus, in Section 4.2 *I introduce the theoretical principles as well as practical algorithms for unambiguous sparse-SRLG failure localization with monitoring trails. I prove that the complexity of unambiguous SRLG failure localization is NP-complete, thus, Integer Linear Programs for code assignment and monitoring structure formation are introduced. Necessary and sufficient conditions are formulated as the basis of unambiguous code assignment. Based on the sufficient conditions, fast, yet efficient heuristic approaches are introduced for sparse-SRLG failure localization, including node failures.*

4.2 Principles and Algorithms for the M-trail Allocation Problem (MAP)

In the dissertation, the mathematical problem for unambiguous SRLG failure localization with the application of (b)m-trails (***M-trail Allocation Problem (MAP)***) is investigated. In specific, the theoretical principles are introduced and practical methods are presented for UFL of type (3) SRLGs with (b)m-trails following the second design principle, which have not been addressed in the literature previously. However, some of the results are also applicable for type (2) SRLGs, which can be treated as a borderline case of type (3) SRLGs.

4.2.1 Computational Complexity of the (bidirectional) M-trail Allocation Problem (MAP) [C8]

The study [77] investigated the redundant alarm reduction problem based on a set of existing working lightpaths (*in-band monitoring*), which aimed to minimize the number of active alarms (or the number of monitors, which is the number of supervisory lightpaths in *out-of-band monitoring*) in the network while maintaining unambiguous single link failure localization. It has been shown that the Redundant Monitor Deactivation Problem (RMDP) is NP-complete. Note that [77] relies on the existing lightpaths for unambiguous failure localization, without allocating any additional S-LPs for the purpose of monitoring.

Similarly to [77], the study in [61] studied the problem of localizing failures in transparent optical networks using active lightpaths for monitoring purposes. The problem of optimal allocation of monitoring devices was investigated, which was solved by using a transparent failure location algorithm (TFLA) for detecting both hard (e.g. cable cut) and soft (e.g. high Bit Error Rate (BER)) failures. Although no additional resources are needed for the monitoring purpose, false positive or false negative could happen since some alarms correspond to multiple SRLGs, which results in ambiguity. The problem relies on the failure model in [71] in which the problem of uniquely diagnose single faults and multiple faults is investigated. It has been shown in [71] that the localization of multiple link faults is an NP-complete

problem if the faults propagate in the network. However, the complexity of optimal m-trail allocation problem for SRLG failure localization has not been addressed in the literature yet. The m-trail allocation problem can be formulated as follows:

Definition 4.2.1 *M-trail allocation problem (MAP) instance: a network $G(V, E)$, a $\mathcal{F}$ set of SRLGs, a positive integer $b < |E|$ representing a limit on the length of the link code.*

The MAP problem is to find a set of $\leq b$ m-trails so as to unambiguously localize all SRLGs in $\mathcal{F}$; i.e., for any pair of SRLGs $e, f \in \mathcal{F}$, (1) there is a m-trail which passes through e but not f or vice versa, and (2) all SRLGs are passed by at least one m-trail.

The complexity proof of the MAP problem takes advantage of a Karp-reduction, which is shown in a form of the Hitting Set problem when the graph is not connected and not all single-link SRLGs are localized. The definition of the hitting set problem can be briefed as follows [30]:

Definition 4.2.2 *Hitting set problem instance: Collection C of subsets of a finite set S, positive integer $K < |S|$.*

The hitting set problem is to find a subset $S' \subseteq S$ with $|S'| \leq K$ such that S' contains at least one element from each subset in C.

Theorem 4.2.3 *The MAP problem is NP-complete.*

Proof: The MAP problem is in NP, as verify whether a candidate solution unambiguously localize all SRLGs with $\leq b$ m-trails or not can be done in polynomial time.

Assume we are given an instance of the hitting set problem, that is, a finite set S with n elements $s_1, s_2, \ldots, s_n$, and a collection of subsets $C_1, C_2, \ldots, C_r$.

The polynomial time transformation is given as follows. We construct a graph with $n + r$ isolated edges, where isolated e_i is assigned for each $s_i, i = 1, \ldots, n$ in the set, and isolated edge f_j is assigned for each subset $C_j, j = 1, \ldots, r$. Note that the graph consists of isolated edges, thus, all m-trails consists of a single edge. An SRLG is defined for all $C_j = \{s_{j_1}, s_{j_2}, \ldots, s_{j_k}\}$ as $\text{SRLG}_j = \{e_{j_1}, e_{j_2}, \ldots, e_{j_k}, f_j\}$, and $\text{SRLG}_{r+j} = \{f_j\}$. In the rest of the proof we show that $2r$ SRLGs can be unambiguously localized with $\leq b = K + r$ m-trails, if and only if ($\Leftrightarrow$) the hitting set problem is solvable with $\leq K$ elements.

($\Leftarrow$) Suppose there exists a solution for the hitting set problem with K elements. In this case, MAP problem has a solution with $b = K + r$ m-trails. In the m-trail solution, all edges f_j for $j = 1, \ldots, r$ are covered by a single m-trail. Besides, the e_{j_k} edges corresponding to element s_{j_k} in the hitting set are also covered by a single m-trail. In this case, the SRLGs are unambiguously localized, because (2) all SRLGs are covered with at least one m-trail. And (1) the m-trail on f_i passes through SRLG_i and SRLG_{i+r}, but none of the other SRLGs. The two SRLGs can be distinguished if there exists an m-trail, which passes through SRLG_{i+r}, but not SRLG_i. Such m-trail exists, as SRLG_{i+r} is passed by another

m-trail on single edge e_{j_k}, which corresponds to the s_{j_k} element in the hitting set (and at least one such element exists).

($\Rightarrow$) Finally, we show how to convert an m-trail solution with $b = K + r$ m-trails into a K element solution of the corresponding hitting set problem. The first observation is that the MAP solution has r m-trails with single links of f_j for $j = 1, \ldots, r$ for cover (2) and unambiguously localize SRLG$_i, i = r + 1, \ldots, 2r$. The second observation is that at least one edge e_i from each SRLG$_j, j = 1, \ldots, r$ are covered by an m-trail in order to distinguish the failure of SRLG$_i$ and SRLG$_{i+r}$. The edges e_i for $i = 1, \ldots, n$ are passed by $K = b - r$ single hop m-trails and the K elements in S corresponding to these m-trails forms a hitting set on the subsets $C_j, j = 1, \ldots, r$. Thus, the MAP is **NP**-complete. ■

As a consequence of this result, finding a fast algorithm that solves an arbitrary input of the MAP problem is unlikely (unless $P = NP$).

4.2.2 Find Optimal Solutions for the M-trail Allocation Problem [J2, C4, C5]

In this section the ILP constraints for the (R1) code assignment phase and the (b)m-trail allocation problem (R2) is formulated for unambiguously localizing a type (3) sparse-SRLG list $\mathcal{F}$ are presented. Furthermore, the code assignment phase using CGT codes with special property is shown for the type (2) dense-SRLG case. Note that the number of (b)m-trails is determined by the length of the CGT codes, which is not necessarily optimal. Thus, the formulations using CGT codes are not generally optimal; it only minimizes the bandwidth cost of the applied (b)m-trail solution. However, the application of CGT codes is the best known solution to handle the highly dependent alarm codes of a given dense-SRLG list $\mathcal{F}$ [J2]. An example is given in [J2], where a suite of rules is provided in order to guide the convergence of a heuristic approach (called Greedy Code Swapping, GCS^d) of the problem solution for better quality when all the SRLGs with up to d links are considered. However, in the event that the SRLGs with significant diversity in terms of the number of links are monitored (i.e. type (3)), it is far from efficient to come up with a specific rule for each type of SRLG scenario; while a more general approach is highly desired.

The formulations shown in this section are the extended version of the ILP formulated for type (1) single link SRLG list presented in [98]. Thus, the main contribution of the following formulations is the ability to handle the dependency between the link codes and SRLG alarm codes, which is not present in the original formulations. The notations used in the ILP are summarized in Table 4.3.

The undirected network is represented by directed edges in the model, but the obtained (b)m-trails are the optimal solution for the undirected network. Our goal is to minimize the cost function presented in Eq. (4.1), namely the number of (b)m-trails and the used wavelength channels for monitoring purposes, formulated in the following objective function:

$$\min \left\{ \gamma \cdot \sum_{\forall j \in J} m_j + \sum_{j \in J} \sum_{\forall (u,v) \in E} c_{u,v} \cdot e^j_{u,v} \right\}. \tag{4.4}$$

Table 4.3: The notations used in the ILP

Notations	Description
$G(V, E)$	The underlying graph with node set V and directed link set E
$\mathcal{F}$	A given set of SRLGs $\mathcal{F} = \{f\}$. We assume that if a directed link (u, v) is in SRLG f, then (v, u) will be in the same SRLG as well.
$\mathcal{F}_s$, $\mathcal{F}_m$	The single link and multiple link SRLGs in the SRLGs list $\mathcal{F}$
J	The maximum number of (b)m-trails allowed in the solution
j	(b)m-trail index, where $j \in \{0, 1, \ldots, J-1\}$
$\mathcal{C}$	Set of Combinatorial Group Testing code in the dense-SRLG code assignment
k_i	A single code in the CGT code list $k_i \in \mathcal{C}$
γ	The cost ratio of a monitor to a supervisory wavelength
$c^j_{u,v}$	The cost of a supervisory wavelength channel on link $u \rightarrow v$, and we assume $c_{uv} = c_{vu}$. $c_{uv} = 1$ if hop count is used as the cost metric. Otherwise, it can be a distance related or TE cost.
δ	A predefined small positive value ($\lvert E\rvert^{-1} \geq \delta > 0$). It is the minimum step of voltage increase along a (b)m-trail
β	A predefined small constant and $2^{-J} \geq \beta > 0$
m^j	Binary variable. It is 1 if t_j is a (b)m-trail, and 0 otherwise
s^j_u	Binary variable. It takes 1 if node u is the source of (b)m-trail t_j, and 0 otherwise
d^j_u	Binary variable. It takes 1 if node u is the sink of (b)m-trail t_j, and 0 otherwise
z^j_u	Binary variable. It takes 1 if node u is traversed by (b)m-trail t_j, and 0 otherwise
$e^j_{u,v}$	Binary variable. It takes 1 if $u \rightarrow v$ is an on-trail vector of (b)m-trail t_j, and 0 otherwise
$q^j_{u,v}$	Fractional variable. It is the voltage of vector $u \rightarrow v$ on (b)m-trail t_j. It takes 0 if $u \rightarrow v$ is not an on-trail vector on t_j.
b^j_g	Binary variable, which takes 1 if any edge in SRLG g is an on-trail vector of (b)m-trail t_j, and 0 otherwise (jth bit of the alarm code of SRLG g)
α_g	General integer variable, which is the decimal alarm code assigned to SRLG $g = \{(u, v), (v, u)\}$.
$f^{i_2}_{i_1}$	Binary variable. For two distinct SRLGs i_1 and i_2, it takes 1 if $a_{i_1} > a_{i_2}$, and 0 if $a_{i_1} < a_{i_2}$
k^j_i	It is a constant with value 1 code k_i has 1 in the jth position, and 0 otherwise
$x^i_{u,v}$	Binary variable. It takes 1 if k_i is assigned to link $u \rightarrow v$, and 0 otherwise

As the number of (b)m-trails corresponds to the hardware cost and signaling complexity in the network, we set $\gamma = 1000$ to emphasize the importance of the monitoring cost. According to the investigated problem, the constraints are presented in the following subsections.

Code assignment (R1) constraints

The sparse- and dense SRLG scenario requires two different approaches, as in the dense-SRLG case assigning CGT codes to the links is a sufficient and efficient method. However, for the sparse-SRLG case CGT codes results insufficiently long codes. Thus, in the sparse-SRLG case we have to care in the ILP about the alarm code uniqueness in the $\underline{\underline{\mathbf{ACT}}}$ between the SRLGs. The ILP formulation in [98] used the sparse code assignment method for single links failure localization to ensure code uniqueness in $\underline{\underline{\mathbf{A}}} = \underline{\underline{\mathbf{ACT}}}$. However, the formulation have to be extended to handle code dependency between link codes in $\underline{\underline{\mathbf{A}}}$ and SRLG alarm codes in $\underline{\underline{\mathbf{ACT}}}$ when type (2) and type (3) SRLGs are considered. The extended constraints in the sparse-SRLG case are:

$$\forall g \in \mathcal{F} : \alpha_g = \sum_{\forall j \in J} 2^j \cdot b_g^j, \tag{4.5}$$

$$\forall g \in \mathcal{F} : \alpha_g \geq 1, \tag{4.6}$$

$$\forall g_1, g_2 \in \mathcal{F}, g_1 \neq g_2 : \beta + \beta \cdot (\alpha_{g_1} - \alpha_{g_2}) \leq f_{g_1}^{g_2}, \tag{4.7}$$

$$\forall g_1, g_2 \in \mathcal{F}, g_1 \neq g_2 : \beta + \beta \cdot (\alpha_{g_2} - \alpha_{g_1}) \leq 1 - f_{g_1}^{g_2}, \tag{4.8}$$

$$\forall j \in J, \forall g \in \mathcal{F}_s : \sum_{\forall (u,v) \in g} e_{u,v}^j = b_g^j, \tag{4.9}$$

$$\forall j \in J, \forall g \in \mathcal{F}_m, \forall f_s \in g : \sum_{\forall (u,v) \in f_s} e_{u,v}^j \leq b_g^j, \tag{4.10}$$

$$\forall j \in J, \forall g \in \mathcal{F}_m : b_g^j \leq \sum_{\forall f_s \in g} b_{f_s}^j. \tag{4.11}$$

Constraint (4.5) assembles the binary alarm code bits and translates them into a decimal alarm code. Constraint (4.6) says that every SRLG must have a positive decimal alarm code, i.e. prevents zero alarm codes. Equations (4.7) and (4.8) ensure distinct decimal alarm codes for any pair of SRLGs, which is equivalent to ensuring unambiguous SRLG failure localization. Finally, Equations (4.9), (4.10) and (4.11) formulate the bitwise `OR` requirement, where a multiple-link SRLG can have 1 in the jth position of its alarm code only if any of the single-link SRLGs with a common edge with it has 1 in the jth position. Note, that the right hand side of the Equations (4.9) and (4.10) in the case of bm-trail formation should be changed to $b_g^j \cdot 2$, as a bm-trail traverses the link in both directions.

When dense-SRLGs (type (2)) are considered, the constraints presented in Eq. (4.5) - (4.11) have to be replaced with the ones introduced in Eq. (4.12) - (4.16). In this case of type (2) SRLG list $\mathcal{F}$ the $\mathcal{C}$ list of CGT codes is given as the part of the input (e.g. generated with the `bktrk` in [27], see Table 4.2),

and the task is to find a minimal cost assignment of the codes to the links, where on each bit position a single trail is formed:

$$\forall i \in \mathcal{C}, \forall (u,v) \in E : x^i_{u,v} = x^i_{v,u}, \tag{4.12}$$

$$\forall (u,v) \in E : \sum_{\forall i \in \mathcal{C}} x^i_{u,v} = 2, \tag{4.13}$$

$$\forall i \in \mathcal{C} : \sum_{\forall (u,v) \in E} x^i_{u,v} \leq 2, \tag{4.14}$$

$$\forall j \in J, \forall (u,v) \in E : 2 \cdot (e^j_{u,v} + e^j_{v,u}) \geq \sum_{\forall i \in \mathcal{C}} k^j_i \cdot x^i_{u,v}, \tag{4.15}$$

$$\forall j \in J, \forall (u,v) \in E : e^j_{u,v} + e^j_{v,u} \leq \sum_{\forall i \in \mathcal{C}} k^j_i \cdot x^i_{u,v}. \tag{4.16}$$

Constraints in Equations (4.12) - (4.14) maps between the directed links in the ILP model and the undirected links of the topology, and ensures that code assignment is injective on the undirected links. Equations (4.15) and (4.16) say that the monitoring structure formed on the jth position traverses a link if and only if its code has 1 in the jth position.

Note, that this formulation can be used for the code assignment phase of type (1) SRLGs as well. In that case, instead of CGT codes $\mathcal{C}$ have to contains arbitrary different codes.

Finally, in both sparse and dense case Eq. (4.17) says that if $|\mathcal{F}|$ SRLGs need to be differentiated than at least $\lceil \log_2 (|\mathcal{F}| + 1) \rceil$ bits (or (b)m-trails) are required:

$$\sum_{\forall j \in J} m_j \geq \lceil \log_2 (|\mathcal{F}| + 1) \rceil , \tag{4.17}$$

Monitoring-trail and bidirectional monitoring-trail formation (R2) constraints

For monitoring structure formation, the following constraints are required:

$$\forall j \in J : \sum_{u \in V} s^j_u \leq 1, \tag{4.18}$$

$$\forall j \in J : \sum_{u \in V} d^j_u \leq 1, \tag{4.19}$$

$$\forall j \in J, \forall u \in V : \sum_{\forall (u,v) \in E} (e^j_{u,v} - e^j_{v,u}) = s^j_u - d^j_u, \tag{4.20}$$

$$\forall j \in J, \forall (u,v) \in E : q^j_{u,v} \leq e^j_{u,v}, \tag{4.21}$$

$$\forall j \in J, \forall u \in V : d^j_u + \sum_{\forall (u,v) \in E} (q^j_{u,v} - q^j_{v,u}) \geq \delta \cdot z^j_u, \tag{4.22}$$

$$\forall j \in J, \forall (u,v) \in E : m_j \geq e^j_{uv}, \tag{4.23}$$

Constraints Eq. (4.18) and Eq. (4.19) allow at most one source-destination pair for each m-trail. Constraint Eq. (4.20) depicts the flow conservation of t_j. The voltage constraint [98] is employed to

ensure that the solution is a single trail, as the aforementioned constraints could result in more than one trail (i.e., a path and a disjoint cycle). Equation (4.21) and Eq. (4.22) says that only an on-trail vector could have nonzero voltage and ensures that the solution will be a single (b)m-trail (or a cycle). The voltages should increase along an m-trail, and with this constraint it is ensured that a single trail is formed. With these constraints, the obtained link set can be covered by a single supervisory lightpath. Constraint Eq. (4.23) identifies the (b)m-trails on each bit position, as only those bit positions form a (b)m-trail, which have at least 1 link covered.

$$\forall j \in J, \forall (u,v) \in E : e^j_{u,v} + e^j_{v,u} \leq 1, \tag{4.24}$$

$$\forall j \in J, \forall u \in V, \forall (u,v) \in E : e^j_{u,v} + e^j_{v,u} \leq z^j_u, \tag{4.25}$$

Constraint (4.24) ensures that an m-trail should traverse an edge only once. Equation Eq. (4.25) stipulates that a node has an inbound or outbound vector, and the vector should be traversed by an m-trail.

In the case of bidirectional m-trail design, Eq. (4.19) have to be removed from the ILP and Eq. (4.24) and Eq. (4.25) should be replaced with the following equations:

$$\forall j \in J, \forall (u,v) \in E : e^j_{u,v} = e^j_{v,u}, \tag{4.26}$$

$$\forall j \in J, \forall u \in V, \forall (u,v) \in E : e^j_{u,v} \leq z^j_u. \tag{4.27}$$

Constraint (4.26) ensures that a bm-trail traverses a link in both directions, and Eq. (4.27) formulates the same constraints for bm-trails as Eq. (4.25) for m-trails.

In summary, the constraints for type (2) and type (3) SRLG are presented. Furthermore, monitoring structure formation for both m-trails and bm-trails are introduced. As a result, four basically different MAP problems can be solved with the constraint, without the special cases for localizing type (1) SRLGs.

4.2.3 Sufficient and Necessary Conditions for Code Assignment [J1, C8]

In this section first necessary and sufficient conditions on $\underline{\underline{\mathbf{A}}}$ are discussed for code uniqueness (R1) of the MAP problems with type (2) SRLGs. Second, Lemma 4.2.4 is introduced as a necessary condition for the SRLG code uniqueness requirement in the sparse-SRLG model. For unambiguous single-link failure localization it is sufficient and necessary that the $\underline{a}_e^T$ link codes are unique. However, for multiple-link SRLG failure localization, this becomes a non-trivial task due to the requirement that the code of an SRLG should be the same as the bitwise OR of the links contained in the SRLG. To resolve this issue, besides the sufficient condition of Theorem 4.2.5 a necessary and sufficient condition in Theorem 4.2.6 is provided on the link codes $\underline{a}_e^T$ in $\underline{\underline{\mathbf{A}}}$ in order to satisfy the design requirement (R1).

The codes of SRLG$_{\{k_1,\ldots,k_n\}}$ and SRLG$_{\{l_1,\ldots,l_m\}}$ are different, iff there exists a position j in their code that is different, i.e. $a_{\{k_1,\ldots,k_n\},j} = 1$ and $a_{\{l_1,\ldots,l_m\},j} = 0$ or vice versa. As (b)m-trails are routed over the network links while SRLGs are simply logical entities, we will mainly deal with the conditions on link codes that can support (R1).

Conditions of Code Uniqueness for Dense-SRLG Model

In this section, the conditions formulated in the literature for the dense-SRLG model applying CGT codes are shown. With type (2) SRLGs, most of the link sets containing up to arbitrary d links are taken as SRLGs; thus, it is efficient to make $\underline{\underline{\mathbf{A}}}$ *$\bar{d}$-separable* to ensure that OR of up to arbitrary d codes in $\underline{\underline{\mathbf{A}}}$ is unique. This has also been investigated in [J2].

A stronger property than *$\bar{d}$-separable* is *d-disjunct*, in which OR of up to d codes does not *contain*[1] any other link code. Here we say a code C_1 *contains* code C_2 if for every bit position $c_{1,i} \geq c_{2,i}$. Lemma 8.1.2 of [25] implies that for a set of $d+1$ *d-disjunct* codes, any code in the set of codes always has a bit position as 1 while 0 at the same bit position of all the other d codes. The following corollary holds:

Corollary 2 *Any arbitrary d rows of a d-disjunct matrix $\underline{\underline{\mathbf{A}}}$ contains a d-by-d submatrix, where the submatrix is a permuted d-order* identity matrix.

The corollary is applicable to the dense-SRLG model where all link sets with up to arbitrary d links are considered as SRLGs. In the sparse-SRLG case, nonetheless, a weaker necessary condition can be obtained to satisfy the SRLG code uniqueness requirement, which will be described in the following subsection.

Conditions of Code Uniqueness for Sparse-SRLG Model

Let a set of sparse-SRLGs be defined in the network. Similarly to Lemma 8.1.2 of [25] for sparse-SRLG we have the following lemma:

Lemma 4.2.4 *The necessary condition to distinguish a multi-link SRLG, $SRLG_{\{f_1,f_2,\ldots,f_d\}}$, from a single-link failure on $SRLG_{\{f_i\}}$, where $1 \leq i \leq d$, is that there exists a bit position l, where $1 \leq l \leq J$, such that the corresponding bit in the link code $a_{f_i,l} = 0$, and $\exists k \neq i, 1 \leq k \leq d : a_{f_k,l} = 1$.*

Proof: It is clear that $\forall j,\ 1 \leq j \leq J$, if $a_{f_i,j} = 1$ we have both $a_{\{f_1,f_2,\ldots,f_d\},j} = 1$ and $a_{\{f_i\},j} = a_{f_i,j} = 1$ due to the OR operation. Thus, the code for $SRLG_{\{f_1,f_2,\ldots,f_d\}}$ contains the code of $SRLG_{\{f_i\}}$. In order to distinguish $a(f_i)$ and $a(f_1, f_2, \ldots, f_d)$, the following must hold: $\exists l$, $1 \leq l \leq J : a_{\{f_1,f_2,\ldots,f_d\},l} = 1$ while $a_{\{f_i\},l} = 0$. As $a(f_1, f_2, \ldots, f_d) = \underline{a}^T_{f_1}$ OR $\underline{a}^T_{f_2}$ OR $\ldots \underline{a}^T_{f_d}$, the only feasible situation is that $\exists \underline{a}^T_{f_k}, k \neq i, 1 \leq k \leq d : a_{f_k,l} = 1$. ■

With Lemma 4.2.4, we can have the following corollary:

Corollary 3 *If the rows $\underline{a}^T_{f_1}, \underline{a}^T_{f_2}, \ldots, \underline{a}^T_{f_m}$ of matrix $\underline{\underline{\mathbf{A}}}$ contains a permuted m-order identity submatrix, then any single-link $SRLG_{\{f_i\}}$ has a different alarm code from that of $SRLG_{\{f_1,f_2,\ldots,f_m\}}$, where $1 \leq i \leq m$.*

[1] In the original terminology, the codes are characteristic vectors of sets.

Note that the existence of an identity matrix is a sufficient but not necessary condition to the uniqueness of $a(f_1, f_2, \ldots, f_d) = \underline{a}_{f_1}^T \texttt{OR}\ \underline{a}_{f_2}^T \texttt{OR}\ \ldots \underline{a}_{f_d}^T$ from any $a_{f_k}^T \forall 1 \leq k \leq d$. A counter example is showed below. Provided with $SRLG_{\{f_1,f_2,f_3\}}$ and its $\underline{\underline{\mathbf{A}}}$ matrix:

$$\underline{\underline{\mathbf{A}}} = \begin{Bmatrix} \underline{a}_{f_1}^T \\ \underline{a}_{f_2}^T \\ \underline{a}_{f_3}^T \end{Bmatrix} = \begin{Bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{Bmatrix}. \tag{4.28}$$

The matrix meets the necessary condition in Lemma 4.2.4 although it is obviously not an identity one. The code uniqueness can be easily verified; for example, since a link code has 1 on the third bit, i.e., $a_{f_2,3} = 1$, the bitwise OR of a_{f_1}, a_{f_2}, and a_{f_3} must be an alarm code with 1 in its third bit position (i.e., $a_{\{f_1,f_2,f_3\},3} = 1$). Further since $a_{\{f_1\},3} = 0$, thus the alarm code of $SRLG_{\{f_1,f_2,f_3\}}$ is different from the alarm code of $SRLG_{\{f_1\}}$.

Thus we can clearly see that Lemma 4.2.4 is not sufficient for meeting the SRLG code uniqueness requirement because it only ensures the unambiguity between a multi-link and a single-link SRLG contained by the multi-link SRLG, while the unambiguity of two multi-link SRLG codes is not addressed. This will be solved in the next subsection.

Strict Sufficient Condition for (R1) for Single- and Multi-Link SRLGs

In this section Theorem 4.2.5 provides a strict sufficient condition for the SRLG code uniqueness requirement in the sparse-SRLG model, which serves as the foundation of the proposed algorithm presented in Section 4.2.4.

Theorem 4.2.5 *All single-link and considered multi-link SRLGs are uniquely coded if the following three conditions hold for the link codes in* $\underline{\underline{\mathbf{A}}}$*:*

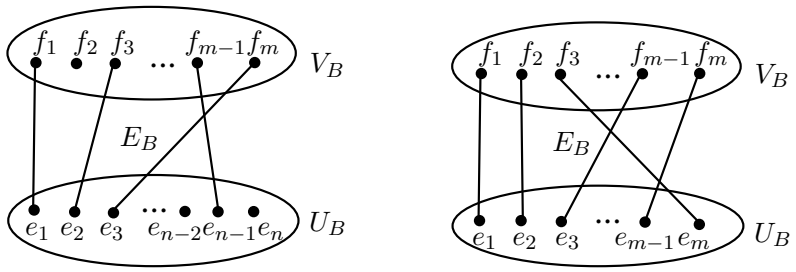
(i) $\forall e \in E : \underline{a}_e^T \neq [0, 0, \ldots, 0]$,

(ii) $\forall e, f \in E : \underline{a}_e^T \neq \underline{a}_f^T$,

(iii) $\forall SRLG_{\{f_1,f_2,\ldots,f_m\}}$ *and* $\forall e \in E$, *if* $\exists l : a_{e,l} = a_{f_i,l} = 1$, *then* $\forall k \neq i : \nexists j$ *with* $a_{e,j} = a_{f_k,j} = 1$, *where* $1 \leq i, k \leq m$ *and* $1 \leq l, j \leq J$.

Proof: In a nutshell, the proof is to demonstrate that every $SRLG_{\{f_1,f_2,\ldots,f_m\}}$ has a unique alarm code provided with the satisfaction of the three conditions on the link codes $\underline{a}_e^T, \forall e \in E$. We will show that two arbitrary and different SRLGs, denoted as $SRLG_{\{f_1,f_2,\ldots,f_m\}}$ and $SRLG_{\{e_1,e_2,\ldots,e_n\}}$, respectively, have different codes.

For easy presentation, the SRLGs are represented as a graph $B = (U_B, V_B, E_B)$ shown in Fig. 4.5, where V_B represents the set of links in $SRLG_{\{f_1,f_2,\ldots,f_m\}}$, U_B represents the set of links in $SRLG_{\{e_1,e_2,\ldots,e_n\}}$, and E_B shows the interconnection among vertices in B. E_B is defined in such a way that any two vertices w_1 and w_2 in B are connected iff $\exists j \in [1, \ldots, J] : a_{w_1,j} = a_{w_2,j} = 1$. Thus, the graph B is a



(a) Some nodes are isolated in B

(b) The edges define a complete match in the graph

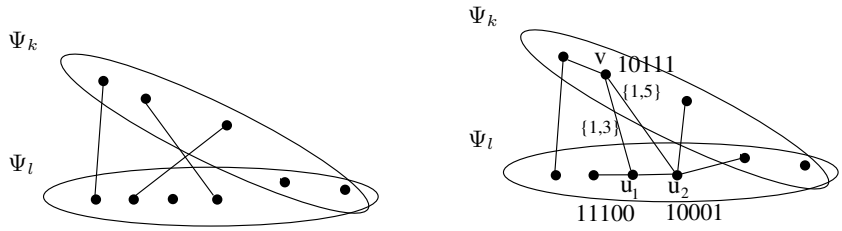
Figure 4.5: Different scenarios on graph B.

bipartite graph according to Condition (iii), which satisfies the necessary condition of Lemma 4.2.4: the codes of two links of a common SRLG cannot be 1 at the same bit position in order to form an m-order identity submatrix. By following Condition (iii), the nodal degree δ of any vertex $u \in U_B$ or $v \in V_B$ is 0 or 1. Thus, the edges in E_B defines *matching* between the two vertex sets, U_B and V_B, of the two SRLGs. Two vertices w_1 and w_2 are *matched* if and only if we can find any bit position j such that $a_{w_1,j} = a_{w_2,j} = 1$.

There are two possible situations according to the matching between the vertices of the two subgraphs in B, which are discussed as follows. The first situation is on partial matching, where at least one vertex in any subgraph has a zero nodal degree. Without loss of generality, we will discuss the case that $\exists e_k \in U_B : \delta(e_k) = 0$. This means a link (i.e., e_k) in U_B is not matched by any vertex in V_B. In this case $\forall j \in [1, \ldots, J]$: if $a_{e_k,j} = 1$, then $a_{e_i,j} = 0, \forall i \neq k, 1 \leq i \leq n$ and $a_{f_l,j} = 0, \forall l, 1 \leq l \leq m$. Thus, it will be 0 at the j^{th} bit position of $SRLG_{\{f_1,f_2,\ldots,f_m\}}$, while 1 for $SRLG_{\{e_1,e_2,\ldots,e_n\}}$.

The second situation is on complete matching, where $m = n$ and each vertex in subgraph U_B finds a matched vertex in the other subgraph V_B. In this case $\forall e_k \in U_B, f_i \in V_B : \delta(e_k) = \delta(f_i) = 1$, and let us choose a matched pair denoted as f_i and e_k. The matching of f_i and e_k means $\exists j \in [1, \ldots, J]$: $a_{f_i,j} = a_{e_k,j} = 1$. From Condition (i) and Condition (ii) we know that each link has a unique non-zero link code, without loss of generality $\exists q \in [1, \ldots, J] : a_{f_i,q} = 1$ and $a_{e_k,q} = 0$. From Condition (iii) $\forall l \neq k, 1 \leq l \leq n : a_{e_l,q} = 0$. Thus, it will be 1 at the q^{th} bit position of $SRLG_{\{f_1,f_2,\ldots,f_m\}}$, while 0 for $SRLG_{\{e_1,e_2,\ldots,e_n\}}$. ■

We give an example to demonstrate the theorem, that it is sufficient but not necessary. The example will show that a link code matrix achieves code uniqueness, but it violates the conditions listed in the theorem. Let us consider all single-link and dual-link SRLGs for the four links with the codes in the



(a) Strict rule on link codes (b) Permissive rule on link codes

Figure 4.6: Different rules on the link codes in the graph-representation of two SRLGs

following link code matrix:

$$\underline{\underline{\mathbf{A}}} = \left\{ \begin{array}{c} \underline{a}_{f_1}^T \\ \underline{a}_{f_2}^T \\ \underline{a}_{f_3}^T \\ \underline{a}_{f_4}^T \end{array} \right\} = \left\{ \begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{array} \right\}. \tag{4.29}$$

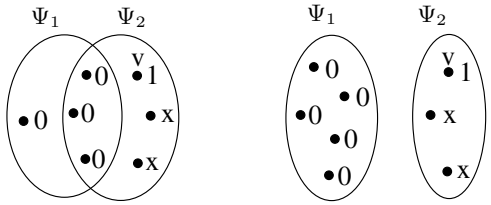
One can easily check that, the code uniqueness holds for all the single-link and dual-link SRLGs. On the other hand, e.g. $SRLG_{\{f_1,f_2\}}$ does not satisfy Condition (iii) because in the link code matrix in Eq. (4.29) exists a link code $\underline{a}_{f_4}^T$, which has two positions, namely $l = 2$ and $j = 3$, where $\underline{a}_{f_1}^T$ and $\underline{a}_{f_2}^T$ from the $SRLG_{\{f_1,f_2\}}$ has bit 1, respectively.

It is possible to find a tighter sufficient condition than that by Theorem 4.2.5; however, the one in Theorem 4.2.5 is simple and can enable a fast and efficient implementation of m-trail code assignment for $\underline{\underline{\mathbf{A}}}$ with alarm code uniqueness of each SRLG. By using the condition in Theorem 4.2.5 for general multi-link SRLGs, we develop an algorithm for m-trail allocation in presence of SRLGs with a single link or with multiple adjacent links.

Permissive Necessary and Sufficient Condition for (R1)

In the following, *an auxiliary graph to represent the SRLGs is introduced*, which is used in the problem formulation. Let Ψ_i be a node set, in which each node represents a link in SRLG$_i$. For example, if SRLG$_k$ consists of two edges $\{e, f\}$, then Ψ_k contains two nodes e and f. In a graph $S = (\Psi_k \cup \Psi_l, E_S)$ two nodes v and u (corresponding to links e and f in the original graph) are connected, iff $\exists j : a_{e,j} = a_{f,j} = 1$. Each link is labeled with the positions $\{j_1, \ldots, j_m\}$, where $\forall j_i : a_{e,j_i} = a_{f,j_i} = 1$. Let $c(v)$ denote the characteristic vector of a link code $\underline{a}_v^T$, which characterizes the bit positions where $\underline{a}_v^T$ is one. For example, if $\underline{a}_v^T = 10111$, then $c(v) = \{1, 3, 4, 5\}$. Further, $c(v, u)$ is defined as $c(v, u) = c(v) \cap c(u)$, and $N(v)$ is defined for node $v \in \Psi_i$ as $N(v) = \cup_{\forall u \in \Psi_j} c(v, u)$. For example, for node v the set is $N(v) = \{1, 3, 5\}$ in Fig. 4.6(b). Obviously, $N(v) \subseteq c(v)$ for all v.

The necessary and sufficient conditions are formulated accordingly as follows:



(a) Intersecting SRLGs (b) Disjoint SRLGs

Figure 4.7: Satisfaction of the strong unambiguity rule at the jth bit position with $a_{\{\Psi_1\},j} = 0$, $a_{\{\Psi_2\},j} = 1$ can be regardless of the assignment of the don't care bits.

Theorem 4.2.6 *The codes of arbitrarily chosen two SRLGs (SRLG$_{\{k_1,\ldots,k_n\}}$ and SRLG$_{\{l_1,\ldots,l_m\}}$) are different, if and only if* $\exists v \in \{(\Psi_k \cup \Psi_l) \setminus (\Psi_k \cap \Psi_l)\} : N(v) \subset c(v)$.

Proof: Without loss of generality, $v \in \Psi_k \setminus (\Psi_k \cap \Psi_l)$. The condition is sufficient, as $a_{\{k_1,\ldots,k_n\},j} = 1$ for all $j \in c(v) \setminus N(v)$, while $a_{\{l_1,\ldots,l_m\},j} = 0$.

The necessity part is shown by indirection. Let assume, that $a(k_1,\ldots,k_n) \neq a(l_1,\ldots,l_m)$, but $\forall v \in \{(\Psi_k \cup \Psi_l) \setminus (\Psi_k \cap \Psi_l)\} : N(v) = c(v)$. As the links in $\Psi_k \cap \Psi_l$ are common, the SRLG$_{\{k_1,\ldots,k_n\}}$ codes could have 1 only on bit positions $c(v)$, $\forall v \in \Psi_k \setminus (\Psi_k \cap \Psi_l)$, where SRLG$_{\{l_1,\ldots,l_m\}}$ has 0, thus their codes are different. Since $\forall v \in \{(\Psi_k \cup \Psi_l) \setminus (\Psi_k \cap \Psi_l)\} : N(v) = c(v)$, SRLG$_{\{k_1,\ldots,k_n\}}$ and SRLG$_{\{l_1,\ldots,l_m\}}$ has 1s (and thus 0s) in the same bit positions. As their codes are identical, it is contradiction with the assumption, which proves the theorem. ■

We give an example to demonstrate the theorem. Let node $v \in \Psi_k$ have a link code $\underline{a}_v^T = 10111$ while $c(v,u_1) = \{1,3\}$ and $c(v,u_2) = \{1,5\}$ in Fig. 4.6(b). In the example, $N(v) = \{1,3,5\} \subset c(v) = \{1,3,4,5\}$. Thus, the two SRLG codes are different on the 4th bit position, i.e. $a_{\{k_1,\ldots,k_n\},4} = 1$ while $a_{\{l_1,\ldots,l_m\},4} = 0$.

Corollary 4 *(Strong unambiguity rule) If there exists a link code $\underline{a}_v^T$ of an arbitrary link v from the symmetric difference ($v \in \{(\Psi_1 \cup \Psi_2) \setminus (\Psi_1 \cap \Psi_2)\}$) of SRLG Ψ_1 and SRLG Ψ_2, without loss of generality for $v \in \Psi_2 : \underline{a}_{v,j}^T = 1$, and $\forall u \in \Psi_1 : \underline{a}_{u,j}^T = 0$, then the alarm codes of the SRLGs are different.*

It is easy to check that $j \in c(v)$ and $j \notin N(v)$, thus, $N(v) \subset c(v)$, that satisfies Theorem 4.2.6. In Fig. 4.7 two examples are shown for the possible situations using the graph representation of the SRLG previously introduced.

Note that Theorem 4.2.6 results in similar structures as provided in Theorem 1 in [4]. However, Theorem 1 in [4] cannot help the design of a corresponding scheme for code construction. Thus, the main difference between the two theorems lies in their perspective toward the targeted scenario: Theorem 1 in [4] formulates a rule on the monitoring structures, while Theorem 4.2.6 provides a rule on the link codes. We will present the proposed Link Code Construction (LCC) heuristic in Section 4.2.5

according to the proposed theorem. In a nutshell, in LCC at the beginning the link code matrix $\underline{\underline{\mathbf{A}}}$ contains codes with arbitrary nonzero Hamming distances without considering any multi-link SRLG, thus, code uniqueness in $\underline{\underline{\mathbf{ACT}}}$ is not necessarily ensured. The proposed LCC iteratively extends $\underline{\underline{\mathbf{A}}}$ with additional columns j in order ot distinguish each SRLG pair in (R1) by using Corollary 4. Each element in the added column j in the link code matrix (i.e., $\underline{\underline{\mathbf{A}}}_{i,j}$) could be either 1, 0, or don't care (x). After the code uniqueness is ensured, bm-trails are formed in each bit position in (R2).

Further note that LCC leaves don't care bits in the jth column after two SRLGs are distinguished using the strong unambiguity rule, only the minimal number of x bits are set. On the other hand, the algorithm following the first design category using Theorem 1 in [4] always sets all bits in the jth bit position, leaving no further space to improve the performance, which is present in LCC.

4.2.4 The Adjacent Link Failure Localization (AFL) Heuristic Approach [J1]

In this section, the MAP problem for UFL of type (3) SRLGs using m-trails is investigated. In the single link failure scenario, an integer linear program (ILP) can be formulated and used to solve small-sized problems with reasonable computation time [95]. In the multiple failure case, the solving of an ILP formulation is intractable even for small inputs of the problem because of the additional OR operations on the link codes (see Section 4.2.2). Thus, we propose an intelligent algorithm, called AFL, to achieve a fast yet efficient m-trail solution for UFL of SRLGs with a single link and multiple adjacent links. Since the failure of all adjacent links implicitly means the node failure, the proposed m-trail solution can be generally claimed for node failure monitoring. Due to the possibly highly heterogeneous SRLGs in term of the number of links, the approaches designed for the dense-SRLG scenario, such as that in [35] and [J2], would be very inefficient.

In this section, the problem of m-trail allocation is formulated into three sub-tasks that will be performed one after the other. The first is to *partition the graph into sub-graphs* in an ***(R0)*** step, where the m-trail formation is performed independently. In the second stage, for each partition *code assignment (R1)* is performed, where each link is first (or tentatively) be assigned with unique alarm code under some given constraints. The third stage is the *m-trail formation (R2)*, which concerns if we can find a set of non-simple paths as m-trails, such that the j^{th} m-trail traverses through all the links with the bit at the j^{th} bit position as 1 while disjoint from any link with the bit at the j^{th} bit position as 0.

The basic idea of the proposed AFL algorithm is to divide the whole m-trail allocation problem into sub-problems, such that each sub-problem can be solved as a single-link UFL case. The division of the problem is by way of a novel topology partitioning method that partitions the whole topology into subgraphs according to the set of SRLGs, such that the dependency of link codes in each partition is removed. Thus, the m-trail formation can be performed via random code swapping (RCS) [85] in each partition. The complete m-trail solution is given by taking all the m-trails in the subgraphs (as the direct sum of $\underline{\underline{\mathbf{A_i}}}$ matrices obtained in the partitions).

Figure 4.8: Adjacent-link Failure Localization (AFL) Algorithm

Input: $G = (V, E), SRLG$
Result: $\underline{\underline{\mathbf{A}}}$

begin
  Initialize $\underline{\underline{\mathbf{A}}}$ empty;
  Form the line graph $L(G) = (V_{L(G)}, E_{L(G)})$;
  for $v = (e, f) \in E_{L(G)}$ **do**
    if $\exists SRLG_i \in SRLG : e, f \in SRLG_i$ **then**
      color $v = (e, f) \in E_{L(G)}$ red;
    end
    else
      color $v = (e, f) \in E_{L(G)}$ blue;
    end
  end
  while $\exists v \in E_{L(G)}$ *with blue color* **do**
    Take a maximum vertex-induced subgraph $L(H)$ with blue links;
    Use RCS to form m-trails in H, results $\underline{\underline{\mathbf{A}}}_H$;
    $\underline{\underline{\mathbf{A}}} := \underline{\underline{\mathbf{A}}} \oplus \underline{\underline{\mathbf{A}}}_H$;
    $\forall v = (e, f) \in E_{L(H)}$: delete v from $E_{L(G)}$;
  end
  while $\exists e \in E : \underline{a}_e^T = [0, 0, \ldots, 0]$ **do**
    $\underline{\underline{\mathbf{A}}}_e = [1]$;
    $\underline{\underline{\mathbf{A}}} := \underline{\underline{\mathbf{A}}} \oplus \underline{\underline{\mathbf{A}}}_e$;
  end
end

Topology Partitioning for SRLG Code Uniqueness

The topology partitioning method, which aims to meet the sufficient condition of SRLG code uniqueness defined in Theorem 4.2.5, is presented in this subsection. Let the network topology be denoted as $G = (V, E)$. The pseudo code of the proposed method is presented at the algorithm in Fig.4.8. In Step (3), the *line graph* of G is formed, denoted as $L(G) = (V_{L(G)}, E_{L(G)})$, which is constructed in such a way that each vertex of $L(G)$ represents an edge of G. Let the set of vertices and edges of $L(G)$ be denoted by $V_{L(G)}$ and $E_{L(G)}$, respectively. Thus, any two vertices in $V_{L(G)}$ are adjacent iff their corresponding links in G are incident to a common node. Fig. 4.9(a) and Fig. 4.9(b) show an example, where the graph G with 4 nodes and 5 links is transferred to a line graph $L(G)$ with 5 vertices and 8 edges.

The topology partitioning process runs on $L(G)$. Each edge in $L(G)$ is marked by red (or blue) if the two end vertices of the edge, which represent two links in G, belong (or do not belong) to a common SRLG (as indicated in Step (6) and Step (9)). An example is shown in Fig. 4.9(a) where all the single-link SRLGs and all adjacent dual-link SRLGs, except $SRLG_{\{(0,1),(1,3)\}}$ and $SRLG_{\{(2,3),(1,3)\}}$, are considered. The line graph $L(G)$ is given in Fig. 4.9(b). The edge $v_1 = ((0,1),(0,3)) \in E_{L(G)}$ is an edge in $L(G)$, and $(0,1)$ and $(0,3)$ belong to $SRLG_{\{(0,1),(0,3)\}}$. Thus, v_1 is colored by red (solid line in the figure) to indicate that $(0,1)$ and $(0,3)$ should not be in the same partition so as to maintain the independence of the codes at each partition. On the other hand, the edge $v_2 = ((0,1),(1,3)) \in E_{L(G)}$ is colored by blue (dashed line in the figure) since the two links in G represented by the vertices $(0,1)$ and $(1,3)$ do not belong to a common SRLG.

In the loop of Steps (13) – (16), the algorithm iteratively identifies each maximal vertex-induced subgraph where all edges colored with blue in $L(G)$, and restores it back to the original graph domain (denoted as H in the algorithm in Fig. 4.8). Then, RCS is performed on H. Detailed descriptions for RCS are given in Section 4.1.2. The m-trail formation on H results in an alarm code matrix $\underline{\underline{\mathbf{A}}}_H$, which will be *direct summed* with the existing alarm code matrix $\underline{\underline{\mathbf{A}}}$. The direct sum $\oplus$ of arbitrary matrices (not necessary quadratic and the same size) $\underline{\underline{\mathbf{A_1}}}, \underline{\underline{\mathbf{A_2}}}, \ldots, \underline{\underline{\mathbf{A_n}}}$ is a block matrix formed as

$$\underline{\underline{\mathbf{A}}} = \bigoplus_{i=1}^{n} \underline{\underline{\mathbf{A_i}}} = \left\langle \underline{\underline{\mathbf{A_1}}}, \underline{\underline{\mathbf{A_2}}}, \ldots, \underline{\underline{\mathbf{A_n}}} \right\rangle = \begin{Bmatrix} \underline{\underline{\mathbf{A_1}}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{\underline{\mathbf{A_2}}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \underline{\underline{\mathbf{A_n}}} \end{Bmatrix},$$

where $\mathbf{0}$ denotes a matrix of proper size whose elements are all 0.

As a result, the final alarm code matrix $\underline{\underline{\mathbf{A}}}$ will contain blocks along its diagonal, by which the requirement of Corollary 3 can always be fulfilled. At the end of the loop, the edges of $L(H)$ are removed from $L(G)$ as shown in Step (16).

The above process iterates on all the blue maximum vertex-induced subgraphs one after the other. Finally, the edges $e \in E$ in G, which were not involved in any H subgraph during the iterations in Steps (13) – (16), will be taken one by one as a partition and assigned by $[1]$ as the 1×1 alarm code matrix

$\underline{\underline{\mathbf{A}}}_e$ in Step (19). Fig. 4.9(c) gives an example on the block diagonal matrix while unique SRLG codes contained in the corresponding $\underline{\underline{\mathbf{ACT}}}$ is shown in Fig. 4.9(d).

The following theorem proves that the proposed partitioning method can remove the code dependency in each partition. This implies that in any subgraph of G corresponding to a partition only single-link SRLGs need to be handled. After coding the partitions independently with methods proposed for single-link SRLG case the final m-trail solution can be obtained by simply including all the m-trails generated in each subgraph, which is equivalent with the direct sum of the partitions link code matrices.

Theorem 4.2.7 *The partitioning and coding method proposed in the algorithm in Fig. 4.8 can sufficiently achieve SRLG code uniqueness.*

Proof: In a nutshell, the proof is to demonstrate that the resulting $\underline{\underline{\mathbf{A}}}$ link code matrix meets all the conditions of Theorem 4.2.5.

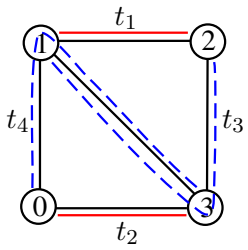
According to the edge coloring in the proposed partitioning process, a red edge will be between two vertices of $L(G)$ which represents arbitrary two links f_i and f_j belonging to $SRLG_{\{f_1,f_2,\ldots,f_m\}}$. Therefore, the two links must be coded in different partitions. Thus, an arbitrary link e is in the same partition with at most one link from an arbitrary $SRLG_{\{f_1,f_2,\ldots,f_m\}}$, without loss of generality f_s (or with itself, if e is part of the SRLG). As the final link code matrix $\underline{\underline{\mathbf{A}}}$ is the direct sum of the partition codes $\underline{\underline{\mathbf{A}}}_k$, e will have 0 in all bit positions, where all other f_i links $i \neq s$ could have 1. Thus, e can have 1 in the same bit position at most f_s (or itself) from the $SRLG_{\{f_1,f_2,\ldots,f_m\}}$, which satisfies Condition (iii) of Theorem 4.2.5.

Note that we can trivially keep $\underline{a}_e^T \neq [0,0,\ldots,0]$ and $\underline{a}_e^T \neq \underline{a}_f^T$ in the partitions with single-link SRLG methods (like RCS), which satisfy Conditions (i) and (ii) in the subgraphs and after direct summing, in $\underline{\underline{\mathbf{A}}}$ too. Therefore, with the proposed method all the three conditions defined in Theorem 4.2.5 can be satisfied, and the code uniqueness of all the SRLGs can be achieved. ■

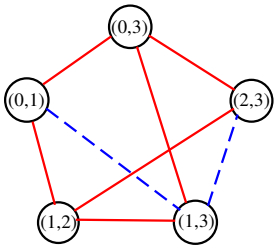
There are two borderline cases of the proposed partitioning method. First, in the event that only single-link SRLGs are considered, the whole line graph is blue and the problem can be solved directly by RCS. On the other hand, when there is no more than one blue edge in any $L(H)$, the resultant alarm code matrix will be an $|E|$-order identity matrix, which means the solution simply leads to link-based monitoring. In this sense, the proposed algorithm can outperform link-based monitoring (i.e., taking less than $|E|$ monitors) if there exist at least one maximal vertex-induced blue subgraph with two or more blue edges, as shown in the example in Figure 4.9.

Complexity Analysis

Creating the line graph in Step (3) takes $O(|E| \cdot \Delta)$ complexity, while coloring the edges in Steps (6) and (9) takes $O(|E|\Delta \cdot |SRLG|\Delta)$, where Δ denotes the maximal nodal degree of graph G, and $|SRLG|$ is at most $O(|E|\Delta + |V|)$ in sparse-SRLG model. The worst complexity of the implemented heuristic for finding the vertex-induced subgraphs in Step (13) is $O(|E|\Delta \cdot |E|)$ since we take $v \in E_{L(G)}$ and



(a) Four-node mesh network G and the m-trail solution



(b) Line graph $L(G)$ of G

$$\underline{\underline{\mathbf{A}}} = \left\{ \begin{array}{c} m-trails \\ \hline\hline \underline{a}^T_{(0,1)} \\ \underline{a}^T_{(1,3)} \\ \underline{a}^T_{(2,3)} \\ \hline \underline{a}^T_{(0,3)} \\ \hline \underline{a}^T_{(1,2)} \end{array} \right\} = \left\{ \begin{array}{cccc} t_4 & t_3 & t_2 & t_1 \\ \hline\hline 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right\}$$

(c) Link code matrix

SRLG	Alarm code matrix
{(0,1)}	1 0 0 0
{(1,3)}	1 1 0 0
{(2,3)}	0 1 0 0
{(0,3)}	0 0 1 0
{(1,2)}	0 0 0 1
{(0,1),(1,2)}	1 0 0 1
{(0,1),(0,3)}	1 0 1 0
{(1,2),(2,3)}	0 1 0 1
{(1,2),(1,3)}	1 1 0 1
{(0,3),(2,3)}	0 1 1 0
{(0,3),(1,3)}	1 1 1 0

(d) Alarm Code Table

Figure 4.9: An example on link code assignment and resulting ACT with the AFL algorithm.

do a greedy check on all the other $e \in V_{L(G)}$ no matter it has a red edge or not. Finally, the worst case complexity of using RCS in Step (14) is $O(|V| \cdot |E|^2 \cdot \Delta^6 \cdot \log^3 |E|)$ as shown in [J2], and it is used at most $|E|$ times. Thus, the code assignment complexity in Steps (13) – (16) is $O(|V| \cdot |E|^3 \cdot \Delta^6 \cdot \log^3 |E|)$, while the code assignment for each individual link in Step (19) at the end is $O(|E|)$. Based on the above, the overall complexity is the complexity of the iteration Step (13) – (16) with $O(|V| \cdot |E|^3 \cdot \Delta^6 \cdot \log^3 |E|)$. This observation is validated by the running time measurement in Section 4.3.2.

4.2.5 The Link Code Construction (LCC) Heuristic Approach [C8]

As the bm-trail allocation for SRLGs has been shown to be NP-complete, in the following a heuristic approach is proposed for UFL of arbitrary SRLG lists $\mathcal{F}$ (including type (3)) following the permissive necessary and sufficient condition in Theorem 4.2.6.

In LCC each SRLG pair is considered sequentially in $\mathcal{F}$, where their codes are derived using the bit-wise OR operation on the codes of the contained links, respectively. Then the algorithm checks whether their codes collide or not. If yes, it is resolved by extending the $\underline{\underline{\mathbf{A}}}$ matrix with a column (e.g., column j) with don't care bits, and apply a minimal cost code assignment following the strong unambiguity rule in Corollary 4. In specific, with any SRLG code collision, the two SRLG codes is made unique in $\underline{\underline{\mathbf{A}}}$ by ensuring $\exists j : a_{\{k_1,k_2,\ldots k_n\},j} = 1, a_{\{l_1,l_2,\ldots l_m\},j} = 0$ or vice versa. Because of the *bitwise OR* relation between the link codes and SRLGs codes, if any don't care bit $\exists k_d : a_{k_d,j} = x$ is set in the future (either 1 or 0) the jth position in the alarm code $a_{\{k_1,k_2,\ldots k_n\},j} = 1$ remains the same. However, if $\exists l_d : a_{l_d,j} = x$, and it is set to 1 in the further iterations, it changes to $a_{\{l_1,l_2,\ldots l_m\},j} = 1$, and possibly makes the codes of the two SRLGs identical. If the strong unambiguity rule formulated in Corollary 4 is true for the two SRLGs, then the alarm codes of the two SRLGs remain different regardless of the assignment of the don't care bits.

The pseudo code of the proposed LCC algorithm is presented in Fig. 4.10. In Step (2) single link UFL is ensured with RCS. In Iteration (5) and Iteration (6), the algorithm compares every pair of SRLG alarm codes, and in case of any code collision, the algorithm makes them distinguished from each other using the strong unambiguity rule shown in Fig. 4.7. Note that each SRLG pair is checked only once, and their codes remain different regardless of the subsequent iterations. To determine which link e, SRLG Ψ_i and bit position pos to perform the strong unambiguity rule, the following `SetCost` function is used:

$$\texttt{SetCost}(e, \Psi_i, pos) = \#\text{S} + \delta \times \#\text{O},$$

where $\#S$ refers to the number of sets of don't care bits to 1 or 0, and $\#O$ refers to the case when we apply the strong unambiguity rule on a newly added don't care column j in $\underline{\underline{\mathbf{A}}}$ which has only don't care bits. The scaling factor δ weights the relative importance of adding a new column in the link code matrix. Since the goal of the method is to minimize the number of bm-trails that is strongly related to the number of colums used for UFL in $\underline{\underline{\mathbf{A}}}$, thus we simply take $\delta = 1000$ in the simulations.

Figure 4.10: Link Code Construction (LCC) Algorithm

Input: $G = (V, E), SRLG$
Result: $\underline{\underline{\mathbf{A}}}$

begin
  Use RCS to form bm-trails in G, results $\underline{\underline{\mathbf{A}}}_{single}$ with J_{single} bit positions;
  $\underline{\underline{\mathbf{A}}} := \underline{\underline{\mathbf{A}}}_{single}$;
  Extend $\underline{\underline{\mathbf{A}}}$ with $J_{max} - J_{single}$ don't care columns;
  for $SRLG_k \in SRLG$ **do**
    for $SRLG_l \in SRLG$ **do**
      if $id(k) < id(l)$ *and* $a(k_1, \ldots, k_n) = a(l_1, \ldots, l_m)$ **then**
        Set x bits in $\underline{\underline{\mathbf{A}}}$ on the bit position pos to 0 $\forall f \in SRLG_i$ and to 1 for $e \in SRLG_j$, $i, j \in \{l, k\}$, where: $\min_{\substack{\forall e \in \Psi_j \setminus \Psi_i, \\ \forall pos \in J}} \texttt{SetCost}(e, \Psi_i, pos)$;
      end
    end
  end
  for $j = J_{single} + 1, \ldots, J_{max}$ **do**
    if $\exists i = 1, \ldots, |E| : \underline{\underline{\mathbf{A}}}_{i,j} \neq x$ **then**
      Set all x on the jth position to 1;
    end
    else
      Remove column j from $\underline{\underline{\mathbf{A}}}$;
    end
  end
  $J_{ca} :=$ number of columns in $\underline{\underline{\mathbf{A}}}$ after code assignment;
  for $j = J_{single} + 1, \ldots, J_{ca}$ **do**
    $J_{pp} := J_{ca}$;
    $CC :=$ The number of connected components in the jth position on the 1 bits;
    if $CC > 1$ **then**
      for $c = 2, \ldots, CC$ **do**
        Add a column $J_{pp} + 1$ to $\underline{\underline{\mathbf{A}}}$, where the elements are set $\forall e_{i_1}, \ldots, e_{i_s} \in c$: $\underline{\underline{\mathbf{A}}}_{i_d, \{J_{pp}+1\}} = 1$, and 0 otherwise;
        Set $\forall e_{i_1}, \ldots, e_{i_s} \in c : \underline{\underline{\mathbf{A}}}_{i_d, j} = 0$;
        $J_{pp} := J_{pp} + 1$;
      end
    end
  end
end

In the initial code assignment phase, bm-trail formation was performed in Step (2) such that each column in the initial $\underline{\underline{\mathbf{A}}}$ matrix corresponds to an eligible bm-trail. However, further efforts on the newly added columns in Step (8) are needed to minimize the number of bm-trails. Thus, Iteration (12) sets the don't care bits to 1 in the newly added columns in Step (8) in order to creating larger connected components on each bit position. Columns containing only don't cares are removed from $\underline{\underline{\mathbf{A}}}$.

Finally, in Iteration (21) in a post-process phase (R2) is conducted, i.e. if in a newly added column j in Step (8) cannot lead to a single trail, new columns $a^{J_{pp}+1}$ are added to $\underline{\underline{\mathbf{A}}}$. Note that if SRLG Ψ_i and SRLG Ψ_j is distinguished on the jth position in Step (8), after (R2) in Iteration (21) the strong unambiguity remains true either on the jth position or on the newly added $a^{J_{pp}+1}$ column. As a result, all SRLGs in the SRLG list $\mathcal{F}$ are unambiguously localized, i.e. each code in $\underline{\underline{\mathbf{ACT}}}$ derived from the link code matrix $\underline{\underline{\mathbf{A}}}$ constructed with the algorithm in Fig. 4.10 is unique.

Receiver and Transmitter Placement

A node failure is equivalent to the failure of all the adjacent links. However, the localization of a node failure concerns more issues than simply localizing a failure event which hits all the adjacent links.

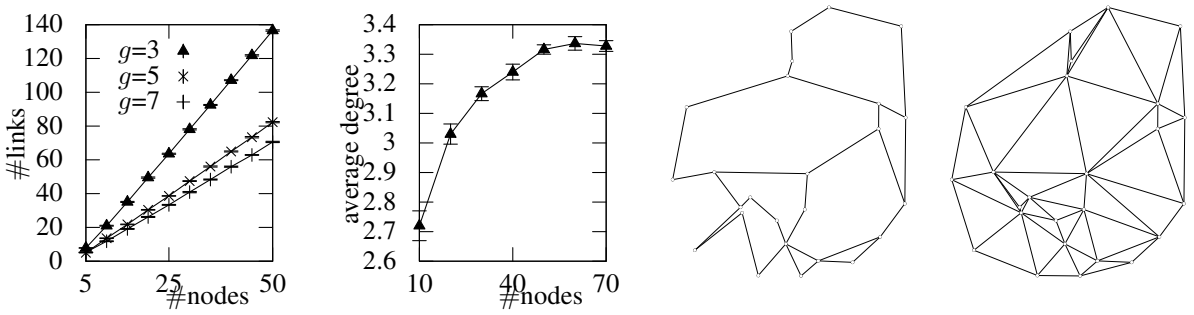
As the LCC design focuses on bm-trails, which is simply a directed Eulerian cycle, and the transmitter and receiver can be possibly placed at the same location (called Monitoring node or *M*-node). If an *M*-node fails, it not only makes all the adjacent links unconnected but also fails to perform expected alarm dissemination. Therefore, a backup M-node (*BM*-node) should be in place along the bm-trail that can identify the status of the *M*-node. As both the *M*-node and *BM*-node can localize exactly the same failures (they are monitoring the status of the same bm-trail), they issue an alarm at the same time. The alarm of the *M*-node suppresses the alarm of the *BM*-node, as in this case the *M*-node is functional. However, if the *BM*-node senses a failure, but has not received the alarm of the *M*-node, then the *BM*-node issues the alarm instead of the *M*-node.

As a bm-trail traverses at least two nodes, the selection of *M* and *BM* is always possible, allowing full coverage of node failure localization.

4.3 Simulation Results

4.3.1 Input Parameters

As the $g(y_{\mathcal{L}})$ solution cost is highly influenced by the network topology [83], a large number of experiments on hundreds of randomly generated planar 2-connected topologies of different connectivity were conducted to verify the proposed algorithms and compare them with previously reported counterparts. The network topologies were generated with `lgf_gen`, a random graph generator of LEMON [1], which randomly generates realistic planar 2-connected networks based on a girth parameter g as described as follows.



(a) Edge number with different girth parameters $g = 3, 5$ and 7

(b) Nodal degree in SRLG analysis

(c) 25-node sparse topology ($g = 7$)

(d) 25-node dense topology ($g = 3$)

Figure 4.11: Statistics of the random topologies generated for the simulation with `lgfgen`

Random Topology Generation with `lgf_gen`

Firstly, nodes are allocated into a unit square with a uniform distribution, and a possible link between every pair of nodes is generated and sorted by their increasing physical link lengths. Each link is added to the graph one by one according to their sorted order. To keep the graph planar links are added only if they do not intersect any other links. Besides, a link is not added if a disjoint path-pair exists between the node pair with no less than g hops. This can be efficiently examined by running Suurballe's algorithm [81] on the node pair. Finally, the algorithm checks each candidate link in an order of descending length, and a link is removed in case the link is part of a disjoint path-pair with more than g hops.

Clearly, a smaller value of g yields a more densely meshed topology, as presented in Fig. 4.11. Topologies generated with $g = 3$ are referred to as *dense topologies*, while the ones generated with $g = 7$ are referred to as *sparse topologies*. Fig. 4.11(a)-(b) show the statistics of the randomly generated topologies used in the simulations. We define the density of an SRLG as shown in Section 3.3.1. Recall, that if only single-link failures are considered ($p = 0\%$), we refer to it as single failure scenario, while in the low, medium and high SRLG scenarios p is chosen to $10\%, 50\%$ and 90%, respectively. With different SRLG densities, the schemes are implemented and compared in terms of the following performance metrics:

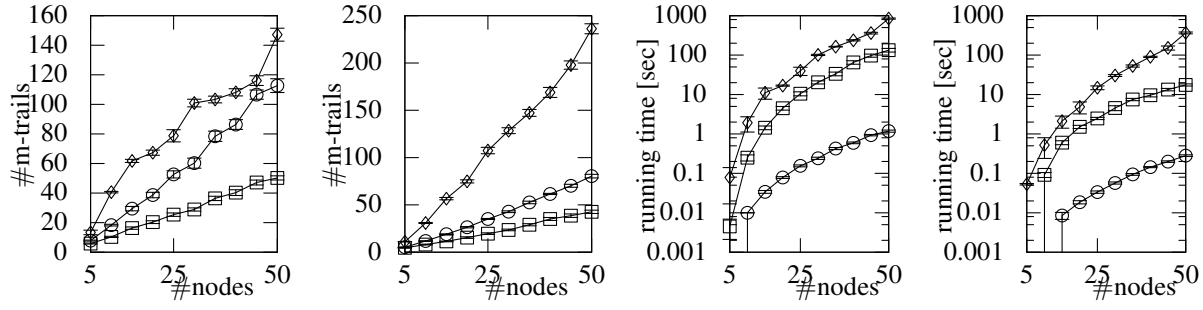
(1) the number of (b)m-trails, (which also stands for the length of the alarm code of each SRLG),,

(2) the average number of (b)m-trails on each link (called normalized cover length, $CL/|E|$),

(3) the total cost $g(y_{\mathcal{L}})$ on full-mesh graphs,

(4) and the running time.

All the performance metrics are examined with respect to different network sizes (i.e., the number of nodes) and topology densities (i.e., g values) which will be presented in the following subsections. The



(a) Dense topologies ($g = 3$) (b) Sparse topologies ($g = 7$) (c) Dense topologies ($g = 3$) (d) Sparse topologies ($g = 7$)

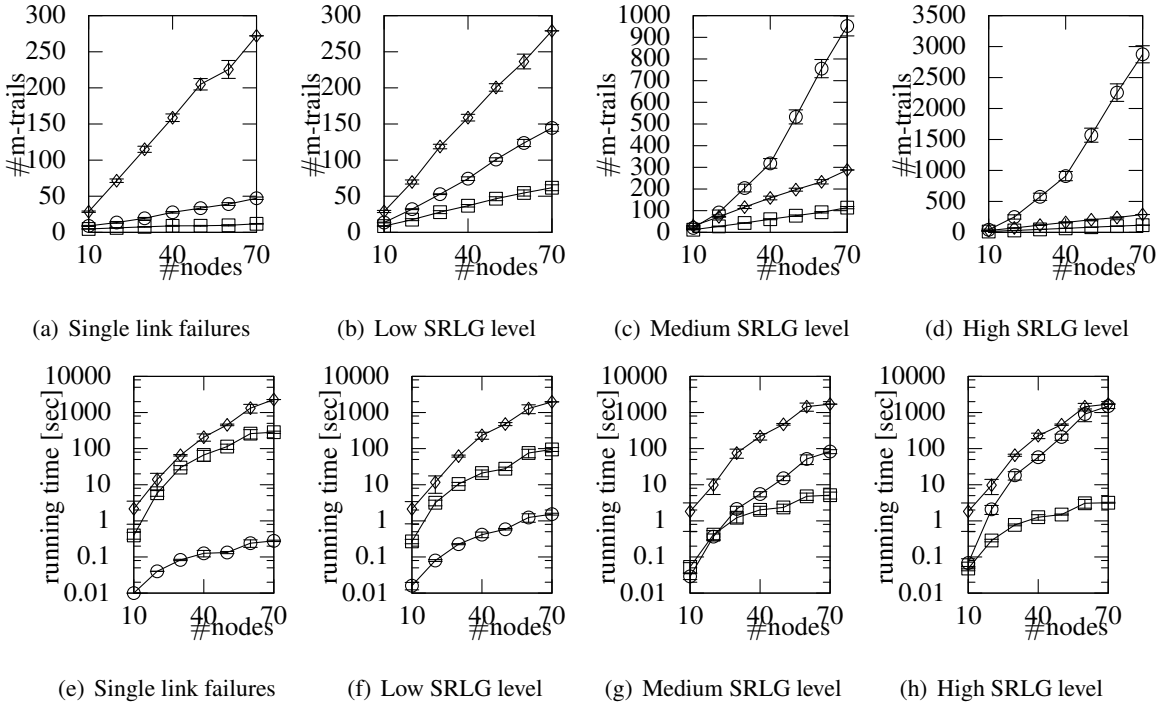
Figure 4.12: The number of m-trails and running times versus the number of nodes with different girth parameters $g = 3$ and 7, with low SRLG level, where AFL, CA and GCS^3 is denoted by □, ○, and ◇, respectively.

simulation has been done on over $1,600$ randomly generated topologies, and each data was obtained by averaging the results from 20 different topologies with a specific g value and a specific number of nodes. 95% confidence interval of each data in the charts is provided.

4.3.2 Number of M-trails versus Network Size

In the simulations corresponding to the number of m-trails in the solution, AFL corresponds to the proposed scheme in the dissertation; CA corresponds to the cycle accumulation method in [4], which allocates monitoring lightpaths in a shape of cycle one by one using Suurballe's algorithm to distinguish each SRLG pair; and GCS^3 is based on $\overline{3}$-separable constructions introduced in [J2], which was originally designed for the dense-SRLG scenario.

The result on the minimum number of m-trails required in the scenario of $p = 10$ is shown in Fig. 4.12. First, we find that the number of m-trails increases when the network size grows, which meets our expectation. In particular, the proposed algorithm can achieve superb scalability, which significantly outperforms the previous work by [4] and [J2] due to the following two aspects. First, compared with CA and GCS^3, AFL based on m-trails can explore the largest problem design space in terms of network topology diversity. This is attested in Fig. 4.12, where CA has achieved far worse performance than AFL when network is dense (i.e., $g = 3$). We have also seen that when network is getting more sparsely connected (i.e., in the case of larger values of g), the performance advantage of the proposed AFL algorithm grows because GCS^3 was not designed for sparse SRLGs. Note that with AFL, the number of m-trails can be well upper bounded by the number of links in the topology. Secondly, the proposed AFL algorithm can take advantage of the precise SRLG information on link code dependency, which is nonetheless absent in the design of CA and GCS^3. Thus, AFL can achieve much better performance especially when the SRLGs are sparse and heterogeneous in terms of the number of links contained in each SRLG.



(a) Single link failures (b) Low SRLG level (c) Medium SRLG level (d) High SRLG level

(e) Single link failures (f) Low SRLG level (g) Medium SRLG level (h) High SRLG level

Figure 4.13: The number of m-trails and running times versus the number of nodes with different SRLG levels, with girth parameter $g = 5$, where AFL, CA and GCS^3 is denoted by □, ○ and ◇, respectively.

The scenarios of different SRLG densities are simulated, and the results are given in Fig. 4.13. Firstly we find that AFL outperforms CA in terms of the number of m-trails in all cases, and the advantage is enlarged when more multi-link SRLGs exist in the network. This is due to a more flexible monitoring structure (which is a free-routed non-simple trail) employed in the proposed AFL. Note that with CA, the routing of the monitoring cycles relies on Suurballe's algorithm and least-cost in nature, which fails in exploration of the logarithmic characteristic between the number of monitoring cycles and the network size. Also from Fig. 4.12 we find that due to the consideration of sparse SRLGs, AFL and CA achieved much better performance than that by GSC^3.

Fig. 4.12(c)-(d) shows the running time for obtaining the data in Fig. 4.12(a)-(b), respectively. It is observed that AFL and CA generally achieves much better computational efficiency for all network topologies. Also, AFL takes longer time than CA in small topologies with low SRLG numbers (Fig. 4.13). Recall that when most SRLGs are single-link (i.e., p is small), a greater blue component is yielded which causes an immediate increase of running time in obtaining an m-trail solution as we have shown in Section 4.2.4. As shown in Fig. 4.13(g) and (h), the running time performance of AFL does not increase much when the network size is growing. In the scenario of medium and high SRLG levels, CA is significantly outperformed by AFL due to more SRLGs with over 3 links.

Table 4.4: Simulation results for the ILPs presented in Section 4.2.2 on three different 8 link networks

scenario	type (2) SRLGs, bm-trails		type (3) SRLGs, m-trails	
method	ILP	GCS^2	ILP	AFL
# (b)m-trails	6	6	3.3	4.6
CL	11	11.6	7.3	6.6
running time (s)	0.4	0	82	0

Optimal Allocation of M-trails

In order to validate the ILP formulations of the MAP problem, two cases were investigated in the simulations to cover all types of constraints. In the first, the optimal dense-SRLG bm-trail solution was tested. In the second scenario, the sparse-SRLG m-trail case with low SRLG level was investigated. The results are presented in Table 4.4. As the ILPs subject to intolerably long computational times, the results are the average results of 3 different randomly generated 8 link networks. The optimal methods are compared with the AFL and GCS^2 heuristics previously proposed for the same (dense/sparse, m-trail/bm-trail) application environments.

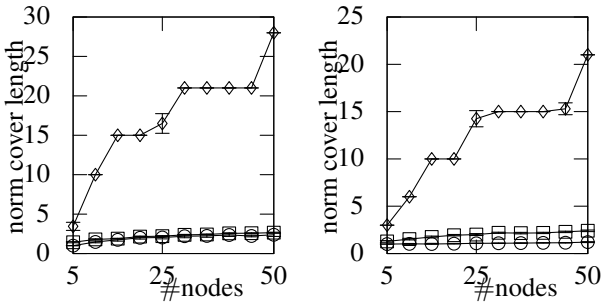
4.3.3 Normalized Cover Length of M-trails

The cover length of an m-trail solution is the total number of used wavelength channels taken by the m-trails. We plotted the normalized cover length of an m-trail solution as the increase of the network size, which is defined as the average number of m-trails passing through a link. Thus, it is trivial to see that the normalized cover length for link-based monitoring is 1 since every link has a single monitoring wavelength channel. As shown in Fig. 4.14, GCS^3 consumed the most monitoring resources among all the three schemes as it has considered much more SRLGs than necessary. In Fig. 4.15, it is observed that CA is only slightly better than the AFL algorithm in the scenario of single-link SRLGs or with a low SRLG level, while being outperformed in other cases.

4.3.4 Total Cost $g(y_{\mathcal{L}})$ on Full-Mesh Graphs

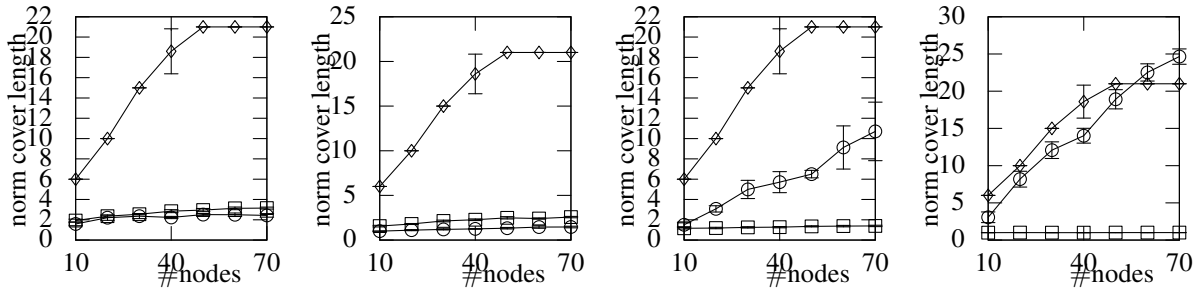
To further look into the performance behavior of the AFL, GCS, CA and link-based monitoring scheme, the section compares the four schemes using complete graphs in terms of the total cost as given in Eq. (1) when $\gamma = 5$ and $\gamma = 1000$. The performance of the link-based monitoring scheme is independent from the applied SRLG level and could be easily calculated from the number of edges: Total cost $= \gamma \cdot |E| + |E|$.

Fig. 4.16 shows the comparison results. As expected, AFL can always yield no larger cost than that by link-based monitoring, due to the fact that link-based monitoring can be treated as a special case of AFL. It is clearly observed that AFL outperforms the other schemes in all the cases except for the



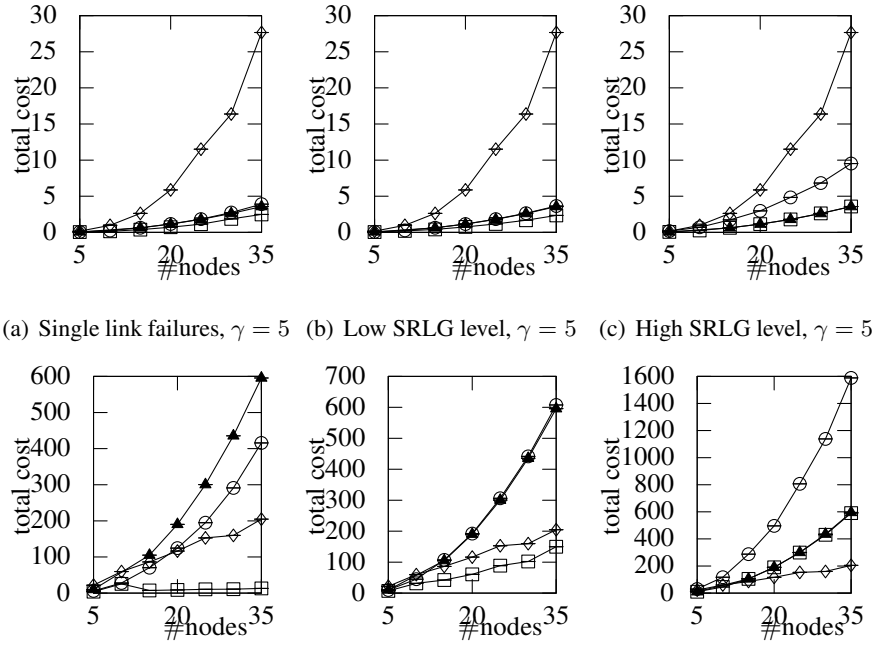
(a) Dense topologies ($g = 3$) (b) Sparse topologies ($g = 7$)

Figure 4.14: The normalized cover length versus the number of nodes with different girth parameters $g = 3$ and 7, and with low SRLG level, where AFL, CA and GCS^3 is denoted by □, ○ and ◇, respectively. The normalized cover length for link-based monitoring is 1 in all figures.



(a) Single link failures (b) Low SRLG level (c) Medium SRLG level (d) High SRLG level

Figure 4.15: The normalized cover length versus the number of nodes with different SRLG levels and with girth parameter $g = 5$, where AFL, CA and GCS^3 is denoted by □, ○ and ◇, respectively. The normalized cover length for link-based monitoring is 1 in all figures.



(a) Single link failures, $\gamma = 5$ (b) Low SRLG level, $\gamma = 5$ (c) High SRLG level, $\gamma = 5$

(d) Single link failures, $\gamma = 1000$ (e) Low SRLG level, $\gamma = 1000$ (f) High SRLG level, $\gamma = 1000$

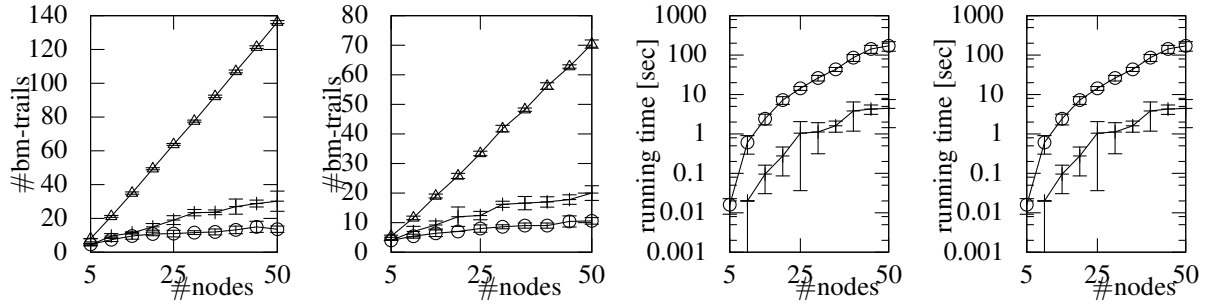
Figure 4.16: The total cost versus the number of nodes with different SRLG levels in full mesh networks, where AFL, CA, GCS and link monitoring is denoted by □, ○, ◇ and △, respectively. The total cost in figures is divided by 1000.

high SRLG scenario with $\gamma = 1000$, where GCS^3 using CGT codes that can deal with all the SRLGs with up to 3 links takes the best advantage in reducing the length of codes (or the number of m-trails). Nonetheless, AFL is not sensitive to the value of γ as that with GCS and can yield superb performance by better exploring the design space in the sparse-SRLG scenario. As shown in Fig. 4.16, GCS^3 is outperformed by all the other schemes under $\gamma = 5$ in which the consumed monitoring resources is more emphasized.

4.3.5 Impact of the Strict and Permissive Condition on the Number of Bm-trails

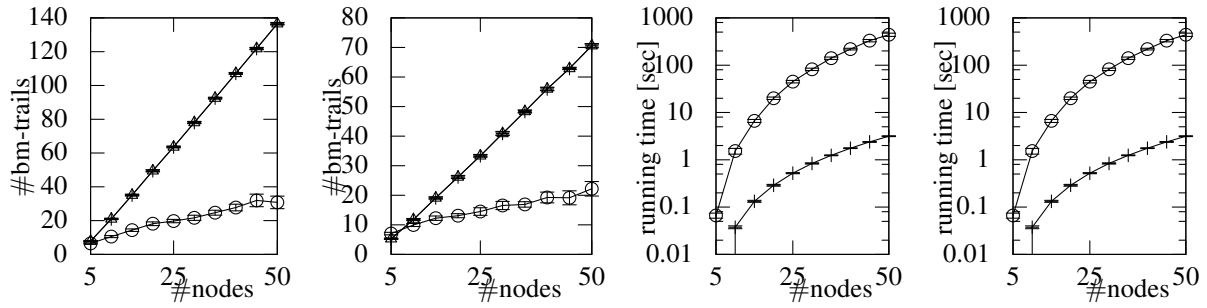
Adjacent dual link failure scenario

The motivation behind the adjacent dual-link failure scenario is that the geographically adjacent links incident to a common node are very likely put into a conduit for some distance and exposed to a common risk of being cut [60] [18]. As expected, in both the proposed LCC algorithm and AFL approach is upper bounded by the link-based monitoring. It is observed in Fig. 4.17 that both AFL and LCC achieve much better performance than link based monitoring, and LCC slightly outperforms AFL as the permissive rule allows more general coding patters as the strict rule (shown in Fig.4.6), based on which AFL was designed.



(a) Dense topologies ($g = 3$) (b) Sparse topologies ($g = 7$) (c) Dense topologies ($g = 3$) (d) Sparse topologies ($g = 7$)

Figure 4.17: The number of bm-trails and running times versus the number of nodes with **10% of adjacent dual SRLGs**, where LCC, AFL, and link-based monitoring is denoted by $\circ$, $+$, and $\triangle$, respectively.



(a) Dense topologies ($g = 3$) (b) Sparse topologies ($g = 7$) (c) Dense topologies ($g = 3$) (d) Sparse topologies ($g = 7$)

Figure 4.18: The number of bm-trails and running times versus the number of nodes with **all single link and node** failure, where LCC, AFL, and link-based monitoring is denoted by $\circ$, $+$, and $\triangle$, respectively.

The running times necessary to obtain the data in Fig. 4.17(a)-(b) measured on a 3GHz Intel Xeon CPU 5160 is presented in Fig. 4.17(c)-(d), respectively. It can be observed that because of the $O(|\mathcal{F}|^2)$ complexity LCC is slower than AFL in all scenarios. However, LCC clearly outperforms AFL in the number of bm-trails corresponding to the hardware cost and failure management complexity.

Single Link and Node Failure

Because of the strict rule on the link codes shown in Fig. 4.6(a), AFL approaches link-based monitoring in this scenario. However, LCC heuristic built on the permissive rule on the link codes shown in Fig. 4.6(b) perfectly fits into this scenario. The results are shown in Fig. 4.18(a)-(b), which clearly shows the superior performance of LCC in terms of the number of required bm-trails for UFL. About the running times in Fig. 4.18(c)-(d) similar observations can be made as in the adjacent-link failure scenario.

Chapter 5

Summary

In this dissertation, two approaches, namely GDP and MAP were investigated for supporting all-optical fault management in phase (i) with a generalized mathematical model and in phase (ii) with failure localization algorithms, respectively.

From a technological point of view restoration strategies may play an important role in future survivable optical backbone design, as soon as fast failure localization is solved in the all-optical environment that can meet the stringent timing requirements (50 ms) of optical layer recovery. At the same time, protection approaches will remain for situations where fast recovery is crucial, while resource efficiency is less important. As the technology evolving and more complex all-optical equipments will be used in practice, (e.g. optical buffering of the signal is solved), near optical packet switching (OPS) and optical burst switching (OBS) optical circuit switching with inverse multiplexing, traffic grooming and network coding could be a competitive technology.

From the view point of survivability, customers are mainly interested in QoS parameters of their connection (e.g. availability) rather than in the applied protection techniques, which is the responsibility of the network operator to choose the proper techniques providing the QoS the customer had paid for. In order to provide well-defined service classes in the SLA for the customer, an availability-aware routing method is crucial. This could be achieved via e.g. an availability-aware implementation of dedicated or shared protection. Availability of the connections can be ensured by building up a proper SRLG list $\mathcal{F}$ in a way that if all SRLGs are protected in the list, the network operator could be sure that the required service availability is provided for the customer without evaluating the availability of the connection. Note that in this case the complex problem of availability evaluation is by-passed, i.e. with the application of this indirection level the problem would be simpler, which could not be generally true. As a result, finding a proper SRLG list $\mathcal{F}$ for a given availability level is still an NP-hard problem. Moreover, the information about link failure dependencies of SRLGs in the same logical hierarchy is inaccurate even at the service provider - who may have a long list of historical failure events. Updating the SRLG database manually could lead to a slow convergence to the current state of the network. Auto-discovery of the SRLG is also an option [73], and could lead to a better network and failure model than manually configured databases.

Table 5.1: Proposed algorithms for the different GDP problems

	IGDP	BGDP	GDP-NC
optimal solution	ILP	-	LP
heuristic solutions	DH, BEA	BFR	-

Even with the application of automatic SRLG discovery method, the accuracy of the SRLG list at the network operator remains the key aspect of the availability-aware methods and pre-planned protection approaches.

5.1 Generalized Dedicated Protection (GDP)

5.1.1 Contribution

We have presented a possible protection method for network coding capable OXCs (generalized dedicated protection) which leads an efficient utilization of resources. In all-optical environment the current architectures proposed for network coding use fiber delay lines to perform coding operations on the signals. The OXCs are equipped with these devices could lead to a finer granularity than current coarse circuit switched WDM networks. The proposed algorithms presented for the different scenarios are shown in Table 5.1. The following properties of the GDP problem were discussed in the dissertation:

Thesis 1.1: (Complexity analysis) I have proved that the non-bifurcated IGDP problem is NP-complete for both in undirected and directed graph. I have proved that an R&R BGDP problem contains IGDP as a subproblem, thus, without network coding the computational complexity of the GDP problem is NP-complete.

Thesis 1.2: (Optimal algorithm) I have given sufficient and necessary condition on the existence of an optimal non-bifurcated GDP solution. I have formulated the non-bifurcated GDP problem as an Integer Linear Program (ILP). I have proposed an iterative algorithm for the BGDP problem in the case when the optimal solution is bifurcated, which uses the ILP formulation proposed for IGDP iteratively.

Thesis 1.3: (Fast heuristic solution) I have proposed a fast, yet efficient heuristic approach to find a feasible IGDP solution based on Dijkstra's shortest path finding algorithm [24]. I have constructed an input type, on which I have proved that the heuristic approach does not approximate the optimal IGDP solution.

Thesis 1.4: (Extension with network coding) I have shown that the polynomial-time solvable linear programming (LP) relaxation of IGDP is resilient but not robust GDP solution without network

coding. Using the network codes proposed in [43] [38] two possible robust codings are shown for the resilient LP solutions, thus, I have shown that an R&R solution for the GDP-NC problem can be found in polynomial time.

5.1.2 Possible Application of the Results

The dedicated protection approach presented in Section 3.2 instead of implementation questions uses a mathematical formulation of the dedicated protection problem. The general formulation enables the network operator to provide the most flexible routing structure for the QoS needs of the customer. Thus, in comparison with the dedicated protection methods in the literature, GDP can significantly reduce the reserved bandwidth while full protection against all failures in $\mathcal{F}$ is maintained. The proposed algorithms can be used in the protection design of any dynamically switched optical network. Based on the equipments available at the network nodes the GDP provides protection solutions for a wide range of networks.

Based on the Technology Readiness Level (TRL) [104] of the available equipments, the proposed GDP approaches can play important role in survivable network design on different time scale. Considering SDH/SONET networks, the required technology for IGDP approaches is present in current networks, thus, it has the highest TRL 9 level. On the other hand, the equipments supporting all-optical network coding (or even all-optical wavelength conversion) are available only in laboratory environment, thus, the optimal GDP-NC has TRL 4 value and can be an alternative in future optical backbone network design. Finally, BGDP assumes the finer granularity of the ngSDH/SONET technology. As most of the equipments are available, BGDP can be classified as TRL 7.

5.1.3 Future Directions

Note that the proposed GDP algorithms are prepared to deal with $\mathcal{F}$ SRLG sets of real networks, which could easily lead to an availability-aware implementation and performance gain of GDP owing to choose the most suitable protection structure to the QoS needs of the customer. However, because of the inaccuracy of SRLG information at this time a proper availability-aware design is hard. A first step was made in [C3] towards an availability-aware GDP implementation, but further elaborations are required on real network data to set a proper input SRLG list $\mathcal{F}$ for each QoS class.

As the technology of IGDP and BGDP are present or can be installed in carrier's network in the near future with a reasonable cost, the technology supporting optical network coding might be expensive. Thus, as a research direction could be to minimize the number of nodes have to be equipped with network-coding capable devices while all merits of GDP-NC is maintained. However, determining a minimal set of the nodes where coding is required is NP-hard, as is its close approximation [46].

Finally, in order to give a thorough mathematical analysis of GDP, besides the proposed optimal and heuristic solutions an approximation algorithm is highly desired. As we have seen, the approximation

Table 5.2: Proposed (b)m-trail solutions for different $\mathcal{F}$ SRLG lists (the ones in parenthesis are not my work)

	(1) single link failure	(2) dense-SRLG	**(3) sparse-SRLG**
optimal solution	(ILP)	ILP	ILP
heuristic solutions	(RCS)	(GCS), LCC	AFL, LCC

ratio of the Steiner Forest problem GDP was reduced to 2, thus, our conjecture is that a 2-approximation algorithm exist for GDP, e.g. via a primal-dual method using cut-based LP formulation of the problem [31]. However, at this time no approximation algorithm exists for GDP.

5.2 M-trail Allocation Problem (MAP)

5.2.1 Contribution

We presented theoretical results and practical approaches for flat, centralized, out-of-band failure localization using the most general structure of supervisory lightpaths, namely m-trails and bm-trails. The proposed algorithms presented for the different scenarios are shown in Table 5.2. The following properties of the MAP problem were discussed in the dissertation:

Thesis 2.1: (Complexity analysis) I have proved that unambiguous SRLG failure localization version of the M-trail Allocation Problem is NP-complete.

Thesis 2.2: (Optimal algorithms) I have introduced the constraints for SRLG code assignment as well as (b)m-trail formation. Based on the scenario under consideration, four different Integer Linear Programs can be constructed.

Thesis 2.3: (Sufficient conditions for code assignment) I have given a strict sufficient and a permissive necessary and sufficient condition for unambiguous code assignment.

Thesis 2.4: (Strict heuristic solution) I have proposed a novel graph partitioning algorithm based on the strict sufficient condition. I have introduced a heuristic (Adjacent-Link Failure Localization) for unambiguously localize SRLGs containing adjacent-link failures using the partitioning approach.

Thesis 2.5: (Permissive heuristic solution) I have proposed a heuristic approach (Link Code Construction) based on the permissive necessary and sufficient condition using bm-trails for unambiguously localizing SRLG failures, including node failures.

5.2.2 Possible Application of the Results

The results in Section 4.2 form the basis of code assignment in failure localization, which can be used as the basis of code construction for other research fields. There is a lack of efficient methods for failure localization in all-optical networks for type (3) SRLG lists. The algorithms in the thesis fills this gap, thus, they can be used in the fault management of optical networks designed for this failure scenario. Using the most general monitoring structures – the bidirectional monitoring trails – the complexity of failure localization can be significantly reduced with the application of the proposed failure localization approaches.

The Technology Readiness Level (TRL) [104] of the m-trail allocation problem is as mature as the optical networks itself. As in current networks all-optical wavelength converters are not present, thus, the all-optical technology as well as the m-trail design has a TRL 6 value. As the optical networks evolve, the TRL value of using m-trails for unambiguous fault localization increase as well. Finally, if bm-trails are used in the network, the technology readiness level is lower, as the loop-back operation has to be solved (TRL 4). However, using bm-trails the failure localization complexity can be reduced.

5.2.3 Future Directions

As a promising way of generalizing the (b)m-trail concept introduced in the dissertation is to add constraints due to specific design premises as the length limitation of (b)m-trails or involve data plane (in-band) information to the (b)m-trail design problem. Involve data plane information to the (b)m-trail design means we are using status information of working lightpaths in the unambiguous localization. However, as we discussed earlier, involve date-plane information would lead to a more frequent reconfiguration of the designed (b)m-trail structure. On the other hand, in those application environments where this is permissible, this approach would lead to a more efficient resource utilization.

Challenging mathematical problems are open in connection with the MAP problem, as the special cases of the (b)m-trail allocation problem, e.g. the (b)m-trail solutions of special graph structures [83, 84]. Another unsolved mathematical problem is the complexity of single-link failure localization for the general case. Our conjecture is that contrary to the general SRLG localization problem it is polynomial-time solvable.

Finally, the (b)m-trail design problem was discussed with a centralized network manager in the dissertation. However, in order to minimize the messages in the control plane distributed localization [96] of failures is desired, i.e. all network nodes are only aware of the status information of the (b)m-trails traversing itself. Such an approach, first, introduces a bunch of implementation and technological questions about the feasibility of the problem.

Bibliography

[1] LEMON: A C++ library for efficient modeling and optimization in networks. [Online] http://lemon.cs.elte.hu Accessed: 2011. 05. 20.

[2] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 134–144. ACM, 1991.

[3] R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[4] S. Ahuja, S. Ramasubramanian, and M. Krunz. SRLG Failure Localization in All-Optical Networks Using Monitoring Cycles and Paths. In *IEEE INFOCOM*, pages 181–185, 2008.

[5] S. Ahuja, S. Ramasubramanian, and M. Krunz. Single-link failure detection in all-optical networks using monitoring cycles and paths. *IEEE/ACM Transactions on Networking*, 17(4):1080–1093, 2009.

[6] O. Al-Kofahi. *Network coding-based survivability techniques for multi-hop wireless networks*. PhD thesis, Iowa State University, 2009.

[7] R. Andersen, F. Chung, A. Sen, and G. Xue. On disjoint path pairs with wavelength continuity constraint in WDM networks. In *IEEE INFOCOM*, volume 1, pages 524–535. Citeseer, 2004.

[8] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

[9] I. Barla, F. Rambach, D. Schupke, and G. Carle. Efficient Protection in Single-Domain Networks using Network Coding. In *IEE GLOBECOM*, pages 1–6. IEEE, 2010.

[10] M. Belzner and H. Haunstein. Network coding in passive optical networks. In *35th European Conference on Optical Communication (ECOC)*, pages 1–2. IEEE, 2009.

[11] M. Bern and P. Plassmann. The Steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 32(4):171–176, 1989.

[12] G. Bernstein, B. Rajagopalan, and D. Saha. *Optical network control: architecture, protocols, and standards*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2003.

[13] M. Bingemann. Optus 3G network goes down. *Australian IT, October*, 24:24545587–15306, 2008.

[14] D. Brungard. Requirements for Generalized Multi-Protocol Label Switching (GMPLS) Routing for the Automatically Switched Optical Network (ASON). [Online] http://tools.ietf.org/html/rfc4258, 2005. Accessed: 2011. 05. 20.

[15] M. Carter et al. Effects of catastrophic events on transportation system management and operations: Baltimore, md-howard street tunnel fire. *US Department of Transportation Report*, 2001.

[16] C. Chekuri, G. Even, A. Gupta, and D. Segev. Set connectivity problems in undirected graphs and the directed Steiner network problem. *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 532–541, 2008.

[17] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WAN's. *IEEE Transactions on Communications*, 40(7):1171–1182, 1992.

[18] H. Choi, S. Subramaniam, and H. Choi. Loopback recovery from neighboring double-link failures in WDM mesh networks. *Information Sciences*, 149(1-3):197–209, 2003.

[19] T. Cinkler. Traffic and λ Grooming. *IEEE Network*, 17(2):16–21, 2003.

[20] T. Cinkler and L. Gyarmati. Mpp: optimal multi-path routing with protection. In *IEEE International Conference on Communications (ICC)*, pages 165–169. IEEE. 2008.

[21] CourierMail. Old lady digging holes looking for scrap metal cut off Georgia-Armenia internet. [Online] http://www.couriermail.com.au/news/technology/scavenging-pensioner-cut-off-g% eorgia-armenia-internet/story-e6frep1o-1226035351932, 2011. Accessed: 2011. 05. 20.

[22] G. Das, D. Papadimitriou, W. Tavernier, D. Colle, T. Dhaene, M. Pickavet, and P. Demeester. Link State Protocol data mining for shared risk link group detection. In *Proc. of 19th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–8. IEEE, 2010.

[23] P. Demeester, M. Gryseels, A. Autenrieth, C. Brianza, L. Castagna, G. Signorelli, R. Clemenfe, M. Ravera, A. Lajszczyk, D. Janukowicz, et al. Resilience in multilayer networks. *IEEE Communications Magazine*, 37(8):70–76, 1999.

[24] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[25] D. Du and F. Hwang. *Combinatorial group testing and its applications*. World Scientific, 2000.

[26] 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*, pages 91–106, January/February 2003.

[27] D. Eppstein, M. Goodrich, and D. Hirschberg. Improved combinatorial group testing for real-world problem sizes. *Algorithms and Data Structures*, pages 86–98, 2005.

[28] M. Feldman, G. Kortsarz, and Z. Nutov. Improved approximating algorithms for directed steiner forest. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 922–931. Society for Industrial and Applied Mathematics, 2009.

[29] A. Fumagalli and L. Valcarenghi. IP restoration vs. WDM protection: is there an optimal choice? *Network, IEEE*, 14(6):34–41, 2000.

[30] M. Garey and D. Johnson. *Computers and intractability. A guide to the theory of NP-completeness. A Series of Books in the Mathematical Sciences*. WH Freeman and Company, San Francisco, Calif, 1979.

[31] M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. In *Proceedings of the third annual ACM-SIAM symposium on Discrete Algorithms*, pages 307–316. Society for Industrial and Applied Mathematics, 1992.

[32] L. Gyarmati, T. Cinkler, and G. Sallai. Srlg-disjoint multi-path protection: When lp meets ilp. In *The 13th International Telecommunications Network Strategy and Planning Symposium, 2008.*, pages 1–15. IEEE, 2008.

[33] A. Haddad, E. Doumith, and M. Gagnaire. A meta-heuristic approach for monitoring trail assignment in WDM optical networks. In *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 601–607. IEEE. 2010

[34] A. Haider and R. Harris. Recovery techniques in next generation networks. *IEEE Communications Surveys & Tutorials*, 9(3):2–17, 2007.

[35] N. Harvey, M. Pătraşcu, Y. Wen, S. Yekhanin, and V. Chan. Non-Adaptive Fault Diagnosis for All-Optical Networks via Combinatorial Group Testing on Graphs. In *IEEE INFOCOM*, 2007.

[36] P.-H. Ho, J. Tapolcai, and T. Cinkler. Segment shared protection in mesh communication networks with bandwidth guaranteed tunnels. *IEEE/ACM Transactions on Networking*, 12(6):1105–1118, December 2004.

[37] T. Ho, M. Médard, and R. Koetter. An information-theoretic view of network management. *IEEE Transactions on Information Theory*, 51(4):1295–1312, 2005.

[38] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52(10):4413–4430, 2006.

[39] J. Q. Hu. Diverse routing in optical mesh networks. *IEEE Transactions on Communications*, 51:489–494, Feb. 2003.

[40] ITU-T G.707 Network node interface for the synchronous digital hierarchy (SDH). ITU-T [Online] http://www.itu.int/rec/T-REC-G.707/en. Accessed: 2011. 05. 20.

[41] ITU-T G.7042 Link capacity adjustment scheme (LCAS) for virtual concatenated signals. ITU-T [Online] http://www.itu.int/rec/T-REC-G/recommendation.asp?lang=en&parent=T-REC-G.7042. Accessed: 2011. 05. 20.

[42] ITU-T G.7041 Generic framing procedure (GFP). ITU-T [Online] http://www.itu.int/rec/T-REC-G/recommendation.asp?lang=en&parent=-T-REC-G.7041. Accessed: 2011. 05. 20.

[43] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, 2005.

[44] A. Kamal. 1+ N network protection for mesh networks: network coding-based protection using p-cycles. *IEEE/ACM Transactions on Networking*, 18(1):67–80, 2010.

[45] V. Kann. On the approximability of np-complete optimization problems. *Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden*, 1992.

[46] M. Kim. *Evolutionary approaches toward practical network coding*. PhD thesis, Massachusetts Institute of Technology, 2008.

[47] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5):782–795, 2003.

[48] K. Kompella and Y. Rekhter. Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS), Internet Draft. [Online] http://tools.ietf.org/html/rfc4202, 2005. Accessed: 2011. 05. 20.

[49] J. Konemann, S. Leonardi, G. Schafer, and S. van Zwam. From primal-dual to cost shares and back: a stronger LP relaxation for the Steiner forest problem. *Automata, languages and programming*, pages 930–942, 2005.

[50] S. LaPerrière. Taiwan earthquake fiber cuts: a service provider view. *NANOG39, Febraury*, 5, 2007.

[51] K. Lee and F. Modiano. Cross-Layer Survivability in WDM-Based Networks. In *IEEE INFOCOM*, pages 1017–1025, 2009.

[52] T. Leighton. Improving performance on the internet. *Communications of the ACM*, 52(2):44–51, 2009.

[53] S. Li, R. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2):371–381, 2003.

[54] D. Lun, M. Médard, R. Koetter, and M. Effros. On coding for reliable communication over packet networks. *Physical Communication*, 1(1):3–20, 2008.

[55] D. Lun, N. Ratnakar, M. Médard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-cost multicast over coded packet networks. *IEEE/ACM Transactions on Networking*, 14(SI):2608–2623, 2006.

[56] M. Maeda. Management and control of transparent optical networks. *IEEE Journal on Selected Areas in Communications*, 16(7):1008–1023, 1998.

[57] S. Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz. Pan-European optical transport networks: an availability-based comparison. *Photonic Network Communications*, 5(3):203–225, 2003.

[58] G. Maier, A. Pattavina, S. De Patre, and M. Martinelli. Optical network survivability: protection techniques in the WDM layer. *Photonic Network Communications*, 4(3):251–269, 2002.

[59] E. Manley, J. Deogun, L. Xu, and D. Alexander. Network Coding for WDM All-Optical Multicast. *CSE Technical reports*, page 10, 2009.

[60] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot. Characterization of failures in an IP backbone. In *Proc. IEEE INFOCOM*, volume 4, pages 2307–2317. Citeseer, 2004.

[61] C. Mas, I. Tomkos, and O. Tonguz. Failure location algorithm for transparent optical networks. *IEEE Journal on Selected Areas in Communications*, 23(8):1508–1519, 2005.

[62] K. Miller, T. Biermann, H. Woesner, and H. Karl. Network Coding in Passive Optical Networks. In *IEEE International Symposium on Network Coding (NetCod), 2010* , pages 1–6. IEEE, 2010.

[63] M. Mohandespour and A. Kamal. 1+N protection in polynomial time: a heuristic approach. In *IEEE GLOBECOM*, pages 1–6. 2010.

[64] H. Mouftah and P. Ho. *Optical networks: architecture and survivability*. Kluwer Academic Publishers, 2003.

[65] S. Orlowski and M. Pioro. On the complexity of column generation in survivable network design with path-based survivability mechanisms. In *International Network Optimization Conference (INOC)*, 2009.

[66] C. Ou, J. Zhang, H. Zang, L. Sahasrabuddhe, and B. Mukherjee. New and improved approaches for shared-path protection in WDM mesh networks. *IEEE/ACM Journal of Lightwave Technology*, 22(5):1223, 2004.

[67] L. Page and J. Perry. Reliability of directed networks using the factoring theorem. *IEEE Transactions on Reliability*, 38(5):556–562, 1989.

[68] D. Papadimitriou. Inference of Shared Risk Link Groups, Internet Draft. [Online] http://tools.ietf.org/html/draft-many-inference-srlg-02, 2002. Accessed: 2011. 05. 20.

[69] D. Papadimitriou, J. Drake, J. Ash, and A. Farrel. L. ong," requirements for generalized mpls (gmpls) signaling usage and extensions for automatically switched optical network (ason). RFC 4139, July, 2005.

[70] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee. Survivable WDM mesh networks. *IEEE/OSA Journal of Lightwave Technology*, 21(4):870–883, 2003.

[71] N. Rao. Computational complexity issues in operative diagnosis of graph-based systems. *IEEE Transactions on Computers*, 42(4):447–457, 1993.

[72] G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122, 2006.

[73] P. Sebos, J. Yates, G. Hjalmtysson, and A. Greenberg. Auto-discovery of shared risk link groups. In *Optical Fiber Communication Conference and Exhibit, 2001.*, volume 3, pages WDD3–1. IEEE, 2001.

[74] X. Shao, Y. Yeo, Y. Bai, J. Chen, L. Zhou, and L. Ngoh. Backup Reprovisioning After Shared Risk Link Group (SRLG) Failures in WDM Mesh Networks. *Journal of Optical Communications and Networking*, 2(8):587–599, 2010.

[75] J. Spragins. Dependent Failures in Data Communication Systems. *IEEE Transactions on Communications*, 25(12):1494–1499, 1977.

[76] S. Stanic, S. Subramaniam, H. Choi, and G. Sahin. On monitoring transparent optical networks. *Proc. International Conference on Parallel Processing Workshops*, pages 217–223, 2002.

[77] S. Stanic, S. Subramaniam, G. Sahin, H. Choi, and H. A. Choi. Active monitoring and alarm management for fault localization in transparent all-optical networks. *IEEE Transactions on Network and Service Management.*, 7(2):118–131, 2010.

[78] J. P. Sterbenz. ResiliNets: Resilient and Survivable Networks. [Online] https://wiki.ittc.ku.edu/resilinets, 2011. Accessed: 2011. 05. 20.

[79] J. Strand, A. Chiu, and R. Tkach. Issues for routing in the optical layer. *IEEE Communications Magazine*, 39(2):81–87, 2001.

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

[81] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.

[82] J. Tapolcai. *Routing algorithms in survivable telecommunication networks*. LAP Lambert Academic Publishing AG & Co KG, 2005.

[83] J. Tapolcai, P.-H. Ho, B. Wu, and L. Rónyai. A novel approach for failure localization in all-optical mesh networks. *IEEE/ACM Transactions on Networking*, 19:275–285, 2011.

[84] J. Tapolcai, L. Rónyai, and P.-H. Ho. Optimal solutions for single fault localization in two dimensional lattice networks. In *Proc. IEEE INFOCOM Mini-Symposium*, 2010.

[85] J. Tapolcai, B. Wu, and P.-H. Ho. On monitoring and failure localization in mesh all-optical networks. In *Proc. IEEE INFOCOM*, pages 1008–1016, Rio de Janero, Brasil, 2009.

[86] TeleGeography. Cable cuts disrupt internet in Middle East and India. [Online] http://www.telegeography.com/cu/article.php?articleid=21528, 2008. Accessed: 2011. 05. 20.

[87] R. Thinniyam, M. Kim, M. Medard, and U. O'Reilly. Network coding in optical networks with O/E/O based wavelength conversion. In *Conference on Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference*, pages 1–3. IEEE, 2010.

[88] I. Tomkos. Dynamically Reconfigurable Transparent Optical Networking Based on Cross-Layer Optimization. *ICTON '07*, 1:327–327, 2007.

[89] M. Tornatore, G. Maier, and A. Pattavina. Availability Design of Optical Transport Networks. *IEEE Journal on Selected Areas in Communications*, 23(8):1520–1532, 2005.

[90] J. Vasseur, M. Pickavet, and P. Demeester. *Network recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann Publishers, 2004.

[91] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger. General availability model for multilayer transport networks. In *Proc. 5th International Workshop on Design of Reliable Communication Networks (DRCN), 2005.*, page 8, 2005.

[92] H. Wang, E. Modiano, and M. Médard. Partial path protection for WDM networks: End-to-end recovery using local failure information. In *Proc. Seventh International Symposium on Computers and Communications (ISCC)*, pages 719–725. IEEE, 2002.

[93] Y. Wen, V. Chan, and L. Zheng. Efficient fault-diagnosis algorithms for all-optical WDM networks with probabilistic link failures. *IEEE/OSA Journal of Lightwave Technology*, 23:3358–3371, 2005.

[94] Laurence A. Wolsey. *Integer Programming*. Wiley-Interscience, 1998.

[95] B. Wu, P.-H. Ho, and K. Yeung. Monitoring Trail: On Fast Link Failure Localization in All-Optical WDM Mesh Networks. *IEEE/OSA Journal of Lightwave Technology*, 27:4175–4185, 2009.

[96] B. Wu, P.-H. Ho, J. Tapolcai, and X. Jiang. A novel framework of fast and unambiguous link failure localization via monitoring trails. In *Proc. IEEE INFOCOM WIP*, San Diego, 2010.

[97] B. Wu, P.-H. Ho, K. Yeung, J. Tapolcai, and H. Mouftah. Optical Layer Monitoring Schemes for Fast Link Failure Localization in All-Optical Networks. *Communications Surveys & Tutorials, IEEE*, 13(1):114-125, 2011.

[98] B. Wu, P.-H. Ho, and K. Yeung. Monitoring Trail: a New Paradiagm for Fast Link Failure Localization in WDM Mesh Networks. *IEEE GLOBECOM*, 2008.

[99] B. Wu, K. Yeung, and P.-H. Ho. Monitoring Cycle Design for Fast Link Failure Localization in All-Optical Networks. *IEEE/OSA Journal of Lightwave Technology*, 27(10):1392–1401, 2009.

[100] T. Wu and A. Somani. Cross-talk attack monitoring and localization in all-optical networks. *IEEE/ACM Transactions on Networking (TON)*, 13(6):1401, 2005.

[101] G. Xue, W. Zhang, T. Wang, and K. Thulasiraman. On the partial path protection scheme for WDM optical networks and polynomial time computability of primary and secondary paths. *MANAGEMENT*, 3(4):625–643, 2007.

[102] H. Zeng and A. Vukovic. The variant cycle-cover problem in fault detection and localization for mesh all-optical networks. *Photonic Network Communications*, 14(2):111–122, 2007.

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

[104] Mankins, J.C. Technology readiness levels. *White Paper, April*, 1995.

[105] S. Huang, C. Martel, and B. Mukherjee. Survivable multipath provisioning with differential delay constraint in telecom mesh networks. *IEEE/ACM Transactions on Networking*, 19(3):644–656, 2011.

Publications

Journal Publications (4.5 p)

[J1] **Péter Babarczi**, János Tapolcai, and Pin-Han Ho. *Adjacent Link Failure Localization with Monitoring Trails in All-Optical Mesh Networks*, **IEEE/ACM Transactions on Networking (ToN)** vol. 19, no. 3, pp. 907–920 ($6/2 = 3$)

[J2] János Tapolcai, Pin-Han Ho, Lajos Rónyai, **Péter Babarczi**, Bin Wu. *Failure Localization for Shared Risk Link Groups in All-Optical Mesh Networks using Monitoring Trails*, **IEEE/OSA Journal of Lightwave Technology (JLT)** vol. 29, no. 10, pp. 1597–1606 ($6/4 = 1.5$)

[J3] **Péter Babarczi**, Péter Soproni, János Tapolcai, Pin-Han Ho, Tibor Cinkler. *Robust Dedicated Protection Approaches to Shared Risk Link Group Failures*, **under submission to Elsevier Computer Communications, Special Issue on Resilient Network-Based Services**

Book Chapters (6 p)

[B1] **Péter Babarczi**, János Tapolcai. *Protection Survivability Architectures: Principles and Challenging Issues*, to appear in *Resilient Optical Network Design: Advances in Fault-Tolerant Methodologies*, Editors: Y. S. Kavian and M. S. Leeson, Publisher: IGI Global ($6/1 = 6$)

Conference Proceedings Publications (10.25 p)

[C1] **Péter Babarczi**, and János Tapolcai. *End-to-end service availability guarantee with generalized dedicated protection*, In *Proc., IEEE CSNDSP*, pp. 511–515, Graz, Austria, 2008. july ($3/1 = 3$)

[C2] János Tapolcai, **Péter Babarczi**, and Pin-Han Ho. *Dedicated protection scheme with availability guarantee*, In *Proc., IEEE NETWORKS*, pp. 1–9, Budapest, Hungary, 2008. october ($3/2 = 1.5$)

[C3] **Péter Babarczi**, János Tapolcai, and Pin-Han Ho. *Availability-Constrained Dedicated Segment Protection in Circuit Switched Mesh Networks*, In *Proc., IEEE RNDM*, pp. 1–6, Saint Petersburg, Russia, 2009. october ($3/2 = 1.5$)

[C4] **Péter Babarczi**, János Tapolcai, Pin-Han Ho, and Bin Wu. *SRLG Failure Localization in Transparent Optical Mesh Networks with Monitoring Trees and Trails*, In *Proc., IEEE ICTON*, pp. 1–4, Munich, Germany, 2010. june ($3/3 = 1$)

[C5] Bin Wu, Pin-Han Ho, János Tapolcai, and **Péter Babarczi**. *Optimal Allocation of Monitoring Trails for Fast SRLG Failure Localization in All-Optical Networks*, In *Proc., IEEE GLOBECOM*, pp. 1–5, Miami, Florida, 2010. december ($3/3 = 1$)

[C6] **Péter Babarczi**, János Tapolcai, Pin-Han Ho, and Muriel Médard. *A Robust Dedicated Protection Approach to Shared Risk Link Group Failures using Network Coding*, **submitted to** IEEE ICC

[C7] Péter Soproni, **Péter Babarczi**, János Tapolcai, Tibor Cinkler, Pin-Han Ho. *A Meta-Heuristic Approach for Non-Bifurcated Dedicated Protection in WDM Optical Networks*, In *Proc., IEEE DRCN*, pp. 110–117, Krakow, Poland, 2011. october ($3/4 = 0.75$)

[C8] **Péter Babarczi**, János Tapolcai, and Pin-Han Ho. *SRLG Failure Localization with Monitoring Trails in All-Optical Mesh Networks*, In *Proc., IEEE DRCN*, pp. 188–195, Krakow, Poland, 2011. october ($3/2 = 1.5$)

Hungarian Journal Publications (0.25 p)

[H1] **Péter Babarczi**, Ferenc Tanai, Levente Csikor, János Tapolcai, and Zalán Heszberger. *Útvonalválasztás késleltetés-toleráns hálózatokban*, Híradástechnika vol. 66, no. 1, pp. 23–31 (in Hungarian) ($1/4 = 0.25$)

Independent Citations

[J1-1] Meng Wang, Weiyu Xu, Enrique Mallada, and Ao Tang. *Sparse Recovery with Graph Constraints: Fundamental Limits and Measurement Construction*, CoRR abs/1108.0443, pp. 1–9, 2011

[C5-1] Mao, M. and Yeung, K.L. *Super Monitor Design for Fast Link Failure Localization in All-Optical Networks*, IEEE International Conference on Communications (ICC), pp. 1–5, 2011

Index