



**HAL**  
open science

# A Method for Joint Routing, Wavelength Dimensioning and Fault Tolerance for any set of simultaneous failures on Dynamic WDM Optical Networks

Nicolas Jara, Reinaldo Vallejos, Gerardo Rubino

► **To cite this version:**

Nicolas Jara, Reinaldo Vallejos, Gerardo Rubino. A Method for Joint Routing, Wavelength Dimensioning and Fault Tolerance for any set of simultaneous failures on Dynamic WDM Optical Networks. *Optical Fiber Technology*, 2017, 38 (Supplement C), pp.30-40. 10.1016/j.yofte.2017.08.001 . hal-01663501

**HAL Id: hal-01663501**

**<https://inria.hal.science/hal-01663501>**

Submitted on 14 Dec 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Method for Joint Routing, Wavelength Dimensioning and Fault Tolerance for any set of simultaneous failures on Dynamic WDM Optical Networks

Nicolás Jara<sup>a,b</sup>, Reinaldo Vallejos<sup>a</sup>, Gerardo Rubino<sup>b</sup>

<sup>a</sup>*UTFSM, Valparaíso, Chile*

<sup>b</sup>*INRIA Rennes Bretagne-Atlantique, Rennes, France*

---

## Abstract

The design of optical networks decomposes into different tasks, where the engineers must basically organize the way the main system's resources are used, minimizing the design and operation costs and respecting critical performance constraints. More specifically, network operators face the challenge of solving routing and wavelength dimensioning problems while aiming to simultaneously minimize the network cost and to ensure that the network performance meets the level established in the Service Level Agreement (SLA). We call this the *Routing and Wavelength Dimensioning* (R&WD) problem.

Another important problem to be solved is how to deal with failures of links when the network is operating. When at least one link fails, a high rate of data loss may occur. To avoid it, the network must be designed in such a manner that upon one or multiple failures, the affected connections can still communicate using alternative routes, a mechanism known as *Fault Tolerance* (FT). When the mechanism allows to deal with an arbitrary number of faults, we speak about *Multiple Fault Tolerance* (MFT).

The different tasks before mentioned are usually solved separately, or in some cases by pairs, leading to solutions that are not necessarily close to optimal ones. This paper proposes a novel method to *simultaneously* solve all of them, that is, the Routing, the Wavelength Dimensioning, and the Multiple Fault Tolerance problems. The method allows to obtain: a) all the primary routes by which each connection normally transmits its information, b) the additional routes, called secondary routes, used to keep each user connected in cases where one or more simultaneous failures occur, and c) the number of wavelengths available at each link of the network, calculated such that the blocking probability of each connection is lower than a pre-determined threshold (which is a network

design parameter), despite the occurrence of simultaneous link failures.

The solution obtained by the new algorithm is significantly more efficient than current methods, its implementation is notably simple and its on-line operation is very fast. In the paper, different examples illustrate the results provided by the proposed technique.

*Keywords:* Dynamic WDM optical networks, Wavelength Conversion, Routing, Wavelength Dimensioning, Fault Tolerance.

---

## 1. Introduction

The numerous new Internet applications arriving nowadays require to transmit very large volumes of information. Think of Social networks, IPTV, HD Streaming, Video on Demand, Online Gaming, Cloud Computing, Internet of Things, Smart Cities, etc. This has caused a considerable  
5 increase in the demand for bandwidth to the communication infrastructure, leading to a significant rise in the use of optical networks based on WDM technologies. Currently, this type of network is operated statically [1], i.e., the route assigned to each user (connection) is permanently assigned from source to destination, regardless of the percentage of time it is used. This operation mode is inefficient in the usage of network resources, especially for low traffic loads, which is the most com-  
10 mon situation. One way to help overcome the inefficiencies of static networks consists of allocating the resources required by each user only when there is enough data to transmit. The resulting system is called a dynamic optical network. The possible lack of resources to successfully transmit a piece of information can happen because dynamic networks are designed based on a statistical commitment: from one side, for economy reasons dynamic networks are designed with the minimum  
15 possible amount of resources; on the other side, for effectiveness they are designed to avoid (or, more generally, to minimize) the occurrence of information losses due to lack of resources when needed (blocking). To achieve a balance between these two opposite goals, the network must be designed so that the blocking probability of any connection is less than or equal to a given threshold or bound, seen as a parameter of network design, which is usually a value close to 0. This can be  
20 refined by looking for a design where each connection  $c$  must have a blocking probability  $BP_c$  less than a given bound  $\beta_c$ , so, a specific restriction for each connection. In this way, we can give more or less importance (or priority) to some of the connections. This more general case is the one considered in this paper. The network cost definition can be addressed as an economic and commercial point of view. For example, representing the cost of how many optical fibers are needed on the

25 network, taking into account how many wavelengths each fiber can handle. However, this approach  
is highly volatile, since it quickly changes overtime, due the fast technology growth and commercial  
strategies. Hence, to measure the network cost in a stable and representative way, we choose to use  
the number of wavelengths on the network, as several works as done in the literature [2]. This is  
because the cost of many components in an optical network is strongly affected by this parameter.  
30 In fact, it determines how many infrastructure resources are needed on the network to achieve the  
network operation. Moreover, if the number of wavelengths required increases by any reason, the  
cost of the network components may be maintained or must be increased. This means that the  
economic network cost is a non-decreasing function of the network cost measured as the number of  
network wavelength.

35 Let us now describe the most important problems that need to be solved in order to design  
dynamic WDM networks taking into account the main services they must provide.

*Routing.* This is a basic component of the network operation: every connection is defined by a pair  
of nodes in the network, the source and the destination, and for each such pair the designer must  
assign a route to be followed by the data to be transmitted.

40 *Wavelength Dimensioning.* It must also be determined how many wavelengths should be assigned to  
each link of the network, in order to achieve the compromise between efficiency and cost previously  
described. As many works on the literature, we assumed each connection needs one wavelength  
at each link it uses, in order to transport its content. Dimensioning the number of wavelength  
varies significantly if the optical switches in the network have, or not, the ability of *wavelength*  
45 *conversion*. The latter means that if a switch receives a signal in a particular wavelength, the  
switch can transmit the signal through any wavelength that is available at that time in its output  
channel. Technology distinguishes some variants of the idea. In *Partial Wavelength Conversion* only  
some output ports in the optical router have the capability of wavelength conversion. In *Sparse*  
*Wavelength Conversion* only a small fraction of the optical routers can perform the operation. The  
50 *Sparse-Partial Wavelength Conversion (SPWC)* network architecture integrates the advantages of  
both variants. In SPWC architectures, from one side the network significantly reduces the number  
of wavelength converters needed, and on the other side, it is flexible enough for the carrier to migrate  
the backbone to support wavelength conversion, either by adding more converters into the router  
ports, or by replacing the common wavelength routers with new ones with conversion capabilities.

55 See [3] for some discussions about how the SPWC scheme has the flexibility to ease network carriers to gradually upgrade the optical networks to support wavelength conversion. Last, a router has *Full Wavelength Conversion* if every output port is equipped with wavelength conversion capacity, and we call *Complete Wavelength Conversion* the case of a network where every optical router on the network has some wavelength switching capacity. The network is of the *Full-Complete*  
60 *Wavelength Conversion* type if both properties hold. It was shown that the blocking probability of the SPWC scheme is pretty close to that obtained with a full-complete wavelength conversion (FCWC) network for the same network topology and the same link capacities (see Figure 5 and Figure 6 of [3]). Consequently, in order to simplify the mathematical analysis (while ensuring a good representation of reality and following the approach of [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]),  
65 we assume that the network has the FCWC capacity.

*Fault Tolerance (FT)*. A third major problem to be addressed is to ensure that the network is still able to provide its transmission service after the failure of one or more of its links. The solution to this problem consists of allowing the necessary infrastructure to rapidly re-establish communication between all source-destination pairs of nodes affected by these link failures. The frequency of link  
70 failures' occurrences is often significant. For instance, [16, 17] reports measures of the mean time between failures of about 367 year/km. This explains why failures on links may significantly impact the performance of the networks. For example, in a 26,000 km-long network such as NSFNet [18] (see Figure 4), there is an average of one fiber cut every 5 days. Moreover, it has been found that the frequency with which two simultaneous network failures occur is high enough to be taken into  
75 account in the design process. For example, in [17] it has been reported that the probability of two simultaneous failures occurring in a network like NSFNet is approximately 0.0027 (a downtime of about 24 hours per year on average), which in addition to the high transmission rate of this kind of networks, implies an unacceptable loss for the operator if the event happens. Even, some special cases of failure as Disaster risk constrains [19] and Shared-Risk-Group (SRG) [20] are important to  
80 be considered since their occurrence severely impact the network performance.

The previous elements justify the need to provide an efficient methodology for multiple fault tolerance, which should ensure (with a certain probabilistic guarantee) successful communications among all network users despite the occurrence of failures in some of the links, and at the lowest possible cost regarding the network infrastructure.

85 In this paper we propose a new procedure for simultaneously solve the routing, wavelength dimensioning and multiple link fault tolerance problems (which we call the *Joint3* method). This new approach obtains all primary and secondary routes assigned to each possible connection. Secondary routes replace the primary ones when the latter are affected by at least one failure, thus keeping users connected as long as the network itself remains connected (that is, for every con-  
90 nection (user) there is at least one path with operational links to reach their destination nodes). In our approach we go one step further, and we take into account the case of an arbitrary set of simultaneous failures. The method also evaluates the number of wavelengths  $W_\ell$  for each link  $\ell$  of the network, ensuring that the blocking probability of any connection request will be lower than the predefined threshold  $B$  previously mentioned, despite the possible occurrence of those simultaneous  
95 link failures.

The remainder of this paper is as follows: Section 2 describes the state of art related to Routing, Wavelength Dimensioning, and Fault Tolerance Capacity on Dynamic WDM Optical Networks. Section 3 presents the new algorithm. Section 4 contains some results obtained by the proposed algorithm, which are compared with those obtained with the current best methods in a set of  
100 different scenarios. Finally, the conclusions of the study are given in Section 5.

## 2. State of the Art

This section describes existent literature to solve the routing, wavelength dimensioning and fault tolerance problems on dynamic WDM optical networks.

*Routing and Wavelength Dimensioning (R $\mathcal{E}$ WD).* Both problems have been extensively studied on  
105 the literature because of their great impact on network cost (CapEx) and on network performance. To ease the difficulty of solving them, they have been historically solved separately. First, a specific procedure builds all the routes in the network. After that, the number of wavelengths on each link is calculated in order to satisfy the maximum blocking probability acceptable by every user (that is, by every connection), when the previously built set of routes is used. This maximal blocking  
110 probability per connection is a network design parameter denoted  $\beta_c$  for connection  $c$ . We will say that the set of capacities is *compatible* with the QoS constraints associated with these given thresholds.

Typically two hypotheses have been considered to design R&WD methods. The first one is used when defining the routes, where the shortest paths are chosen under the assumption that this  
115 minimizes the resource consumption [21, 22]. The second hypothesis is that selecting, within the set of shortest paths, a subset that allows to balance the load of each link in the network [23, 24] reduces the global blocking probability of the network. Consequently, the method mostly used to establish routes is the “Shortest Path Balanced Routing” (SPBR) one [25, 26, 27, 12, 28, 29, 30, 31, 32, 33]. However, these hypotheses present two problems: First, the strategy of balancing the load assumes  
120 that all network links need the same number of wavelengths associated with, something that rarely occurs in real networks. Under this assumption, it is not surprising that the SPBR method obtains less blocking probability than another routing procedure, because if a link receives more traffic load than the rest it will necessarily have a higher blocking probability, which in turn will determine the blocking level of the whole network. Using the just described assumptions, generally the wavelength  
125 dimensioning has been made assigning the number of wavelengths on each link homogeneously, this means, the same number of wavelength on each link. Solving separately the Routing and the Wavelength Dimensioning, it is possible to achieve good local solutions to both problems, but the approach misses the opportunity to find a good global solution found when solving them simultaneously. Since the wavelength dimensioning has been usually solved by simulation, the time  
130 consuming task involved does not allowed to explore every possible solution, requiring to divide the R&WD problem. One way to overcome this problem is to solve the dimensioning by using heuristics, but they should be validated extensively. Recently in [14], an exhaustively validated heuristic strategy called CPR (Cheapest Path Routing) was applied, solving both routing and wavelength dimensioning jointly. Its performance was better than local strategies approaches.

135 *Fault Tolerance (FT)*. The fault tolerance methods proposed so far have been generally devoted to finding alternative paths in case that a single failure occurs across the network. Then, the number of wavelengths in the network is dimensioned in order to tolerate this setting [34, 35, 17, 36, 37, 38, 39]. However, as observed above, the probability of occurrence of two or more simultaneous failures is high enough to be considered important, and therefore it should be taken into account in the network  
140 design. Some studies have focused on this scenario; see [40, 41, 42, 43, 44, 28, 30, 45]. Also, some studies have consider special cases of failure, such as Disaster risk constrains and Shared-Risk-Group scenarios. Disaster risk constrains [19, 46] considers the possible service disruptions in case of a

natural disaster or a targeted attack. On the other hand, Shared-Risk-Group (SRG) [20] considers the possibility that some fibers are placed physically together, even if they are connecting different  
145 optical nodes. This scenario makes them all liable to physical cuts, since they can be cut together at the same time.

Let us briefly list next the most common methods currently used to provide fault tolerance. One of the most frequent ways considered to address simple and double fault tolerance, called “1+1”, can be found in [36, 47, 35, 37, 38, 48]. In this technique, a secondary route is associated with each  
150 primary one, designed disjoint with the latter (in the sense that they don’t share any link), and the information is transmitted simultaneously through both of them. To dimension the number of wavelengths of each link (task usually done by simulation), each secondary route is considered as just another network route with a load equal to the corresponding primary one. The 1+1 method is scalable to provide tolerance to  $K \geq 1$  simultaneous failures. In this case, for each connection,  
155  $K + 1$  supplementary disjoint routes must be found, one to be designed as the primary route and the remaining  $K$  as secondary routes. Observe that a necessary and sufficient condition for this is that the graph defined by the set of nodes and links is  $(K + 1)$ -connected.

Another strategy is described in [49], where a routing technique is proposed to provide fault tolerance while sharing the resources of the network. This method is known as “*Shared Path Protection*” (SPP) [50, 49, 44]. In this method the extra resources (wavelengths) assigned to the  
160 secondary routes can be shared by different connections, and are assigned only when a fault occurs. It can be executed in two different ways. The first one consists of running the algorithm off-line, where the routes are calculated prior to the operation of the network (off-line SPP). The second way is the on-line implementation (on-line SPP). In this last case, the method is executed every time  
165 there is a change in the network (link failure or traffic load variation). In the SPP online mode, the primary routes are specified before the network is operating, but in order to find new routes to the affected connections it must be executed again every time that one or more simultaneous failures occur. For this reason, we say that this is a proactive and a reactive approach at the same time.

Finally, in [51, 52, 40, 42, 28, 30] a method of fault tolerance called “*p-cycle*” is used, allowing  
170 sharing resources through fixed secondary routes which have a cyclic form. These routes are shared between several primary routes. One problem with this approach is that the applicability of the idea is very dependent on the size of the network, introducing excessive additional delay for a connection in protection state on large networks. Also, to perform multiple fault tolerance, this



method requires a large amount of cycles (e.g. hundreds of cycles for the 11 nodes pan-European COST 239 network [52]), which is impractical from various points of view.

### 3. The Proposed Algorithm

This section contains the main contribution of the paper. We present first the model used and the related assumptions. Then, we describe the main sub-procedures necessary to our technique. Last, we present the algorithm in two steps, first without any fault tolerant capability, second including general fault tolerance.

#### 3.1. Model and assumptions

The network topology is represented by a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ , where  $\mathcal{V}$  is the set of network nodes or vertices and  $\mathcal{L}$  is the set of unidirectional links (the arcs in  $\mathcal{G}$ ), with respective cardinalities  $|\mathcal{V}| = V$  and  $|\mathcal{L}| = L$ . The set of connections  $\mathcal{X} \subseteq \mathcal{V}^2$ , with cardinality  $|\mathcal{X}| = X$ , is composed by all the source-destination pairs with communication between them. These connections are also called “users” in the text.

To represent the traffic between a given source-destination pair, an ON-OFF model is used. Consider connection  $c$ . During any of its ON periods, whose average length is  $t_{ON_c}$ , the source transmits at a constant rate (which is the rate associated to the wavelength). During an OFF period, with average length  $t_{OFF_c}$ , the source refrains from transmitting data.

To simplify the presentation, the constant transmission rate during the ON periods is determined by the used technology, that will be our rate unity. Consequently, the traffic load of connection  $c$ , denoted by  $\varrho_c$ , is given by the following expression:

$$\varrho_c = \frac{t_{ON_c}}{t_{ON_c} + t_{OFF_c}}. \quad (1)$$

Observe that we address here the general case where the load can be different for each connection, the so-called *heterogeneous* situation, instead of the usual homogeneous case where the load is assumed to be the same for all users [25, 26, 27, 12, 28, 29, 30, 31, 32, 33].

The specific protocols used in these types of networks also depend on many other systems’ characteristics (e.g. memory, administration type, etc.) than considered here. Hence, we restricted our work to obtain a methodology for the routes assignment and calculation of wavelength number

for each network link. The rest remains to be defined according to the system's practical aspects, and it's out of the scope of this paper.

Let  $\mathcal{R} = \{r_c \mid c \in \mathcal{X}\}$  be the set of routes that enable communications among the different users, where  $r_c$  is the route associated with connection  $c \in \mathcal{X}$ . The set  $\mathcal{R}$  is also denominated set of *primary* routes, because this set alone does not offer any fault tolerance to the possible failure of network links. The set of connections passing through link  $\ell$  in the graph is denoted by  $\mathcal{T}_\ell$ , and we denote by  $N_\ell$  their number, that is,  $N_\ell = |\mathcal{T}_\ell|$ .

Let  $\mathcal{W} = \{W_\ell \mid \ell \in \mathcal{L}\}$  be the set containing the number of wavelengths associated with each unidirectional network link, where  $W_\ell, \ell \in \mathcal{L}$ , is the number of wavelengths on link  $\ell$ . The value  $W_\ell$ , for every  $\ell \in \mathcal{L}$ , will be evaluated so that the blocking probability  $BP_c$  of each connection  $c \in \mathcal{X}$  is less than or equal to a given pre-specified threshold  $\beta_c$  and the total number of available network wavelengths is as small as possible.

The chosen number of wavelengths per link varies significantly if the optical switches in the network have or have not the ability of wavelength conversion. When the network nodes are not capable of wavelength conversion, the order in which the wavelengths are used on each link is important, since the total number of wavelengths required by the network is affected by it [53]. However, when the switches have full wavelength conversion capacity, the order of wavelengths assignment does not affect the number of necessary wavelengths. As mention before, this paper focuses on the full wavelength conversion case, therefore only the number of necessary wavelengths is important regardless of their order.

As in [53, 2, 14], in this work the total network cost  $C_{net}$  is defined as the sum of all wavelengths of all network links, that is,  $C_{net} = \sum_{\ell \in \mathcal{L}} W_\ell$ . As we are considering fault tolerance capabilities, this includes all the extra wavelengths needed to provide tolerance to multiple link failures.

Consider a set of every possible failure scenarios  $\mathcal{H}$  on the network to be considered. Each of these scenarios is a subset of failed network links  $\mathcal{F}$ , where  $\mathcal{F} \subset \mathcal{L}$  with cardinality  $|\mathcal{F}| \leq K$ . This subset  $\mathcal{F}$  corresponds to a failure scenario, where every links belonging to  $\mathcal{F}$  is non operational (failed). Therefore, this method can be applied to any possible case of failures in the network, e.g.: every possible single failure scenario ( $|\mathcal{F}| = 1$ ) which corresponds to the case of single link failure; every possible double failure scenario ( $|\mathcal{F}| = 2$ ); node failure (corresponding to a special case of multiple failure, where all the links connected to that node are non operational), etc. Similarly, we can also handle disaster risk constrains [19, 46] and Shared-Risk-Group constrains scenarios [20].

Disaster risk constraints can be considered in our method by including in the failure scenario  $\mathcal{F}$  all the links where the disaster has taken place. Also, Shared-Risk-Group (SRG) can be considered also  
230 as a special case of multiple link failures, where  $\mathcal{F}$  is composed by every link that can be affected by the same physical cut.

Assume that a set of routes  $\mathcal{R}$  and a set of capacities  $\mathcal{W}$  are given. Assume some failure scenario  $\mathcal{F}$  is also given (or observed), that is, assume that the links in  $\mathcal{F}$  are failed, and those that don't belong to  $\mathcal{F}$  work. Some additional definitions required for this method are presented in the  
235 following list:

- $\mathcal{G}^{-\mathcal{F}} = (\mathcal{V}, \mathcal{L} \setminus \mathcal{F})$ , is the partial graph of  $\mathcal{G}$  (same nodes, part of the edges), containing only the non-failed links;
- $\mathcal{X}^{\mathcal{F}} = \{c \in \mathcal{X} \mid \text{there exists } \ell \in \mathcal{F} \text{ with } \ell \in r_c\}$ , is the set of all network connections whose routes pass through at least one failed link;
- 240 •  $\mathcal{A}^{\mathcal{F}} = \{r_c \in \mathcal{R} \mid \text{there exists } \ell \in \mathcal{F} \text{ with } \ell \in r_c\}$ , is the set of routes that are affected when all links in  $\mathcal{F}$  fail; that is,  $\mathcal{A}^{\mathcal{F}}$  is the subset of the routes in  $\mathcal{R}$  that are associated with the connections in  $\mathcal{X}^{\mathcal{F}}$ ;
- $\mathcal{R}^{\mathcal{F}}$  is a set of routes that replace those in  $\mathcal{A}^{\mathcal{F}}$  when all links in  $\mathcal{F}$  are failed; for each of these routes, the corresponding blocking probability must be less than or equal to the corresponding  
245 given upper bound;
- $\mathcal{S}^{\mathcal{F}}$  is the total set of routes guaranteeing fault tolerance to the failure event “all links in  $\mathcal{F}$  fail”, that is, the set defined by  $\mathcal{S}^{\mathcal{F}} = (\mathcal{R} \setminus \mathcal{A}^{\mathcal{F}}) \cup \mathcal{R}^{\mathcal{F}}$ .

### 3.2. Sub-procedures needed by our Joint3 method

The method needs a few sub-procedures to work. They are described now. Since the given  
250 graph and the set of connections (or users) are fixed data that never change, as well as the upper bounds  $\beta_c$  of the blocking connection probabilities, we will omit them in the list of the parameters of the procedures. Also, when we refer to the network cost, we will write simply  $C_{net}$  because we must in general change many times during the computational process the capacities of the links.

255 *InitialRoutes()*: *initial computation of routes*. Our method starts by computing an initial set of routes. This can be done by any available technique. Typically, for each connection  $(s, t)$  we look for the shortest route from  $s$  to  $t$  in the graph, implicitly assuming that the capacity of all involved links (that is, the number of wavelengths available at each arc of the graph) is infinite. Let us symbolically write  $\mathcal{R} := \text{InitialRoutes}()$  to represent the execution of this sub-procedure.

260 *Dimensioning()*: *given the routes and the thresholds  $\beta_c$ , compute the capacities*. We call *Dimensioning* this sub-procedure. The problem consists in finding, for each link  $\ell \in \mathcal{L}$ , a capacity  $W_\ell$  such that the end-to-end blocking probability  $BP_c$  of every user  $c \in \mathcal{X}$  passing through the link  $\ell$  is less than the given threshold or upper bound  $\beta_c$ .

The dimensioning of each network link is obtained in 2 steps. First, for each link in the network we compute a threshold  $\delta_\ell$  such that, under specific assumptions (later explained), if the blocking probability of link  $\ell$ ,  $LBP_\ell$ , satisfies  $LBP_\ell \leq \delta_\ell$ , then for all connection  $c$ ,  $BP_c \leq \beta_c$ . Second, an iterative procedure is executed in order to find an appropriate number of wavelengths using the computed bounds  $\delta_\ell$ s on the blocking probability of the links.

Let us explain now the first step. Consider connection  $c$  and suppose for a moment that its blocking probability satisfies  $BP_c \leq \beta_c$ . Suppose also that the blocking of links belonging to connection  $c$  are independent of each other. This is obviously false in general, but it is an usual and useful trick, and it has been “numerically validated” [54], that is, it leads to very accurate approximations (see below). Under these conditions, we have:

$$BP_c = 1 - \prod_{\ell \in r_c} (1 - LBP_\ell) \leq \beta_c.$$

Let us *define* a new threshold  $\alpha_c = 1 - (1 - \beta_c)^{1/|r_c|}$ , that is, a threshold depending on the connection only. It is now easy to check that if for each link  $\ell \in r_c$  we have  $LBP_\ell \leq \alpha_c$ , then for all  $c$ ,  $BP_c \leq \beta_c$ :

$$\begin{aligned} BP_c &= 1 - \prod_{\ell \in r_c} (1 - LBP_\ell) && \text{(links are assumed to behave independently)} \\ &\leq 1 - \prod_{\ell \in r_c} (1 - \alpha_c) \\ &= 1 - (1 - \alpha_c)^{|r_c|} \\ &= 1 - (1 - \beta_c) \\ &= \beta_c. \end{aligned}$$

Now, observe that, in general, a same link will belong to many connections. To take into account this, the final upper bound  $\delta_\ell$  that we will use on the link blocking probabilities is:

$$\delta_\ell = \min_{c: r_c \ni \ell} \{\alpha_c\} = \min_{c: r_c \ni \ell} \{1 - (1 - \beta_c)^{1/|r_c|}\}.$$

It remains to specify the procedure used to compute the link blocking probabilities. For this purpose, we use the method proposed in [55], a fast and simple numerical evaluation considering heterogeneous traffic loads on each network connection.

To find the capacities of each of the links, observe first that if  $N_\ell$  is the number of routes

```

function Dimensioning( $\mathcal{R}$ )
// --- input: the graph (the network), the connections
//      and the bounds on the blocking probabilities,
//      all seen as global variables
//      and a set  $\mathcal{R}$  of routes
// --- output: a set  $\mathcal{W}$  of wavelengths per link
//      satisfying the QoS constraints

for all link  $\ell$ 
    // --- start with capacity equal to # of passing connections
     $W_\ell := N_\ell$ ;
    // --- decrease  $W_\ell$  until QoS constraint violated
     $LBP_\ell := LinkBlockingProbability(\ell)$ ;
    while  $LBP_\ell \leq \delta_\ell$ 
         $W_\ell := W_\ell - 1$ ;
         $LBP_\ell := LinkBlockingProbability(\ell)$ ;
    // --- assign last value satisfying the QoS constraint
     $W_\ell := W_\ell + 1$ ;

return  $\mathcal{W}$ 

```

Figure 1: Dimensioning the links: this procedure assigns a number  $W_\ell$  of wavelengths to link  $\ell$ , for each  $\ell$ , such that the blocking probability of connection  $c$  is less than the beforehand specified bound  $\beta_c$ , for each  $c$ .

using link  $\ell$  (easily computed from the set of routes), then we necessarily have  $W_\ell \leq N_\ell$ . Now, the

idea is to initially assign to each link  $\ell$  a capacity  $W_\ell = N_\ell$ , in which case obviously there is no blocking, that is, all link blocking probabilities are equal to zero. Then, we start diminishing the capacities while checking that the link blocking probabilities are less than the thresholds  $\delta_\ell$ s. The process continues until the inequalities  $LBP_\ell \leq \delta_\ell$  are not anymore satisfied, for all  $\ell \in \mathcal{L}$ . Put in simply algorithmic form, this can be written as shown in Figure 1.

In the pseudo-code (Figure 1), *LinkBlockingProbability*( $\ell$ ) refers to the procedure described in [55] used to calculate the blocking probability at a specific link (link  $\ell$  here). Symbolically, the execution of the whole dimensioning sub-procedure will be written  $\mathcal{W} := \textit{Dimensioning}(\mathcal{R})$ , since its output is the set of wavelengths.

*NewRoutes*( $\cdot$ ): *given the routes and the capacities, find new routes.* The problem is the following: we have a set of routes and the capacities of the links compatible with the QoS constraints given by the thresholds  $\beta_c$ s, that is, satisfying that for all  $c \in \mathcal{X}$  we have  $BP_c \leq \beta_c$ . The resulting network has a total cost  $C_{net}$ , equal to the sum of the capacities of all its links. We would like to find a new set of routes leading to a total network cost less than  $C_{net}$ . In other words, we would like to find new routes such that the sum of the wavelengths assigned to the different links is smaller than with the initially given ones, while still satisfying the QoS required by each connection.

The idea of the sub-procedure is as follows. From the initial set of routes  $\mathcal{R}$ , we define, for each link  $\ell$ , its *relative cost*  $\gamma_\ell$  as the ratio  $\gamma_\ell = W_\ell/N_\ell \leq 1$  (recall that  $N_\ell$  is the number of connections that pass through link  $\ell$ ).

Observe that  $\gamma_\ell$  is the amount of resources (wavelengths or a wavelength's fraction) needed by the system to serve one user (with the quality of service required). This means that if we use this number to weight link  $\ell$ , and if we define the *relative length* of a path as the sum of the relative costs of its links, then short paths, using this definition, are not always better than long ones.

The search for new routes is then done as follows. We run Dijkstra's algorithms looking, for each user  $c \in \mathcal{X}$ , for the cheapest route where the link costs are now given by the relative costs  $\gamma_\ell$ . This produces a new set of routes that we denote  $\tilde{\mathcal{R}} = \{\tilde{r}_c \mid c \in \mathcal{X}\}$ . In pseudo-algorithmic form, we write  $\tilde{\mathcal{R}} := \textit{NewRoutes}(\mathcal{R}, \mathcal{W})$ .

*BestOf*( $\cdot$ ): *given the ancient routes and the new ones provided by function *NewRoutes*( $\cdot$ ), find the best of the two sets.* The idea is to choose which one of the two sets of routes  $\mathcal{R}$  and  $\tilde{\mathcal{R}}$  has the smallest cost, that is, the smallest total number of wavelengths (this is the sum of the wavelengths

used by each of the network links); the one with the smallest cost is called the “best” of the two, denoted by  $\mathcal{R}'$ . Symbolically, we write  $\mathcal{R}' := \text{BestOf}(\mathcal{R}, \tilde{\mathcal{R}})$ .

305 Observe that this doesn’t guarantee that the cost of the network with the new routes  $\mathcal{R}'$  will be less than before executing the procedure. More explicitly, if we dimension again the links (since the routes have changed) and we obtain a new set of capacities  $\mathcal{W}'$  (that is, if we execute  $\mathcal{W}' := \text{Dimensioning}(\mathcal{R}')$ ), then we can have  $C_{net}(\mathcal{W}') \geq C_{net}(\mathcal{W})$ . To see this, observe that we are in a finite model, meaning that there is a global minimal cost. Then, if the new routes together  
 310 with their dimensioning always led to strictly smaller costs, we would have a contradiction with the existence of the global optimum.

### 3.3. Algorithm without any fault tolerance capability

Let us present first the case where we don’t consider secondary routes for ensuring fault tolerance to links’ failures. The idea is to start by any set of routes, as indicated above, for which we find  
 315 a first assignment of capacities (wavelengths), that is, for which we execute a first dimensioning step. Then, we continue by iterating the following actions: we find a new set of routes ( $\mathcal{R}'$ ) by means of the *NewRoutes()* procedure, then we choose the best (cheapest) set of routes between the new routes and the previous routes ( $\mathcal{R}$ ) by means of *BestOf()* storing them on  $\mathcal{R}''$ , then we dimension again finding new capacities. We continue this scheme until some criteria is satisfied.  
 320 In our method, we stop iterating when the last  $M$  iterations couldn’t diminish the network cost, where  $M$  is a supplementary parameter of the global procedure. Since the best routes are chosen as a group, we use this criteria to ensure that there is no better solution considering our relative costs on every link  $\gamma_\ell$ .

In algorithmic form, the algorithm chains the seen procedures as shown in Figure 2.

325 Observe that the method allows looking for new routes using the relative cost concept, suggesting the use of a fixed point equation (that is, an iterative process) that strongly improves the global optimization process (see the numerical illustrations below).

The complexity of the procedure (Fig 2) is proportional to the evaluation of the cheapest routes for every connection in  $\mathcal{X}$ . For example, if  $\mathcal{X}$  is composed by all the possible source-destination  
 330 pair of nodes, then the complexity is given by the Floyd-Warshall’s algorithm, which is  $O(N^3)$ . Additionally, the algorithm complexity is proportional to the number of iterations required to find the cheapest routes (this parameter, denoted as  $I$ , in practice turns out to be less than 10). Then,

```

function R&WDwithoutFT()
// --- input: the graph (the network), the connections
//      and the bounds on the blocking probabilities,
//      all seen as global variables
// --- output: the routes and the wavelengths per link

 $\mathcal{R}$  := InitialRoutes(); // shortest paths
 $\mathcal{W}$  := Dimensioning( $\mathcal{R}$ ); // --- with  $\mathcal{W}$ , QoS is satisfied
 $cmin$  := Cost( $\mathcal{W}$ ); // ---  $cmin = W_1 + \dots + W_L$ 
counter := 0;
do
     $\mathcal{R}'$  := NewRoutes( $\mathcal{R}, \mathcal{W}$ ); // - cheapest paths; weights = relative costs
     $\mathcal{R}''$  := BestOf( $\mathcal{R}, \mathcal{R}'$ );
    if  $\mathcal{R} == \mathcal{R}''$  exit; // - leave the do loop
     $\mathcal{W}''$  := Dimensioning( $\mathcal{R}''$ );
    if  $C_{net}(\mathcal{W}'') < cmin$  then
        // --- improvement
         $cmin$  :=  $C_{net}(\mathcal{W}'')$ ;  $\mathcal{R} := \mathcal{R}''$ ;  $\mathcal{W} := \mathcal{W}''$ ; counter := 0;
    else
        counter += 1;
until counter =  $M$ 
return ( $\mathcal{R}, \mathcal{W}$ )

```

Figure 2: Slightly more explicit description of function R&WDwithoutFT() presented in Figure 2.

the complexity of the method proposed to solve R&WD without any fault tolerance (considering all the possible pair of nodes in  $\mathcal{X}$ ) is order  $O(N^3I)$ . For example, when the procedure is executed on a network topology as the ARPANET, which has 20 nodes, 67 links, and is solved with a mean number of iterations  $I = 5$ , the total number of operations required is in the order of  $10^4$  operations, which is executed in less than 0.01 second in a current standard PC (Windows 10 64 bits, Intel Core i7 2.60GHz processor and 8GB RAM).



Let us now look at the general case, where we will also look for auxiliary routes in case of failures, always respecting the QoS constraints.

### 3.4. Multiple Fault Tolerance Capacity

```

function R&WDwithFTmultiple()
// --- input: the graph (the network), the connections,
//   the bounds on the blocking probabilities,
//   and  $H$  subsets  $\mathcal{F}_1, \dots, \mathcal{F}_H$  of links
//   where at most one of the events ‘all links in  $\mathcal{F}_h$ 
//   fail simultaneously’ occurs, all seen as global variables
// --- output: the primary routes, the secondary routes
//   and the wavelengths per link

 $(\mathcal{R}, \mathcal{W}) := R\&WDwithoutFT()$ ;
for  $h := 1..H$ 
     $\mathcal{S}_h := SecondaryRoutes(\mathcal{R}, \mathcal{W}, \mathcal{F}_h)$ ;
     $\mathcal{W}' := Dimensioning(\mathcal{S}_h)$ ;
     $cmin := Cost(\mathcal{W})$ ;
    do
         $\mathcal{S}_h := SecondaryRoutes(\mathcal{S}_h, \mathcal{W}, \mathcal{F}_h)$ ;
         $\mathcal{W}' := Dimensioning(\mathcal{S}_h)$ ;
        if  $Cost(\mathcal{W}') < cmin$  then
            // --- improvement
             $cmin := Cost(\mathcal{W}')$ ;  $\mathcal{W}^h := \mathcal{W}'$ ;
        until no improvement in network cost in the last  $M$  iterations
    for  $\ell := 1..L$ 
         $W_\ell := Max(\mathcal{W}^\ell, \mathcal{W}_\ell^1, ..\mathcal{W}_\ell^H)$ 
return  $(\mathcal{R}, \mathcal{S}, \mathcal{W})$ 

```

Figure 3: Algorithm for solving the R&WD problem, providing alternative routes if the links of one specific subset of links fail (all together) belonging to a list of possible subsets of links that can fail.

We first use previous algorithm to obtain a set of routes  $\mathcal{R}$  and a set of capacities  $\mathcal{W}$  compatible with the QoS constraints  $\beta_c$ ,  $c \in \mathcal{X}$ , assuming no failures. Then, we consider that we have a set of every possible failure scenarios  $\mathcal{H}$ , where each of these scenarios is a subset of failed network links  $\mathcal{F}$  that can have up to  $K \geq 1$  simultaneous link failures (obviously,  $K$  is usually 1 or 2, but our method is general). To explain how the procedure works, just assume that the only possible failure event is the simultaneous failures of all links in a specific subset  $\mathcal{F}$  of  $\mathcal{L}$ .

We first start by finding replacement routes in case of the failure of all links in a single subset of links  $\mathcal{F}$ . If a route  $r_c$  doesn't use any link of  $\mathcal{F}$ , it is not changed. But for all connections  $c$  whose route  $r_c \in \mathcal{R}$  uses at least one link of  $\mathcal{F}$  (that is, for all  $c \in \mathcal{X}^{\mathcal{F}}$ ), we must find a new route that avoids the links of  $\mathcal{F}$ . As in previous subsection, for every link  $\ell \in \mathcal{L} \setminus \mathcal{F}$ , we define its relative cost  $\gamma_\ell$  using the capacities in  $\mathcal{W}$  by the expression  $\gamma_\ell = W_\ell/N_\ell$ , and then, with these  $\gamma_\ell$ s as weights, we run Dijkstra's algorithm to find the cheapest route for each connection  $c \in \mathcal{X}^{\mathcal{F}}$ , which we denote by  $r'_c$  here. The set of all these routes will be denoted by  $\mathcal{R}^{\mathcal{F}}$ .

Symbolically, and adding more parameters to *NewRoutes()* since we are changing, for instance, the link set, we execute the call  $\mathcal{R}^{\mathcal{F}} := \text{NewRoutes}(\mathcal{L} \setminus \mathcal{F}, \mathcal{X}^{\mathcal{F}}, \mathcal{R}, \mathcal{W})$ . Observe that  $\mathcal{R}$  is indexed on  $\mathcal{X}$  but the function uses only the routes in  $\mathcal{A}^{\mathcal{F}}$ , that is, those whose indexes in  $\mathcal{R}$  belong to set  $\mathcal{X}^{\mathcal{F}}$ . Similarly,  $\mathcal{W}$  is indexed on  $\mathcal{L}$  but we are concerned here only by the links in  $\mathcal{L} \setminus \mathcal{F}$ .

Then, define the set of routes:

$$\mathcal{S}^{\mathcal{F}} = (\mathcal{R} \setminus \mathcal{A}^{\mathcal{F}}) \cup \mathcal{R}^{\mathcal{F}}.$$

In words,  $\mathcal{S}^{\mathcal{F}}$  is the set of routes to use when all links in  $\mathcal{F}$  are failed. Now, we must dimension again the links because we must always respect the QoS constraints. For this purpose, we restrict the analysis to the graph  $\mathcal{G}^{-\mathcal{F}}$ , that is, we remove the links in  $\mathcal{F}$  from  $\mathcal{L}$ , and we run a dimensioning phase as in previous section. In pseudo-algorithmic form, we execute the function call  $\mathcal{W}^{\mathcal{F}} := \text{Dimensioning}(\mathcal{L} \setminus \mathcal{F}, \mathcal{S}^{\mathcal{F}})$ .

Next, we proceed as in the algorithm without any fault tolerance capability: since we have now new relative costs for the links in  $\mathcal{L} \setminus \mathcal{F}$ , we can: run again Dijkstra, compute new routes, then dimension again the links in  $\mathcal{L} \setminus \mathcal{F}$ , recompute relative costs, and so on, stopping the process when a convergence criteria is satisfied. For this purpose, each time we dimension again the links in  $\mathcal{L} \setminus \mathcal{F}$ , we compute the new network cost  $C_{net}$  and we compare it to its previous value.

To finish this process, since we obtain now new capacities for the links in each possible  $\mathcal{L} \setminus \mathcal{F}$ ,

370 stored in  $\mathcal{W}_\ell^{\mathcal{F}}$ , for every possible failure scenario in  $\mathcal{H}$ , we compare each  $W_\ell^{\mathcal{F}}$  (and the  $W_\ell$  obtained in the no failure scenario), for all  $\ell \in \mathcal{L}$ , and the final capacity of the link will be the maximum between them. Formally, we add a procedure  $Max()$  that performs this task, and we symbolically write  $\mathcal{W}_\ell := Max(\mathcal{W}_\ell, \mathcal{W}_\ell^1, \dots, \mathcal{W}_\ell^{\mathcal{H}})$ . Finally, putting together each final link capacity  $\mathcal{W}_\ell \in \mathcal{L}$  will conform the final dimensioning set  $\mathcal{W}$ .

375 The corresponding pseudo-code is presented in Figure 3.

The complexity of the fault tolerance procedure presented on Fig 3 is proportional to the number of link failures scenarios in  $\mathcal{H}$ , the complexity of Dijkstra's algorithm ( $O(N^2)$ ) to find the cheapest routes for every connection in  $\mathcal{X}^{\mathcal{F}}$  and the number of iterations required for the algorithm to find the cheapest routes (again defined as  $I$ , which in practice turns out to be less than 10). The set  $\mathcal{X}^{\mathcal{F}}$  380 cardinality is given by  $X$  the total number of network connections, multiplied by the mean number of hops of each connection route, which is denoted by  $\bar{r}$ , divided by the total number of network links  $L$ . Then, the procedure complexity is order  $O(HN^2 \frac{X \cdot \bar{r}}{L} I)$ . For example, in the case of single link failure (in this case the number of link failures scenarios  $H$  is given by the  $L$  possible single link failure) the complexity of the proposed method is order  $O(LN^2 \frac{X \cdot \bar{r}}{L} I)$ . On the other hand, if the 385 algorithm is used to solve the problem of double link failures, the link failure scenarios are  $H = L^2$ ; therefore the algorithm complexity is order  $O(L^2 N^2 \frac{X \cdot \bar{r}}{L} I)$ .

For example, to execute the method on the ARPANET network topology (20 nodes and 67 links), considering that the mean number of iterations  $I = 5$  and the mean number of hops on the network routes  $\bar{r}$  is equal to 3, the total number of operations required to solve single and double 390 link failure is in the order of  $10^8$  operations, which is executed in a time close to 1 second in a current normal PC. This theoretical complexity agrees strongly with the practical values obtained in practice (see section 4).

#### 4. Examples

To compare different methods, it is necessary to evaluate their performances with respect to 395 relevant metrics that enable to assess the advantages/disadvantages of each of them. The most important metrics for the routing, dimensioning and fault tolerance algorithms are: the cost of the network and the delay in the restoration procedure in case of the occurrence of failures, when this is relevant (in our approach this delay is negligible since the computations are done off-line).

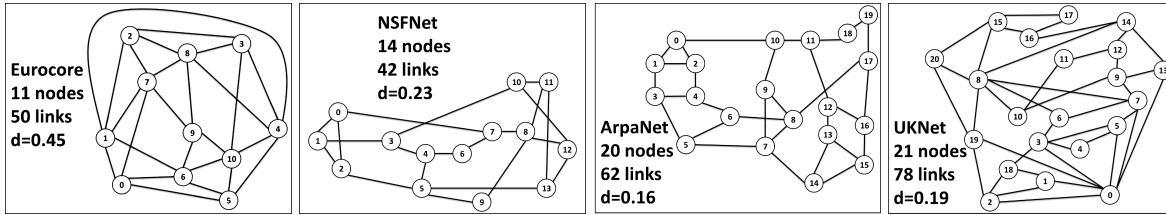


Figure 4: Mesh networks evaluated. The number of links refers to the number of bidirectional arcs. Observe that in the picture we see the edges (for instance, the picture shows the Eurocore topology with 25 edges, which corresponds to 50 arcs). The parameter  $d$  is a measure of density: if the graph has  $a$  arcs (twice the number of edges) and  $n$  nodes,  $d = a / (n(n - 1))$ .

Ideally, one of the things that should be done when evaluating any method in this area is to  
 400 compare its output with the (an) optimal solution, but this is impossible due to the high cost in time  
 and memory of the execution of the necessary optimization procedure, even for small topologies  
 [56, 57]. Because of this, the selected alternative was to compare the Joint3 method with the most  
 frequently referenced current techniques.

When reviewing the current methods of Routing and Wavelength Dimensioning, it was noted  
 405 that the algorithm most commonly referenced today, and considered the best so far, is SPBR  
 [25, 26, 27, 12, 28, 29, 30, 31, 32, 33] (see Section 1). Consequently, this was the method chosen for  
 comparison.

There are several types of fault tolerance algorithms proposed so far, such as Shared Path  
 Protection,  $p$ -cycle and 1+1. In the sequel we discuss the pertinence in comparing Joint3 with each  
 410 of these different types of algorithms.

*Shared Path Protection (SPP) Method.* This strategy provides tolerance to multiple network links  
 failure. There are two methods for implementing this algorithm (SPP on-line and SPP off-line).  
 Both methods require between 40 to 80% of additional wavelengths (compared to the case without  
 fault tolerance) to provide single link fault tolerance capability [49]. Another aspect that must  
 415 be considered is that the SPP off-line method has the additional weakness that the percentage of  
 restorability obtained (percentage of connections that remain connected in case of link failure) is  
 very low (80% to 90% [49]). Therefore, it is not comparable to the method proposed in this work,  
 which ensures that the blocking probability pre-established by the network designer (i.e.  $10^{-6}$ ) is  
 satisfied. On the other hand, the implementation of the SPP-online method requires to run on

420 demand a route search algorithm (whenever one or more links fail) in order to find an alternative route to each affected connection. Evidently, this on-line strategy causes a slow re-routing, which added to the fact that many of the applications that use computer networks require very fast on-line responses in case of failures [1], make that this type of method does not represent a practical fault tolerant mechanism for many practical applications. Due to the facts just commented, the SPP  
425 method was not considered for comparison with the method proposed in this paper.

*The p-cycle Method.* To provide tolerance to multiple failures, this method requires a large quantity of cycles (which implies a high cost when defining secondary routes), so it is not scalable for multiple faults. Given the fact that in this paper we consider the multiple fault tolerant case, it is unreasonable to compare our method with the  $p$ -cycle one.

430 *Method 1+1.* This method provides tolerance to multiple failures, using as many disjoint routes as simultaneous link failures it can tolerate. It solves the problem of primary and secondary routes prior to the network dimensioning (off-line) sub-task. Then, the number of wavelengths is computed, having as a constraint to provide enough resources to all routes, and guaranteeing sufficient information to re-route each connection in case of failure. Consequently, 1+1 is the most suitable  
435 fault tolerance method to compare with our algorithm.

In summary, the most appropriate methods for comparison are SPBR in order to generate the primary routes and then, 1+1 for the fault tolerance mechanism. These two methods together are denoted SPBR1+1 in the paper. To evaluate the corresponding blocking probabilities, the mathematical method proposed in [55] was used in both SPBR1+1 and Joint3.

440 To evaluate the performance of the methods under different scenarios, the algorithms were executed for different real network topologies, having different sizes and different degrees of connection  $d$ , where  $d$  is the average number of neighbors of a node in the network. Some of the selected topologies and their respective parameters  $N$ ,  $L$  and  $d$  are shown in Figure 4.

In Figure 5 we show the total cost  $C_{net}$  obtained by the Joint3 and SPBR1+1 methods for the  
445 case of a single link failure, as a function of the traffic load, for different network topologies, and for a maximum acceptable blocking connection of  $10^{-6}$ . Additionally, Figure 6 shows the  $C_{net}$  value for the same methods, but for double link failures. Note that the total cost  $C_{net}$  is given by the total number of wavelengths necessary to perform in both cases, with or without fault tolerance capability, that is, including the primary and secondary routes of every connection  $c$  (but only taking

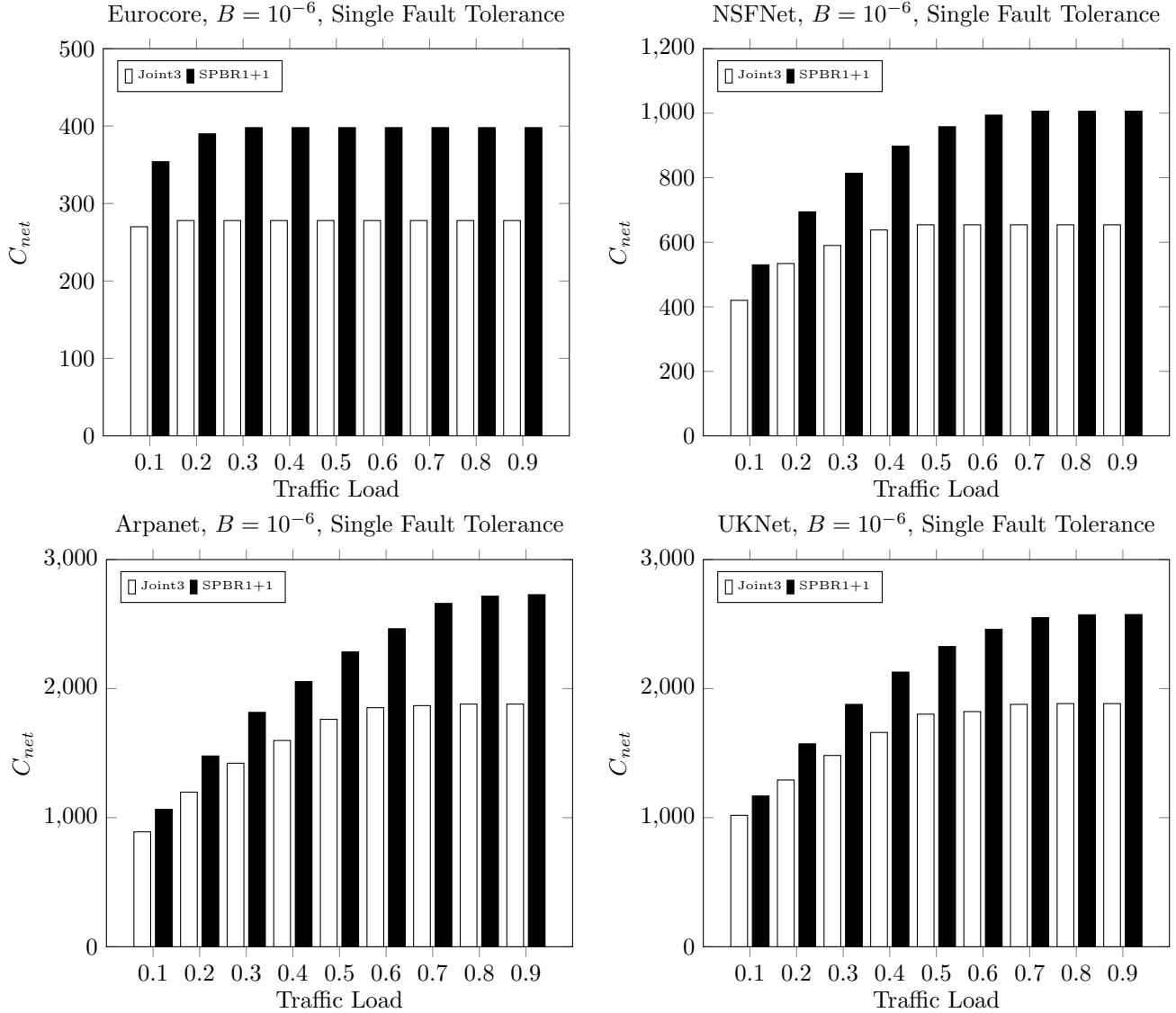


Figure 5: The total number of wavelengths  $C_{net}$  obtained with our method (Joint3) and SPBR1+1 on Eurocore, NSFNet, Arpanet and UKNet real mesh network topologies, for different connection traffic loads, in the single fault tolerance case.

450 into account the actual traffic that can transmit each possible scenario). Also, it is important to notice that in this section only single and double link failure scenarios are shown. This is because to represent another kind of failures scenarios, such as SRG and Disaster Risk constrains, the total

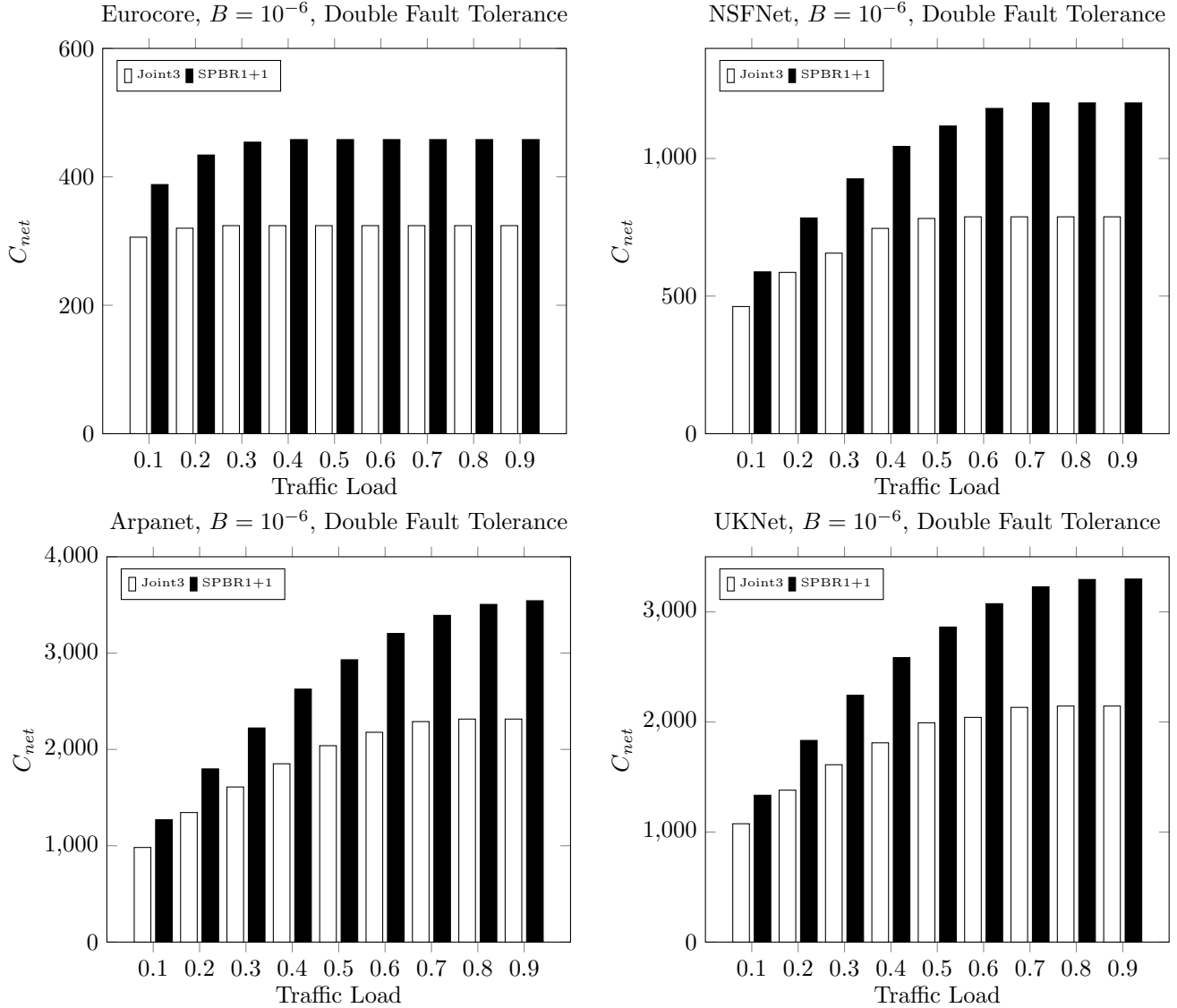


Figure 6: The total number of wavelengths  $C_{net}$  obtained with our method (Joint3) and SPBR1+1 on Eurocore, NSFNet, Arpanet and UKNet real mesh network topologies, for different connection traffic loads, in the double fault tolerance case.

number of different scenarios could be unmanageable, and any subset selection could be arbitrary.

As it can be seen in Figure 5, in the case of tolerance to a single failure, the Joint3 method  
455 performs clearly better. In fact, for all the scenarios evaluated in our experiments, the SPBR1+1

Parameters	$C_{net}$ no FT		$C_{net}$ Single FT				$C_{net}$ Double FT			
	Joint3	SPBR1+1	Joint3	$EX$	SPBR1+1	$EX$	Joint3	$EX$	SPBR1+1	$EX$
Eurocore	174	174	278	60%	398	128%	324	86%	454	160%
NSFNet	384	390	590	51%	656	108%	671	68%	926	137%
ArpaNet	912	922	1422	54%	1816	100%	1610	74%	2223	140%
UKNet	910	974	1482	52%	1878	92%	1611	65%	2242	130%

Table 1: Total number of wavelengths required by the Joint3 and SPBR1+1 methods: no fault tolerance ( $C_{net}$  no FT), single fault ( $C_{net}$  Single FT), and simultaneous double fault ( $C_{net}$  Double FT), for Eurocore, NSFNET, ARPANET, UKNet and Eurolarge networks, considering a maximum blocking probability of  $10^{-6}$  and a load of traffic 0.3. Additionally, the extra percentage of wavelength (with respect to the no failure case), needed to achieve single and double fault tolerance by each method, respectively, denoted as  $EX$  is presented.

method requires in general 30% more wavelengths (for  $\rho = 0.3$  which is a representative network load [1]) than the cost of the method proposed herein. In the case of tolerance to two simultaneous failures of links (Figure 6), the Joint3 method also significantly outperforms the SPBR1+1 technique. In this last case, the SPBR1+1 method requires in the order of 40% more wavelengths (always for  $\rho = 0.3$  [1]) than our proposal. Notice that each scenario presented herein achieves to connect the same users (connections) with the same QoS requirements (maximum acceptable blocking probability), but our proposal requires less resources than SPBR1+1 to do so.

To better exemplify such differences, in Table 1, the total cost  $C_{net}$  is shown for the following cases: without fault tolerance ( $C_{net}$  no FT); single link failure ( $C_{net}$  Simple FT); and simultaneous double link failure ( $C_{net}$  Double FT). The results shown were obtained for the same network topologies as in Figure 5, and considering a maximum blocking probability of  $10^{-6}$  for a traffic load of 0.3. Additionally, the single and double fault tolerance cases include the extra percentage of wavelength (with respect to the no failure case) needed to achieve single and double fault tolerance by each method, respectively, denoted as  $EX$ .

In Table 1 it is clear that our method performs much better than SPBR1+1. For example, to achieve the same users and QoS requirements, the Joint3 method requires only a 50% to 60% extra number of wavelengths than the no fault tolerance case, to provide single fault tolerance; meanwhile, the SPBR1+1 requires from 90% to 120% more wavelengths than the no fault tolerance scenario. On the other hand, in the double link failure case, our proposal requires between 60% to 90% extra wavelengths, in contrast to the SPBR1+1 method which requires almost 150% more wavelengths



than the no fault tolerance scenario requires. A notable consequence of this performance is that our method achieves double fault tolerance capacity with the same (or even less) resources than the ones required using the SPBR1+1 method to provide just single fault tolerance.

Let us discuss here some qualitative reasons that justify the better results of Joint3 over  
480 SPBR1+1:

- By simultaneously solving the 3 problems (primary and secondary routing, wavelengths dimensioning and fault tolerance), a global solution is obtained, which is a more efficient strategy than to solve these problems separately.
- Joint3 uses a wavelength dimensioning policy without distinction between primary and secondary routes. Therefore, it better exploits statistical multiplexing among all requests for  
485 connection/disconnection.
- Our technique never dimensions the cases that will never happened. For example, when dimensioning a determined failure case, it does not consider the load of the primary routes affected by the link failures. Although this fact may seem trivial, this is not considered by  
490 the 1+1 method.
- Joint3 uses a fixed point algorithm (do-until loop on Figures 2 and 3), which begins with an approximate initial solution and repeats an iterative cycle, generating a succession of improved solutions, until obtaining the best possible outcome (i.e., until it converges to a solution) within the criteria used.
- The usual blocking probability evaluation is based on simulation techniques, which is a slow  
495 procedure. Consequently, the evaluation process is limited to only a few restricted scenarios. So, to find the best possible solution, researchers apply some criteria to restrict the solutions search (heuristics). For example, using SPBR to select the primary routes, and similar criteria to find secondary routes. However, Joint3 was able to outperform these methodologies, so we  
500 can conclude that such heuristic solutions are not using completely accurate assumptions and decisions. We think that one of the key aspects why our proposal achieves better solutions is due to the fact that we can quickly explore a bigger solution space to find the best possible solution.

In addition to the increased efficiency compared to existing methods, the Joint3 method has  
505 other desirable attributes:

- a) It is scalable to any set of simultaneous failures, as long as the network remains connected after those multiple failures.
- b) The on-line operation of the method is very fast and simple. This is due to the fact that the re-routing information is stored in routing tables, which contain the secondary routes for each  
510 possible failure scenario. This means that the mechanism of fault tolerance can be executed quickly on demand.
- c) The off-line execution time of the Joint3 method takes only a few seconds using a standard current PC (Windows 10 64 bits, Intel Core i7 2.60GHz processor and 8GB RAM).

## 5. Conclusions

515 A novel method was proposed to jointly solve three main problems in optical network design. These concern routing, wavelength dimensioning and fault tolerance capacity on any set of link failures.

The methodology differs considerably from those published so far. Its main contributions are the following: a) Our approach jointly solves three problems at the same time: it determines the  
520 set of primary and secondary routes, it evaluates the minimum number of wavelengths necessary on each network link so that the blocking probability of each user is lower than a certain pre-specified threshold (which is a design parameter of the network), and it solves the fault tolerance problem on any set of possible link failures. b) By simultaneously attacking the three problems, a global solution can be obtained, which is often more efficient than that obtained when solving them separately.  
525 The solution is more efficient than current methods in terms of necessary number of wavelengths. c) Our technique dimensions the number of wavelengths without making any distinction between primary and secondary routes. In this way, it better exploits statistical multiplexing among all connection requests. d) The method is scalable to any set of simultaneous link failures, as long as the network topology allows reconnection via the links that remain operating. e) The main work of  
530 the procedure is executed before the network is operating (off-line) and the on-line operation of our procedure is simple and fast, since the routes (both primary and secondary) are stored in routing

tables that are consulted only under demand. f) It requires a few seconds to fulfill the routing and the wavelength dimensioning subtasks, so our approach can be adapted to traffic load variations during network operation.

## 535 6. Acknowledgments

This work received financial support from FONDEF ID14I10129, Stic/Amsud-Conicyt project “DAT”, USM PIIC Project, CONICYT PhD and CONICYT PhD internship scholarships. These projects and institutions are then gratefully acknowledged.

- 540 [1] A. A. M. Saleh, J. M. Simmons, Technology and architecture to enable the explosive growth of the Internet, *Communications Magazine, IEEE* 49 (1) (2011) 126–132.
- [2] R. Ramaswami, K. Sivaraman, G. Sasaki, *Optical Networks: A Practical Perspective*, 3rd Edition, 3rd Edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009.
- [3] X. Chu, J. Liu, Sparse-partial wavelength conversion: Converter placement and wavelength assignment, *Optical Switching and Networking* 7 (2) (2010) 66 – 74.
- 545 [4] C. Qiao, M. Yoo, Optical burst switching (OBS)—a new paradigm for an Optical Internet, *Journal of high speed networks* 8 (1) (1999) 69–84.
- [5] A. Ge, F. Callegati, L. S. Tamil, On optical burst switching and self-similar traffic, *Communications Letters, IEEE* 4 (3) (2000) 98–100.
- [6] I. de Miguel, M. Düser, P. Bayvel, Traffic load bounds for optical burst-switched networks with dynamic wavelength allocation, in: *ONDM, 2001*, pp. 209–226.
- 550 [7] X. Yu, Y. Chen, C. Qiao, Study of traffic statistics of assembled burst traffic in optical burst-switched networks, in: *ITCom 2002: The Convergence of Information Technologies and Communications*, International Society for Optics and Photonics, 2002, pp. 149–159.
- [8] G. Hu, K. Dolzer, C. Gauger, Does burst assembly really reduce the self-similarity?, in: *Optical Fiber Communications Conference, 2003. OFC 2003, 2003*, pp. 124–126 vol.1.
- 555 [9] M. Zukerman, E. Wong, Z. Rosberg, G. M. Lee, H. L. Vu, On teletraffic applications to OBS, *Communications Letters, IEEE* 8 (2) (2004) 116–118.

- [10] A. Zalesky, E. Wong, M. Zukerman, H. L. Vu, R. Tucker, Performance analysis of an OBS edge router, *Photonics Technology Letters, IEEE* 16 (2) (2004) 695–697.
- 560 [11] N. A. Jara, A. Beghelli, Blocking probability evaluation of end-to-end dynamic WDM networks, *Photonic Network Communications* 24 (1) (2012) 29–38.
- [12] E. Donato, J. Celestino Júnior, A. Vieira, A. Patel, A proposal of dynamic RWA using ant colony in optical burst switched networks, in: *ICN 2012, The Eleventh International Conference on Networks, 2012*, pp. 246–252.
- 565 [13] Z. Xu, J. Huang, Z. Zhou, Z. Ding, T. Ma, J. Wang, A novel grooming algorithm with the adaptive weight and load balancing for dynamic holding-time-aware traffic in optical networks, *Optical Fiber Technology* 19 (5) (2013) 392 – 399.
- [14] R. Vallejos, N. Jara, Join routing and dimensioning heuristic for dynamic WDM optical mesh networks with wavelength conversion, *Optical Fiber Technology* 20 (3) (2014) 217 – 223.
- 570 [15] G. Kavitha, T. Indumathi, Situational analysis of significant research contribution in Optical Network, *International Journal of Computer Applications* 112 (15).
- [16] M. To, P. Neusy, Unavailability analysis of long-haul networks, *Selected Areas in Communications, IEEE Journal on* 12 (1) (1994) 100–109.
- 575 [17] D. A. Schupke, A. Autenrieth, T. Fischer, Survivability of multiple fiber duct failures, in: *Third International Workshop on the Design of Reliable Communication Networks (DRCN, 2001*, pp. 7–10.
- [18] R. Ramaswami, K. Sivarajan, Design of logical topologies for wavelength-routed all-optical networks, in: *INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE, 1995*, pp. 1316–1325 vol.3.
- 580 [19] F. Dikbiyik, A. S. Reaz, M. De Leenheer, B. Mukherjee, Minimizing the disaster risk in optical telecom networks, in: *Optical Fiber Communication Conference, Optical Society of America, 2012*, pp. OTh4B–2.

- [20] H. Zang, C. Ou, B. Mukherjee, Path-protection routing and wavelength assignment in wdm mesh networks under shared-risk-group constraints, in: Asia-Pacific Optical and Wireless Communications Conference and Exhibit, International Society for Optics and Photonics, 2001, pp. 49–60.
- [21] G. R. Ash, Dynamic Routing in Telecommunications Networks, 1st Edition, McGraw-Hill Professional, 1997.
- [22] S. Keshav, An engineering approach to computer networking: ATM networks, the Internet, and the telephone network, Reading MA 11997.
- [23] W. Van Parys, B. Van Caenegem, P. Demeester, Reduction of blocking in arbitrary meshed WDM networks through a biased routing approach, in: Optical Fiber Communication Conference and Exhibit, 1998. OFC'98., Technical Digest, IEEE, 1998, p. 94.
- [24] J. Teng, G. Rouskas, Traffic engineering approach to path selection in optical burst switching networks, J. Opt. Netw. 4 (11) (2005) 759–777.
- [25] M. Brunato, R. Battiti, E. Salvadori, Dynamic load balancing in WDM networks (2003).
- [26] M. Klinkowski, J. Pedro, D. Careglio, M. Pióro, J. Pires, P. Monteiro, J. Solé-Pareta, An overview of routing methods in optical burst switching networks, Optical Switching and Networking 7 (2) (2010) 41–53.
- [27] C. Yahaya, A. Latiff, A. Mohamed, A review of routing strategies for optical burst switched networks, International Journal of Communication Systems.
- [28] R. Yadav, R. Yadav, H. Singh, Intercycle switching (ICS)-based dynamic reconfiguration of  $p$ -cycle for dual-failure survivability of WDM networks, Photonic Network Communications 24 (2) (2012) 160–165.
- [29] Dewiani, K. Hirata, Y. Higami, S.-Y. Kobayashi, Dynamic routing and wavelength assignment in multifiber WDM networks with sparse wavelength conversion, in: ICT Convergence (ICTC), 2012 International Conference on, 2012, pp. 567–572.
- [30] L. Tang, M. Cai, B. Li, R. Wu, A novel multi-link fault-tolerant algorithm for survivability in multi-domain optical networks, Photonic Network Communications 24 (2) (2012) 77–85.

- [31] L. Belgacem, I. Charon, O. Hudry, A post-optimization method for the routing and wavelength assignment problem applied to scheduled lightpath demands, *European Journal of Operational Research*.
- [32] A. Ebrahimzadeh, A. G. Rahbar, B. Alizadeh, Binary quadratic programming formulation for routing and wavelength assignment problem in all-optical WDM networks, *Optical Switching and Networking* 10 (4) (2013) 354–365.
- [33] C.-C. Hsu, H.-J. Cho, S.-C. Fang, Routing and wavelength assignment in optical networks from maximum edge-disjoint paths, in: *Genetic and Evolutionary Computing*, Vol. 238 of *Advances in Intelligent Systems and Computing*, Springer International Publishing, 2014, pp. 95–103.
- [34] G. Mohan, C. Murthy, Routing and wavelength assignment for establishing dependable connections in WDM networks, in: *Fault-Tolerant Computing, 1999. Digest of Papers. Twenty-Ninth Annual International Symposium on*, 1999, pp. 94–101.
- [35] G. Mohan, C. Murthy, Lightpath restoration in WDM optical networks, *Network, IEEE* 14 (6) (2000) 24–32.
- [36] S. Ramamurthy, L. Sahasrabudde, B. Mukherjee, Survivable WDM mesh networks, *Lightwave Technology, Journal of* 21 (4) (2003) 870–883.
- [37] P. C. Srikanth, M. V. Sathyanarayana, D. Balaram, T. Srinivas, Fault tolerance in survivable WDM optical networks using hybrid scheme: An analytical approach, in: *TENCON 2009 - 2009 IEEE Region 10 Conference*, 2009, pp. 1–6.
- [38] H. Singh, J. Prakash, D. Arora, A. Wason, Fault tolerant congestion based algorithms in OBS network, *International Journal of Engineering (IJE)* 5 (5).
- [39] F. S. H. Souza, D. L. Guidoni, G. R. Mateus, A column generation-based heuristic for the GRWA with protection and QoS in WDM optical networks, in: *Computers and Communications (ISCC), 2013 IEEE Symposium on*, 2013, pp. 000922–000927.
- [40] D. Schupke, W. Grover, M. Clouqueur, Strategies for enhanced dual failure restorability with static or reconfigurable  $p$ -cycle networks, in: *Communications, 2004 IEEE International Conference on*, Vol. 3, 2004, pp. 1628–1633.

- [41] M. Clouqueur, W. Grover, Mesh-restorable networks with enhanced dual-failure restorability properties, *Photonic Network Communications* 9 (1) (2005) 7–18.
- 640 [42] D. Mukherjee, C. Assi, A. Agarwal, Alternate strategies for dual failure restoration using  $p$ -cycles, in: *Communications, 2006. ICC'06. IEEE International Conference on*, Vol. 6, 2006, pp. 2477–2482.
- [43] Y. Kavian, R. Rejeb, O. Strobel, Dual-link failure covering in DWDM optical networks using genetic algorithms, in: *Transparent Optical Networks (ICTON), 2010 12th International*  
645 *Conference on*, 2010, pp. 1–4.
- [44] A. Wason, R. Kaler, Fault-tolerant routing and wavelength assignment algorithm for multiple link failures in wavelength-routed all-optical WDM networks, *Optik - International Journal for Light and Electron Optics* 122 (2) (2011) 110 – 113.
- [45] D. S. Yadav, S. Rana, S. Prakash, A mixed connection recovery strategy for surviving dual  
650 link failure in WDM networks, *Optical Fiber Technology* 19 (2) (2013) 154 – 161.
- [46] S. Ferdousi, F. Dikhiyik, M. F. Habib, B. Mukherjee, Disaster-aware data-center and content placement in cloud networks, in: *2013 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), IEEE, 2013*, pp. 1–3.
- [47] S. Chalasani, V. Rajaravivarma, Survivability in optical networks, in: *System Theory, 2003. Proceedings of the 35th Southeastern Symposium on*, 2003, pp. 6–10.  
655
- [48] M. Wang, S. Li, E. W. M. Wong, M. Zukerman, Performance analysis of circuit switched multi-service multi-rate networks with alternative routing, *Journal of Lightwave Technology* 32 (2) (2014) 179–200. doi:10.1109/JLT.2013.2289925.
- [49] D. Schupke, R. Prinz, Capacity efficiency and restorability of path protection and rerouting  
660 in WDM networks subject to dual failures, *Photonic Network Communications* 8 (2) (2004) 191–207.
- [50] J. Zhang, B. Mukherjee, A review of fault management in WDM mesh networks: basic concepts and research challenges, *Network, IEEE* 18 (2) (2004) 41–48.

- 665 [51] D. A. Schupke, C. G. Gruber, A. Autenrieth, Optimal configuration of  $p$ -cycles in WDM networks, in: In IEEE International Conference on Communications, 2002, pp. 2761–2765.
- [52] D. Schupke, Multiple failure survivability in WDM networks with  $p$ -cycles, in: Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on, Vol. 3, 2003, pp. III-866–III-869 vol.3.
- 670 [53] A. Zapata-Beghelli, P. Bayvel, Dynamic versus static wavelength-routed optical networks, Lightwave Technology, Journal of 26 (20) (2008) 3403–3415.
- [54] R. Vallejos, A. Zapata, M. Aravena, Fast and effective dimensioning algorithm for end-to-end optical burst switching networks with ON-OFF traffic model, in: Optical Network Design and Modeling, Springer, 2007, pp. 378–387.
- 675 [55] R. Vallejos, J. Olavarría, N. Jara, Blocking evaluation and analysis of dynamic WDM networks under heterogeneous on/off traffic, Optical Switching and Networking 20 (2016) 35 – 45.
- [56] R. Vallejos, A. Zapata, V. Albornoz, Optimal Routing for Minimum Wavelength Requirements in End-to-End Optical Burst Switching Rings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 448–457.
- 680 [57] R. Vallejos, A. Zapata-Beghelli, V. Albornoz, M. Tarifeño, Joint routing and dimensioning of optical burst switching networks, Photonic Network Communications 17 (3) (2009) 266–276.