Placing Regenerators in Optical Networks to Satisfy Multiple Sets of Requests

George B. Mertzios
School of Engineering and Computing Sciences, Durham University, UK.
Ignasi Sau
AlGCo group, CNRS, LIRMM, Montpellier, France.
Mordechai Shalom
TelHai College, Upper Galilee, 12210, Israel.
Shmuel Zaks
Department of Computer Science, Technion, Haifa, Israel.

***Abstract*—The placement of regenerators in optical networks has become an active area of research during the last years. Given a set of lightpaths in a network G and a positive integer d, regenerators must be placed in such a way that in any lightpath there are no more than d hops without meeting a regenerator. The cost function we consider is given by the *total* number of regenerators placed at the nodes, which we believe to be a more accurate estimation of the real cost of the network than the number of *locations* considered in *[Flammini et al., IEEE/ACM Trans. Netw., 2011]*. Furthermore, in our model we assume that we are given a finite set of p possible traffic patterns (each given by a set of lightpaths), and our objective is to place the minimum number of regenerators at the nodes so that each of the traffic patterns is satisfied. While this problem can be easily solved when $d = 1$ or $p = 1$, we prove that for any fixed $d, p \geq 2$ it does not admit a PTAS, even if G has maximum degree at most 3 and the lightpaths have length $\mathcal{O}(d)$. We complement this hardness result with a constant-factor approximation algorithm with ratio $\ln(d \cdot p)$. We then study the case where G is a path, proving that the problem is polynomial-time solvable for two particular families of instances. Finally, we generalize our model in two natural directions, which allows us to capture the model of *[Flammini et al., IEEE/ACM Trans. Netw., 2011]* as a particular case, and we settle some questions that were left open therein.**

***Index Terms*—Optical networks, regenerators, over-provisioning, approximation algorithms, hardness of approximation.**

A preliminary conference version of this work appeared in the *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP), volume 6199 of LNCS, pages 333-344, Bordeaux, France, July 2010. Best paper award of Track C*. This research was supported by the Israel Science Foundation (grant No. 1249/08) and by the British Council (grant No. UKTELHAI09).

I. Introduction

A. Background

In modern optical networks, high-speed signals are sent through optical fibers using WDM (Wavelength Division Multiplexing) technology. Networks with each fiber typically carrying around 80 wavelengths are operational, whereas networks with a few hundreds wavelengths per fiber are already experimental. As the energy of the signal decreases with the traveled distance, optical amplifiers are required every some fixed distance (a typical value being around 100 km). However, optical amplifiers introduce noise into the signal, so after a certain number of amplifications, the optical signal needs to be regenerated in order to keep the SNR (Signal-to-Noise Ratio) above a specified threshold. In current technology, the signal is regenerated as follows. An ROADM (Reconfigurable Optical Add-Drop Multiplexer) has the capability of inserting/extracting a given number of wavelengths (typically, around 4) to/from the optical fiber. Then, for each extracted wavelength, an optical regenerator is needed to regenerate the signal carried by that wavelength. That is, at a given optical node, one needs as many regenerators as wavelengths one wants to regenerate. See Fig. 1 for a simplified illustration of the aforementioned devices in the case when the network is a path and the fiber carries 3 wavelengths.

The problem of placing regenerators in optical networks has attracted the attention of several recent research works [1]–[8]. Mostly, these articles propose heuristics and run simulations in order to reduce the

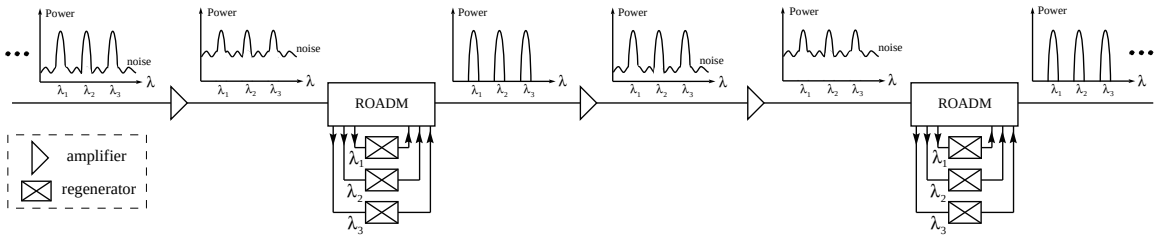

Fig. 1. A simplified optical network: amplifiers introduce noise into the signal, which needs to be regenerated after at most $d = 3$ hops. When the signal is regenerated through an ROADM, a different regenerator is needed for each wavelength.

number of regenerators, but no theoretical analysis is presented. Recently, the first theoretical study of the problem has been done by Flammini *et al*. in [9]. In the next paragraph we discuss how our model differs from the one studied in [9].

Nowadays the cost of a regenerator is considerably higher than the cost of an ROADM (as an example, $\$160$K vs $\$50$K). Moreover, the regenerator cost is per wavelength, as opposed to ROADM cost that is payed once per several wavelengths. Therefore the *total* number of regenerators seems to be the right cost to minimize. Another possible criterion is to minimize the number of *locations* (that is, the number of nodes) in which optical regenerators are placed. This measure is the one assumed in [9], which makes sense when the dominant cost is given by the set-up of new optical nodes, or when the equipment to be placed at each node is the same for all nodes. Nevertheless, the total number of regenerators seems to be a more accurate estimate of the real cost of the network, and therefore we consider this cost in this article.

It is worth mentioning here that when all the connection requests are known a priori, minimizing the number of regenerators is an easy task. Indeed, suppose that the maximum number of hops a lightpath can make without meeting a regenerator is an integer d (in the example of Fig. 1, we have $d = 3$). Then, for each lightpath ℓ, we need to place one regenerator every d consecutive vertices in ℓ, to get an optimal solution.

Unfortunately, when designing a network, it is usually the case that the traffic requests are not known in advance. For instance, the traffic in a given network may change dramatically depending on whether in the foreseeable future an Internet supplier or an email storage server opens or closes a site within the area of the network. In such a situation of uncertain traffic forecast, a common approach in order to minimize capital expenses is to predeploy (or overprovision) resources [10]–[13]. That is, the network is designed to satisfy several possible traffic patterns. A similar setting arises in networks in which there are several possible traffic configurations that alternate according to some phenomena, like the weather, the season, an overflow of the capacity of another network, or a breakdown. In that case, the network must be designed so that it can satisfy each of the traffic configurations independently.

In our model, we assume that we are given a finite set of p possible traffic patterns (each given by a set of lightpaths), and our objective is to place the minimum total number of regenerators at the nodes so that each of the traffic patterns is satisfied. That is, the number of regenerators that must be placed at a node of the network is the maximum of the number of regenerators needed by any of the traffic patterns at that node. We aim at minimizing the total number of regenerators placed at the network. We formally define the problem in Section I-B.

B. Definitions

Given an undirected underlying graph $G = (V, E)$ that corresponds to the network topology, a *lightpath* is a simple path in G. That is, we assume that the routing of the requests is given (see [9] for complexity results when the routing of the requests is not given). We also assume that lightpaths sharing an edge use different wavelengths. That is, we deal with optical networks without traffic grooming [14].

The *length* of a lightpath is the number of edges it contains. We consider *symmetric* lightpaths, that is, a lightpath with endpoints u and v consists of a request from u to v and a request from v to u. The *internal vertices* (resp. *edges*) of a lightpath or

a path ℓ are the vertices (resp. edges) in ℓ different from the first and the last one. Given an integer d, a lightpath ℓ is *d-satisfied* if there are no d consecutive internal vertices in ℓ without a regenerator. A set of lightpaths is *d-satisfied* if each of its lightpaths is d-satisfied. Given p sets of lightpaths $L_1, \ldots, L_p$, with $L_i = \{\ell_{i,j} \mid 1 \leq j \leq x_i\}$ (that is, x_i is the number of lightpaths in the set L_i), we consider the union of all lightpaths in the p sets $\cup L_i = \{\ell_{i,j} \mid 1 \leq i \leq p, 1 \leq j \leq x_i\}$. An *assignment* of regenerators is a function $\text{reg} : V \times \cup L_i \to \{0, 1\}$, where $\text{reg}(v, \ell) = 1$ if and only if a regenerator is used at vertex v by lightpath ℓ.

We study the following problem: given $p \geq 1$ sets of lightpaths, and a distance $d \geq 1$, determine the smallest number of regenerators that d-satisfy each of the p sets. Formally, for two fixed integers $d, p \geq 1$, the optimization problem we study is defined as follows.

(d,p)-TOTAL REGENERATORS ((d,p)-TR)
Input: An undirected graph $G = (V, E)$ and p sets of lightpaths $\mathcal{L} = \{L_1, \ldots, L_p\}$.
Output: A function $\text{reg} : V \times \cup L_i \to \{0, 1\}$ s.t. each lightpath in $\cup L_i$ is d-satisfied.
Objective: Minimize $\sum_{v \in V} \text{reg}(v)$, where $\text{reg}(v) = \max_{1 \leq i \leq p} \sum_{\ell \in L_i} \text{reg}(v, \ell)$.

Note that, as mentioned in Section I-A, in the case $p = 1$ (that is, when there is a single set of requests) the problem is trivially solvable in polynomial time, as the regenerators can be placed for each lightpath independently. The case $d = 1$ is not interesting either, as for each internal vertex $v \in V$ and each $\ell \in \cup L_i$, $\text{reg}(v, \ell) = 1$, so there is only one feasible solution, which is optimal.

C. Our contribution

In this article we provide hardness results and approximation algorithms for the (d,p)-TOTAL REGENERATORS problem ((d,p)-TR for short). We first prove in Section III that for any two fixed integers $d, p \geq 2$, (d,p)-TR does not admit a PTAS (see definition in Section II) unless P = NP, even if the underlying graph G has maximum degree at most 3, and the lightpaths have length at most $2d$. In Section IV we complement this hardness result with a constant-factor approximation algorithm with ratio $\min\{p, H_{d \cdot p} - 1/2\}$, where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ is the n-th harmonic number. Section V is devoted to the case where the underlying graph is a path: we prove that (d,p)-TR is polynomial-time solvable in paths when all the lightpaths share the first (or the last) edge, as well as when the maximum number of lightpaths sharing an edge is bounded. In Section VI we generalize the model presented in Section I-B in two natural directions. This generalization allows us to capture the model of [9] as a particular case, and to settle some complexity issues that were left open in [9]. (Since we need some further definitions, we defer the precise statement of these results to Section VI.) Finally, in Section VII we conclude the article and present a number of interesting avenues for further research. We first provide in Section II some standard preliminaries.

II. PRELIMINARIES

We use standard terminology concerning graphs, complexity, and algorithms; see for instance [15]–[17], respectively.

Graphs. All the graphs considered in this article are simple and undirected. Given a graph G we denote by $V(G)$ and $E(G)$ the sets of vertices and edges of G, respectively. If H is a subgraph of G, we denote it by $H \subseteq G$. Given a graph G and $F \subseteq E(G)$, we denote by $G[F]$ the subgraph of G induced by the edges in F together with their endpoints. Given a subset $S \subseteq V(G)$, we define $N_G[S]$ to be the set of vertices of $V(G)$ at distance at most 1 from at least one vertex of S. If $S = \{v\}$, we simply use the notation $N_G[v]$. We also define $N_G(v) = N_G[v] \setminus \{v\}$. The *degree* of a vertex $v \in V(G)$ is defined as $\deg_G(v) = |N_G(v)|$. A graph is *cubic* if all its vertices have degree 3. The *maximum degree* of G is defined as $\Delta(G) = \max_{v \in V(G)} \deg_G(v)$. A *matching* in a graph is a set of disjoint edges, and a *vertex cover* is a set of vertices that contains at least one endpoint of every edge. The *girth* of a graph is the length of a shortest cycle. Given an edge $e = \{u, v\}$, by *subdividing* e we denote the operation of deleting the edge $e = \{u, v\}$, adding a new vertex w, and making it adjacent to both u and v.

Complexity and approximation algorithms. Given an NP-hard minimization problem Π, we

say that a polynomial-time algorithm $\mathcal{A}$ is an α-approximation algorithm for Π, with $\alpha \geq 1$, if for any instance of Π, algorithm $\mathcal{A}$ finds a feasible solution with cost at most α times the cost of an optimal solution. For instance, a maximal matching constitutes a 2-approximation algorithm for the MINIMUM VERTEX COVER problem. In complexity theory, the class APX (Approximable) contains all NP-hard optimization problems that can be approximated within a constant factor. The subclass PTAS (Polynomial Time Approximation Scheme) contains the problems that can be approximated in polynomial time within a ratio $1+\varepsilon$ for *any* fixed $\varepsilon > 0$. In some sense, these problems can be considered to be *easy* NP-hard problems. Since, assuming P $\neq$ NP, there is a strict inclusion of PTAS in APX (for instance, MINIMUM VERTEX COVER $\in$ APX $\setminus$ PTAS), an APX-hardness result for a problem implies the non-existence of a PTAS unless P = NP.

III. HARDNESS RESULTS FOR GENERAL GRAPHS

In this section we prove that, unless P = NP, (d,p)-TR does not admit a PTAS for any $d, p \geq 2$, even if the underlying graph G has maximum degree at most 3 and the lightpaths have length $\mathcal{O}(d)$. Before this, we need two technical results to be used in the reductions.

MINIMUM VERTEX COVER is known to be APX-hard in cubic graphs [18]. By a simple reduction, we prove in the following lemma that MINIMUM VERTEX COVER is also APX-hard in a class of graphs with degree at most 3 and high girth, which will be used in the proofs of Proposition 1 and Theorem 1.

Lemma 1: MINIMUM VERTEX COVER is APX-hard in the class of graphs $\mathcal{H}$ obtained from cubic graphs by subdividing each edge twice.

Proof: Given a cubic graph G, let H the graph obtained from G by subdividing each edge twice. That is, each edge $\{u,v\}$ gets replaced by 3 edges $\{u,u_e\}$, $\{u_e,v_e\}$, and $\{v_e,v\}$, where u_e, v_e are two new vertices. We now claim that

$$\text{OPT}_{\text{VC}}(H) \;=\; \text{OPT}_{\text{VC}}(G) + |E(G)| \;, \qquad (1)$$

where OPT_{VC} indicates the size of a minimum vertex cover. Indeed, let $S_G \subseteq V(G)$ be a vertex cover of G. We proceed to build a vertex cover S_H of H of size $|S_G| + |E(G)|$. First, include in S_H all the vertices in S_G. Then, for each 3 edges $\{u,u_e\}$, $\{u_e,v_e\}$, and $\{v_e,v\}$ of H corresponding to edge $\{u,v\} \in E(G)$, the edge $\{u_e,v_e\}$ is *not* covered by S_G, and at least one of $\{u,u_e\}$ and $\{v_e,v\}$ is covered by S_G. Therefore, adding either u_e or v_e to S_H covers the three edges $\{u,u_e\}$, $\{u_e,v_e\}$, and $\{v_e,v\}$. This procedure defines a vertex cover of H of size $|S_G| + |E(G)|$. Conversely, let $S_H \subseteq V(H)$ be a vertex cover of H, and let us construct a vertex cover S_G of G of size at most $|S_H| - |E(G)|$. We shall see that we can construct S_G from S_H by decreasing the cardinality of S_H by at least one for each edge of G. Indeed, consider the three edges $\{u,u_e\}$, $\{u_e,v_e\}$, and $\{v_e,v\}$ of H corresponding to an edge $e = \{u,v\} \in E(G)$. Note that at least one of u_e and v_e belongs to S_H. If both $u_e, v_e \in S_H$, add either u or v to S_G if none of u, v was already in S_G. Otherwise, if exactly one of u_e and v_e (say, u_e) belongs to S_H, then at least one of u and v must also belong to S_H, and do not add any new vertex to S_G.

Note that as G is cubic, each vertex in a solution S_G covers exactly 3 edges, so $|E(G)| \leq 3 \cdot \text{OPT}_{\text{VC}}(G)$.

In order to prove the lemma, assume for contradiction that there exists a PTAS for MINIMUM VERTEX COVER in $\mathcal{H}$. That is, for any $\varepsilon > 0$, we can find in polynomial time a solution $S_H \subseteq V(H)$ such that $|S_H| \leq (1+\varepsilon) \cdot \text{OPT}_{\text{VC}}(H)$. By the above discussion, we can find a solution $S_G \subseteq V(G)$ such that

$$\begin{aligned} |S_G| &\leq |S_H| - |E(G)| \\ &\leq (1+\varepsilon) \cdot \text{OPT}_{\text{VC}}(H) - |E(G)| \\ &= (1+\varepsilon) \cdot (\text{OPT}_{\text{VC}}(G) + |E(G)|) - |E(G)| \\ &= (1+\varepsilon) \cdot \text{OPT}_{\text{VC}}(G) + \varepsilon \cdot |E(G)| \\ &\leq (1+\varepsilon) \cdot \text{OPT}_{\text{VC}}(G) + 3\varepsilon \cdot \text{OPT}_{\text{VC}}(G) \\ &= (1+4\varepsilon) \cdot \text{OPT}_{\text{VC}}(G) \;, \end{aligned}$$

where we have used Equation (1) and the fact that $|E(G)| \leq 3 \cdot \text{OPT}_{\text{VC}}(G)$. That is, the existence of a PTAS for MINIMUM VERTEX COVER in the class of graphs $\mathcal{H}$ would imply the existence of a PTAS in the class of cubic graphs, which is a contradiction by [18] unless P = NP. ∎

It is known that the edges of any cubic graph can be two-colored such that each monochromatic connected component is a path (of any length) [19]. In fact, solving a conjecture of Bermond *et al.* [20], Thomassen proved [21] a stronger result: the edges

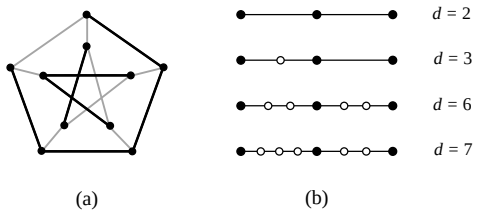

Fig. 2. (a) A two-coloring of the edges of the Petersen graph (grey and black) such that each monochromatic component is a path of length at most 5. (b) Construction of the lightpaths from a path of length 2 for several values of d, in the proof of Theorem 1. Full dots correspond to vertices of the VERTEX COVER instance (called *black* in the proof).

of any cubic graph can be two-colored such that each monochromatic connected component is a path of length at most 5 (see Fig. 2(a) for an example). In addition, the aforementioned colorings can be found in polynomial time [19], [21]. Note that in such a coloring of a cubic graph, each vertex appears exactly once as an endpoint of a path, and exactly once as an internal vertex of another path. We next show that these results can be easily strengthened for the family of graphs $\mathcal{H}$ defined in Lemma 1.

Lemma 2: Let $\mathcal{H}$ be the class of graphs obtained from cubic graphs by subdividing each edge twice. The edges of any graph in $\mathcal{H}$ can be two-colored in polynomial time such that each monochromatic connected component is a path of length at most 2.

Proof: Let $H \in \mathcal{H}$ be a graph obtained from a cubic graph G by subdividing each each twice. That is, edge $\{u, v\}$ of G gets replaced by 3 edges $\{u, u_e\}$, $\{u_e, v_e\}$, and $\{v_e, v\}$ in H. Find a two-coloring of the edges of G such that each monochromatic connected component is a path, using [19] or [21]. To color the edges of H, do the following for each edge $\{u, v\}$ of G: color $\{u, u_e\}$ and $\{v_e, v\}$ with the same color as $\{u, v\}$, and color $\{u_e, v_e\}$ with the other color. It is then easy to check that each monochromatic connected component of the obtained two-coloring of H is a path of length at most 2. ■

We are now ready to announce the main results of this section. For the sake of presentation, we first present in Proposition 1 the result for $d = p = 2$, and then we show in Theorem 1 how to extend the reduction to any fixed $d, p \geq 2$.

Proposition 1: $(2, 2)$-TR does not admit a PTAS unless P = NP, even if G has maximum degree at most 3 and the lightpaths have length at most 4.

Proof: The reduction is from MINIMUM VERTEX COVER (VC for short) in the class of graphs $\mathcal{H}$ obtained from cubic graphs by subdividing each edge twice, which does not admit a PTAS by Lemma 1 unless P = NP. Note that by construction any graph in $\mathcal{H}$ has girth at least 9. Given a graph $H \in \mathcal{H}$ as instance of VC, we proceed to build an instance of $(2, 2)$-TR. We set $G = H$, so G has maximum degree at most 3.

To define the two sets of lightpaths L_1 and L_2, let $\{E_1, E_2\}$ be the partition of $E(H)$ given by the two-coloring of Lemma 2. Therefore, each connected component of $H[E_1]$ and $H[E_2]$ is a path of length at most 2. Each such path in $H[E_1]$ (resp. $H[E_2]$) will correspond to a lightpath in L_1 (resp. L_2), which we proceed to define. A key observation is that, as the paths of the two-coloring have length at most 2, if any endpoint v of such a path P had one neighbor in $V(P)$, it would create a triangle, a contradiction to the fact that the girth of H is at least 9. Therefore, as the vertices of H have degree 2 or 3, any endpoint v of a path P has at least one neighbor in $V(H) \backslash V(P)$.

We are now ready to define the lightpaths. Let P be a path with endpoints u, v, and let u' (resp. v') be a neighbor of u (resp. v) in $V(H) \setminus V(P)$, such that $u' \neq v'$ (such distinct vertices u', v' exist because P has length at most 2 and H has girth at least 9; in fact we only need H to have girth at least 5). The lightpath associated with P consists of the concatenation of $\{u', u\}$, P, and $\{v, v'\}$. Therefore, the length of each lightpath is at most 4. This completes the construction of the instance of $(2, 2)$-TR. Observe that since we assume that $d = 2$, regenerators must be placed in such a way that all the internal edges of a lightpath (that is, all the edges except the first and the last one) have a regenerator in at least one of their endpoints. We can assume without loss of generality that no regenerator serves at the endpoints of a lightpath, as the removal of such regenerators does not alter the feasibility of a solution. Note that in our construction, each vertex of G appears as an internal vertex in at most two lightpaths, one (possibly) in L_1 and the other one (possibly) in L_2, so we can assume that $\text{reg}(v) \leq 1$ for any $v \in V(G)$.

We now claim that $\text{OPT}_{\text{VC}}(H) = \text{OPT}_{(2,2)-\text{TR}}(G, \{L_1, L_2\})$.

Indeed, let first $S \subseteq V(H)$ be a vertex cover of H.

Placing one regenerator at each vertex belonging to S defines a feasible solution to $(2,2)$-TR in G with cost $|S|$, as at least one endpoint of each internal edge of each lightpath contains a regenerator. Therefore, $\text{OPT}_{\text{VC}}(H) \geq \text{OPT}_{(2,2)-\text{TR}}(G, \{L_1, L_2\})$.

Conversely, suppose we are given a solution to (2,2)-TR in G using r regenerators. Since E_1 and E_2 are a partition of $E(G) = E(H)$ and the set of internal edges of the lightpaths in L_1 (resp. L_2) is exactly E_1 (resp. E_2), the regenerators placed at the endpoints of the internal edges of the lightpaths constitute a vertex cover of H of size at most r. Therefore, $\text{OPT}_{\text{VC}}(H) \leq \text{OPT}_{(2,2)-\text{TR}}(G, \{L_1, L_2\})$.

Summarizing, since $\text{OPT}_{\text{VC}}(H) = \text{OPT}_{(2,2)-\text{TR}}(G, \{L_1, L_2\})$ and any feasible solution to $\text{OPT}_{(2,2)-\text{TR}}(G, \{L_1, L_2\})$ using r regenerators defines a vertex cover of H of size at most r, the existence of a PTAS for $(2,2)$-TR would imply the existence of a PTAS for VERTEX COVER in the class of graphs $\mathcal{H}$, which is a contradiction by Lemma 1, unless P = NP. ■

Theorem 1: (d,p)-TR does not admit a PTAS for any $d \geq 2$ and any $p \geq 2$ unless P = NP, even if the underlying graph G satisfies $\Delta(G) \leq 3$ and the lightpaths have length at most $2d$.

Proof: The case $d = p = 2$ was proved in Proposition 1. We next prove the result for $p = 2$ and arbitrary $d \geq 2$. Again, the reduction is from VERTEX COVER in the class of graphs $\mathcal{H}$ defined in Lemma 1. Given a graph $H \in \mathcal{H}$ as instance of VERTEX COVER, we partition $E(H)$ into E_1 and E_2 according to the two-coloring given by Lemma 2.

In order to build G, we associate a *parity* to the edges of H as follows. Recall that the vertices of H have degree 2 or 3. From the set of paths $\mathcal{P}$ given by Lemma 2, we build a set of paths $\mathcal{P}'$ as follows. If a vertex v appears in $\mathcal{P}$ as an endpoint of two paths P_1 and P_2 (necessarily, of different color), we merge them to build a new longer path, and add it to $\mathcal{P}'$. We orient each path $P \in \mathcal{P}'$ arbitrarily, and define the parity of the edges of P accordingly (the first edge being odd, the second even, and so on). This defines the parity of all the edges in $E(H)$.

We now subdivide the edges of H as follows. We distinguish two cases depending on the value of d. For each $P \in \mathcal{P}$:

- If $d \geq 2$ is even, we subdivide $\frac{d}{2} - 1$ times each edge of P (that is, we introduce $\frac{d}{2} - 1$ new vertices for each edge of P).
- If $d \geq 3$ is odd, we subdivide $\frac{d-1}{2}$ times each odd edge of P, and $\frac{d-3}{2}$ times each even edge of P.

This completes the construction of G. Note that $\Delta(G) \leq 3$. We call the vertices of G corresponding to vertices of H *black*, the other ones being *white*. An example of this construction is illustrated in Fig. 2(b) for several values of d in a path P with 2 edges. We now have to define the two sets of lightpaths. Again, each path of $H[E_1]$ (resp. $H[E_2]$) will correspond to a lightpath in L_1 (resp. L_2), but now we have to be more careful with the first and last edges of the lightpaths. Namely, we will construct the lightpaths in such a way that the parities of the corresponding edges of H alternate.

Let P be a path of the two-coloring of $E(H)$ with endpoints u and v. We will argue about u, and the same procedure applies to v. We distinguish two cases according to the degree of u in H. In both cases, we will associate a vertex $u' \in V(H)$ with u. First, if u has degree 2 in H, let u' be the neighbor of u in $V(H) \setminus V(P)$ (recall that $u' \in V(H) \setminus V(P)$ as H has girth at least 9). Note that by the definition of the parity of the edges of H, the edge $\{u', u\}$ has different parity from the edge of P containing u. Otherwise, u has degree 3 in H, and let u' and u'' be the two neighbors of u in $V(H) \setminus V(P)$. By the properties of the two-coloring given by Lemma 2, $\{u', u\}$ and $\{u'', u\}$ are two consecutive edges in a path of the two-coloring, hence they have different parity. Let without loss of generality $\{u', u\}$ have different parity from the edge of P containing u. Equivalently, the same discussion determines another vertex $v' \in V(H)$ associated with v (note that $u' \neq v'$ due to the high girth of H).

Then the lightpath associated with P consists of the concatenation of the edges in G corresponding to $\{u', u\}$, P, and $\{v, v'\}$. This completes the construction of the instance of $(d,2)$-TR. Note that for both d even or odd, the length of the lighpaths is at most $2d$. Note also that the case $d = 2$ is consistent with the proof of Proposition 1. As in the case $d = 2$, we now claim that $\text{OPT}_{\text{VC}}(H) = \text{OPT}_{(d,2)-\text{TR}}(G, \{L_1, L_2\})$.

Let $S \subseteq V(H)$ be a vertex cover of H. Place one regenerator at each black vertex of G corresponding to a vertex in S; this defines a feasible solution to $(d,2)$-TR with cost $|S|$. Indeed, at least one of every 2 consecutive black vertices of each

lightpath ℓ hosts a regenerator, so the maximum distance in a lightpath without meeting a regenerator is bounded by the distance between the first and the last black vertex in a sequence of 3 consecutive black vertices, which is exactly d for both d even and odd (see Fig. 2(b)). Therefore, $\text{OPT}_{\text{VC}}(H) \geq \text{OPT}_{(d,2)-\text{TR}}(G, \{L_1, L_2\})$.

Conversely, given a solution to $(d, 2)$-TR in G using r regenerators, we perform the following transformation to each lightpath ℓ: let v_1 and v_2 be two consecutive black vertices in ℓ, and assume that v_1 is on the left of v_2 in the chosen orientation of the path corresponding to ℓ. If there are any regenerators at the white vertices between v_1 and v_2, we remove them and put a regenerator at v_1, if there was no regenerator before. We perform this operation for any two consecutive black vertices of each lightpath, inductively from right to left. This defines another feasible solution to $(d, 2)$-TR in G using at most r regenerators, since there is no lightpath with two consecutive black vertices without a regenerator. Indeed, if there were two consecutive black vertices without a regenerator after the described transformation, it would imply that the original solution was not feasible, a contradiction. The latter property implies that the regenerators at the black vertices constitute a vertex cover of H of size at most r. Therefore, $\text{OPT}_{\text{VC}}(H) \leq \text{OPT}_{(d,2)-\text{TR}}(G, \{L_1, L_2\})$.

That is, the existence of a PTAS for $(d, 2)$-TR would imply the existence of a PTAS for VERTEX COVER in the class of graphs $\mathcal{H}$, which is a contradiction by Lemma 1, unless P = NP.

For $p \geq 2$, it suffices to further refine in an arbitrary way the partition of $E(H)$ given by Lemma 2 into p sets of edges, which correspond to the p sets of lightpaths. For instance, if $p = 5$, we can partition $E(H_1)$ (resp. $E(H_2)$) into 2 (resp. 3) sets of paths. Then, the same proof presented above carries over to any $p \geq 2$. ■

IV. APPROXIMATION ALGORITHMS FOR GENERAL GRAPHS

We have seen in Section III that (d, p)-TR does not admit a PTAS for $d, p \geq 2$ unless P = NP. In this section we complement this result with a constant-factor approximation algorithm for (d, p)-TR in general graphs.

Theorem 2: For any fixed $d, p \geq 2$, there is a polynomial-time approximation algorithm for the (d, p)-TR problem with ratio $\min\{p, H_{d \cdot p} - 1/2\}$, where $H_{d \cdot p} = \sum_{i=1}^{d \cdot p} \frac{1}{i}$.

Proof: As mentioned in Section I, $(d, 1)$-TR is trivially solvable in polynomial time for any $d \geq 1$. Note also that any solution to (d, p)-TR must d-satisfy all the lightpaths in L_i, for $1 \leq i \leq p$. Therefore, for any instance $(G, \mathcal{L} = \{L_1, ..., L_p\})$, $OPT_{(d,p)-\text{TR}}(G, \mathcal{L}) \geq \max\{OPT_{(d,1)-\text{TR}}(G, L_i); 1 \leq i \leq p\}$. That is, $\sum_{i=1}^{p} OPT_{(d,1)-\text{TR}}(G, L_i) \leq p \cdot \max\{OPT_{(d,1)-\text{TR}}(G, L_i); 1 \leq i \leq p\} \leq p \cdot OPT_{(d,p)-\text{TR}}(G, \mathcal{L})$, so the union of the optima to the p instances defined by the sets $L_i, 1 \leq i \leq p$, constitutes a p-approximation to (d, p)-TR.

A better approximation ratio for most values of d and p can be obtained by reducing (d, p)-TR to MINIMUM SET COVER. The algorithm is simple, so for the sake of intuition we provide a high-level description rather than a technical one. Indeed, the universe of objects to be covered is the union of the edges of the lightpaths in the sets $L_1, \ldots, L_p$. These edges are covered by regenerators placed at vertices. Since we assume that the traffic requests are symmetric, each lightpath can be arbitrarily oriented, and then each regenerator covers edges only on one side. Namely, each regenerator covers d edges of at most p lightpaths, each lighpath belonging to a different set L_i. Each set of the MINIMUM SET COVER instance is made of a vertex $v \in V(G)$ together with a choice of the d covered edges of at most one lightpath from each L_i that goes through v. The number of possible sets for each v is at most $\prod_{i=1}^{p} |L_i|$, so the total number of sets is at most $|V(G)| \cdot \prod_{i=1}^{p} |L_i|$, which is polynomial in the input size since p is fixed. Each such set has cost 1, which corresponds to the cost of a regenerator placed at a vertex and used by these lightpaths. Clearly, a set cover of minimum cost corresponds to a placement of the minimum number of regenerators d-satisfying all the lightpaths. The algorithm of [22] achieves an approximation ratio of $H_k - 1/2$, where k is the maximum size of a set and $H_k = \sum_{i=1}^{k} \frac{1}{i}$. In our case, each regenerator covers at most d edges of at most p lightpaths, so the size of the sets is at most $d \cdot p$. Therefore, the approximation ratio of the algorithm is at most $H_{d \cdot p} - 1/2$. ■

Note that for big d, p, $H_{d \cdot p} \approx \ln d + \ln p + 1/2$, so comparing both approximation ratios, we have that $p < \ln d + \ln p$ when $d = \Omega(2^p)$.

V. The case of the path

While our investigation presented in this paper is restricted to special cases of traffic on the specific network topology of a path, they are valuable in several ways. As always, studying elemental cases of a problem can give us insights into more general versions of the problem. It is also true that insight is obtained by investigating the border of tractability and intractability in a problem, as we do in this paper. More practically, solution methods for simple cases can often be used to obtain good solution methods for general cases. We

In this section we focus on the case where the network topology is a path, which is one of the most studied topologies in theoretical networking (see for instance [23]–[26]), as well as one of the most natural and apparently simplest underlying graphs to study. Our investigations on the path are valuable in several ways. As always, studying elemental cases of a problem can give us insights into more general versions of it. It is also true that studying the border of tractability and intractability in a problem, as it seems to be the case of the path in our problem, usually provides intuition about how to approach it in a more general setting. More practically, solution methods for simple cases can often be used to obtain good solution methods for general cases.

We present polynomial-time optimal algorithms for two particular families of instances. Namely, we study in Section V-A the case when all the lightpaths go through the first or the last edge of the path, and in Section V-B the case when the *load* of the path (that is, the maximum number of lightpaths in any set L_i crossing an edge of the path) is bounded by a logarithmic function of the input size. We would like to stress here that it was claimed in the conference version of this paper that the (d, p)-TR problem is NP-hard in paths for any $d, p \geq 2$; unfortunately, the proof of this claim contained a flaw, and the computational complexity of the problem in paths remains still open.

A. Edge instances

In an *edge instance* there is an edge $e \in E(G)$ that is used by all the lightpaths.

Proposition 2: For any fixed $d, p \geq 2$, there is a polynomial-time algorithm solving the (d, p)-TR problem for edge instances in a path where all the lightpaths share the first edge.

Proof: Let P be the path with $V(P) = \{0, 1, \ldots, n\}$ and $E(P) = \{\{j, j+1\} \mid 0 \leq j \leq n-1\}\}$. By assumption, all the lightpaths share the edge $\{0, 1\}$. We first claim that there is an optimal solution using regenerators only at vertices $d, 2d, 3d, \ldots$. Indeed, consider an optimal solution in which this property is not satisfied and consider the first (leftmost) vertex with index not divisible by d containing a regenerator. We can replace the regenerators at this vertex with regenerators in the next vertex with index divisible by d (satisfying the same lightpaths than the previous regenerators were satisfying), and get a solution with the same cost. Therefore, the solution recursively obtained in this way is also optimal, and satisfies the claimed property. In the rest of the proof we confine ourselves to solutions that use regenerators only at vertices that are multiples of d.

For a vertex j and an index i, let load(j, i) be the number of lightpaths in L_i using edge $\{j, j+1\}$, and let load$(j) = \max_{1 \leq i \leq p}$ load(j, i). We shall see that the following simple algorithm is optimal.

Algorithm 1

procedure

For each vertex j which is a positive multiple of d do:

Place load(j) regenerators at v.

For each i, $1 \leq i \leq p$, do:

Associate arbitrarily one of these regenerators with each lightpath in L_i using edge $\{j, j+1\}$.

end procedure

Algorithm 1 constructs a feasible solution, as by definition of load(j) there are enough regenerators to d-satisfy each set of lightpaths. The optimality of the algorithm follows from the fact that any feasible solution uses at least $\sum_{j \text{ multiple of } d, j>0}$ load(j) regenerators, since we can assume that regenerators are only placed at vertices j which are multiples of d, and at least load(j) regenerators are needed at such vertex j. ∎

It is natural to ask whether Algorithm 1 is optimal for general edge instances, that is, even if the edge shared by all lightpaths is an internal one. Unfortunately, the answer is negative even for $p = 1$, as

shown by the following simple example for $d = 2$ and a path on six vertices $0, 1, \ldots, 5$. Let ℓ_1 (resp. ℓ_2) be a lightpath from vertex 0 to vertex 4 (resp. vertex 1 to vertex 5), so all lightpaths share, for instance, the edge $\{2, 3\}$. If we place regenerators only at multiples of $d = 2$, we need at least 3 regenerators (recall that $p = 1$), but we can do better by 2-satisfying ℓ_1 (resp. ℓ_2) with a regenerator at vertex 2 (resp. vertex 3), therefore using only 2 regenerators.

B. Bounded load

It turns out that if we impose that the load of the path is bounded by an appropriate function of the size of the instance, then the problem is solvable in polynomial time. Intuitively, this special case of instances is in the opposite extreme of the edge instances, where there is an edge with unbounded load.

Proposition 3: For any fixed $d, p \geq 2$, (d, p)-TR is polynomial-time solvable in paths if the load is $\mathcal{O}\left(\frac{\log |I| - \log p}{2p \cdot \log d}\right) = \mathcal{O}(\log |I|)$, where $|I|$ is the size of the instance.

Proof: The algorithm uses standard dynamic programming techniques. Again, let P be the path with $V(P) = \{0, 1, \ldots, n\}$ and $E(P) = \{\{j, j+1\} \mid 0 \leq j \leq n-1\}\}$. Let $L_i^j \subseteq L_i$ be the subset of L_i consisting of lightpaths crossing the edge $\{j, j+1\}$, and let $L^j = \cup_i L_i^j$ be the set of lightpaths crossing $\{j, j+1\}$. (Note that $|L_i^j| = \text{load}(j, i)$ and that $|L^j| \geq \text{load}(j)$.) We denote by $[d]$ the set of integers $\{1, \ldots, d\}$. We consider the set of all vectors $\mathcal{U}_j = [d]^{L^j}$. Namely, a vector $U \in \mathcal{U}_j$ is an assignment of an element of $[d]$ to every lightpath crossing edge $\{j, j+1\}$. In our algorithm, such vectors are used to denote for each lightpath $\ell \in L^j$ the distance to the closest (that is, rightmost) regenerator on the left of vertex j used by ℓ. If $X \subseteq L^j$ and $U \in \mathcal{U}_j$, we denote by $U|_X$ the vector U restricted (or projected) to the index set X. In particular, if $X = \{\ell\}$ is a single lightpath, then we denote – in the usual way – by U_ℓ the entry at index ℓ of U, which is a positive integer.

For each vertex j and each vector $U \in \mathcal{U}_j$, we store two values in the tables, namely $\text{cost}_{j,U} \in \mathbb{N}$ and $\text{best}_{j,U} \in \mathcal{U}_{j-1}$. The value $\text{cost}_{j,U}$ is the minimum cost of covering all the edges on the left of $j+1$ by regenerators, such that for each lightpath $\ell \in L^j$ the rightmost regenerator of ℓ is at vertex $j - d + U_\ell$. The vector $\text{best}_{j,U} \in \mathcal{U}_{j-1}$ is a vector achieving $\text{cost}_{j,U}$.

We visit each vertex $0 \leq j \leq n-1$. For $j = 0$ we have $\forall U \in \mathcal{U}_0, \text{cost}_{0,U} = 0$, because every lightpath using edge $\{0, 1\}$ starts at vertex 0, and therefore it is covered without any additional regenerators. We set also $\forall U \in \mathcal{U}_0, \text{best}_{0,U} = \emptyset$ indicating the same fact.

For each $j > 0$ and for each $U \in \mathcal{U}_j$, we calculate $\text{cost}_{j,U}$ and $\text{best}_{j,U}$ as follows:

$$X = \{\ell \in L^j \mid U_\ell < d\}, \quad \overline{X} = \{\ell \in L^j \mid U_\ell = d\}. \tag{2}$$

Note that $X \cup \overline{X} = L^j$. Then we compute $\text{cost}_{j,U}$ as

$$\min_{\substack{U' \in \mathcal{U}_{j-1} \text{ such that} \\ U'|_X = U|_X + \mathbf{1}}} \left(\text{cost}_{j-1,U'} + \max_{1 \leq i \leq p} |\overline{X} \cap L_i^j| \right) , \tag{3}$$

where $\mathbf{1}$ is a vector of ones of appropriate size. Then, $\text{best}_{j,U}$ is set to some vector $U' \in \mathcal{U}_{j-1}$ achieving the minimum in the above expression.

Finally, we construct an optimum solution as follows:

Set $U^{n-1} = \textbf{argmin}_U \ \text{cost}_{n-1,U}$.

For $j = n-2$ to 0 do: $U^j = \text{best}_{j+1,U^{j+1}}$.

For each vertex j and each lightpath $\ell \in L^j$, we assign a regenerator at vertex j to lightpath ℓ if and only if $U_\ell^j = d$, i.e., vector U^j has value d in the entry ℓ.

Correctness. We shall now see that Equation (3) is correct. Consider a vertex j and a vector $U \in \mathcal{U}_j$. Consider a solution corresponding to U and its cost. This solution defines some vector $U' \in \mathcal{U}_{j-1}$. We want to express U in terms of U'. Let X and $\overline{X}$ be as in (2), and consider a lightpath $\ell \in L^j$. If $\ell \in X$, then by definition $U_\ell < d$, i.e., the rightmost regenerator serving lightpath ℓ is at vertex $j - d + U_\ell < j$. Therefore, for U' we have $U'_\ell = U_\ell + 1$. This is true for every index $\ell \in X$, and thus $U'|_X = U|_X + \mathbf{1}$. This is exactly the set of values U' over which the expression in parentheses is minimized. Now we show that for each possible value of U the value of this expression is actually the cost of the solution corresponding to U. For any $\ell \notin X$ we have $U_\ell = d$, i.e., the rightmost regenerator serving lightpath ℓ is at vertex $j - d + U_\ell = j$. Therefore, the number of regenerators at vertex j serving lightpaths of L_i is $|\overline{X} \cap L_i^j|$. The number of regenerators at j is the maximum of this value over all L_i. This is the cost

incurred by the solution at vertex j. Adding this to the cost of U', we obtain the cost of U. Therefore the algorithm chooses in Equation (3) the best U' for a given U over all possible values of U'.

Running time. Let β be an upper bound on the load of the instance. Then we have $|L_i^j| \leq \beta$, and thus $|L^j| = \text{load}(j) \leq \beta \cdot p$ and $|\mathcal{U}_j| \leq d^{\beta \cdot p}$. The computation of the above minimum takes at most $|\mathcal{U}_{j-1}| \leq d^{\beta \cdot p}$ steps, each of which takes p steps in order to compute the maximum, so we have at most $p \cdot d^{\beta \cdot p}$ steps overall. After initialization, we iterate over $j = 1, 2, \ldots, n$, and then over the at most $d^{\beta \cdot p}$ values of U. The total number of steps is $n \cdot p \cdot d^{2\beta \cdot p} < |I| \cdot p \cdot d^{2l \cdot p}$. If for some integer c, it holds $\beta \leq \frac{(c-1) \cdot \log |I| - \log p}{2p \cdot \log d} = \mathcal{O}(\log |I|)$, the running time of the presented algorithm is bounded by the polynomial $|I|^c$. ■

VI. More general settings

In this section we generalize the (d, p)-TR problem in two natural directions. Namely, in Section VI-A we allow the number p of traffic patterns to be unbounded, and in Section VI-B we introduce a parameter k that bounds the number of regenerators that can be placed at a vertex. Technologically, the latter constraint captures the fact of having a bounded number of ROADMs per vertex, as the number of wavelengths (and therefore, the number of regenerators) an ROADM can handle is usually not too big (see Section I-A).

A. Unbounded number of sets of lightpaths

If p is part of the input, then (d, p)-TR contains as a particular case the model studied in [9] (the so-called *location* problem, denoted RPP/∞/+ in [9]). Indeed, if each set of lightpaths consists of a single lightpath (that is, when p is the number of lightpaths), then the objective is to place the minimum number of regenerators such that each lightpath is satisfied. Therefore, the hardness results stated in [9] also apply to this more general setting, in particular an approximation lower bound of $\Omega(\log(d \cdot p))$ unless NP can be simulated in subexponential time. Note that this hardness bound matches the approximation ratio given by Theorem 2. Nevertheless, note also that the approximation algorithm presented in Theorem 2 runs in polynomial time only for bounded p.

We now reformulate the problem studied in [9] using our terminology. Let $d \geq 1$ be a fixed integer.

d-Regenerators Location (d-RL)
Input: An undirected graph $G = (V, E)$ and a set of lightpaths L.
Output: A function $\text{reg} : V \times L \to \{0, 1\}$ s.t. each lightpath $\ell \in L$ is d-satisfied.
Objective: Minimize $\sum_{v \in V} \text{reg}(v)$, where $\text{reg}(v) = \max_{\ell \in L} \text{reg}(v, \ell)$.

Note that in the above problem, $\text{reg}(v) \in \{0, 1\}$. We now focus on the case $d = 2$ of d-RL.

Remark 1: Given an instance of 2-RL in a graph G, the problem can be reduced to a Minimum Vertex Cover problem in a subgraph of G. Indeed, given a set of lightpaths L, remove the first and the last edge of each lightpath, and let H be the subgraph of G defined by the union of the edges in the modified lightpaths. It is then clear that the minimum number of regenerators to 2-satisfy all the lightpaths in L equals the size of a minimum vertex cover of H.

By Remark 1 and König's theorem [15], it follows that 2-RL can be solved in polynomial time in bipartite graphs. This result extends the results of [9] for $d = 2$, where it is proved that for any $d \geq 2$, d-RL is polynomial-time solvable in trees and rings. Finally, it also follows from Remark 1 that 2-RL admits a PTAS in planar graphs [27] and, more generally, in any family of minor-free graphs [28].

B. Bounded number of regenerators per vertex

From a technological point of view, it makes sense to introduce a parameter k that limits the number of regenerators that can be used at a single vertex. Adding this restriction to the d-RL problem, we get the following problem, which is actually the so-called k-*location* problem and denoted RPP/k/+ in [9].

Again, we restate the problem using our terminology. Let $d, k \geq 1$ be two fixed integers.

(d,k)-REGENERATORS LOCATION ((d,k)-RL)

Input: An undirected graph $G=(V,E)$ and a set of lightpaths L.

Output: A function $\text{reg}: V \times L \to \{0,1\}$ s.t. each lightpath $\ell \in L$ is d-satisfied and $\text{reg}(v) \leq k$, where $\text{reg}(v) = \sum_{\ell \in L} \text{reg}(v,\ell)$.

Objective: Minimize $|\{v \in V \mid \text{reg}(v) > 0\}|$.

We now resolve two questions that were left open in [9]. Namely, it is proved in [9] that given an instance of $(3,1)$-RL, it is NP-complete to decide whether there exists a feasible solution for it, which in particular implies that the $(3,1)$-RL problem itself is NP-hard to approximate within any ratio. In the following we prove that, surprisingly, the situation changes for $d=2$ and $k=1$. More precisely, it is in P to decide whether there exists a feasible solution for an instance of $(2,1)$-RL, while finding an optimal one is NP-hard.

Proposition 4: Given an instance of $(2,1)$-RL, it can be decided in polynomial time whether there exists a feasible solution for it.

Proof: Given an instance $\alpha=(G,L)$ of $(2,1)$-RL, we reduce it to an instance of 2-SAT as follows. We introduce boolean variables $R_{v,\ell}$, for every $v \in V(G)$ and every $\ell \in L$, which correspond to $\text{reg}(v,\ell)$, namely $R_{v,\ell}$ is true if and only if there is a regenerator at vertex v of lightpath ℓ. We construct the boolean expression ϕ with the following clauses (with two literals each) that capture exactly the constrains of $(2,1)$-RL:

$$\forall v \in V(G),\ \forall \ell_i, \ell_j \in L \qquad (\overline{R_{v,\ell_i}} \vee \overline{R_{v,\ell_j}})$$
$$\forall \ell \in L, \forall \{u,v\} \text{ internal edge of } \ell \qquad (R_{u,\ell} \vee R_{v,\ell})$$

The first family of clauses imposes that there is no vertex v with two lightpaths ℓ_i, ℓ_j being served by a regenerator in v ($k=1$), and the second family of clauses imposes that every internal edge of every lightpath is covered by some regenerator; in other words, that every lightpath is 2-satisfied. Thus, there exists a feasible solution for α if and only if ϕ is satisfiable, which can be checked in polynomial time [17]. ■

Finally, we prove the following result by reduction from MINIMUM VERTEX COVER.

Proposition 5: The $(2,1)$-RL problem is NP-hard.

Proof: We prove it by reduction from the MINIMUM VERTEX COVER (VC for short) problem. Given an instance $G=(V,E)$ of VC, we construct an instance $(G'=(V',E'),L)$ of $(2,1)$-RL as follows.

For each vertex v of G, we add $2 \cdot \deg_G(v) + 1$ vertices $v_1, v_2, \ldots, v_{2\cdot\deg_G(v)+1}$ to G'. Note that there are $\deg_G(v)$ vertices with even indices and $\deg_G(v)+1$ vertices with odd indices. We also add a lightpath ℓ_v of length $2 \cdot \deg_G(v)$ through vertices $v_1, v_2, \ldots, v_{2\cdot\deg_G(v)+1}$, in this order. These lightpaths are termed *long lightpaths*.

For each edge $e=\{u,v\}$ of G, we add a lightpath of length 1, termed *short lightpath*. The endpoints of this lightpath are a vertex v_i and a vertex u_j with even indices i,j such that neither v_i nor u_j is already an endpoint of a short lightpath. There are always two such available vertices, since there are $\deg_G(v)$ vertices with even indices. Finally, we extend both ends of each lightpath by adding an extra vertex to G for every endpoint of a lightpath, and an additional edge connecting the current endpoint of the lightpath to the new vertex (the reason to do this transformation is that no regenerator needs to be placed in the endpoints of a lightpath). This construction is depicted in Fig. 3, where for simplicity the extra vertices and edges added in the last step are not shown. Square (resp. circle) nodes correspond to vertices in G (resp. in the network); the colors in the figure are used in the sequel.

Note that there is exactly one way to 2-satisfy a long lightpath ℓ_v optimally, namely by placing $\deg_G(v)$ regenerators at the vertices with even indices, any other solution needing at least $\deg_G(v)+1$ regenerators to 2-satisfy ℓ_v. We claim that there is a vertex cover of G with cardinality at most t if and only if there is a solution of (G',L) with cost at most $3|E|+t$.

Indeed, let first $S \subseteq V$ be a vertex cover of G of size t. For each vertex $v \in S$, we put regenerators at all the odd vertices of the long lightpath ℓ_v (that is, $\deg_G(v)+1$ regenerators) and one regenerator in every short lightpath intersecting ℓ_v at the vertex in their intersection (recall that there is exactly one such lightpath for every even vertex of ℓ_v). For each vertex $u \notin S$, we put regenerators at all the even vertices of the long lightpath ℓ_u (that is, $\deg_G(u)$ regenerators).

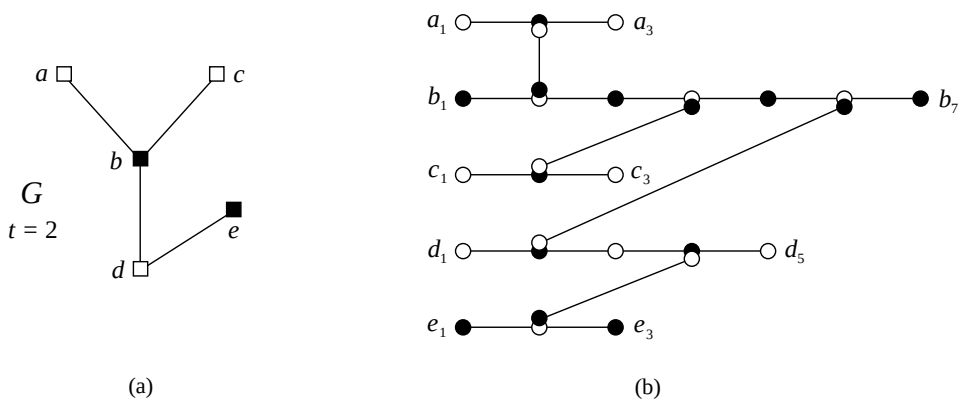

Fig. 3. Reduction from VC to $(2,1)$-RL in the proof of Proposition 5. (a) Instance G of VC, where the squares indicate a vertex cover. (b) Instance of $(2,1)$-RL constructed from G with $|V(G)|+|E(G)|$ lightpaths, where for simplicity the first and last edge of each lightpath are not depicted. Black vertices correspond to regenerators associated with a vertex and a lightpath, defining a feasible solution for this instance.

If a short lightpath has two regenerators (one at each endpoint), we remove one of them arbitrarily. It can be verified that every lightpath is 2-satisfied, and that at each vertex there is at most one regenerator. This solution uses $|E|$ regenerators for the short lightpaths, and $\sum_{v\in S}(\deg_G(v)+1)+\sum_{u\notin S}\deg_G(u)$ regenerators for the long lightpaths. Therefore, its cost is $3|E|+|S| = 3|E|+t$ regenerators.

Conversely, consider a solution SOL to the instance (G', L) using at most $3|E|+t$ regenerators. Each short lightpath has a regenerator in at least one endpoint v_i. Therefore SOL has at least $|E|$ regenerators to satisfy the short lightpaths. Let X be the set of vertices of V containing at least one of these regenerators. The set X is a vertex cover of G, since each edge of G corresponds to a short lightpath in L and one of its endpoints contains a regenerator. Consider a lightpath ℓ_v with $v \in X$. It cannot be covered with $\deg_G(v)$ regenerators, since at least one of the even vertices cannot be used, because this vertex already contains a regenerator for a short lightpath. Therefore ℓ_v needs at least $\deg_G(v)+1$ regenerators. For a lightpath ℓ_v with $v \notin X$, we need at least $\deg_G(v)$ regenerators. Therefore, SOL uses at least $3|E|+|X|$ regenerators. It holds $3|E|+|X| \leq \text{cost}(SOL) \leq 3|E|+t$, and thus $|X| \leq t$.

The above proof is illustrated in Fig. 3; in Fig. 3(a) black squares indicate a vertex cover of G, and in Fig. 3(b) black vertices correspond to regenerators. ■

VII. Conclusions and further research

In this article we presented a theoretical study of the problem of placing regenerators in optical networks, so that on each lightpath we must put a regenerator every at most d hops. The cost is the total number of regenerators. We considered the case when p possible traffic patterns are given (each by a set of lightpaths), and the objective is to place the minimum number of regenerators satisfying each of these patterns. This setting arises naturally when designing real networks under uncertain traffic forecast. The problem is called (d,p)-Total Regenerators problem, or (d,p)-TR for short. We now summarize our results and propose a number of lines for further research.

We proved that for any fixed $d, p \geq 2$, (d,p)-TR does not admit a PTAS unless P = NP, even if the network topology has maximum degree at most 3 and the lightpaths have length at most $2d$, by reduction from Minimum Vertex Cover in graphs of maximum degree 3. It would be interesting to determine which is the explicit approximation lower bound given by Theorem 1. The recent results of Austrin *et al.* [29] about the hardness of Minimum Vertex Cover in graphs of bounded degree may shed some light on this question. We provided an approximation algorithm for (d,p)-TR with constant ratio $\ln(d \cdot p)$, by reducing it to Minimum Set Cover. Finding a polynomial-time approximation algorithm matching the hardness lower bound given by Theorem 1 seems to be a challenging task.

We proved that (d,p)-TR is polynomial-time solv-

able in paths when all lightpaths use the first (or the last) edge of the path. It remains to settle the complexity of the case when the edge shared by all lightpaths is an internal edge of the path, which could be polynomial or NP-hard. Still in the path, but in the opposite extreme of the type of instances, we also proved that (d, p)-TR can be solved in polynomial time when the maximum number of lightpaths using an edge is logarithmically bounded by the size of the instance. It may be possible to extend our dynamic programming approach to trees with instances having this property, and even to graphs with bounded treewidth.

The computational complexity of (d, p)-TR in paths (for general instances) is still unknown. Very recently, it has been proved in [30] that there is a polynomial-time algorithm to solve $(2, p)$-TR in paths for any fixed $p \geq 1$; the case $d \geq 3$ remains open.

We generalized our model by allowing the number of sets of lightpaths to be unbounded, and by introducing a parameter k that bounds the number of regenerators that can be placed at a node. This way, the model studied in [9] becomes a particular case. We settled several complexity questions that were left open in [9] concerning the case $k = 1$ and $d = 2$. As future work, it seems to be of high importance to consider the parameter k in the original statement of our (d, p)-TR problem.

As mentioned in [9], other interesting avenues for further research are to consider the online setting (that is, when the lightpaths are not given in advance) and the weighted version of the problem (see also [31]), in the sense that the edges of the network have an associated weight, and the distance constraint is replaced with the corresponding weighted distance.

Considering the parameterized complexity of the (d, p)-TR problem is a promising approach in order to better understand its complexity. A natural choice for a parameter could be the number of regenerators used by a feasible solution. The powerful techniques of the theory of parameterized complexity [32] could be very helpful in designing efficient and practical algorithms for finding optimal solutions in real networks, even if (d, p)-TR is NP-hard.

Finally, we assumed that we are given a discrete (finite or infinite) set of possible traffic patterns. Even if this model can be applied in a variety of contexts, in some cases the traffic distribution may be more complicated. In this spirit, a possible direction is to consider a probability distribution (discrete or continuous) over the space defined by possible sets of lightpaths, the objective being to satisfy the lightpaths with high probability, or to satisfy a given fraction of them.

Acknowledgement. We would like to express our gratitude to Ori (Ornan) Gerstel from CISCO, Israel, for sharing with us the new trends concerning the placement of regenerators in optical networks, and especially for suggesting this research direction concerning multiple sets of requests.

REFERENCES

[1] S. Chen, I. Ljubic, and S. Raghavan, "The regenerator location problem," *Networks*, vol. 55, no. 3, pp. 205–220, 2010.

[2] S. W. Kim and S. W. Seo, "Regenerator placement algorithms for connection establishment in all-optical networks," *Communications, IEE Proceedings*, vol. 148, no. 1, pp. 25–30, 2001.

[3] S. Pachnicke, T. Paschenda, and P. M. Krummrich, "Physical Impairment Based Regenerator Placement and Routing in Translucent Optical Networks," in *Optical Fiber Communication Conference and Exposition and The National Fiber Optic Engineers Conference (Optical Society of America, paper OWA2)*, 2008.

[4] K. Sriram, D. Griffith, R. Su, and N. Golmie, "Static vs. dynamic regenerator assignment in optical switches: models and cost trade-offs," in *Workshop on High Performance Switching and Routing (HPSR)*, 2004, pp. 151–155.

[5] 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, pp. 39–53, 2005.

[6] ——, "Dynamic routing in translucent WDM optical networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2002, pp. 955–971.

[7] R. Fedrizzi, G. M. Galimberti, O. Gerstel, G. Martinelli, E. Salvadori, C. V. Saradhi, A. Tanzi, and A. Zanardi, "Traffic Independent Heuristics for Regenerator Site Selection for Providing Any-to-Any Optical Connectivity," in *Proceedings of IEEE/OSA Conference on Optical Fiber Communications (OFC)*, 2010, paper OTuG4.

[8] ——, "A Framework for Regenerator Site Selection Based on Multiple Paths," in *Prooceedings of IEEE/OSA Conference on Optical Fiber Communications (OFC)*, 2010, paper OTuG7.

[9] M. Flammini, A. Marchetti-Spaccamela, G. Monaco, L. Moscardelli, and S. Zaks, "On the complexity of the regenerator placement problem in optical networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 498–511, 2011.

[10] O. Gerstel and H. Raza, "Predeployment of Resources in Agile Photonic Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 22, no. 10, pp. 2236–2244, 2004.

[11] S. K. Korotki and K. N. Oikonomou, "Scaling of Most-Likely Traffic Patterns of Hose- and Cost-Constrained Ring and Mesh Networks," in *Prooceedings of IEEE/OSA Conference on Optical Fiber Communications (OFC)*, 2006.

[12] S. K. Korotki, "An Overview of the Global Network Expectation Model," in *Prooceedings of IEEE/OSA Conference on Optical Fiber Communications (OFC)*, 2004, invited paper.

[13] C. S. R. Murthy and S. Sinha, "Information Theoretic Approach to Traffic Adaptive WDM Networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, pp. 881–894, 2005.

[14] O. Amini, S. Pérennes, and I. Sau, "Hardness and Approximation of Traffic Grooming," *Theoretical Computer Science*, vol. 410, no. 38-40, pp. 3751–3760, 2009.

[15] R. Diestel, *Graph Theory*. Springer-Verlag, 2005.

[16] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[17] V. V. Vazirani, *Approximation algorithms*. Springer-Verlag, 2001.

[18] P. Alimonti and V. Kann, "Some APX-completeness results for cubic graphs," *Theoretical Computer Science*, vol. 237, no. 1-2, pp. 123–134, 2000.

[19] J. Akiyama and V. Chvátal, "A Short Proof of the Linear Arboricity for Cubic Graphs," *Bulletin of Liberal Arts & Sciences, Nippon Medical School*, vol. 2, pp. 1–3, 1981.

[20] J.-C. Bermond, J. L. Fouquet, M. Habib, and B. Péroche, "On linear k-arboricity," *Discrete Mathematics*, vol. 52, pp. 123–132, 1984.

[21] C. Thomassen, "Two-Coloring the Edges of a Cubic Graph Such That Each Monochromatic Component Is a Path of Length at Most 5," *J. Comb. Theory, Ser. B*, vol. 75, no. 1, pp. 100–109, 1999.

[22] R. Duh and M. Fürer, "Approximation of k-Set Cover by Semi-Local Optimization," in *Proceedings of the 29th ACM Symposium on Theory of Computing (STOC)*, 1997, pp. 256–264.

[23] S. Huang, R. Dutta, and G. N. Rouskas, "Traffic grooming in path, star, and tree networks: complexity, bounds, and algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. S-4, pp. 66–82, 2006.

[24] J.-C. Bermond, D. Coudert, J. Moulierac, S. Pérennes, H. Rivano, I. Sau, and F. Solano Donado, "Mpls label stacking on the line network," in *Proceedings of the 8th International Networking Conference*, ser. LNCS, vol. 5550, 2009, pp. 809–820.

[25] J.-C. Bermond, L. Braud, and D. Coudert, "Traffic grooming on the path," *Theoretical Compututer Science*, vol. 384, no. 2-3, pp. 139–151, 2007.

[26] P. Iyer, R. Dutta, and C. D. Savage, "Minimizing transceivers in optical path networks," *Journal of Optical Networking*, vol. 8, no. 5, pp. 454–461, 2009.

[27] B. S. Baker, "Approximation algorithms for NP-complete problems on planar graphs," *Journal of the ACM*, vol. 41, no. 1, pp. 153–180, 1994.

[28] E. Demaine, M. Hajiaghayi, and K.-i. Kawarabayashi, "Algorithmic Graph Minor Theory: Decomposition, Approximation and Coloring," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005, pp. 637–646.

[29] P. Austrin, S. Khot, and M. Safra, "Inapproximability of Vertex Cover and Independent Set in Bounded Degree Graphs," in *Proceedings of the 24th IEEE Computational Complexity Conference (CCC)*, 2009, pp. 74–80.

[30] A. Voloshin, *personal communication*, 2011.

[31] A. Sen, S. Banerjee, P. Ghosh, S. Murthy, and H. Ngo, "Brief announcement: on regenerator placement problems in optical networks," in *Proceedings of the 22nd Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, 2010, pp. 178–180.

[32] R. G. Downey and M. R. Fellows, *Parameterized Complexity*. Springer-Verlag, 1999.

George B. Mertzios received his Diploma (with Honors) in Mathematics and Informatics from the Technische Universitaet Muenchen in 2005, and his PhD in Computer Science from the RWTH Aachen University in 2009. During 2010-11 he worked as a postdoctorate researcher at the Department of Computer Science in Technion - Israel Institute of Technology and at the Caesarea Rothschild Institute for Computer Science in the University of Haifa, Israel. Since September 2011 he is a Lecturer in Computer Science in Durham University, England. His research areas include Efficient Graph Algorithms, Computational and Parameterized Complexity, Evolutionary Graph Theory, and Algorithmic Game Theory.

Ignasi Sau received his Degree in Mathematics in 2005 and his Degree in Telecommunications Engineering in 2006, both from Universitat Politècnica de Catalunya (UPC) of Barcelona. In 2009 he received a joint PhD in Mathematics and Computer Science from UPC and Université de Nice-Sophia Antipolis (France), respectively. In 2010 he was a postdoc at the Computer Science Department of the Technion (Haifa, Israel), and since October 2010 he is a Chargé de Recherche (Junior Researcher) of CNRS at LIRMM (Montpellier, France). His research interests focus mainly on Algorithmic Graph Theory, in particular Parameterized and Computational Complexity, as well as Approximation and Exact Algorithms.

Mordechai Shalom holds a B.A. degree in Architecture from ITU, Technical University of Istanbul, from which he graduated at 1981. He completed his studies towards M.Sc. degree at 1986 in the Faculty of Computer Science, Technion, from which he received his Ph.D. degree at 2006. He is currently a Lecturer in the Tel Hai Academic college and a Teaching Assistant at the Technion. His research interests are Optimization Problems in Optical Networks, Distributed Algorithms, and Approximation Algorithms for Graph Theoretical problems.

Shmuel Zaks received his BSc (cum laude) and MSc degrees in Mathematics from the Technion, Haifa, Israel, in 1971 and 1972, respectively, and his PhD degree in Computer Science from the University of Illinois at Urbana-Champaign in 1979. He is a Professor in the Department of Computer Science at the Technion, Haifa, Israel, and is the incumbent of the Joan Callner-Miller Chair in Computer Science. Professor Zaks is the author of over 150 journal and conference publications. His research areas include Distributed Computing, Graph and Combinatorial Algorithms, Discrete Mathematics, and Theory of Networking (especially Optical Networks).