

Reoptimization of Dynamic Flexgrid Optical Networks After Link Failure Repairs

M. Żotkiewicz, M. Ruiz, M. Klinkowski, M. Pióro, and L. Velasco

Abstract—In dynamic flexgrid optical networks, the usage of capacity may not be optimal due to the permanent process of setting up and tearing down connections, which, if not controlled, leads to spectrum fragmentation and, as a result, to increase of connection blocking. On top of this, a restoration mechanism that is launched in reaction to a link failure (cable cut) restores the affected lightpaths. Eventually, when the cable is repaired and its capacity becomes available for new connections, the unbalance between lightly and heavily loaded links increases, thus further decreasing the probability of finding optical paths with continuous and contiguous spectrum for future connection requests. In this paper we study the effects of re-optimizing the lightpath connections after a link failure has been repaired (namely, the AFRO problem) as an effective way for both reducing and balancing capacity usage and, by these means, for improving network performance. To solve AFRO a column generation decomposition method is presented. Illustrative numerical results show that AFRO allows to significantly decrease the request blocking probability in realistic dynamic network scenarios. Moreover, the proposed column generation algorithm delivers quasi-optimal solutions in reasonable times. Besides, traffic disruptions resulting from lightpath rerouting are practically negligible. Finally, we show that it is sufficient to apply AFRO only for a selected set of link failures in order to achieve high network performance.

Index Terms—integer programming; column generation; routing and spectrum assignment; flexgrid optical networks, network survivability.

I. INTRODUCTION

The progress in flexgrid optical technology [1] achieved in the last few years makes it possible to fulfil operators' and customers' expectations in terms of improved handling in both short-term and long-term traffic. Such features as fine spectrum granularity (a result of dividing the spectrum into narrow frequency *slices*, e.g., of the 6.25 GHz width [2]), advanced optical modulation formats, and filtering and signal processing, constitute the basis for introducing flexibility and elasticity to network operation [3]. In the context of dynamic networks, the Application-Based Network Operations (ABNO) architecture [4] based on a centralized active stateful path

computation element (PCE) [5] can be used to control dynamic operations such as: solving the routing and spectrum assignment (RSA) problem for incoming connection requests [6], restoring lightpaths to deal with network failures [7], and performing in-operation network planning [8], e.g. spectrum defragmentation [3], [9].

In case of a link (cable) failure, the use of bulk path computation was recently proposed to increase the amount of restored lightpaths without dramatically increasing restoration times, which are kept in the order of hundreds of milliseconds [7]. Note that between the restoration instant and the failure repair instant, the network operates with reduced available capacity, thus increasing the blocking probability of new connection requests. Once the link is repaired, the overall available capacity of the network increases. However, the difference in terms of free available capacity between the highest and the lowest loaded links becomes a problem when solving RSA for new connection requests. This is caused by the difficulty of finding routes with a sufficient number of continuous and contiguous frequency slices and, therefore, the connection blocking probability remains high after the failure repair.

In order to address the above discussed problem, in this paper we focus on a dynamic network scenario in which lightpath connections are reoptimized during network operation. The reoptimization is performed after a link has been repaired from a failure - which we refer to as *after-failure-repair optimization* (AFRO) [10] - and it concerns rerouting of some existing lightpaths with the objective to improve network performance. Note that the AFRO problem was recently used to experimentally validate network reoptimization using the ABNO architecture to control a flexgrid optical network [11]. The effectiveness of AFRO relies on the fact that the lightpaths are rerouted from highly loaded links to the recently restored link what results in a more balanced network load and, by these means, increases the possibility to accept further connection requests. It is worth mentioning that in the literature there have been proposed some network reoptimization techniques based on lightpath reallocation with the aim to improve the performance of a dynamic flexgrid optical network (e.g., see [3], [9]). Also, there can be found several works that focus on reoptimization of network resources after network-state updates in conventional fixgrid wavelength division multiplexing (WDM) networks, e.g., see [12], [13]. However, to the best of our knowledge, there are no prior works that address the optimization of a flexgrid optical network after failure repairs.

It is clear that even without the application of AFRO the imbalance in capacity between the repaired link and other net-

Manuscript submitted on April 11, 2014. A part of this work was presented at the OFC 2014 conference [10].

Mateusz Żotkiewicz (mzotkiew@tele.pw.edu.pl) and Michał Pióro (mpp@tele.pw.edu.pl) are with the Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland. Michał Pióro is also with the Lund University, P.O. Box 118, SE-22100 Lund, Sweden.

Marc Ruiz (mruiz@ac.upc.edu) and Luis Velasco (lvelasco@ac.upc.edu) are with the Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, 08034 Barcelona, Spain.

Miroslaw Klinkowski (M.Klinkowski@itl.waw.pl) is with National Institute of Telecommunications, Szachowa 1, 04-894 Warsaw, Poland.

work links will be mitigated after some time as old lightpaths are torn down and new are established. Therefore, in very dynamic scenarios in which the connection holding time is much smaller than the mean time between two consecutive link failures, the gains from applying AFRO, in terms of the improved network performance, seem to lose its importance. Moreover, in such scenarios, the need to suspend the connection setup/release operations during network reoptimization and possible traffic disruptions during lightpath rerouting may deteriorate network performance. However, if the connection holding time increases, the benefits from applying the reoptimization mechanism increases as well.

Thus, when the connections remain for days, weeks, or even months (which is more than the usual duration in certain real world scenario), the application of AFRO allows a rapid improvement in the balancing of network capacity. The reoptimization not only increases the possibility to admit following connection requests, but it also improves the effectiveness of the lightpath restoration algorithm used after link failures. In order to illustrate it, let us consider a national network with a total of about 10,000 km of fiber and a link failure rate of $2.72 \cdot 10^{-3}$ per km per year [14], which corresponds to about 30 failures in a year. If we assume this number and that the connections remain active for weeks or months, the probability that a connection is affected by more than one failure (and therefore, by more than one restoration) is high. The accumulation of several restorations without the ability to reoptimize the network can decrease the effectiveness of the restoration mechanisms. Therefore, AFRO can be considered as a problem of network reoptimization that aims at improving load distribution so that to: a) reduce the blocking of incoming connection requests, and b) improve the efficiency of the restoration algorithm. The main objective of this paper is to assess the performance of AFRO, through a comprehensive numerical analysis, with the aim to determine the conditions under which AFRO offers its best performance.

In this paper, we continue and considerably extend the study on AFRO that we initiated in our conference paper [10]. The particular contribution of this work concerns modeling and formulation of the AFRO optimization problem as an integer programming (IP) problem, a proposal and numerical evaluation of different alternative options for the AFRO optimization algorithm and, above all, performing simulation experiments of AFRO in a dynamic network scenario. Our optimization algorithm is based on a column generation technique that allows to reduce the complexity of IP formulations by considering a reduced set of problem variables without significantly affecting the quality of the obtained solution [15]. In our recent works [16] and [17], we have developed a column generation algorithm for a generic RSA problem and, here, we adapt it for the AFRO problem.

The rest of the paper is organized as follows. In Section II, we describe the AFRO problem, which we then formally model in Section III. Then, in Section IV we present an efficient optimization algorithm for solving AFRO. The algorithm is evaluated in Section V by means of numerical and simulation experiments. Eventually, in Section VI we conclude the work.

II. AFTER-FAILURE-REPAIR OPTIMIZATION PROBLEM

A. Problem description

In order to describe the problem addressed in this paper, in Fig. 1 we show a process a dynamic network undergoes after being subject to a certain link failure. Moreover, in Fig. 2, which complements Fig. 1, we show an illustrative example of the network states at different time instants. Here, we consider a dynamic network that serves a set of lightpaths under normal operation (denoted as state t_0). After an event of a link failure, the restoration mechanism finds an alternative route for each of the lightpaths affected by the fiber cut (state t_1). Specifically, Fig. 2a) shows a network with three lightpaths, namely p_1 , p_2 , and p_3 , just before the event of a fiber cut in link 2 – 7. After that, lightpaths p_1 and p_2 are rerouted by the restoration algorithm (Fig. 2b)).

Once the solution of the restoration algorithm is implemented, the network operates without the failed link until it is restored (6–12 hours are usually needed to repair a fiber link), which is denoted as state t_2 . For the ease of presentation, in Fig. 2 we assume that $t_1 = t_2$, i.e., neither setups nor tear-downs of lightpaths occur during the period of link restoration. At this point, we can see that after the link is repaired, the network capacity increases, however, its performance is not necessarily improved. Namely, in Fig. 2b) a new connection requesting 4 slices between nodes 1 and 6 can be established neither when the link is broken nor when the capacity of the repaired link is available again.

With the aim to improve network performance and make use of restored link resources we consider that lightpath connections are reoptimized after the link is repaired (i.e., in state t_2). To this end, we solve an after-failure-repair optimization (AFRO) problem, which at the input gets the network state information, in particular, concerning routing and spectrum allocation of lightpaths established in the network, and at the output returns a list of rerouted lightpaths with their new routes and spectrum allocation. Since AFRO tries to make use of the recently restored link capacity, new routes are forced to use the repaired link. The lightpaths that cannot be routed over the restored link are kept without changes and are not considered for reoptimization. This constraint allows to reduce significantly the amount of lightpath reallocations in the network. After solving AFRO and implementing its solution (state t_3), the network is ready to operate and its performance is expected to be improved. As can be seen in Fig. 2c), connection p_4 between nodes 1 and 6 can be established now.

Note that during the AFRO processing time the process of establishing new connection requests is stopped, similarly to other dynamic network reoptimization operations such as spectrum defragmentation. For this reason, the time needed for solving AFRO and implementing network changes should be short in order to reduce the delay of that process. Indeed, to coordinate with other operations and to reduce even more the impact over new incoming traffic, AFRO might be scheduled in a time slot with less dynamic activity (e.g. night hours) and, therefore, not applied immediately after the failure repair.



Fig. 1: Dynamic evolution of network states (t) with link failure and link repair events.

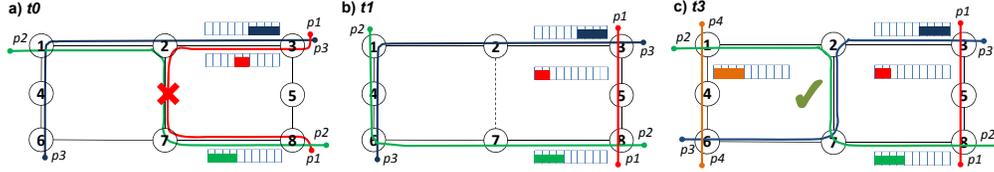


Fig. 2: An example of link occupancy in different network states.

B. Problem statement

The AFRO problem can be formally stated as follows:

Given:

- 1) flexgrid optical network represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{D})$ where \mathcal{V} is the set of optical nodes, \mathcal{E} is the set of fiber links, and \mathcal{D} is the set of (traffic) demands,
- 2) optical spectrum (given at each link) which is divided into a set $\mathcal{S} = \{s_1, s_2, \dots, s_S\}$ of frequency slices of a given width,
- 3) set of lightpaths currently established in the network,
- 4) link $e^r \in \mathcal{E}$, representing the link recently repaired that was previously unavailable.

Find: a subset of demands from set \mathcal{D} which will be rerouted and, for each of these demands, a new lightpath over the network such that it uses the repaired link e^r , aiming at utilizing recently restored link capacity, with the *objective* to optimize the usage of spectrum in the network.

III. MODELING AND FORMULATION OF AFRO

In this section, we focus on IP modeling of the AFRO problem in a flexgrid optical network. As already stated in Section II-B, the AFRO problem consists in deciding, for each demand $d \in \mathcal{D}$, if it should be rerouted to a new lightpath, and if so, in determining a route over the network that uses the recently repaired link e^r . Moreover, spectrum allocation should be optimized. A goal of such reoptimization is to improve the allocation of slices on network links in order to provide the network with free spectrum resources for future connection requests. Afterwards, in Section IV, we use the model to develop an efficient optimization algorithm for solving AFRO.

We assume that regeneration requirements are met for all potential paths. Moreover, we assume that only one link can fail and be repaired at a time. However, the proposed method is not restricted to solely single link failure scenarios, and can be successively used to handle multiple failures, i.e., when e^r is not just a single link but a set of recently repaired links. To handle multiple failures the algorithm of Section IV-B3 has to

be appropriately modify, i.e., Algorithm 2 has to be executed for all affected demands independently.

A. Notation

For the problem considered in the paper, links and demands are directed. Let \mathcal{V}^2 denote the set of all two-element subsets of the set of nodes \mathcal{V} . Each link $e \in \mathcal{E}$ is represented by its end nodes $s(e)$ and $t(e)$ with $\{s(e), t(e)\} \in \mathcal{V}^2$.

The slices of a link are used by connections in the form of frequency slots (see [2]). Slot c of capacity n is a set of contiguous (i.e., consecutive) slices of the form $c = \{s_i, s_{i+1}, \dots, s_{i+n-1}\}$ for some i between 1 and $S - (n - 1)$. In the sequel, \mathcal{C} will denote the set of all possible slots and $n(c), c \in \mathcal{C}$, the number of slices in slot c . Obviously, the number of all possible slots (for all $n = 1, 2, \dots, S$) is equal to $|\mathcal{C}| = \frac{S(S+1)}{2}$.

A lightpath (denoted by p) is a route in \mathcal{G} that uses the same slot c on all its links. The set of links composing the route of lightpath p is denoted by $\mathcal{E}(p) \subseteq \mathcal{E}$, and the slot assigned to this lightpath by $c(p) \in \mathcal{C}$.

Each demand $d \in \mathcal{D}$ is represented by its end nodes $s(d)$ and $t(d)$ with $\{s(d), t(d)\} \in \mathcal{V}^2$, and is characterized by demand bit-rate $h(d)$ and the currently used lightpath $p(d)$ with the route linking nodes $s(d)$ and $t(d)$. We use $n(d)$ to denote the bandwidth (i.e., a portion of the spectrum, in terms of the number of slices) able to serve bit-rate $h(d)$. Besides, each demand d is assigned a set of allowable lightpaths denoted by $\mathcal{P}(d)$. The set $\mathcal{P}(d)$ consists of the current lightpath $p(d)$ and new lightpaths. The new lightpaths have to fulfill a basic condition that each of them has to use the repaired link e^r , i.e., $e^r \in \mathcal{E}(p)$ for each $d \in \mathcal{D}, p \in \mathcal{P}(d) \setminus \{p(d)\}$.

Note that the set $\mathcal{C}(d) \subseteq \mathcal{C}$ of slots that can be used for demand $d \in \mathcal{D}$ is defined as follows: $c \in \mathcal{C}(d)$ if and only if $n(d) \leq n(c)$, which means that lightpaths allowable for demand $d \in \mathcal{D}$ can use only those slots whose capacity $b(n(c))$ is sufficient to carry the demand volume $h(d)$, i.e., $n(d) \leq n(c(p))$ for each $d \in \mathcal{D}, p \in \mathcal{P}(d)$.

TABLE I: Notation.

Objects and parameters	
\mathcal{V}	set of nodes, $V = \mathcal{V} $
\mathcal{E}	set of links, $E = \mathcal{E} $
$s(e)$	source node of link e
$t(e)$	termination node of link e
e^r	recently repaired link, $e^r \in \mathcal{E}$
\mathcal{S}	set of spectrum slices, $S = \mathcal{S} $
\mathcal{C}	set of frequency slots, $C = \mathcal{C} = \frac{S(S+1)}{2}$
$\mathcal{S}(c)$	set of (contiguous) slices composing slot $c \in \mathcal{C}$
$n(c)$	number of slices used by slot $c \in \mathcal{C}$
$b(n)$	bit-rate carried in a slot consisting of n slices
\mathcal{D}	set of demands, $D = \mathcal{D} $
$s(d)$	source node of demand d
$t(d)$	termination node of demand d
$h(d)$	bit-rate of demand $d \in \mathcal{D}$
$n(d)$	number of slices required to carry $h(d)$, $d \in \mathcal{D}$
$p(d)$	current lightpath used by demand $d \in \mathcal{D}$
$\mathcal{C}(d) \subseteq \mathcal{C}$	set of slots allowable for demand $d \in \mathcal{D}$, $c \in \mathcal{C}(d) \Leftrightarrow n(d) \leq n(c)$
$\mathcal{P}(d)$	set of lightpaths (new and current) allowable for demand $d \in \mathcal{D}$
$\mathcal{P} = \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$	set of all allowable lightpaths
$\mathcal{Q}(d, e, s) \subseteq \mathcal{P}(d)$	set of lightpaths for $d \in \mathcal{D}$ using slice $s \in \mathcal{S}$ on link $e \in \mathcal{E}$
$\mathcal{Q}(d, e) \subseteq \mathcal{P}(d)$	set of lightpaths for $d \in \mathcal{D}$ using link $e \in \mathcal{E}$
$\mathcal{E}(p)$	set of links traversed by lightpath $p \in \mathcal{P}$
$d(p)$	demand $d \in \mathcal{D}$ realized by lightpath $p \in \mathcal{P}$
$c(p)$	slot occupied by lightpath $p \in \mathcal{P}$, $c(p) \in \mathcal{C}(d)$
$n(p) = n(c(p))$	number of slices occupied by lightpath $p \in \mathcal{P}$
$L(p)$	cost of lightpath p , $L(p) = \mathcal{E}(p) \cdot n(p)$
$H(p)$	highest index of any slice used by lightpath p , $1 \leq H(p) \leq S$
$\mathcal{S}(p) = \mathcal{S}(c(p))$	set of slices used by lightpath $p \in \mathcal{P}$
κ_{es}	cost of using slice s in link e
\mathcal{K}	set of different sizes of demands in terms of a number of requested slices
$n(k)$	number of requested slices for demand size $k \in \mathcal{K}$
Problem variables	
x_{dp} , $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$	binary variables, $x_{dp} = 1$ when lightpath p carries its demand d
z_e , $e \in \mathcal{E}$	integer variables, maximum index of any used slice in link e

B. IP formulations of the AFRO problem

We present two alternative IP formulations of AFRO. Although both formulations aim at improving the usage of spectrum in the network, still they implement different objective functions. The first formulation, referred to as MIN-SUM, aims at minimizing the total number of slices occupied on all network links. The second formulation, referred to as MIN-MAX, minimizes a sum of maximum identification numbers of used slices over all links. In Section V, we evaluate the effectiveness of both optimization objectives in improving the network performance.

The list of objects, parameters, and problem variables required for the problem formulation is given in Table I.

1) *MIN-SUM formulation*: Our first IP formulation of the problem can be written down in the following way:

$$\min \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d)} L(p) \cdot x_{dp} \quad (1a)$$

$$[\alpha_d] \quad \sum_{p \in \mathcal{P}(d)} x_{dp} = 1 \quad d \in \mathcal{D} \quad (1b)$$

$$[\beta_{es}] \quad \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{Q}(d,e,s)} x_{dp} \leq 1 \quad e \in \mathcal{E}, s \in \mathcal{S} \quad (1c)$$

$$x_{dp} \in \{0, 1\} \quad d \in \mathcal{D}, p \in \mathcal{P}(d), \quad (1d)$$

where variables in brackets, i.e. α_d and β_{es} , are dual variables.

Objective function (1a) minimizes the sum of all slices used on all links. Constraint (1b) assigns a lightpath (either new or the current one) to a demand. Here, we assume that both the current paths and new possible paths for rerouting are in appropriate sets $\mathcal{P}(d)$. Constraint (1c) makes sure that occupancy of the slices is not violated. Finally, constraint (1d) assures that the variables are binary.

Observe that formulation (1) is based on path-flows and therefore it assumes a given set of allowable lightpaths \mathcal{P} . Clearly, the formulation is non-compact as it requires an exponential number of variables x_{dp} , $d \in \mathcal{D}$, $p \in \mathcal{P}(d)$, in order to consider all possible lightpaths for rerouting [15].

2) *MIN-MAX formulation*: Our second IP formulation of AFRO that we study in this paper is the following:

$$\min \sum_{e \in \mathcal{E}} z_e \quad (2a)$$

$$[\alpha_d] \quad \sum_{p \in \mathcal{P}(d)} x_{dp} = 1 \quad d \in \mathcal{D} \quad (2b)$$

$$[\beta_{es}] \quad \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{Q}(d,e,s)} x_{dp} \leq 1 \quad e \in \mathcal{E}, s \in \mathcal{S} \quad (2c)$$

$$[\gamma_{edp}] \quad z_e \geq H(p) \cdot x_{dp} \quad e \in \mathcal{E}, d \in \mathcal{D}, p \in \mathcal{Q}(d,e) \quad (2d)$$

$$x_{dp} \in \{0, 1\} \quad d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (2e)$$

where variables in brackets are dual variables.

The objective function (2a) minimizes the sum of maximum identification numbers of used slices over all links. By minimizing the sum of z_e we assure that there are $S - z_e$ contiguous slices, with indices between $z_e + 1$ and S , available in link e and these slices can be assigned to future connection requests. Constraint (2b) assigns a lightpath to a demand. Constraint (2c) makes sure that occupancy of the slices is not violated. Constraint (2d) finds the maximum index of any used slice in link e . Finally, constraint (2e) assures that the variables are binary.

Formulation (2) describes our problem exactly. However, we formulate it also in an alternative way that allows for a simpler column generation, as it does not contain dual variables indexed by both edges and demands. The alternative formulation is as follows:

$$\min \sum_{e \in \mathcal{E}} z_e \quad (3a)$$

$$[\alpha_d] \quad \sum_{p \in \mathcal{P}(d)} x_{dp} = 1 \quad d \in \mathcal{D} \quad (3b)$$

$$[\beta_{es}] \quad \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{Q}(d,e,s)} x_{dp} \leq 1 \quad e \in \mathcal{E}, s \in \mathcal{S} \quad (3c)$$

$$[\gamma_{es}] \quad z_e \geq \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{Q}(d,e,s)} H(p) \cdot x_{dp} \quad e \in \mathcal{E}, s \in \mathcal{S} \quad (3d)$$

$$x_{dp} \in \{0, 1\} \quad d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (3e)$$

where again, variables in brackets are dual variables, and the purpose of each constraint is as in (2). Notice that now constraint equivalent to (2d), i.e., (3d) is indexed only by $e \in \mathcal{E}$ and $s \in \mathcal{S}$.

IV. OPTIMIZATION ALGORITHM

In this section, we develop an efficient algorithm for solving AFRO. In Fig. 3 we present its main processing steps. Starting from the set of lightpaths established in t_2 , some of them are selected for reoptimization. In particular, at this step we decide for which demands it is potentially worth to reroute their lightpaths. The pre-selection of these demands is described in Section IV-A. Next, for the selected subset of demands, the AFRO problem is solved using an algorithm presented in Section IV-B and a reoptimized set of lightpaths is obtained for the selected demands. The generation of new lightpaths is traffic disruption-aware and, to this end, a procedure described in Section IV-C is applied. Finally, in a post-processing step, the reoptimized lightpaths are arranged appropriately in an order in which the lightpath rerouting will be implemented in the network. The aim of this ordering is to minimize traffic disruptions that might occur during the rerouting. The lightpath ordering procedure is described in Section IV-D.

A. Selecting demands to be processed

The first issue, which we have to face while solving the described problem, is to define a set of demands to be taken into account while reoptimizing a network. Here the two extremities exist. On the one hand, we can consider all the demands served in a network. This approach assures us that

we do not miss an optimal solution. However, it keeps a size of the problem large, thus making it difficult to solve. On the other hand, we can improve the tractability by limiting a set of considered demands, and sacrificing some parts of the feasibility region. This leads us to the second extremity which is selecting only one demand in the beginning, and solving a rather simple problem of reoptimizing a single lightpath.

In our research we have developed two procedures that both place themselves between the two extremities described above. The procedures are independent and can be used interchangeably. The procedures were designed for the MIN-SUM formulation only, as according to numerical results of Section V, this formulation proved to be more efficient in practice than the MIN-MAX formulation.

The first approach is to consider all demands that can theoretically improve the objective function. This way we are certain that all demands excluded from the set of considered demands can by no mean be used to improve the objective function. On the other hand the procedure has to be simple enough from the complexity theory viewpoint not to put a significant computational burden on the whole optimization process. We refer to the first method as *complete* in the rest of the paper. On the contrary, to the second procedure we refer to as *reduced*, because it limits the feasibility region more severely. However, in practice the second approach works quite well as shown in Section V. Let us now present both procedures in details.

1) *Complete procedure*: Recall that a number of slices used by demand d is denoted by $n(d)$, while its current lightpath is denoted by $p(d)$. By $\hat{p}(d)$ we denote the shortest path from $s(d)$ to $t(d)$ that uses e^r . Then, by $C(p)$ we denote a cost of a lightpath p understood as a sum of slices it occupies on all links, e.g., $C(p(d)) = n(d) \cdot |p(d)|$, while $C(\hat{p}(d)) = n(d) \cdot |\hat{p}(d)|$. For a given network state we can easily compute costs of all current lightpaths and costs of the best possible lightpath for each demand after reoptimization understood as a length of the shortest path between demand's source and sink node that uses an affected link multiplied by a number of slices requested by the demand.

Having those values we can compute a maximum possible gain (minimum loss), denoted by $G(d)$, for each demand. The gain is equal $C(p(d)) - C(\hat{p}(d))$ for a given demand d .

Then, using dynamic programming we solve the knapsack problem where objects are demands, their profits are $G(d)$, weights are $n(d)$, and a knapsack capacity varies from 1 to S . Denote the result by $K(c)$, where c is a capacity of a knapsack.

Now for each demand we can compute if reoptimizing a lightpath for it can be profitable. If reoptimizing a lightpath for demand d can produce a direct gain, i.e., $G(d) > 0$, then the demand should be considered. In other case the demand is considered only if the losses are smaller than the gain that can be produced by reoptimizing other demands using the rest of available slices on the affected link, i.e., demand d is considered if $G(d) + K(S - n(d)) > 0$. We do so, because reoptimizing one lightpath (even when it produces direct losses) can give place for reoptimizing other demands, which would not be possible if the first lightpath was not rearranged. Notice that the procedure does not guarantee that

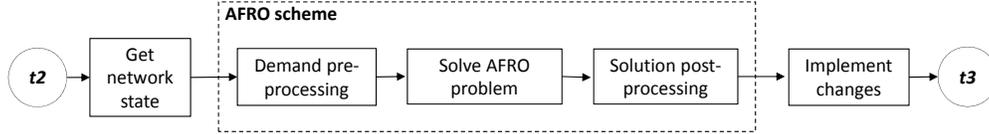


Fig. 3: Processing steps of the AFRO optimization algorithm.

rerouting of each demand considered for reoptimization can be profitable. However, it guarantees that if a demand is not considered, then rerouting this demand definitely cannot work towards cost minimization.

2) *Reduced procedure*: The above rule seems to be too rigid because it allows an excessive number of demands to the reoptimization process. Although it is theoretically sound, in practice conditions that justify it hardly occur. Therefore, we decided to design another method that will allow less demands to the reoptimization process. The second procedure we have developed is common-sense based. Its core is similar to the core of the first procedure. However, now we only consider demands that in the most optimistic case cannot generate direct losses, i.e., their current lightpaths are not shorter than the shortest possible lightpaths that can be used for them in reoptimization.

B. Solving AFRO with a column generation algorithm

In order to efficiently solve AFRO for real networks with hundreds of lightpaths, we have designed a dedicated column generation algorithm based on the work recently proposed in [16] and [17]. The column generation algorithm allows for optimal solving of IP formulations, such as the ones presented in Section III, for the RSA-based problems in an affordable way, by considering a reduced set of problem variables (corresponding to the lightpaths) without significantly affecting the quality of the obtained solution [15].

In details, in our column generation algorithm, a linear programming (LP) relaxation of our IP problem is initiated and solved with a small, feasible set of allowable lightpaths (see Section IV-B1), denoted as \mathcal{P} . Set \mathcal{P} is then extended iteratively with new lightpaths. A key element of column generation is to formulate and solve a *pricing* problem (see Section IV-B2), which consists of finding such a new lightpath that, when included into the problem formulation, it can lead to the improvement in the objective function value of the LP problem relaxation. To achieve this, the pricing problem makes use of the current LP solution and, in particular, of the values of its dual variables. If such a lightpath exists, it is included into set \mathcal{P} and LP problem is solved again, otherwise, the column generation procedure is terminated. Below, we present a pseudo-code of the column generation algorithm.

Once set \mathcal{P} is found using column generation, it is used as a set of allowable lightpaths in the IP problem and this problem is now solved using branch-and-bound. Since set \mathcal{P} is much smaller than the set containing all possible lightpaths, our IP problem can be solved in shorter times. Note that although the optimality of the LP problem relaxation is guaranteed

Algorithm 1 Column generation.

- 1: Initialize set \mathcal{P} with current lightpaths;
 - 2: Solve the LP relaxation of the MIP problem for the set \mathcal{P} of allowable lightpaths;
 - 3: Find new allowable lightpath p by solving the pricing problem using the dual variables of the current LP solution;
 - 4: If p exists, include it into set \mathcal{P} and go to step 2, otherwise, terminate the algorithm.
-

after column generation, still the obtained set of allowable lightpaths does not necessarily lead to an optimal solution of the IP problem.

In the following, we discuss in details the processing steps of the column generation algorithm.

1) *Initial set of lightpaths*: In our AFRO problem, the initial set of allowable lightpaths \mathcal{P} contains just single lightpaths (i.e., current lightpaths of the network). This set is enlarged solely for demands selected for reoptimization in Section IV-A by lightpaths containing link e^r using the lightpath generation algorithm of Section IV-B3.

2) *Solving pricing problem*: In order to formulate the pricing problem, we first derive the problem dual to the LP relaxation of the primal problem, i.e., problem (1) and (3), respectively, for formulation MIN-SUM and MIN-MAX. To this end, we use the dual variables specified on the left-hand sides of constraints (1b)-(1c) and (3b)-(3d), respectively. The dual to (1) is as follows:

$$\min \sum_{d \in \mathcal{D}} \alpha_d - \sum_{e \in \mathcal{E}} \sum_{s \in \mathcal{S}} \beta_{es} \quad (4a)$$

$$\alpha_d - \sum_{e \in \mathcal{E}(p)} \sum_{s \in c(p)} (\beta_{es} + 1) \leq 0 \quad d \in \mathcal{D}, p \in \mathcal{P}(d), \quad (4b)$$

and the dual to (3) is the following:

$$\min \sum_{d \in \mathcal{D}} \alpha_d - \sum_{e \in \mathcal{E}} \sum_{s \in \mathcal{S}} \beta_{es} \quad (5a)$$

$$\alpha_d - \sum_{e \in \mathcal{E}(p)} \sum_{s \in c(p)} (\beta_{es} + H(p) \cdot \gamma_{es}) \leq 0 \quad (5b)$$

$$\begin{aligned} d \in \mathcal{D}, p \in \mathcal{P}(d), \\ \sum_{s \in \mathcal{S}} \gamma_{es} \leq 1 \quad e \in \mathcal{E}, \end{aligned} \quad (5c)$$

where $\alpha_d \in \mathbb{R}, d \in \mathcal{D}$ and $\beta_{es} \geq 0, \gamma_{es} \geq 0$ for $e \in \mathcal{E}, s \in \mathcal{S}$.

The formulas at the left-hand sides of constraints (4b) and (5b) represent the so-called reduced costs of primal variables x_{dp} , respectively, for formulation MIN-SUM and MIN-MAX.

The pricing problem consists in finding such a new lightpath p for which its reduced cost is positive. When found, new variable x_{dp} representing this lightpath is included into the primal problem. Note that for a given LP solution of the primal, the values of dual variables are also given—we use these values to calculate the reduced costs.

As we can see, for given demand d , the minuend of the reduced cost (i.e., α_d) is fixed for any lightpath realizing this demand. On the contrary, the subtrahend (i.e., $\sum_{e \in \mathcal{E}(p)} \sum_{s \in c(p)} \kappa_{es}$, where $\kappa_{es} = \beta_{es} + 1$ for MIN-SUM and $\kappa_{es} = \beta_{es} + H(p) \cdot \gamma_{es}$ for MIN-MAX) depends on lightpath p . Therefore, since in the pricing problem we are looking for a lightpath with a positive reduced cost, it is enough to look for a lightpath for which the sum $\sum_{e \in \mathcal{E}(p)} \sum_{s \in c(p)} \kappa_{es}$ is the smallest. If we assume that κ_{es} represents the cost of using slice s in link e , our new lightpath has to be the cheapest (i.e., shortest) one with respect to these costs. In Section IV-B3, we present a procedure for generating such lightpaths.

3) *Generating new lightpaths*: In the constraint generation problem we are looking for a directed loop-less lightpath containing link e^r . The lightpath, denoted by p , should be the shortest with respect to metric π_e for all $e \in \mathcal{E}$, where $\pi_e = \sum_{s \in c(p)} \kappa_{es}$. The lightpath has to be searched for each demand $d \in \mathcal{D}$ and each slot. Those paths could be found using the Suurballe's algorithm [18] if the considered network and demands were undirected. However, in our directed case the problem is \mathcal{NP} -hard, as it can be reduced to 2DIV-PATHS (finding node-disjoint paths from s_1 to t_1 and from s_2 to t_2) in a directed graph [19].

As we cannot afford solving to optimality an \mathcal{NP} -hard problem $2 \cdot |\mathcal{D}| \cdot |\mathcal{S}|$ times in each iteration, we have proposed a different approach. Instead of generating shortest loop-less paths, we generate shortest paths that can contain loops, and if they do so, we do not include them to the set of potential lightpaths. Such an approach, although not exact, is very promising, as it can be efficiently implemented in practice. Not only the generating algorithm is polynomial now, but also instead of running the Dijkstra's algorithm for $2 \cdot |\mathcal{D}| \cdot |\mathcal{S}|$ times, it is possible to invoke the algorithm solely $2 \cdot |\mathcal{K}| \cdot |\mathcal{S}|$ times, where \mathcal{K} is a set of different sizes of demands in terms of a number of requested slices. The number of requested slices will be denoted by $n(k)$, where $k \in \mathcal{K}$. Notice that $|\mathcal{K}| \ll |\mathcal{D}|$. The method can be implemented in this way because, as shown in Section IV-B2, reduced costs of links are the same for all the demands. Notice that it would not be the case, if we had stuck to the initial MIN-MAX formulation (2).

We denote the Dijkstra's algorithm [20] run from node n by $P = DA(n)$. It returns a set of paths P , where $P(m)$ denotes the cost of the shortest path from node n to node m . The procedure looks as follows:

The procedure takes advantage of a well known feature of Dijkstra's algorithm, which calculates shortest paths from one source to all other nodes as fast as a single shortest path between a pair of nodes.

C. Disruption-aware lightpath generation

One of potential problems that may occur during network reconfiguration is disruption of traffic on rerouted lightpaths.

Algorithm 2 Lightpath generation.

```

1: for each  $s \in \mathcal{S}$  do
2:   for each  $k \in \mathcal{K}$ ,  $s + n(k) < \mathcal{S}$  do
3:     Calculate costs of links  $\pi_e = \sum_{l=0}^{n(k)-1} \kappa_{e(s+l)}$ 
4:      $P^1 = DA(s(e^r))$  with links reversed
5:      $P^2 = DA(t(e^r))$ 
6:     for each  $d \in \mathcal{D}$  of size  $k$  do
7:       Check if a cost  $P^1(s(d)) + P^2(t(d)) + \pi_{e^r}$  for  $d$ 
         is the best;

```

In order to minimize this effect, in the optimization algorithm we apply two techniques: 1) when generating lightpaths, and 2) in post-processing. These techniques are described, respectively, in this and in the next subsection.

According to our preliminary research a vast majority of disruptions result from a situation when a newly reoptimized lightpath cannot be established before the one it replaces is torn down, because both of them are using some common slices on one or more links. We can prevent such situations from happening by not generating lightpaths that overlap with the lightpaths they are supposed to replace both in terms of routes and spectrum assignment. However, in order to make the problem less complex and coherent with the lightpath generation algorithm we presented in Section IV-B3, our excluding rule is slightly more harsh. In our experiments we decided to generate lightpaths that use no common slices with the current lightpath. However, they can use links utilized by the current lightpath. Such an approach does not take into consideration feasible lightpaths that share one or more slices with the current lightpath, but are link disjoint. Although less exact than the model approach described earlier, this approach can be easily adapted by the applied generation algorithm, as it just requires to consider only an appropriate subset of demands for each s and k that will not contain demands which occupy slices that are common with the spectrum defined by s and k .

D. Solution post-processing

The post-processing procedure aims at ordering and grouping the lightpaths that have to be rerouted in such a way that all the changes can be performed in a minimum number of bundles without disruption. We define a bundle without disruption as a subset of reroutings that can be performed simultaneously (i.e., in parallel) and by means of a make-before-break procedure [21]. The make-before-break procedure consists in establishing the new lightpath and rerouting the traffic from the old lightpath to the new one before tearing down the old lightpath. It can be assumed that the signalling related to the setup and tear down of n lightpaths in a bundle can be performed in parallel, so that the time required to reroute all lightpaths in a bundle corresponds to the time necessary to reroute just one lightpath.

However, it may happen that the make-before-break strategy cannot be applied to all the rerouted lightpaths, especially, if it is necessary to remove some existing lightpaths so that to establish some others. In such case, it will lead to traffic disruptions on the affected lightpaths. Therefore, the process of ordering and grouping the lightpaths into bundles should aim

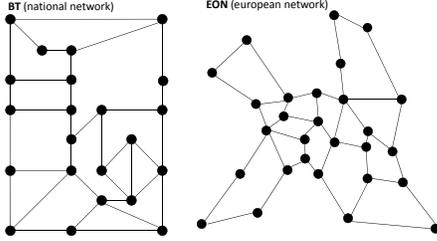


Fig. 4: Network topologies.

at reducing such situations. For solutions without disruptions or with the same number of disruptions, the secondary objective is to minimize the number of bundles so that to minimize the time required to perform lightpath rerouting.

In order to address the above objectives, we have implemented a simple iterative heuristic in which, given a sequence of lightpaths to be rerouted, we create and fill the bundles by including into them the lightpaths in the order in which they appear in the sequence. Therefore, in the first iteration the first bundle of lightpaths is created by checking all lightpaths in the sequence and including into the bundle such lightpaths that can be rerouted without disruptions (i.e., with the make-before-break strategy). After that, the first bundle is closed and a new one is open. Now, in consecutive iterations, the procedure is repeated for the rest of lightpaths (i.e., not included into the previously filled bundles) until either there are no lightpaths left or for the rest of lightpaths it is not possible to perform their rerouting with the make-before-break strategy. In the latter case, we assume that such lightpaths will be disrupted.

Eventually, in order to diversify the set of solutions, several random sequences of lightpaths are generated, then the above described heuristic is run for all of them, and finally the best solution is selected.

V. NUMERICAL AND SIMULATION RESULTS

In this section, we evaluate the performance of a dynamic flexgrid network operating with AFRO. To this end, in Section V-B we compare different options for the AFRO optimization algorithm, which are described in Section IV, in order to find best performing settings. Then, considering these settings, in Section V-C we apply the AFRO framework in a dynamic network scenario and evaluate it by means of network simulations. Eventually, in Section V-D we study the impact of applying AFRO only for a selected set of network links, instead of using it in all network links, with the aim to reduce the AFRO operational complexity.

A. Evaluation scenario

To evaluate the performance of AFRO, we use two representative core network topologies: the 22-node British Telecom (BT) and the 28-node European (EON) networks (see Fig. 4). The details of the networks are presented in Table II. We consider the fiber links with the spectrum width equal to 2 THz, divided into frequency slices of 6.25 GHz (320 slices per link). We assume a link failure rate of $2.72 \cdot 10^{-3}$ per

km per year [14]. Taking this into account and considering the total length of fiber links in BT and EON (as presented in Table II), the average number of cuts per year is equal to 14 and 70, respectively. Accordingly, the mean time to failure (MTTF) is equal to 625 and 125 hours, respectively, for BT and EON. We assume that the mean time to repair (MTTR) is the same for both networks and equal to 12 hours. The AFRO problem is solved using the CPLEX v.12.5 optimizer [22] on a 2.4GHz Quad-core machine with 8GB RAM. Simulations are run on an ad-hoc event-driven simulator implemented in OMNET++ [9].

Connection requests are generated following a Poisson process and lightpaths are torn down after an exponentially distributed holding time (HT). The source and destination nodes are uniformly chosen. We consider that the requested bit-rate can be equal to 40, 100, or 400 Gbit/s, and the share of each connection type is equal to 66.7%, 26.7%, and 6.6%, respectively. Such traffic profile combines several bit-rates and takes advantage of high spectrum granularity of the flexgrid [9]. Assuming QPSK as the applied modulation format, with spectrum efficiency equal to 2 bit/s/Hz, the number of slices requested for each connection type is 4, 8, and 32, respectively. One of the performance metrics that we use in the evaluation is the blocking probability (BP) of incoming connection requests weighted by the requested bit-rate.

B. Evaluation of the AFRO algorithm

In this section, our main objective is to find best performing settings of the AFRO optimization algorithm. To this end, we generate network instances in the following way:

- 1) we run a network simulation, realizing connection setup and tear down requests, until reaching a steady state in terms of BP—after that the network is in state t_0 ;
- 2) we select a link, which is subject to failure, and run the bulk restoration algorithm proposed in [7] for affected lightpaths—the network is in state t_1 ;
- 3) the link is immediately repaired after restoration—the network is now in state t_2 ;
- 4) we run AFRO—after that the network is in state t_3 ;

After performing the above steps, we calculate network statistics (discussed in details later). The evaluation is performed for such traffic load, proper for each studied network, that it results in BP equal to 1% in state t_0 . Presented results are averaged over 100 randomly generated network instances.

1) *Efficiency of IP formulations*: First, we compare the results obtained for two different formulations of the AFRO problem, i.e., MIN-SUM and MIN-MAX. We remind that these formulations differ in the objective function that is subject to optimization. In this analysis, we consider that all affected lightpaths are subject to optimization and the mechanisms minimizing traffic disruptions are not used.

In Table III, we present the results of network link usage, in terms of the number of allocated slices, both averaged over all network links and obtained for the most occupied link in the network. Also, we focus on link entropy, which is a metric related to the fragmentation of the free spectrum and which is computed by means of the Shannon link entropy

TABLE II: Network parameters.

Topology	Nodes	Links	Total Link Length [km]	Av. # of cuts per year	MTTF [h]	MTTR [h]
BT	22	35	5145	~ 14	625	12
EON	28	41	25625	~ 70	125	12

TABLE III: Link usage and link entropy observed in different network states.

Scenario		Link usage						Link entropy					
		$t0$		$t2$		$t3$		$t0$		$t2$		$t3$	
Network	Formulation	av.	max.	av.	max.	av.	max.	av.	max.	av.	max.	av.	max.
BT	MIN-SUM	79.40	226.04	81.22	231.92	79.31	227.56	1.0755	1.7355	1.0543	1.7506	1.0519	1.7307
	MIN-MAX	79.40	226.04	81.22	231.72	81.18	231.04	1.0755	1.7355	1.0541	1.7465	1.0435	1.7386
EON	MIN-SUM	77.97	215.68	79.76	229.33	77.87	219.60	1.1232	1.7427	1.1034	1.7500	1.0930	1.7302
	MIN-MAX	77.97	215.68	79.75	229.36	79.63	228.36	1.1232	1.7427	1.1063	1.7500	1.1011	1.7450

formula proposed in [23]. This entropy value is lower when the fragmentation of free spectrum is lower.

We can see that in all studied cases the application of AFRO (see $t3$) improves both studied metrics when compared to the network without AFRO (see $t2$). Moreover, AFRO allows to reoptimize the network so that its link statistics are similar to those observed during normal network operation (compare $t3$ with $t0$). Finally, the optimization of overall spectrum usage (MIN-SUM) is more effective than optimization of the sum of highest indices of used slices over all links (MIN-MAX). Therefore, in the remainder of the paper we consider that AFRO makes use of the MIN-SUM formulation.

2) *Evaluation of the column generation algorithm:* Next, we analyze previously proposed strategies for selecting demands to be processed, namely, *complete* and *reduced*, as well as a strategy in which all demands in the network are processed (denoted as *full*). We consider that the mechanisms minimizing traffic disruptions are not applied in the analysis.

In Table IV, we present averaged results obtained for the considered strategies. We focus on the number of demands which are subject to reoptimization ($|\mathcal{D}'|$), the number of allowable lightpaths obtained in the column generation algorithm ($|\mathcal{P}|$), the number of rerouted lightpaths ($|\mathcal{P}^r|$), the link entropy, a relative optimality gap between the integer problem solution obtained for the generated set of lightpaths and the LP relaxation (*gap*), and the algorithm processing time (T).

We can see that the reduced strategy allows to decrease considerably the number of demands selected for optimization when compared to the other two strategies. At the same time, the number of generated and rerouted lightpaths as well as the link entropy is comparable for all strategies. Note that the optimality gap is very low (below 0.03% in almost all cases). It shows that the quality of generated lightpaths is very high and for these lightpaths near-optimal integer solutions can be found. We can also see that the processing time is very similar in all analyzed cases and is much less than 1 minute. We can conclude that the algorithm is fast enough for the scenarios with traffic dynamics that are assumed in Section V-C.

In the rest of the paper, we assume that the column generation algorithm is implemented with the *reduced* strategy,

since it provides the best trade-off between solution quality and execution time.

3) *Analysis of traffic disruptions:* Eventually, we study the effectiveness of the traffic disruption-aware mechanisms discussed in Sections IV-C and IV-D.

In Table V, we present averaged results of the number of rerouted lightpaths, the number of lightpath disruptions, and the number of lightpath bundles. We remind that lightpath disruption occurs if the lightpath cannot be rerouted with the make-before-break mechanism. The number of bundles corresponds to the signalling overhead of the lightpaths rerouting.

We can see that both the number of lightpaths selected to be rerouted, which is the result of the optimization algorithm, and the signalling overhead are not affected by using our traffic disruption-aware mechanisms. At the same time, the number of disrupted lightpaths is significantly reduced, up to one order of magnitude for EON.

In the following experiments, the traffic disruption-aware mechanisms are applied.

C. Dynamic performance of AFRO

Here, we study the performance of AFRO by means of dynamic network simulations.

In Fig. 5 we show the results of BP in a function of the offered load, which is normalized to the load for which BP is equal to 1% in a network operating without failures (i.e., for $MTTF = \infty$). The results are obtained for different values of connection holding time (HT). We can see that the application of AFRO leads to lower values of BP in all studied scenarios when compared to a network without AFRO. Moreover, the relative gain in terms of BP after applying AFRO increases with traffic load as well as with higher values of HT. It can be concluded that if a network has to guarantee certain target level of BP (e.g., BP=1%), the application of AFRO allows to increase the network throughput. Finally, we can see that the difference between the cases with AFRO and w/o AFRO is more prominent in the EON topology than in the BT topology.

In order to evaluate the gain in supported traffic load (i.e., throughput), in Fig. 6 we present these results obtained for different values of HT at BP=1%. Again, we can see that the

TABLE IV: Performance of the column generation algorithm.

Network	Demand selection strategy	$ \mathcal{D}' $	$ \mathcal{P} $	$ \mathcal{P}^r $	link entropy	gap [%]	T [s]
BT	full	255.28	723.90	13.12	1.0520	0.021	21.567
	complete	226.75	717.86	13.08	1.0516	1.010	21.686
	reduced	45.58	747.18	13.07	1.0520	0.021	22.458
EON	full	263.53	645.43	13.37	1,0928	0.027	15.946
	complete	238.14	639.62	13.12	1,0934	0.023	15.636
	reduced	43.48	625.43	13.16	1,0928	0.025	15.404

TABLE V: Disruption analysis.

Network	Disruption-aware CG	Rerouted lightpaths	Number of disruptions	Number of bundles
BT	No	13.40	2.29	2.15
	Yes	14.19	0.47	2.49
EON	No	13.18	2.89	2.36
	Yes	13.51	0.26	2.52

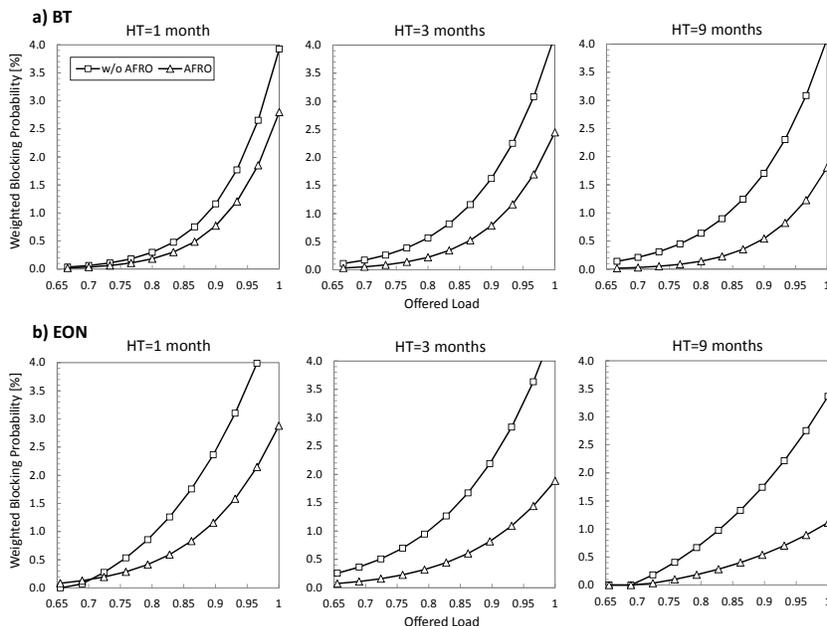


Fig. 5: Weighted blocking probability vs. normalized network load.

higher value of HT the higher gain in network throughput after applying AFRO. Also, in order to achieve a 10% gain, the connection holding time can be shorter in EON ($HT \approx 1.5$ month) than in BT ($HT \approx 4$ months).

Eventually, in Table VI we evaluate the impact of AFRO on the effectiveness of the restoration mechanism used in our network. In particular, we can see that in both BT and EON topologies the percentage of restored connections in a network operating with AFRO is not only preserved but it is even slightly better than in a network w/o AFRO.

D. Solving AFRO after selected link repairs

The last set of experiments concerns network scenarios in which AFRO is run only after the recovery of a selected set of links, instead of using it for all network links. In particular, in

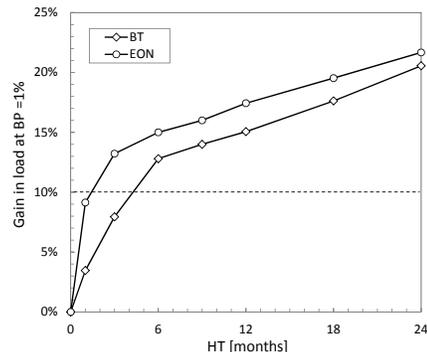
Fig. 6: Gain in supported load at $BP = 1\%$ vs. connection holding time.

TABLE VI: Effectiveness of connection restoration in a dynamic network without and with AFRO.

Network	Scenario		% of restored connections	
	HT	HT/MTTF	w/o AFRO	AFRO
BT	4	4.6	99.58	99.65
EON	1.5	8.7	99.48	99.52

this study AFRO is applied for a set of links which are most loaded during normal network operation.

In Figure 7, for the BT topology and HT=9 months, we present the performance of AFRO, in terms of weighted BP, assuming different percentage of network links which are considered for reoptimization (denoted as ρ). Note that AFRO is not applied if $\rho = 0\%$, is always used if $\rho = 100\%$, and for instance for $\rho = 40\%$ it is applied for 40% of (most loaded) network links. Apart from that we calculate the AFRO relative gain, which represents the improvement in BP with respect to the scenario without using AFRO.

In Figure 7, we can see that it is not necessary to run AFRO for all network links in order to obtain its maximal efficiency. For instance, if we restrict AFRO to be applied after the recovery of links which are among 60% of most loaded links, the AFRO relative gain is already equal to 100%. Moreover, it is enough to use it only for 20% of links so that to achieve a 50% reduction of BP. These results indicate that AFRO can be applied with lower frequency, thus reducing its operational and signaling complexity, without affecting significantly its effectiveness.

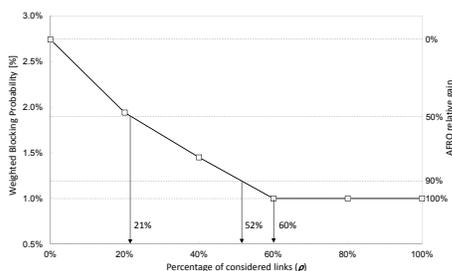


Fig. 7: Weighted blocking probability and AFRO relative gain vs. percentage of links considered for reoptimization.

VI. CONCLUSION AND FUTURE WORK

In the paper we presented AFRO—an online reoptimization problem useful for improving traffic performance of a flexgrid optical network subject to link failures and employing dynamic lightpath restoration. Specifically, when a failure is repaired, AFRO considers rerouting the traffic from some existing lightpaths to new lightpaths established through the repaired link, thus making use of the restored capacity. Through this, a better distribution of network load and, consequently, a decrease in the connection blocking probability can be achieved.

The AFRO problem was defined by means of two alternative link-path IP formulations. Due to the AFRO requirement to use the repaired link by new lightpaths and in order to avoid the

solution of large problem instances, i.e., with a large set of pre-computed lightpaths, column generation was applied in a form of a time-wise acceptable procedure to obtain near-optimal solutions. Based on our previous work on column generation for a basic RSA problem in flexgrid optical networks, specific procedures for both presented formulations were developed. Moreover, several strategies for reducing the computational effort, including the reduction of the amount of lightpaths considered for reoptimization, as well as some extensions to minimize traffic disruptions when migrating from old to new lightpaths were proposed.

Numerical results obtained for a national network and a European network show that AFRO reduces both the amount of used capacity and the spectrum fragmentation (measured in terms of link entropy) with respect to the network state before applying AFRO. Additionally, the proposed column generation algorithm provides quality solutions in execution times acceptable for a real dynamic scenario. The effectiveness of AFRO in improving the performance of a dynamic network in operation is demonstrated by two main results: (i) a significant blocking probability reduction ($> 10\%$) for reasonable values of MTTF, MTTR, and HT is achieved; (ii) application of AFRO allows improving the performance of the dynamic restoration algorithm by increasing the restorability of connections affected by a failure. Finally, it is shown that the operational overhead of AFRO can be reduced by using its mechanism only for a selected set of links ($\geq 60\%$ of all links) without a negative impact on its performance.

The presented models are developed for a network operating with a single modulation format. Further extensions corresponding to adaptive transponders, i.e., making use of multiple modulation formats in accordance to lightpath characteristics, are left for further study.

ACKNOWLEDGMENT

The presented work was supported by the FP7 project IDEALIST (grant agreement no. 317999). The work was also supported by National Science Center (Poland) under grant 2011/01/B/ST7/02967 (M. Pióro and M. Żotkiewicz) and 2011/01/D/ST7/05884 (M. Klinkowski). M. Żotkiewicz was additionally supported by the EC European Social Fund through the Warsaw University of Technology Development Programme. Also, M. Ruiz and L. Velasco were supported by the Spanish Ministry of Science through the TEC2011-27310 ELASTIC project. The authors wish to thank Alberto Castro for his valuable collaboration.

REFERENCES

- [1] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and scalable elastic optical path network:

- Architecture, benefits, and enabling technologies,” *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 66–73, Nov. 2009.
- [2] ITU-T Recommendation G.694.1 (ed. 2.0), “Spectral grids for WDM applications: DWDM frequency grid,” Feb. 2012.
- [3] F. Paolucci, A. Castro, F. Fresi, M. Imran, A. Giorgetti, B. Bhowmik, G. Berrettini, G. Meloni, F. Cugini, L. Velasco, L. Poti, and P. Castoldi, “Active PCE Demonstration performing Elastic Operations and Hitless Defragmentation in Flexible Grid Optical Networks,” *Phot. Netw. Commun.*, 2014.
- [4] D. King and A. Farrel, “A PCE-based architecture for application-based network operations,” IETF draft, 2014.
- [5] E. Crabbe, I. Minei, S. Sivabalan, and R. Varga, “PCEP extensions for PCE-initiated LSP setup in a stateful PCE model,” IETF draft, 2013.
- [6] L. Velasco, A. Castro, M. Ruiz, and G. Junyent, “Solving Routing and Spectrum Allocation Related Optimization Problems: from Off-Line to In-Operation Flexgrid Network Planning,” *IEEE/OSA J. Lightw. Technol.*, vol. 32, pp. 2780–2795, 2014.
- [7] A. Castro, L. Velasco, J. Comellas, and G. Junyent, “Dynamic restoration in multi-layer IP/MPLS-over-flexgrid networks,” in *Proc. of DRCN*, Budapest, Hungary, 2013.
- [8] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. Lopez, “In-Operation Network Planning,” *IEEE Commun. Mag.*, vol. 52, pp. 52–60, 2014.
- [9] L. Gifre, F. Paolucci, A. Aguado, R. Casellas, A. Castro, F. Cugini, P. Castoldi, L. Velasco, and V. Lopez, “Experimental Assessment of In-Operation Spectrum Defragmentation,” *Phot. Netw. Commun.*, vol. 27, pp. 128–140, 2014.
- [10] M. Ruiz, M. Żotkiewicz, A. Castro, M. Klinkowski, L. Velasco, and M. Pióro, “After failure repair optimization in dynamic flexgrid optical networks,” in *Proc. of OFC*, San Francisco, USA, Mar. 2014.
- [11] L. Gifre, F. Paolucci, L. Velasco, A. Aguado, F. Cugini, P. Castoldi, and V. Lopez, “First Experimental Assessment of ABNO-driven In-Operation Flexgrid Network Re-Optimization,” *IEEE/OSA J. Lightw. Technol.*, 2014.
- [12] L. Song, J. Zhang, and B. Mukherjee, “A Comprehensive Study on Backup-Bandwidth Reprovisioning After Network-State Updates in Survivable Telecom Mesh Networks,” *IEEE/ACM Trans. on Netw.*, vol. 16, no. 6, pp. 1366–1377, 2008.
- [13] J. Ahmed, F. Solano, P. Monti, and L. Wosinska, “Traffic Re-Optimization Strategies for Dynamically Provisioned WDM Networks”, in *Proc. of ONDM*, Bologna, Italy, Feb. 2011.
- [14] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger, “General availability model for multilayer transport networks,” in *Proc. of DRCN*, Ischia Island, Italy, Oct. 2005.
- [15] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann, 2004.
- [16] M. Ruiz, M. Żotkiewicz, L. Velasco, and J. Comellas, “A column generation approach for large-scale RSA-based network planning,” in *Proc. of IEEE ICTON*, Cartagena, Spain, 2013.
- [17] M. Ruiz, M. Pióro, M. Żotkiewicz, M. Klinkowski, and L. Velasco, “Column generation algorithm for RSA problems in flexgrid optical networks,” *Phot. Netw. Commun.*, vol. 26, no. 2-3, pp. 53–64, 2013.
- [18] J. W. Suurballe, “Disjoint paths in a network,” *Networks*, vol. 4, no. 2, p. 125145, 1974.
- [19] S. Fortune, J. Hopcroft, and J. Wyllie, “The directed subgraph homeomorphism problem,” *Theoretical Computer Science*, vol. 10, no. 2, pp. 111 – 121, 1980.
- [20] E. W. Dijkstra, “A note on two problems in connection with graphs,” *Numerische Math.*, vol. 1, pp. 269–271, 1959.
- [21] D. Awduche, L. Berger, D. Gan, T. Li, G. Swallow, and V. Srinivasan, “RSVP-TE: Extensions to RSVP for LSP tunnels,” IETF RFC-3209, Dec. 2001.
- [22] IBM, “ILOG CPLEX optimizer,” 2012, <http://www.ibm.com>.
- [23] P. Wright, M. C. Parker, and A. Lord, “Simulation results of shannon entropy based flexgrid routing and spectrum assignment on a real network topology,” in *Proc. of ECOC*, London, UK, Sep. 2013.