

Impairment-aware Path Selection and Regenerator Placement in Translucent Optical Networks

Fernando Kuipers*, Anteneh Beshir*, Ariel Orda[†], and Piet Van Mieghem*

*Network Architectures and Services
Delft University of Technology
Mekelweg 4, 2628 CD Delft
The Netherlands

{F.A.Kuipers, A.A.Beshir, P.F.A.VanMieghem}@tudelft.nl

[†]Department of Electrical Engineering
Technion

Israel Institute of Technology
Haifa, Israel 32000

ariel@ee.technion.ac.il

Abstract—Physical impairments, such as noise and signal distortions, negatively affect the quality of information transfer in optical networks. The effect of physical impairments predominantly augments with distance and bit rate of the signal to the point that it becomes detrimental to the information transfer. To reverse the effect of physical impairments, the signal needs to be regenerated at nodes that have regeneration capabilities. Regenerators are costly and are, therefore, usually only sparsely placed in the network, in which case it is referred to as a translucent network. This paper deals with two problems in translucent networks, namely: (1) how to incorporate impairment awareness in the routing algorithms, and (2) how many regenerators to place inside the network and where. We propose exact and heuristic algorithms for impairment-aware path selection and, through simulations, show that our heuristic *TIARA* is computationally efficient and performs very close to our exact algorithm *EIARA*. Subsequently, we propose a greedy algorithm for placing regenerators that, contrary to previous proposals, is suitable for multiple impairment metrics, has polynomial complexity for a single impairment metric, and is cheaper in terms of the number of regenerators needed.

Index Terms—Optical Signal Impairments, Regenerator Placement, Impairment-aware Routing, Translucent Networks.

I. INTRODUCTION

Optical networks using wavelength-division multiplexing (WDM) technology are being widely deployed in long-haul and metro/regional networks. In transparent optical networks, the signal is transmitted in the optical domain from the source to the destination, without any conversion to the electrical domain. If the signal is not regenerated at intermediate nodes, noise and signal distortions are accumulated along the physical path. The noise and signal distortions are known as *physical impairments* and degrade the quality of the received signal. Especially for long distances and high bit rates, the signal degradation may lead to an unacceptable bit error rate (BER).

To overcome physical impairments, re-amplification, re-shaping, and re-timing, which are collectively known as 3R regeneration, are used at intermediate nodes. Since regenerators are costly, it is practical to have only sparse regeneration capacity in the network. A network that uses sparse regeneration is known as a *translucent optical network*.

This paper deals with two problems in translucent optical

networks, namely: (1) how to incorporate impairment awareness in the routing algorithms, and (2) how many regenerators to place inside the network and where.

The outline of this paper is as follows. In Sec. II, we present a model of an optical transport network and discuss the key physical impairments that can be encountered in such a network. In Sec. III, we briefly overview the work related to impairment-aware routing and regenerator placement. In Sec. IV, we propose impairment-aware path selection algorithms in translucent networks. In Sec. V, we deal with the regenerator placement problem. We conclude in Sec. VI.

II. IMPAIRMENTS MODEL

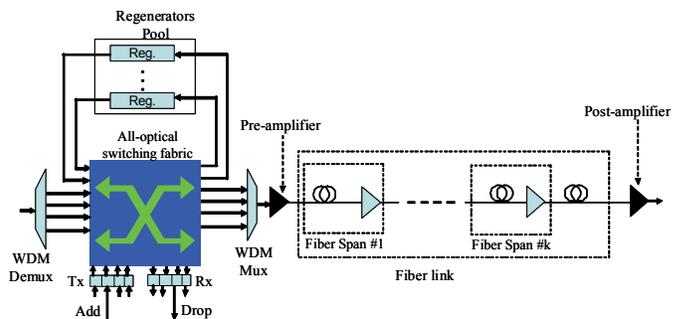


Fig. 1. Network components that make up a translucent WDM OTN.

Fig. 1 shows the components that make up a typical transmission system of a translucent optical transport network (OTN). In this model, a node is mainly composed of an all-optical switch, optional regenerators, transponders, multiplexers, demultiplexers, and pre- and post-amplifiers; whereas a fiber link is a WDM line system comprising of fibers and amplifiers. At a given node, transponders modulate electrical signals onto distinct wavelengths. These wavelengths are then multiplexed and the multiplexed signal is pre-amplified before being propagated through the WDM line. At the receiver end, the signal is post-amplified and de-multiplexed into individual signals. A link has a single fiber in each direction, each containing a number of wavelengths to be used by lightpaths.

In Fig. 1, only one direction is shown. We assume that there is no conversion at any of the nodes, which implies that a lightpath should use the same wavelength on all of its links.

At each node, there are add and drop ports for data to locally enter and leave the network. Each incoming signal is demultiplexed and switched inside a translucent node using an all-optical switching fabric, which can switch an optical signal from any input port to any other output port. In a translucent network, certain wavelengths may pass through the pool of regenerators so that the quality of their signals is restored through regeneration. We assume that the optical switches at the translucent nodes have enough ports to support incoming signals as well as regenerated signals. It is possible that some nodes do not have regenerators, thus providing only the service of locally adding and dropping of wavelengths. We define a *regeneration segment* of a lightpath to be a transparent segment (i.e., one or more links) between two regenerator nodes (including source and destination nodes) of the lightpath. A lightpath can be made up of multiple regeneration segments. There is no need for a lightpath to be regenerated at the source and destination nodes. After a transit signal is regenerated, its original physical features are restored. Thus, from a physical impairment point of view, the effect of physical impairments along the path followed to reach the regenerator node is completely removed. Each lightpath is assumed to require a single wavelength and each request represents a single lightpath (otherwise, each lightpath can be considered independently).

According to RFC 4054 [29], physical impairments can be classified into two categories: linear and non-linear impairments. Linear impairments are independent of signal power and affect wavelengths individually. Non-linear impairments generate dispersion on channels and crosstalk between channels. We shall present the main impairments listed in [29].

- *Polarization Mode Dispersion (PMD)* is a form of modal dispersion where two different polarizations of light in a waveguide travel at different speeds due to imperfections and asymmetries, causing random spreading of optical pulses. PMD is expressed in ps/\sqrt{km} , which means that its square value is additive with distance.
- *Amplifier Spontaneous Emission (ASE)* refers to the emission of radiation (photons) due to the presence of an electromagnetic field. ASE degrades the optical signal to noise ratio (OSNR) and is reflected in that measure. In practice, vendors generally provide bounds on the length of the transparent segment and number of spans in order to ensure an acceptable level of OSNR. Assuming the same output power at all amplifiers along a segment, the constraint on the number of spans H is computed as

$$\sum_{j=1}^H n_{sp}(j)(\gamma(j) - 1) \leq \frac{P_L}{hvB_oSNR_{\min}},$$

where $n_{sp}(j)$ and $\gamma(j)$ are the spontaneous emission factor and the amplifier gain of the j -th amplifier, respectively; P_L is the average optical power, h is Planck's

constant, v is the carrier frequency, and B_o is the optical bandwidth.

- Other linear impairments, like *Polarization Dependent Loss (PDL)*, *Chromatic Dispersion*, *Crosstalk*, and *Effective Passband*, can be approximated by a domain-wide margin on the OSNR, plus in some cases a bound on the number of networking elements along the path [29].
- Incorporating non-linear impairments is much more complex and requires a detailed knowledge of the physical network. Strand and Chiu [29] suggested to trade-off accuracy for simplicity and to assume that the non-linear impairments are bounded and implicitly reflected in a maximum number of spans.

The major physical impairments can, therefore, be well approximated by one or more link-based additive metric(s) and corresponding constraint(s) (for instance on length and number of spans) [24]. In the literature, different types of cost functions have been suggested for links and nodes to represent their physical impairments during the path-selection process. These include the distance of a link [35], a logical distance [10], a combination of distance and hopcount [4], a cost that is a function of the Four-Wave Mixing (FWM) crosstalk [19], the signal quality Q -factor [7] [35], an aggregated cost of monitored link information [23], and the noise variance [11]. Approaches dealing with multiple metrics explicitly have also been considered (e.g., [20], [24]). These metrics may represent measured or computed physical impairment values. Our work is independent of the impairment cost function used, and is applicable to single or multiple additive link metrics.

III. RELATED WORK

There has been an increasing interest in dealing with physical impairments in optical networks. Most of the related work is either directed to (1) studying the problem of finding feasible paths that satisfy a given set of impairment constraints or (2) studying the optimal placement of regenerators in a network. We shall briefly discuss the work in both areas.

Azodolmolky *et al.* [2] have surveyed impairment-aware routing and wavelength assignment (RWA) algorithms. These impairment-aware algorithms commonly fall into two categories. In one category, the path and the wavelength of a lightpath are computed in the traditional way without taking into account the physical impairments, and subsequently the quality of the selected lightpath is tested against physical impairments [1], [12], [18], [21]. Then, new paths are computed if the candidate paths do not meet the physical impairment thresholds. In the second category, the physical impairment values are considered in the routing and/or wavelength assignment process [7], [11], [19], [22], [24], [26]. In these works, information pertaining to physical impairments is incorporated in finding a suitable path. However, most of them do not account for the presence of regenerator nodes during path computation. In order to achieve optimal use of regenerators, the physical impairment values of segments between regenerator nodes should be verified during the path computation process (as in our approach), and not after the path is finally

computed. We provide a detailed study into the complexity of this problem and propose both exact and heuristic algorithms.

There are several papers dealing with the placement of regenerators, e.g. [15], [16], [21], [25], [27], [30], [31], [32]. Some of the approaches depend on the type of physical impairments and some of them are specific to certain networks. In addition, most of them consider a single physical impairment metric. Chen *et al.* [5] have studied the regenerator placement problem for a single metric (i.e., distance) with the objective of finding the minimum number of nodes, where regenerators are to be placed so that there is a feasible path between any pair of nodes in the given network. They have shown that this problem is NP-complete and have provided heuristic algorithms. Flammini *et al.* [8] have considered different variants of the regenerator placement problem (also for a single metric) under the assumption that all links have the same cost, thus the impairment threshold is basically determined by the hopcount of the path. The main goal in the aforementioned regenerator placement studies is to minimize the number of nodes where regenerators are placed (*regenerator nodes*). However, this approach has two major drawbacks. Firstly, minimizing the number of *regenerator nodes* does not necessarily minimize the total number of *regenerators* needed, which directly affects the cost associated with regeneration. Secondly, minimizing the number of *regenerator nodes* leads to an NP-complete problem (even for a single metric). Therefore, in this paper, we study the regenerator placement problem where the main objective is to minimize the total number of *regenerators* used in the network, and we show that this problem is polynomially solvable for a single metric.

IV. IMPAIRMENT-AWARE PATH SELECTION

In this section, we shall assume that a network is given with a scarce amount of regenerators in place and that requests arrive in an online fashion, i.e. without prior knowledge of when and between which nodes these requests are made. We define the impairment-aware routing problem as follows.

Problem 1: The impairment-aware routing problem: The physical optical network is modeled as a graph $G(\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of N nodes and \mathcal{L} is the set of L links. Associated with each fiber link $(u, v) \in \mathcal{L}$ are m physical impairments $r_i(u, v)$, $i = 1, \dots, m$. $\mathcal{N}_R \subseteq \mathcal{N}$ represents the set of R nodes that have (spare) regeneration capacity. A request is represented by the tuple $(s, d, \vec{\Delta})$, where $s, d \in \mathcal{N}$ are the source and destination nodes of the request and $\vec{\Delta} = \{\Delta_1, \dots, \Delta_m\}$ represents m threshold values for the m physical impairments. The impairment-aware routing problem is to find a route from source to destination that does not exceed any of the thresholds Δ_i , $i = 1, \dots, m$ on any of its regeneration segments.

We illustrate this problem for $m = 1$ impairment using the example network in Fig. 2(a) for a request $(s, d, 5)$. In this example, the shortest path from s to d goes via the direct link (s, d) , but this path violates the impairment threshold, i.e., $r(s, d) = 6 > \Delta$. The only feasible path is $s - t - d$, where t is a regenerator node, because for the regeneration

segments $P_{s \rightarrow t} = s - t$ and $P_{t \rightarrow d} = t - d$, it holds that $r(P_{s \rightarrow t}) = r(P_{t \rightarrow d}) = 5 \leq \Delta$.

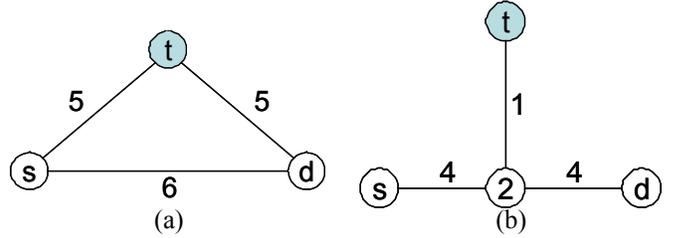


Fig. 2. Example networks with request $(s, d, 5)$. (a) There is a feasible path through regeneration node t . (b) There is no feasible simple path.

Consider now the instance in Fig. 2(b), where, given a request $(s, d, 5)$, there is a feasible *walk* $s - 2 - t - 2 - d$, but there is no feasible simple *path* for the impairment-aware routing problem. However, it is of interest to consider a variant of the impairment-aware routing problem where only simple paths are admitted as solutions. Such restrictions may be due to scarcity of resources (link or node capacities) or management considerations.

We first present a Polynomial-time Impairment-Aware Routing Algorithm (*PIARA*) for finding a path from source s to destination d subject to a single impairment threshold Δ and for the case that loops are allowed (i.e., nodes and links may be revisited). *PIARA* will serve as a module for our loopless impairment-aware routing algorithms.

Algorithm 1 *PIARA*($G, s, d, \mathcal{N}_R, \Delta$)

- 1) For each pair of nodes $u, v \in \mathcal{N}_R \cup \{s, d\}$, find the shortest (w.r.t. impairment) path $P_{u \rightarrow v}^*$.
 - 2) Make a graph G' consisting of nodes in $\mathcal{N}_R \cup \{s, d\}$. There is a link between nodes $u, v \in \mathcal{N}_R \cup \{s, d\}$ in G' if $r(P_{u \rightarrow v}^*) \leq \Delta$.
 - 3) Assign a cost to each link (u, v) in G' (e.g., a cost equal to $r(P_{u \rightarrow v}^*)$).
 - 4) Find a (shortest) path from s to d in G' and substitute the links of the path in G' with the corresponding subpaths in G .
-

By replacing the shortest path algorithm in Step 1 of *PIARA* with a multi-constrained path algorithm (like SAM-CRA [33]), we can deal with multiple impairments. However, since multi-constrained path selection is a (weakly) NP-complete problem [17], *PIARA* will no longer be of polynomial complexity. *PIARA* assumes that link or node capacities are not confining, even when traversed multiple times. When link or node capacities are confining we may need to find loop-free paths, which is considered in the remainder of this section. Although it is clear that the problem is NP-complete for $m > 1$ impairments, we shall demonstrate in the following that the problem is NP-complete for $m = 1$ as well.

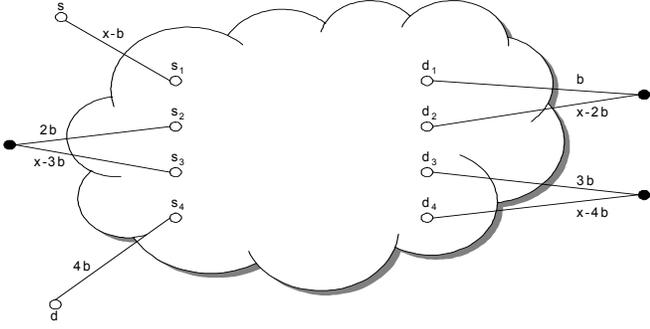


Fig. 3. Transformation of an MLBDP instance to an impairment-aware routing instance. Black nodes are regenerator nodes.

Theorem 1: The impairment-aware loopless routing problem is strongly NP-complete¹.

To prove that the problem is strongly NP-complete, we shall use the Maximum Length-Bounded Disjoint Paths (MLBDP) problem [9], which is defined as follows.

Problem 2: The maximum length-bounded disjoint paths problem: Given an undirected graph G , source s and destination d , and positive integers b and K , does G contain K or more mutually node-disjoint paths from s to d , none involving more than b links?

The MLBDP problem was proven to be NP-complete for $b \geq 5$ by Itai *et al.* in [13] and later proven to be APX-hard² for $b \geq 5$ by Bley in [3].

Proof: When we are given a path it is easy to verify whether it obeys the threshold Δ or not. The problem is therefore in NP. We shall provide a reduction to the MLBDP problem to prove strong NP-completeness.

Any instance of the MLBDP problem can be transformed in polynomial time to an impairment-aware routing instance as follows. The source node is split into K source nodes s_1, \dots, s_K and the destination node is split into K destination nodes d_1, \dots, d_K . Each of these source (destination) nodes is connected to the same nodes as the original source (destination) node. So far all links have a weight of 1. We add a new source and connect it to s_1 with a link of weight $x - b$. For each pair of source nodes (s_{2i}, s_{2i+1}) , for $i = 1, \dots, \lfloor \frac{K-1}{2} \rfloor$, we add a new regenerator node and link it to s_{2i} with weight $2ib$ and to s_{2i+1} with weight $x - (2i + 1)b$. For each pair of destination nodes (d_{2i-1}, d_{2i}) , for $i = 1, \dots, \lceil \frac{K-1}{2} \rceil$, we add a new regenerator node and link it to d_{2i-1} with weight $(2i-1)b$ and to d_{2i} with weight $x - 2ib$. The last node (either s_K or d_K) is connected to a new destination node through a link with weight Kb . Fig. 3 visualizes this construction for $K = 4$. If we choose $\Delta = x + b$ and $x > 2Kb$, then solving the impairment-

¹“Strongly NP-complete” indicates that the problem remains NP-complete even if the link weights are bounded by a polynomial in the length of the input. Unlike weakly NP-complete problems, these problems do not admit pseudo-polynomial time solutions.

²APX-hard problems can be approximated within some constant factor, but not every constant factor (as with polynomial-time approximation schemes), unless P=NP.

aware routing problem in the new graph provides a solution to the MLBDP problem. Moreover, since $Kb \leq 2(N - 1)$, we have that $\Delta = O(N)$, which on its turn means that the impairment-aware routing problem is strongly NP-complete. ■

Since good approximation schemes are unlikely to exist, as indicated by Theorem 1, we focus in the following sections on exact and heuristic solutions.

A. Problem variants

Depending on how regenerator nodes are used, associating an objective with solving the impairment-aware routing problem can lead to several problem variants. We shall first focus on the case of $m = 1$ impairment, after which we present our algorithms for the general case of $m \geq 1$ impairments.

Variante 1: Find the shortest (in terms of physical impairment) feasible path. Regenerators can be used at no extra cost.

Variante 2: Given that each used regenerator has a cost of usage that will be added to the total path length, find the shortest feasible path.

Variante 3: Find a feasible path that uses the fewest number of regenerators. In case of a tie, the one with shortest length is returned.

Problem variants 2 and 3 can be transformed into problem variant 1 by splitting each regenerator node in the input graph G into four nodes as shown in Fig. 4(a) for undirected networks and Fig. 4(b) for directed networks. In these figures, the link weight x equals the cost of using the given regenerator in problem variant 2, while $x = \Delta$ in problem variant 3. We will focus on solving problem variant 1.

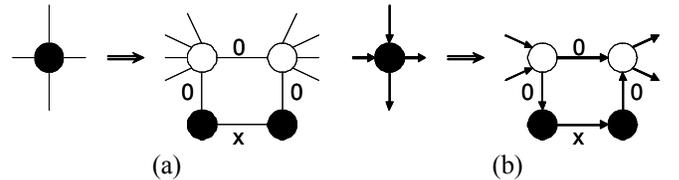


Fig. 4. Regenerator node splitting in (a) an undirected network, and (b) a directed network. Black nodes are regenerator nodes.

In solving the impairment-aware shortest path routing problem, we have to take into account two parameters during the search process:

- 1) The total length $r(P)$ of a (sub)path P accumulated since the source node.
- 2) The length $r'(P)$ since the last used regenerator node (or the source node) along a (sub)path P .

The fact that $r'(P)$ does not reflect an end-to-end property prevents a simple adoption of multi-parameter algorithms like SAMCRA [33]. Two search-space reducing techniques that are used in SAMCRA are the concept of *non-dominance* (or *Pareto optimality*) and the concept of *look-ahead* (or A^*). We will demonstrate that, while the concept of non-dominance cannot be used, we can apply the look-ahead concept with some modifications.

1) **Non-dominance:** When solving multi-constrained routing problems, at any intermediate node, it does not make sense to consider a (sub)path that has worse weights (i.e., higher or equal in every metric) than another (sub)path. Such paths are said to be *dominated* and are discarded, thereby reducing the search space. This non-dominance technique fails in impairment-aware routing as shown in Fig. 5. In this example, the request is $(s, d, 9)$ and t is the only regenerator node. At node 3, the subpath $P_1 = s - 3$ with $r(P_1) = r'(P_1) = 8$ is dominated by the subpath $P_2 = s - 1 - 2 - 3$ with $r(P_2) = r'(P_2) = 7$. However, P_1 cannot be discarded since it is part of the only feasible path $s - 3 - t - 2 - 1 - d$.

Assuming non-negative link weights, the non-dominance principle prevents loops along a path. In its absence, we will have to check for loops explicitly.

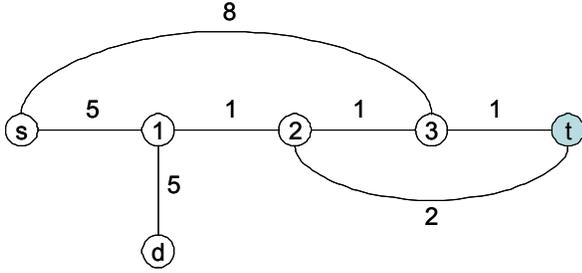


Fig. 5. An example wherein the concept of non-dominance fails for impairment-aware routing. The request is $(s, d, 9)$ and node t is a regenerator.

2) **Look-ahead:** Look-ahead refers to finding lower bounds on the weights of the remaining subpath towards the destination in order to predict whether the current subpath will exceed any of the constraints. For multi-constrained routing, this information is built by computing for each metric, the shortest paths tree rooted at the destination node to each node in the network. For the impairment-aware routing problem, we employ two types of look-ahead values for each node, i.e., the length of the node to its nearest regenerator node and the shortest length of the node to the destination node. The former is used to calculate whether the current segment of the given subpath will lead to a length higher than Δ , while the latter is used to assess whether the lower bound on the end-to-end length of the given subpath exceeds $(R + 1)\Delta$, since any feasible path can use a maximum of R regenerators, where R is the total number of regenerator nodes in the network.

B. Exact Impairment-Aware Routing Algorithm (EIARA)

As in Dijkstra's algorithm, our Exact Impairment-Aware Routing Algorithm (EIARA) records information pertaining to subpaths leading up to intermediate nodes in the path selection process. However, for each intermediate node, unlike Dijkstra's algorithm, which stores only a single subpath, EIARA maintains a list of several feasible subpaths with their corresponding impairment values and sets of regenerator nodes. We now describe in detail how EIARA works.

In order to prevent, if possible, the more expensive operations in the latter parts of the code, EIARA calls algorithm

Algorithm 2 EIARA($G, s, d, \mathcal{N}_R, \vec{\Delta}$)

```

1: for  $i = 1, \dots, m$  do
2:    $P_i \leftarrow PIARA(G, s, d, \mathcal{N}_R, \Delta_i)$ 
3:   if  $P_i = NULL$  then
4:     STOP  $\rightarrow$  return no path found!
5:   else
6:      $loop \leftarrow CHECK\_LOOP(P_i)$ 
7:     if (NOT( $loop$ )) then
8:       for  $j = 1, \dots, m$  do
9:          $infeasible \leftarrow CHECK\_CONS(P_i, \Delta_j)$ 
10:        if ( $infeasible$ ) then
11:          goto Step 1
12:        STOP  $\rightarrow$  return ( $P_i$ )
13:
14:   for each  $u \in \mathcal{N}$  do
15:      $counter[u] \leftarrow 0$ 
16:     for  $i = 1, \dots, m$  do
17:        $Rnear_i[u] \leftarrow \min_{v \in \mathcal{N}_R \cup \{d\}} \{r_i(P_{u \rightarrow v}^i)\}$ 
18:        $Rbound_i[u] \leftarrow r_i(P_{u \rightarrow d}^i)$ 
19:
20:   queue  $Q \leftarrow \emptyset$ 
21:    $counter[s] \leftarrow counter[s] + 1$ 
22:   INSERT( $Q, s, counter[s], \vec{0}$ )
23:
24:   while ( $Q \neq \emptyset$ ) do
25:      $P[u, k] \leftarrow EXTRACT-MIN(Q)$ 
26:     if ( $u = d$ ) then
27:       STOP  $\rightarrow$  return  $P[u, k]$ 
28:     else
29:       for each  $v \in adj[u]$  do
30:          $infeasible \leftarrow BACKTRACK(P[u, k], v)$ 
31:         for  $i = 1, \dots, m$  do
32:            $RL'_i \leftarrow r'_i(P[u, k]) + r_i(u, v) + Rnear_i[v]$ 
33:            $RL_i \leftarrow r_i(P[u, k]) + r_i(u, v) + Rbound_i[v]$ 
34:           if ( $RL'_i > \Delta_i$  OR  $RL_i > (R + 1) * \Delta_i$ ) then
35:              $infeasible \leftarrow 1$ 
36:           if (NOT( $infeasible$ )) then
37:              $counter[v] \leftarrow counter[v] + 1$ 
38:             for  $i = 1, \dots, m$  do
39:                $r_i[v[counter[v]]] \leftarrow r_i(P[u, k]) + r_i(u, v)$ 
40:                $r'_i[v[counter[v]]] \leftarrow r'_i(P[u, k]) + r_i(u, v)$ 
41:               if ( $v \in \mathcal{N}_R$ ) then
42:                  $r'_i[v[counter[v]]] \leftarrow 0$ 
43:                $\pi[v[counter[v]]] \leftarrow u[k]$ 
44:             INSERT( $Q, v, counter[v], \vec{RL}$ )

```

PIARA for each impairment metric i in Lines 1-12. If PIARA fails to find a path for metric i , then EIARA exits in Line 4. However, if PIARA returns a path, this path is returned as a feasible solution only if it does not contain loops and satisfies all the other constraints. Hence, the flag *loop* is set to *true* if the path contains loops in Line 6, and *infeasible* is set to *true* if the path fails to satisfy any of the other constraints in Line 9.

For each node u , $Rnear_i[u]$ and $Rbound_i[u]$ are computed in Lines 14-18, which represent the look-ahead lengths of node u to its nearest regenerator node (or destination node) and the destination node, respectively for metric i . The shortest path between two nodes u and v of metric i is denoted as $P_{u \rightarrow v}^i$. In addition, $counter[u]$, which represents the number of subpaths maintained for each node u , is set to zero. In Line 20, the queue Q that stores all the computed subpaths in the network is initialized to an empty set. The path counter of the source node s (i.e., $counter[s]$) is incremented in Line 21, and in Line 22 the subpath that contains only node s is inserted into the queue with a value of 0 for all impairment metrics.

Lines 24-44 search for the solution as long as the queue Q is not empty (otherwise, there is no feasible path). In Line 25, EXTRACT-MIN extracts the best subpath (e.g., the one with the smallest $\max_i \{r_i(P)\}$) in Q . Let the extracted subpath be the k -th subpath of a node u , which is denoted as $P[u, k]$. If node u is the destination node, then subpath $P[u, k]$ is returned as the solution by concatenating the predecessor list π in Line 27. If node u is not the destination node, each node adjacent to node u is considered in Lines 29-44. In Line 30, the function BACKTRACK returns *true* if adjacent node v has already been encountered along this subpath, and *false* otherwise. In Line 32, RL'_i , which is the predicted length from the last regenerator node along the current subpath to the nearest regenerator node of node v in metric i is computed. In Line 33, RL_i , which is the predicted end-to-end length (i.e., source to destination node) in metric i of the current subpath is computed. If a cycle is not detected along the current subpath, and the values of RL'_i and RL_i do not exceed Δ_i and $(R + 1)\Delta_i$, respectively, the path counter of node v is incremented in Line 37. The corresponding information associated with the new subpath, i.e. r_i (the length of the subpath in metric i), r'_i (the length since the last regenerator node in metric i), and π (the predecessor list) are assigned in Lines 38-43. If node v is a regenerator node, the length since the last regenerator node along the current subpath is set to zero in Line 42. This does not necessarily mean that this regenerator node will be used along the current subpath. Instead, after *EIARA* finds the final solution, it identifies the regenerator nodes where regeneration is absolutely necessary. This can be accomplished by only using regenerators that are farthest, but within Δ_i for each $i = 1, \dots, m$, from the previous regenerator or source. Finally, the subpath is inserted into Q in Line 44. Since *EIARA* is essentially a brute-force approach that only prunes paths from the search space (via the look-ahead concept) that are provably infeasible, *EIARA* is guaranteed to be exact.

The complexity of *EIARA* can be computed as follows (disregarding $O(1)$ operations). Lines 1-12 have a complexity of $O(mRL + mRN \log N)$ and the operations in Lines 14-18 have the same complexity. Let k_{\max} be the maximum number of subpaths that are computed for any intermediate node. Then, the queue Q contains at most $k_{\max}N$ subpaths. When using a Fibonacci or Relaxed Heap to structure the queue, selecting the best subpath takes at most $O(\log(k_{\max}N))$ time

[6]. Since each node can be selected at most k_{\max} times from the queue, the EXTRACT-MIN function in Line 25 takes at most $O(k_{\max}N \log(k_{\max}N))$ time. Constructing the path in Line 27 takes at most $O(N)$ time. The **for** loop starting in Line 29 is invoked at most k_{\max} times for each side of each link in the graph, resulting in $O(k_{\max}L)$ time. The BACKTRACK function in Line 30 takes $O(N)$ time and the **for** loop in Line 31 takes $O(m)$ time. Thus, the total running time of Lines 29-44 is $O(k_{\max}N \log(k_{\max}N) + k_{\max}LN + k_{\max}Lm)$. Combining the running times of all the operations in *EIARA* results in the following computational complexity:

$$C_{EIARA} = O(mRN \log N + k_{\max}N \log(k_{\max}N) + k_{\max}LN)$$

C. Heuristics

In this section, we provide two heuristics. Our first heuristic is named *TIARA*, i.e., Tunable Impairment-Aware Routing Algorithm, and it is identical to *EIARA* except that the maximum number of subpaths k_{\max} that can be computed for any node is now bounded by a fixed k that is part of the input³. If $k = 1$, as set in the simulations, the complexity of *TIARA* is $O(mRN \log N + LN)$. The second heuristic is called the Loop Avoidance Heuristic *LAH*.

Algorithm 3 $LAH(G, s, d, \mathcal{N}_R, \vec{\Delta})$

- 1) Create graph $G'(\mathcal{N}', \mathcal{L}')$ such that $\mathcal{N}' = \mathcal{N}_R \cup \{s, d\}$ and $\mathcal{L}' = \emptyset$.
 - 2) For each impairment i :
 - a) For each pair of nodes $u, v \in \mathcal{N}_R \cup \{s, d\}$, $u \neq v$, and link $(u, v) \notin \mathcal{L}'$:
 - i) Find the shortest path $P_{u \rightarrow v}^i$ in graph G using metric i as the cost.
 - ii) For each impairment metric j , if $r_j(P_{u \rightarrow v}^i) > \Delta_j$ go to Step 2 for the next metric.
 - iii) Set $P_{u \rightarrow v}^* = P_{u \rightarrow v}^i$ and add link (u, v) to \mathcal{L}' .
 - b) Set the cost of link (u, v) in G' equal to the number of other such paths that $P_{u \rightarrow v}^*$ shares a segment (i.e., a link or more) with.
 - c) Find the shortest path from s to d in G' and substitute each link (u, v) in the shortest path with the corresponding subpath $P_{u \rightarrow v}^*$ in G to obtain the solution.
 - d) Return the path if it exists and is loop-free, else go to Step 2 for the next metric.
-

Algorithm $LAH(G, s, d, \mathcal{N}_R, \vec{\Delta})$, as in algorithm $PIARA(G, s, d, \mathcal{N}_R, \Delta)$ given earlier, computes the shortest paths between the regenerator nodes (including s and d) in creating graph G' . The difference is that *LAH* tries to avoid loops by assigning link weights in G' that reflect the “criticality” of links, which in this case relates to the number

³The multi-constrained path selection heuristic TAMCRA is analogously derived from its exact counterpart SAMCRA [33].

of paths associated with a link. Other measures of criticality could also be used.

D. Simulation Results

In this subsection, we tested the algorithms under a wide range of instances. Specifically, we present case-by-case comparisons of the three algorithms, where we create thousands of graphs from a particular class, and for each graph we run the algorithms for a single request. The performance metrics that we use to compare the algorithms are the *success ratio* (i.e., the ratio of requests with feasible paths to the total number of requests) and the *average time* an algorithm takes to find a feasible path (only requests for which feasible paths are found by all the three algorithms are considered).

TABLE I
SIMULATION PARAMETERS FOR A GIVEN CARRIER'S BACKBONE NETWORK.

Parameter	Value
$OSNR_{\min}$	20 dB
Average optical power	4 dBm
ASE factor (n_{sp})	2.5
Average span length	75 km
Fiber Loss	0.2 dB/km
$h\nu B_0$	-58 dBm
Amplifier gain (γ)	15 dB
Bit rate	10 Gb/s

We first give simulation results on a carrier's backbone network that has been used in other works [24], [34]. The network has 28 nodes and 43 links. The source and destination nodes of each request are randomly selected. Similarly, the regenerator nodes are randomly chosen. Table I shows the parameters used in this simulation, which are similar to those suggested in [24] and [28]. As in [24], we assume that ASE is the dominant physical impairment. We compare the performance of our three algorithms and a K -shortest path approach that has been used in [18], where at most K shortest paths are computed and the path that fits first is selected. In Figure 6, it can be seen that the K -shortest paths approach performs poorly for small values of K . The success ratio of the K -shortest paths approach becomes comparable to those of our algorithms only for very large values of K (e.g., $K = 100$), making it less suitable for impairment-aware path selection in translucent optical networks.

We have also performed extensive simulations on random and lattice networks. The link weights were uniformly distributed within a scaled and normalized range of $(0, 1]$. In practice, we expect a more positive correlation between links individually and also between the m impairment values of a link, which would simplify the problem. Also, without loss of generality, we assume that the threshold values are equal, i.e., $\Delta_i = \Delta$ for $i = 1, \dots, m$. For each request, the source and destination nodes are randomly assigned. In addition, the regenerator nodes are placed randomly in such a way that there are no two adjacent nodes with regeneration capacity. If two regenerator nodes are directly connected by a link with feasible link weights, then they can be merged to form a "super

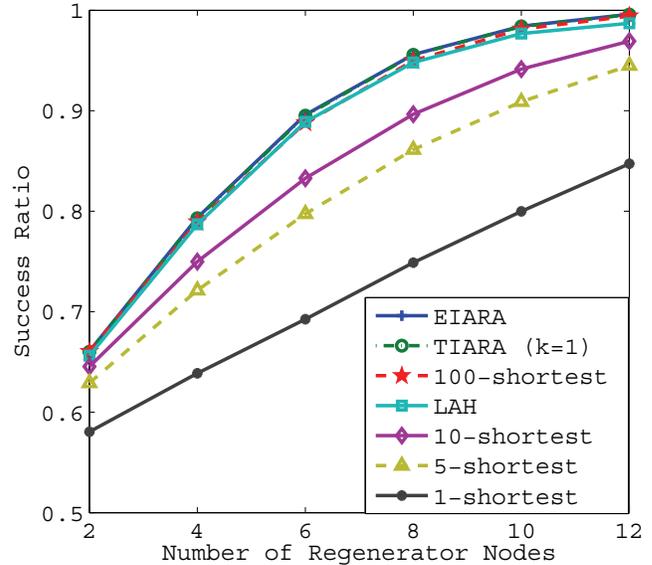


Fig. 6. Case-by-case comparison of the success ratios of our three algorithms and a K -shortest path approach.

regenerator node" without affecting the feasibility of any path. Such preprocessing of the graph could reduce the network size and make the problem relatively easier, which is avoided in our set-up.

We have observed that the results for the lattice networks have been worse than for the random graphs, when the network size and link-weight distribution are the same. The reason is that a larger expected hopcount in lattice networks increases the probability that the impairment thresholds will be violated. We shall only present the results for the lattice networks. We have also observed that *EIARA* is generally fast when a feasible path exists, but in some cases it can take a long time to decide that no solution exists. As noted earlier, we only present the computation times for requests that are accepted by all the three algorithms.

1) *Different impairment thresholds*: Figure 7 shows the success ratios of the three algorithms for lattice networks of $N = 49$ by varying the impairment threshold Δ . It can be seen that an increase in the impairment threshold leads to a relaxation of the problem, and consequently increases the probability that a feasible path exists (and is found). In fact, after $\Delta = 2.4$, both *EIARA* and *TIARA* find feasible paths for all requests.

2) *Different number of regenerator nodes*: Figures 8 and 9 show the success ratios and average times of the three algorithms for lattice networks of $N = 49$ by varying the number of regenerator nodes R . Since increasing the number of regenerator nodes improves the possibility of finding a feasible path, the success ratio grows fast with the number of regenerator nodes. The average time of finding a feasible path also increases with the number of regenerator nodes. This is because, in *EIARA* and *TIARA*, the number of feasible

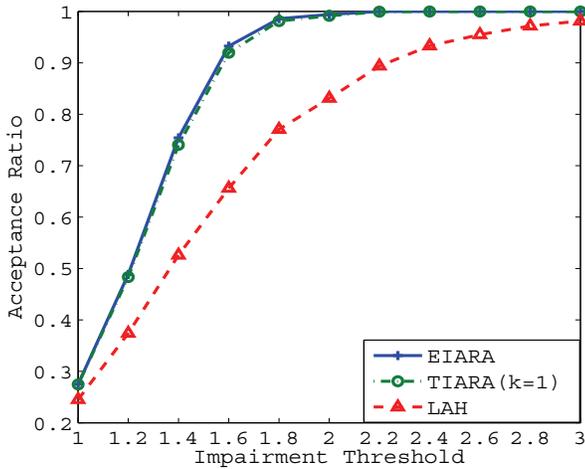


Fig. 7. Case-by-case comparison of the success ratio for lattice networks ($N = 49$) and different impairment thresholds (Δ). (Number of networks = 10000, $R = 12$, and $m = 2$).

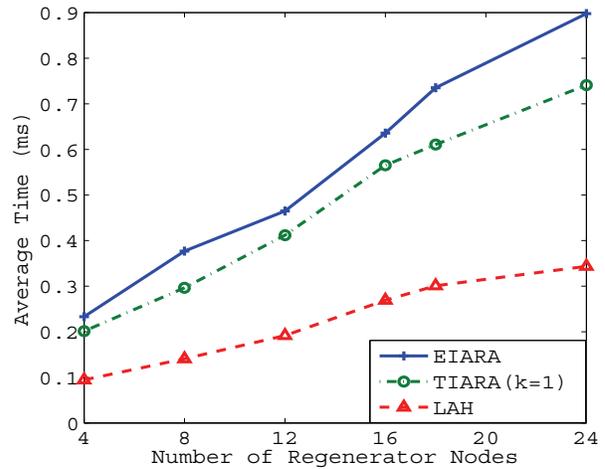


Fig. 9. Case-by-case comparison of the average time for lattice networks ($N = 49$) and different number of regenerator nodes (R). (Number of networks = 10000, $\Delta = 1$, and $m = 2$).

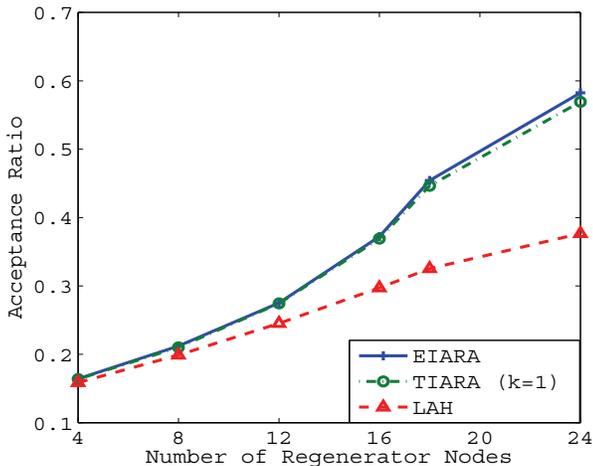


Fig. 8. Case-by-case comparison of the success ratio for lattice networks ($N = 49$) and different number of regenerator nodes (R). (Number of networks = 10000, $\Delta = 1$, and $m = 2$).

subpaths that are considered in the path computation process increases, while in *LAH*, the number of links in \mathcal{L}' increases.

In general, the simulations show that although *LAH* is somewhat faster than *TIARA*, *TIARA* is also fast and always outperforms (often considerably) *LAH*. Moreover, the quality of *TIARA*'s solutions are quite close to the exact solutions of *EIARA*. Hence, *TIARA* is our preferred choice, offering close-to-optimal performance within a reasonable

computational complexity.

V. THE REGENERATOR PLACEMENT PROBLEM

In Section IV we considered online requests arriving to a network with regenerators in place. In this section we shall consider the offline case, where all the source and destination nodes are known in advance. This case also allows for optimally allocating regenerators, which will be the focus of this section. Moreover, we argue that all nodes in a network are already regularly maintained, and minimizing the number of regenerator nodes due to maintenance costs is therefore of less importance than minimizing the number of regenerators. We assume that only the network and a set of requests are given. As in [5] and [8], we assume that there are enough wavelengths so that the wavelength continuity constraint is relaxed.

Problem 3: The Regenerator Placement Problem: The physical optical network is modeled as a graph $G(\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of N nodes and \mathcal{L} is the set of L links. Associated with each fiber link $(u, v) \in \mathcal{L}$ are m physical impairments $r_i(u, v)$, $i = 1, \dots, m$. Given are a set of requests with request j represented by the tuple $(s_j, d_j, \vec{\Delta})$, where $s_j, d_j \in \mathcal{N}$ are the source and destination nodes of request j and $\vec{\Delta} = \{\Delta_1, \dots, \Delta_m\}$ represents m threshold values for the m physical impairments. The regenerator placement problem is to minimize the total number of regenerators such that each request is assigned a simple path that does not exceed the respective thresholds on any of its regeneration segments.

We note that the above problem, namely minimizing the total number of regenerators, is different than that of minimizing the number of regenerator nodes in a network. The latter is shown to be NP-hard [8] for the case $m = 1$.

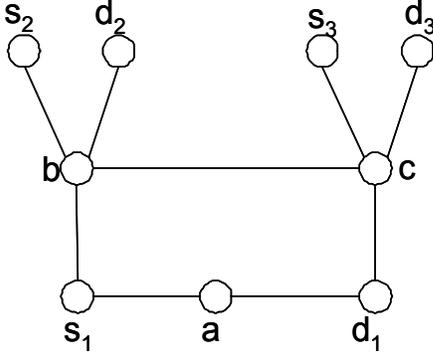


Fig. 10. An example network where minimizing the number of regenerator nodes does not result in the minimum number of regenerators.

The difference between the two problems is illustrated in the example depicted in Figure 10 for $m = 1$. In this example, there are three requests (s_1, d_1, Δ) , (s_2, d_2, Δ) and (s_3, d_3, Δ) , and each link has a cost that is equal to the impairment threshold Δ . If the objective is to minimize the number of regenerator nodes, the regenerators are placed at nodes b and c . The total number of regenerators needed in this case is four, i.e., one for (s_2, d_2, Δ) and one for (s_1, d_1, Δ) at node b , and one for (s_3, d_3, Δ) and one for (s_1, d_1, Δ) at node c . However, when the objective is minimizing the total number of regenerators, we need only three regenerators, i.e., one for (s_1, d_1, Δ) at node a , one for (s_2, d_2, Δ) at node b , and one for (s_3, d_3, Δ) at node c . Since the total regeneration cost in a network depends mainly on the number of regenerator nodes needed, our approach leads to a cheaper solution than that of minimizing the number of regenerator nodes in the network.

Since at a given time, a regenerator can only be used by a single lightpath, the allocation of regenerators to each request can be considered independently. Thus, the regenerator placement problem is reduced to solving a regenerator placement problem for individual requests. For each request, the problem is then to assign a feasible simple path by allocating the necessary regenerators, while minimizing the total number of regenerators needed. We call this problem the Single Request Regenerator Placement (SRRP) problem. For $m > 1$, the SRRP problem is obviously NP-complete (it includes the multi-constrained path problem), however we show that the problem is polynomially solvable for $m = 1$. We first provide an exact algorithm called *ESRRP*, and subsequently prove its exactness. In Step 1 of *ESRRP*, a multi-constrained path is computed between each pair of nodes in the network. For $m > 1$, an exact multi-constrained path algorithm, such as SAMCRA [33], can be employed. However, for $m = 1$, the *ESRRP* algorithm is polynomial, since Step 1 is basically finding the shortest paths between all pairs of nodes.

The notations used in the following theorems are given in the *ESRRP* algorithm.

Theorem 2: The minimum number of regenerators required by any path from s to d is at least $R = \ell(P'_{s \rightarrow d}) - 1$.

Algorithm 4 *ESRRP*(G, s, d, Δ)

- 1) For each pair of node, $u, v \in \mathcal{N}$, find a (shortest) path $\{P_{u \rightarrow v}^*\}$ such that $r_i(P_{u \rightarrow v}^*) \leq \Delta_i$ for $i = 1, \dots, m$.
 - 2) Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' = \{(u, v) \mid r_i(P_{u \rightarrow v}^*) \leq \Delta_i \quad \forall i = 1, \dots, m\}$ and assign a cost of 1 to each link.
 - 3) Find the shortest path $P'_{s \rightarrow d}$ from s to d in G' . Let $\ell(P'_{s \rightarrow d})$ represent the cost/hopcount of path $P'_{s \rightarrow d}$.
 - 4) Substitute the links of $P'_{s \rightarrow d}$ with the corresponding subpaths $P_{u \rightarrow v}^*$ in G to obtain $P_{s \rightarrow d}$.
 - 5) Remove all loops in $P_{s \rightarrow d}$ to obtain the optimal solution.
-

Proof: We prove by contradiction. Assume that there is a path $P''_{s \rightarrow d}$ in G from s to d that needs only $k < R$ regenerators and let $\{n_1, n_2, \dots, n_k\}$ be the nodes (in that order) in $P''_{s \rightarrow d}$ where the regenerators are placed. Hence, there should be links $(s, n_1), (n_1, n_2), \dots, (n_{k-1}, n_k), (n_k, d)$ in graph G' . Then, there is a path from s to d through these nodes with a hopcount of $\ell(P''_{s \rightarrow d}) = k + 1 < \ell(P'_{s \rightarrow d})$ in graph G' , which in turn implies $P''_{s \rightarrow d}$ is not the shortest path in G' . ■

Corollary 1: A loopless path that uses only R regenerators can be obtained from $P'_{s \rightarrow d}$.

Proof: Consider a loop in path $P'_{s \rightarrow d}$ starting and ending at a given node b . Let a and c represent the nearest regenerator nodes at both sides of node b outside the loop. Thus, $r_i(P_{a-b}) \leq \Delta_i$ and $r_i(P_{b-c}) \leq \Delta_i$ for $i = 1, \dots, m$. If $r_i(P_{a-b}) + r_i(P_{b-c}) \leq \Delta_i$ for $i = 1, \dots, m$, then removing the loop will not affect the feasibility of the path and the total number of regenerators required. But if $r_i(P_{a-b}) + r_i(P_{b-c}) > \Delta_i$ for any physical impairment i , then there is a regenerator in the loop (the regenerator can be either at b or at any other node in the loop). Then, placing the regenerator at node b and removing the loop will not affect the feasibility of the path and the total number of regenerators required. This process is repeated until all the loops in $P'_{s \rightarrow d}$ are removed in order to obtain a loopless path of R regenerators. ■

For $m > 1$, the complexity of *ESRRP* is entirely dominated by the multi-constrained path computation in Step 1. For example, if SAMCRA is used, the complexity of *ESRRP* will be $O(k_{\max} N \log(k_{\max} N) + k_{\max}^2 m L)$, where k_{\max} is the maximum number of paths that are computed for any node. On the other hand, for $m = 1$, the complexity of *ESRRP* is determined by three major operations in the algorithm: constructing graph G' , finding the shortest path between s and d in G' , and removing the loops of the path in G . The construction of graph G' involves finding the shortest paths from each node to all other nodes. This can be implemented with $O(N^2 \log N + NL)$ complexity using Johnson's algorithm [14]. Finding the shortest path in G' can be implemented using a Breadth First Search (BFS) with $O(L')$ complexity, which is $O(N^2)$ for dense graphs. The path obtained in G' has $O(N)$ hopcount in the worst case and at each node there can be a loop with $O(N)$ hopcount in G , thus the total complexity of removing loops is $O(N^2)$.

Therefore, the total complexity of *ESRRP* for $m = 1$ is $O(N^2 \log N + NL)$.

VI. CONCLUSIONS

In optical networks, physical impairments, such as noise and signal distortions, degrade the quality of the signal. These impairments become more severe with distance and bit rate, unless the signal is regenerated timely. Since regenerators are costly, they are generally sparsely deployed, and such networks are called translucent networks. In this paper we have studied two important problems pertinent to routing in translucent networks, namely: (1) the problem of selecting a path that meets one or more impairment constraints, and (2) the problem of placing regenerators for a given set of requests. We have shown that the impairment-aware path selection problem is NP-complete, even for the case of a single impairment. Subsequently, we have provided an exact algorithm *EIARA* and heuristic algorithms. Through simulations we have demonstrated that our heuristic *TIARA* is computationally efficient and offers close-to-optimal solutions.

In the regenerator placement problem, we have focused on minimizing the total number of regenerators needed, instead of minimizing the number of regenerator nodes, which was the focus of previous studies. Our motivation has been that the number of regenerators directly affects the cost associated with regeneration. We have shown that the problem is polynomially solvable for $m = 1$ and is NP-complete for $m > 1$, and we have established an exact algorithm (*ESRRP*) for it.

REFERENCES

- [1] M. Ali Ezzahdi, S. Al Zahr, M. Koubaa, N. Puech, and M. Gagnaire, "LERP: a quality of transmission dependent heuristic for routing and wavelength assignment in hybrid WDM networks," *Proc. of ICCCN 2006*, pp. 125–136, 2006.
- [2] S. Azodolmolky, M. Klinkowski, E. Marin, D. Careglio, J.S. Pareta, and I. Tomkos, "A survey on physical layer impairments aware routing and wavelength assignment algorithms in optical networks," *Computer Networks*, vol. 53, no. 7, pp. 926–944, 2009.
- [3] A. Bley, "On the complexity of vertex-disjoint length-restricted path problems," *Computational Complexity*, vol. 12, pp. 131–149, 2003.
- [4] T.J. Carpenter, R.C. Menendez, D.F. Shallcross, J.W. Gannett, J. Jackel, and A.C. Von Lehmen, "Cost-conscious impairment-aware routing," *Proc. of IEEE/OSA OFC 2004*, Feb. 2004.
- [5] S. Chen, I. Ljubic, and S. Raghavan, "The regenerator location problem," *Networks*, vol. 55, no. 3, pp. 209–220, Dec. 2009.
- [6] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd edition, MIT Press, Cambridge, MA, 2001.
- [7] T. Deng and S. Subramaniam, "Adaptive QoS routing in dynamic wavelength-routed optical networks," *Proc. of 2nd International Conference on Broadband Networks*, pp. 184–193, 2005.
- [8] M. Flammini, A.M. Spaccamela, G. Monaco, L. Moscardelli, and S. Zaks, "On the complexity of the regenerator placement problem in optical networks," *Proc. of SPAA'09*, pp. 154–162, 2009.
- [9] M.R. Garey and D.S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
- [10] R. Hayashi, K. Shimizu, I. Inoue, and K. Shiimoto, "Novel impairment-aware traffic engineering in all-optical networks," *Proc. of PACRIM'09*, pp. 152–157, 2009.
- [11] J. He, M. Brandt-Pearce, Y. Pointurier, and S. Subramaniam, "QoS-aware routing in impairment-constrained optical networks," *Proc. of IEEE GLOBECOM 2007*, pp. 2269–2274, 2007.
- [12] Y. Huang, J.P. Heritage, and B. Mukherjee, "Connection provisioning with transmission impairment consideration in optical WDM networks with high-speed channels," *Journal of Lightwave Technology*, vol. 23, no. 3, pp. 982–993, March 2005.
- [13] A. Itai, Y. Perl, and Y. Shiloach, "The complexity of finding maximum disjoint paths with length constraints," *Networks*, vol. 12, no. 3, pp. 277–286, 1982.
- [14] D.B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, no. 1, pp. 1–13, Jan. 1977.
- [15] K.M. Katrinis, A. Tzanakaki, and G. Markidis, "Impairment-aware WDM network dimensioning with optimized regenerator placement," *National Fiber Optic Engineers Conference (NFOEC)*, March 22, 2009.
- [16] S.-W. Kim and S.-W. Seo, "Regenerator placement algorithms for connection establishment in all-optical networks," *Proc. of IEE Communications*, vol. 148, issue 1, pp. 25–30, Feb. 2001.
- [17] F.A. Kuipers and P. Van Mieghem, "Conditions that impact the complexity of QoS routing," *IEEE/ACM Transaction on Networking*, vol. 13, no. 4, pp. 717–730, August 2005.
- [18] G. Markidis, S. Sygletos, A. Tzanakaki, and I. Tomkos, "Impairment aware based routing and wavelength assignment in transparent long haul networks," *Proc. of IFIP ONDM*, pp. 48–57, May 2007.
- [19] A. Marsden, A. Maruta, and K. Kitayama, "FWM-aware dynamic routing and wavelength assignment for wavelength-routed optical networks," *IEICE Transaction of Communication*, vol. E91-B, no. 7, pp. 2145–2151, July 2008.
- [20] D. Monoyios, K. Vlachos, M. Aggelou, and I. Tomkos, "On the use of multi-objective optimization algorithms for solving the Impairment Aware RWA problem," *Proc. of ICC'09*, 2009.
- [21] P.M. Pachnicke, S. Paschenda, and T. Krummrich, "Physical impairment based regenerator placement and routing in translucent optical networks," *Proc. of the Optical Fiber communication/National Fiber Optic Engineers Conference, OFC/NFOEC 2008*, pp. 1–3, Feb. 2008.
- [22] G.S. Pavani, L.G. Zuiliani, H. Waldman, and M.F. Magalhães, "Distributed approaches for impairment-aware routing and wavelength assignment algorithms in GMPLS Networks," *Computer Networks*, vol. 52, no. 10, pp. 1905–1915, July 2008.
- [23] C. Pinart and H. Harai, "On using optical-layer link information parameters in distributed impairment constraint-based routing," *Proc. of the Optical Fiber communication/National Fiber Optic Engineers Conference, OFC/NFOEC 2007*, pp. 1–3, March 2007.
- [24] S. Rai, C.-F. Su, and B. Mukherjee, "On provisioning in all-optical networks: an impairment-aware approach," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1989–2001, 2009.
- [25] G. Shen, W.D. Grover, T.H. Cheng, and S.K. Bose, "Sparse placement of electronic switching nodes for low blocking in translucent optical networks," *OSA Journal of Optical Networking*, vol. 1, no. 12, pp. 424–441, Dec. 2002.
- [26] N. Shinomiya, T. Hoshida, Y. Akiyama, H. Nakashima, and T. Terahara, "Hybrid link/path-based design for translucent photonic network dimensioning," *J. Lightwave Technol.* vol. 25, no. 10, pp. 2931–2941, Oct. 2007.
- [27] K. Sriram, D. Griffith, R. Su, and N. Golmie, "Static vs. dynamic regenerator assignment in optical switches: models and cost trade-offs," *Proc. of High Performance Switching and Routing, HPSR 2004*, pp. 151–155, 2004.
- [28] J. Strand, A.L. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 81–87, Feb. 2001.
- [29] J. Strand and A. Chiu, "Impairments and Other Constraints on Optical Layer Routing," RFC 4054, May 2005.
- [30] X. Yang and B. Ramamurthy, "Sparse regeneration in translucent wavelength-routed optical networks: architecture, network design and wavelength routing," *Photonic Network Communications*, vol. 10, no. 1, July 2005.
- [31] X. Yang and B. Ramamurthy, "Dynamic routing in translucent WDM optical networks," *Proc. of ICC'02*, vol. 5, pp. 2796–2802, 2002.
- [32] E. Yetginer and E. Karasan, "Regenerator placement and traffic engineering with restoration in GMPLS networks," *Photonic Network Communications*, vol. 6, no. 2, pp. 139–149, Sept. 2003.
- [33] P. Van Mieghem and F.A. Kuipers, "Concepts of exact quality of service algorithms," *IEEE/ACM Transaction on Networking*, vol. 12, no. 5, pp. 851–864, Oct. 2004.
- [34] K. Zhu, H. Zang, and B. Mukherjee, "A comprehensive study on nextgeneration optical grooming switches," *IEEE J. of Selected Areas in Commun.*, vol. 21, pp. 1173–1186, Sep. 2003.
- [35] S. Zsigmond, G. Németh, and T. Cinkler, "Mutual impact of physical impairments and grooming in multilayer networks," *Proc. of IFIP ONDM 2007*, pp. 38–47, May 2007.