On the Complexity of the Regenerator Placement Problem in Optical Networks

Michele Flammini
Dipartimento di Informatica
Università degli Studi dell'Aquila
L'Aquila, Italy
flammini@di.univaq.it

Alberto Marchetti Spaccamela
Dipartimento di Informatica e Sistemistica
Sapienza Università di Roma
Italy
alberto@dis.uniroma1.it

Gianpiero Monaco
Dipartimento di Informatica
Università degli Studi dell'Aquila
L'Aquila, Italy
gianpiero.monaco@di.univaq.it

Luca Moscardelli
Dipartimento di Scienze
Università degli Studi di Chieti-Pescara
Italy
moscardelli@sci.unich.it

Shmuel Zaks
Department of Computer Science
Technion
Haifa, Israel
zaks@cs.technion.ac.il

ABSTRACT

Placement of regenerators in optical networks has attracted the attention of recent research works in optical networks. In this problem we are given a network, with an underlying topology of a graph G, and with a set of requests that correspond to paths in G. There is a need to put a regenerator every certain distance, because of a decrease in the power of the signal. In this work we investigate the problem of minimizing the number of locations to place the regenerators. We present analytical results regarding the complexity of this problem, in four cases, depending on whether or not there is a bound on the number of regenerators at each node, and depending on whether or not the routing is given or only the requests are given (and part of the solution is also to determine the actual routing). These results include polynomial time algorithms, NP-complete results, approximation algorithms, and inapproximability results.

Categories and Subject Descriptors

F.2.0 [**ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY**]: General

General Terms

Algorithms

Keywords

Optical Networks, Wavelength Division Multiplexing (WDM), Regenerators, Approximation Algorithms, Complexity

1. INTRODUCTION

We deal with optical networks, where we are given requests for connections to be established between pairs of nodes. These connections are established by using lightpaths, which correspond to paths in the underlying network topology. For given nodes a and b in the network, we might get as an input either a path connecting a and b, or only the request (=pair of nodes) (a, b), and in this latter case part of the problem is also to determine the actual routing. In addition, since a lightpath looses its power along its way, we are given a bound $d \geq 1$ on the length of a lightpath. A connection between nodes a and b can thus be done by a single lightpath of length $\leq d$ connecting a and b (in which the signal is sent in the optical domain). Alternatively, it can be sent by a sequence of lightpaths each of length $\leq d$ connecting a and v_1, v_1 and $v_2, \ldots, v_{t-1}$ and v_t, and v_t and b. In this case in each of the intermediate nodes $v_1, v_2, \ldots, v_{t-1}, v_t$ we put a *regenerator* (each regenerator serves a unique connection); the signal from a to v_1 is transformed at v_1 to the electronic domain, then again to the optical domain from v_1 to v_2, then it is transformed again at v_2 to the electronic domain, and again to the optical domain from v_2 to v_3, and so on. To each lightpath we assign a color (which is the wavelength assigned to it). Two lightpaths that share an edge must get different colors. Note that if a connection is made by a sequence of lightpaths, they are not all necessarily with the same color. Anyway, since we do not consider bounds on the number of colors and our goal is independent of it, in the sequel of the paper we never refer to colors or wavelengths. In fact, our goal is to minimize the number of locations (i.e. nodes) where regenerators are used.

We consider this problem in four cases, depending on whether or not there is a bound on the number of regenerators at each node, and depending on whether or not the routing is given or only the requests are given (and part of the solution is also to determine the actual routing). Moreover, we first deal with the case in which only simple paths are allowed, and then we extends our results to the case of non-simple paths.

This Regenerator Placement Problem (RPP) in optical networks has attracted the attention of recent research works (see, e.g., [4, 11, 12, 14, 15, 17, 18]). Moreover, other papers (see, e.g., [6, 10, 16]) studied related problems dealing with hardware optimization in optical networks with wavelength conversion. An all-optical network is also called a transparent optical network, and the opposite of a transparent optical network is the opaque optical network (see, e.g., [2, 13]). An opaque optical network employs *3R regeneration* (reamplification, reshaping and retiming) at every intermediate node of a path to regenerate the signal and improve transmission quality. We note that currently 3R regeneration is realized only through optical-to-electronic-to-optical (O/E/O) conversion. Most of today's optical networks are fully opaque, where O/E/O conversions take a major fraction of network cost. This study is thus highly important for this application area.

This is the first work where a systematic theoretical study is made for this RPP. Most of the above-mentioned works discuss the technological aspects of the problem, and include heuristic algorithms for it. This work is the first to present a theoretical framework to deal with these problems. Our results include polynomial time algorithms, NP-complete results, approximation algorithms, and inapproximability results.

1.1 Definitions

We are given a network whose underlying topology is an undirected graph $G = (V, E)$. We assume that $V = \{1, 2, \cdots, n\}$. A path in the graph is a sequence of nodes that follow edges of the graph; formally, a path $[a, b]$ of length $t \geq 1$ connecting vertices $a, b \in V$ is $[a, b] = \langle a = v_0, v_1, v_2, \cdots, v_t = b \rangle$, where $(v_i, v_{i+1}) \in E$ for every $i = 0, 1, \cdots, t-1$. We are given $d > 0$, which corresponds to the maximal number of edges that a signal can travel before it needs a regenerator. Each connection thus consists of a sequence of lightpaths (of length at most d) whose concatenation forms a path connecting a_i and and b_i.

We need to establish a given set of requests by putting regenerators in as few locations as possible. A *solution* consists of a subset of nodes $U \subseteq V$, such that by putting regenerators at these nodes all the constraints (on the length of the lightpaths) can be satisfied.

Regarding the connections to be established we consider two cases:

- We are given the actual routing. We will denote these given m paths by $\{[a_1, b_1], [a_2, b_2], \cdots, [a_m, b_m]\}$, $a_i \neq b_i$, $a_i, b_i \in V$. In this case we have to determine a set of nodes $U \subseteq V$ where regenerators can be put. This set U has the property that at least one among any $d+1$ consecutive internal nodes of any path $[a_i, b_i]$ must belong to U. This corresponds to the constraint that each path must be assigned a regenerator every at most d hops.

- We are given only requests, defined by a *demand matrix* $A_{n \times n}$, where $a_{i,j}$ is the number of connections to be established between i and j.

  In this case we have to determine the routing for each of the requests, and the set U as above.

These two cases will be denoted as 'rt' (when the routing is given) and 'req' (when the routing is not given).

We consider two cases, depending on whether there is a given bound k on the number of regenerators to be put in a single node, or this number is unbounded ($k = \infty$). The number of regenerators in a node v is denoted by $reg(v)$.

We thus consider the following four optimization problems:

`Problem RPP/`∞`/rt`
`Input:` A network $G = (V, E)$, a set of paths $\{[a_1, b_1], [a_2, b_2], \cdots, [a_m, b_m]\}$, $a_i \neq b_i$, $a_i, b_i \in V$, $d > 0$.
`Output:` A solution $U \subseteq V$.

`Problem RPP/k/rt`
`Input:` A network $G = (V, E)$, a set of paths $\{[a_1, b_1], [a_2, b_2], \cdots, [a_m, b_m]\}$, $a_i \neq b_i$, $a_i, b_i \in V$, $d, k > 0$.
`Output:` A solution $U \subseteq V$ such that $reg(v) \leq k$ for every $v \in U$.

`Problem RPP/`∞`/req`
`Input:` A network $G = (V, E), |V| = n$, a matrix $A_{n \times n}$, $d > 0$.
`Output:` A routing for the requests and a solution $U \subseteq V$.

`Problem RPP/k/req`
`Input:` A network $G = (V, E)$, $G = (V, E), |V| = n$, a matrix $A_{n \times n}$, $d, k > 0$.
`Output:` A routing for the requests and a solution $U \subseteq V$ such that $reg(v) \leq k$ for every $v \in U$.

For all the problems, we also have:
`Measure of a solution:` $|U|$, i.e. the number of locations hosting regenerators.
`Objective:` Minimizing the measure of a solution.

1.2 Our contribution

Our results are summarized in Table 1.

We remark that *Exp–Apx* is the class of problems admitting an approximation algorithm with an exponential approximation factor.

We first deal, in Section 2, with the case in which only simple paths are allowed.

	complexity	approximability
`RPP/`∞`/rt`	polynomial for trees, rings NP-hard for general network	$\Theta(\log m + \log d)$
`RPP/k/rt`	NP-hard $(k = 1, d = 3)$	$\notin$ *Exp–Apx*
`RPP/`∞`/req`	NP-hard ([4], for d=1 and all-to-all)	$\Theta(\log m)$ (for the all-to-all case)
`RPP/k/req`	NP-hard $(k = 1, d = 1)$	$\notin$ *Exp–Apx*

Figure 1: A summary of results for routing by simple paths

In Section 3 we extend our results to the case in which the routing can be done by using non-simple paths.

Finally, in Section 4 we summarize our results and discuss possible extensions of our studies.

2. THE SIMPLE PATHS CASE

In this section we study the problem in the case in which the routing is done using simple paths. In particular, the cases where we are given the paths are treated in Section 2.1 for case where there is no bound on the number of regenerators that can be placed in one node (`RPP/`∞`/rt`) and in Section 2.2 for the bounded case (`RPP/k/rt`). The cases where we are given only the requests are treated in Section 2.3 for case where there is no bound on the number of regenerators (`RPP/`∞`/req`), and in Section 2.4 for the bounded case (`RPP/k/req`).

2.1 Problem `RPP/`∞`/rt`

We now study the problem `RPP/`∞`/rt`. For the topologies of trees and rings we show polynomial constructions in Section 2.1.1. For the general problem we present approximability results in Section 2.1.2.

2.1.1 Tree and ring networks

We show that `Problem RPP/`∞`/rt` is polynomially solvable if the graph is a tree. We denote by $dist(x, y)$ the distance between vertices x and y and we assume w.l.o.g. that the tree is rooted at an arbitrary vertex and vertices are numbered according to DFS postorder visit (i.e. we perform a DFS search from the root and we number a vertex the last time it is visited).

Given a tree and a set of paths P the algorithm will modify paths in P whenever a regenerator is placed.

```
Algorithm Reg − Trees
Input P:  set of paths

for i from 1 to n do
  let P_i be the set of paths including
   node i
  for all paths [y,w] from P_i do
    if max(y,w) < i and dist(y,w) > d
    then
       place a regenerator at i
       eliminate [y,w] in P
    endif
    if max(y,w) ≥ i and
       dist(min(y,w),i) = d then
       place a regenerator at i
       eliminate [y,w] from P
       if dist(i,max(y,w)) > d then
          add [i,max(y,w)] to P
       endif
    endif
  endforall
endfor
```

THEOREM 2.1. *Algorithm $Reg - Trees$ finds an optimal solution to* `Problem RPP/`∞`/rt` *if the given graph is a tree.*

PROOF. It is easy to check that the algorithm finds a feasible solution. To prove optimality we proceed by induction on k, the number of locations where regenerators are placed by the algorithm. Given i let T_i be the subtree rooted at i.

The basis of the induction is trivially verified for $k = 0$.

It remains to prove the induction step: for any $k \geq 1$, we assume the claim true for $k - 1$ and we prove it for k. Let i be the first location in which the algorithm places a regenerator. It is easy to check that, if no regenerator is placed in T_i, since there exists a path $[a, b]$ such that either *1)* $[a, b]$ is completely contained in T_i and $dist(a, b) > d$ or *2)* a belongs to T_i, b does not belong to T_i and $dist(a, i) = d$, then any solution that does not use a regenerator in one of the vertices of T_i is not feasible. Therefore, the optimal solution uses regenerators at a least 1 location in T_i.

Moreover, let $\hat{P}$ be the set of paths obtained from P by

(i) eliminating from P all paths completely contained in T_i, and

(ii) replacing each path $[a, b] \in P$, such that a belongs to T_i and b does not belong to T_i with path $[i, b]$.

Since $\hat{P}$ induces a new instance of the problem for which the algorithm uses $k-1$ locations, by the induction hypothesis $k-1$ is also the value of an optimal solution for such an instance.

Therefore, the value of an optimal solution for the initial instance is $1 + (k - 1) = k$. $\square$

Note that if the graph is a path then Algorithm $Reg-Trees$ is equivalent to the greedy algorithm that sweeps the path from left to right and puts a regenerators whenever deemed.

The previous theorem is the basis to show that `Problem RPP/∞/rt` is polynomially solvable if the graph is a ring as shown in the following theorem.

THEOREM 2.2. *`Problem RPP/∞/rt` is polynomially solvable if the given graph is a ring.*

PROOF. First observe that if all paths have length less than d then clearly the problem is trivial.

If there exists at least one path $[a, b]$ of length greater than d then arbitrarily choose a segment Z of d consecutive vertices of $[a, b]$. For each vertex i of Z we obtain a solution U_i as follows:

(i) put a regenerator in i;

(ii) apply Algorithm $Reg-Trees$ to the path obtained by cutting the ring at vertex i.

It is easy to see that in this way U_i is an optimal solution with the additional constraint that there is a regenerator at i. It follows that the optimal solution in the ring is equal to the best solution among the d solutions U_i, $i \in Z$. □

2.1.2 General networks

In this section we first show that the problem `RPP/∞/rt` for general topologies is NP-hard and is not approximable in polynomial time with an approximation factor $(1-\epsilon)\log m$ (unless NP has slightly superpolynomial time algorithms), and than we provide a general approximation algorithm with approximation ratio $O(\log m + \log d)$.

THEOREM 2.3. *Problem `RPP/∞/rt` is NP-hard and any polynomial time approximation algorithm has an approximation factor at least $\log m$ and $\Omega(\log d)$, unless $NP \subset TIME(m^{O(\log\log m)})$.*

PROOF. We use a reduction from the Set Cover optimization problem:

`Set Cover`

`Input:` A set $A = \{1, \ldots, |A|\}$ of elements, a collection S of subsets of A, $S = \{s_1, s_2, \cdots, s_{n'}\}$, and $k > 0$.

`Output:` A subcollection $U \subseteq S$ of subsets covering the elements in A, i.e. such that $\cup_{u \in U} u = A$.

`Measure of a solution:` $|U|$, i.e. the cardinality of the subcollection U.

`Objective:` Minimizing the measure of the returned solution.

The `Set Cover` problem is known to be NP-hard and not approximable in polynomial time with an approximation factor $(1-\epsilon)\log|A|$, unless $NP \subset TIME(|A|^{O(\log\log|A|)})$ [7]. Given an instance of `Set Cover`, we construct an instance of `RPP/∞/rt` as follows. We first describe the network $G = (V, E)$. (See example in Figure 2).

Informally, for each set $s_i \in S$ $(i = 1, \ldots, n')$, we add a *main node* $v_{0,i}$ and $|A| - |s_i|$ *additional nodes* $v_{a,i}$ such that $a \notin s_i$, corresponding to the $|A| - |s_i|$ elements in A not belonging to s_i. Formally, $V = V_{start} \cup \hat{V} \cup V_{end}$, where $V_{start} = \{start_a | a = 1, \ldots, |A|\}$, $V_{end} = \{end_a | a = 1, \ldots, |A|\}$ and $\hat{V} = \bigcup_{s_i \in S}\{v_{0,i}\} \cup \bigcup_{s_i \in S}\{v_{a,i} | a = 1, \ldots, |A|, a \notin s_i\}$.

In order to describe the edge set E, we need the following function:

$$row(a, i) = \begin{cases} 0 & \text{if } a \in s_i \\ a & \text{otherwise.} \end{cases}$$

The edge set is $E = E_{start\to 1} \cup E_{1\to 2} \cup \cdots \cup E_{(n'-1)\to n'} \cup E_{n'\to end}$, where $E_{start\to 1} = \{\{start_a, v_{row(a,1),1}\} | a = 1, \ldots, |A|\}$, $E_{n'\to end} = \{\{v_{row(a,n'),n'}, end_a\} | a = 1, \ldots, |A|\}$ and finally, for $i = 1, \ldots, n'-1$, $E_{i\to(i+1)} = \{v_{row(a,i),i}, v_{row(a,i+1),i+1}\} | a = 1, \ldots, |A|\}$.

The set P of paths contains $|A|$ paths, one for each element in A; the path L_a corresponding to element $a \in A$ is $L_a = \langle start_a, v_{row(a,1),1}, v_{row(a,2),2}, \ldots, v_{row(a,n'),n'}, end_a \rangle$.

In order to complete the reduction, we set $d = n'$, so that each path needs at least one regenerator.

If there exists a solution for the `Set Cover` optimization problem with measure k and subsets $s_{i_1}, \cdots, s_{i_k}$, then, since one regenerator for each path is enough, it is easy to check that placing regenerators at the nodes $v_{0,i_1}, \cdots, v_{0,i_k}$ is a solution having the same measure k for the corresponding instance of the `RPP/∞/rt` problem.

Conversely, assume we are given a solution for the `RPP/∞/rt` problem using regenerators at k locations. Now we show that in this case it is possible to obtain a solution for the `Set Cover` optimization problem with measure at most k, thus proving the claim. We now show that any solution using regenerators at k locations can be converted into another solution using regenerators at $k' \leq k$ locations corresponding uniquely to *main* nodes $v_{0,i}$ $(i = 1, \ldots, n')$. First, notice that regenerators at nodes in V_{start} and V_{end} can be eliminated because such nodes are only endpoints for the paths. Moreover, if a regenerator used by a path L_a is located at an *additional* node (notice that by construction each additional node is crossed by only one path), it can be moved to a main node crossed by L_a (such a node there must exist since the sets cover all elements). Clearly, this process cannot increase the number of locations where regenerators are placed. Therefore, we have obtained a solution for the `RPP/∞/rt` problem using regenerators at $k' \leq k$ locations corresponding only to *main* nodes $v_{0,i}$ $(i = 1, \ldots, n')$; clearly, a solution for the `Set Cover` optimization problem with measure k' can be obtained by choosing the sets corresponding to such main nodes.

Finally, since in the work of [7] the instances proving the result have $|A| = \Theta(n')$, and in the reduction $d = n'$, the same proof implies the $\Omega(\log d)$ approximability result. □

We now provide an approximation algorithm almost matching the inapproximability result.

THEOREM 2.4. *It is possible to find a solution for problem `RPP/∞/rt` with approximation ratio $O(\log m + \log d)$.*

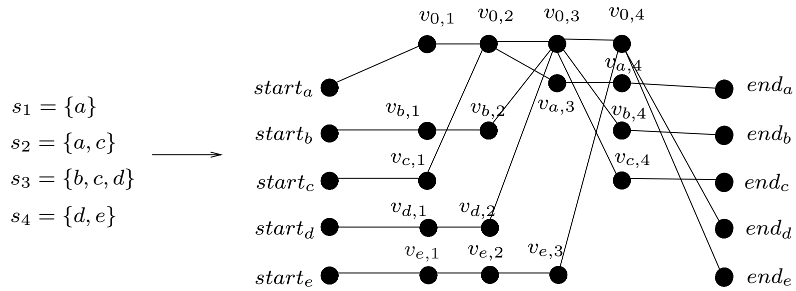

Figure 2: A reduction from Set Cover to RPP/∞/rt

PROOF. A given instance of the problem consists of a network $G = (V, E)$, a set of paths $\{[a_1, b_1], [a_2, b_2], \cdots, [a_m, b_m]\}$, $a_i \neq b_i$, $a_i, b_i \in V$, $d > 0$. We reduce it to the above-mentioned Set Cover problem.

We distinguish two cases: d odd and d even.

If d is odd, let $A = \{a_{v,i} | v \in V, v \in [a_i, b_i]\}$, and $S = \bigcup\{s_v | v \in V\}$, where for every v the set s_v is defined as follows. Let P_v be the set of paths crossing node v. For each path $p \in P_v$, we add to set s_v all the elements in A corresponding to the nodes of p at distance in p at most $\frac{d-1}{2}$ from v; moreover, for any endpoint of such paths being at distance at most d from v, we also add to set s_v all the elements in A corresponding to all the nodes of p between v and such an endpoint.

Now we apply the Set Cover algorithm of [5]. If the solution obtained for the set cover is $\{s_{v_1}, \ldots, s_{v_t}\}$, then clearly by placing regenerators at the nodes $\{v_1, \ldots, v_t\}$ we get a solution to the RPP/∞/rt problem with the same measure. Conversely, given a solution for the RPP/∞/rt problem with regenerators at the nodes $\{v_1, \ldots, v_t\}$, it is easy to check that $\{s_{v_1}, \ldots, s_{v_t}\}$ is a solution for the Set Cover problem with the same measure.

Since the size of each of the sets of S is bounded by md, it follows that the approximation ratio is $\log m + \log d + 1$.

If d is even, let $A = \{a_{j,i} | e_j \in E, e_j \in [a_i, b_i]\}$, and $S = \bigcup\{s_v | v \in V\}$, where for every v the set s_v is defined as follows. Let P_v be the set of paths crossing node v. For each path $p \in P_v$, we add to set s_v all the elements in A corresponding to the edges of p at distance in p at most $\frac{d}{2}$ from v (we consider the edges incident to a node at distance 1 from it); moreover, for any endpoint of such paths being at distance at most d from v, we also add to set s_v all the elements in A corresponding to all the edges of p between v and such an endpoint.

The claim follows by exploiting the same arguments of the previous (d odd) case. □

2.2 Problem RPP/k/rt

In this section we study the problem RPP/k/rt. Clearly if $d = 1$ then the problem is polynomial time solvable. We prove that it is NP-Complete to find a feasible solution for the problem if $k = 1$ and $d = 3$. This result is presented in Theorem 2.5.

THEOREM 2.5. *Given an instance of the* RPP/k/rt *problem, the problem of deciding whether there exists a feasible solution is NP-Complete when $k = 1$ and $d = 3$.*

Since it is difficult even to find a feasible solution for the considered problem, the following corollary holds.

COROLLARY 2.6. *The* RPP/k/rt *problem is not approximable in polynomial time, i.e.* RPP/k/rt $\notin$ Exp–Apx *unless $P = NP$.*

Moreover, by using a reduction from Graph d-Colorability, $d > 3$, it is possible to show that the RPP/k/rt problem is NP-Complete for every value of $d \geq 3$.

2.3 Problem RPP/∞/req

In this section we first show that the problem RPP/∞/req is NP-hard and is not approximable in polynomial time with an approximation factor $(1 - \epsilon) \log m$, even in the all-to-all case (unless NP has slightly super-polynomial time algorithms). Then, we provide an approximation algorithm with approximation ratio $O(\log m)$ for the all-to-all case.

In [4] the problem was proved to be NP-complete for the all-to-all case, where $d = 1$. This reduction was from the Minimum Vertex Cover problem. But it implies that the all-to-all problem cannot be approximated better than $\frac{7}{6}$ (see [1]). We use a different proof, that enables us to derive a much stronger inapproximability result; namely, we prove that the all-to-all problem cannot be approximated within $\log m$.

THEOREM 2.7. *Problem* RPP/∞/req *is NP-hard and is not approximable in polynomial time with an approximation factor $(1 - \epsilon) \log m$, unless $NP \subset TIME(n^{O(\log \log m)})$, even when $d = 1$. Moreover, the problem is still not approximable in polynomial time with an approximation factor $(\frac{1}{2} - \epsilon) \log m$, unless $NP \subset TIME(n^{O(\log \log m)})$, if the demand matrix is all-to-all and $d = 1$.*

PROOF. We use a reduction from the above-mentioned Set Cover problem. This problem is known to be NP-hard and not approximable in polynomial time with an approximation factor $(1 - \epsilon) \log |A|$, unless $NP \subset TIME(|A|^{O(\log \log |A|)})$ [7].

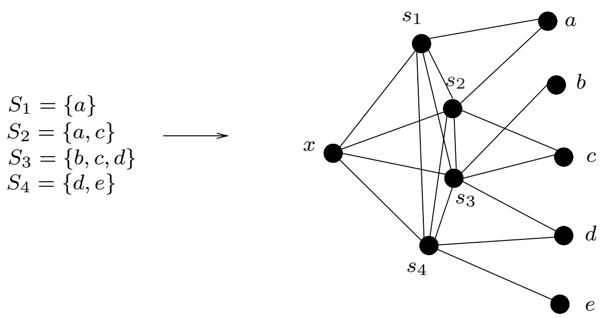

Figure 3: A reduction from Set Cover to RPP/∞/req

Given an instance of `Set Cover`, we construct an instance of `RPP/`∞`/req` as follows: the network is $G = (V, E)$, where $V = \{x\} \cup A \cup \{s | s \in S\}$. $E = \{(x, s) | s \in S\} \cup \{(a, s) | a \in s\} \cup \{(s, s') | s, s' \in S, s \neq s'\}$ and finally $d = 1$. (See example in Figure 3).

We first provide the proof for the all-to-all demand matrix case. In such a case, since in the work of [7] the instances proving the inapproximability have $|A| = \Theta(|S|)$, the number of requests turns to be $m = \Theta(|A|^2)$, an therefore $|A| = \Theta(\sqrt{m})$. Therefore, the reduction implies the $(\frac{1}{2} - \epsilon) \log m$ inapproximability result.

If there exists a solution for the `Set Cover` optimization problem with measure k and subsets $s_{i_1}, \cdots, s_{i_k}$, then placing regenerators at the nodes $s_{i_1}, \cdots, s_{i_k}$ is a solution having the same measure k for the corresponding instance of the `RPP/`∞`/req` problem in which the paths are selected as follows: the paths between the nodes directly connected by an edge are composed by such a unique edge (and do not need any regenerator); the path between node x and a node in $a \in A$ is $\langle x, s_{i_p}, a \rangle$, where p is such that $a \in s_{i_p}$; the path between two nodes in $a, b \in A$ is $\langle a, s_{i_p}, s_{i_{p'}}, b \rangle$, where p is such that $a \in s_{i_p}$ and p' is such that $b \in s_{i_{p'}}$; the path between node s_j and a node in $a \in A$ such that $a \notin s_j$ is $\langle s_j, s_{i_p}, a \rangle$, where p is such that $a \in s_{i_p}$. Notice that all the chosen paths requiring regenerators have as intermediate nodes only the nodes $s_{i_1}, \cdots, s_{i_k}$, i.e. the nodes hosting a regenerator.

Conversely, assume we are given a solution for the `RPP/`∞`/req` problem using regenerators at k locations. Now we show that in this case it is possible to obtain a solution for the `Set Cover` optimization problem with measure at most k, thus proving the claim. First of all, we show that any solution using regenerators at k locations can be converted into another solution using regenerators at $k' \leq k$ locations corresponding uniquely to nodes s_i. Recall that all the nodes s_i $(i = 1, \ldots, n)$ form a clique; if a regenerator is located at a node y such that $y = x$ or $y \in A$, since such a node has to be an intermediate node for all the paths using such a regenerator, it is possible to remove all the regenerators from y by short-cutting all such paths. More specifically, each path $\langle \ldots, s_i, y, s_j, \ldots \rangle$ can be converted in a new path $\langle \ldots, s_i, s_j, \ldots \rangle$ directly connecting s_i and s_j. In this way we have obtained a new solution for the `RPP/`∞`/req` problem using regenerators at $k' \leq k$ locations corresponding uniquely to nodes s_i. Finally, since in order to be connected to x any node $a \in A$ has to cross at least one node s_i in which the path requires a regenerator, we have that there exist $k' \leq k$ sets spanning all the elements in A, and thus constituting a solution for the `Set Cover` optimization problem having measure $k' \leq k$.

In order to prove that in the case of general demand matrix the problem is not approximable in polynomial time with an approximation factor $(1 - \epsilon) \log m$, unless $NP \subset TIME(n^{O(\log \log m)})$, it suffices to consider a demand matrix in which there are $|A|$ requests between node x and all the nodes in A. The proofs proceeds with the same arguments exploited in the all-to-all demand matrix case. □

THEOREM 2.8. *It is possible to find a solution for the* `RPP/`∞`/req` *problem with approximation ratio* $\frac{3}{2} \log m + 1$*, when the demand matrix is all-to-all.*

PROOF. In order to provide a $\frac{3}{2} \log m + 1$ approximation algorithm for the `RPP/`∞`/req` problem, we first apply the Set Cover greedy algorithm of [5] to the following Set Cover instance: $A = V$, and $S = \bigcup \{s_v | v \in V\}$, where for every v the set s_v be the sets of nodes $u \in V$ such that the distance in G between u and v is at most d.

Given a solution for an instance of the `RPP/`∞`/req` problem with regenerators at the nodes $\{v_1, \ldots, v_t\}$, it is easy to check that $\{s_{v_1}, \ldots, s_{v_t}\}$ is a solution for the corresponding instance of the `Set Cover` problem with the same measure; thus, the optimum for an instance of the `RPP/`∞`/req` problem is at least the optimum for the corresponding instance of the `Set Cover` problem. Recall that the greedy algorithm of [5] is $\log |A|$ approximating and $|A| = |V| = \frac{1+\sqrt{1+8m}}{2} \leq 1 + \sqrt{2m}$. Therefore, since $\log |A| \leq \log 1 + \sqrt{2m} \leq \frac{1}{2} \log m + 1$, in order to prove the claim it remains to show that, given an instance I of the `RPP/`∞`/req` problem and the corresponding instance I' of the `Set Cover` problem, to any solution $U = \{s_{v_1}, \ldots, s_{v_t}\}$ of I' can be associated a solution of I with measure at most $3t - 2$.

Let $G' = (V, E')$ be the graph having the same node set of G and the edge set $E' \subseteq E$ defined as follows: for each set in $s_{v_i} \in S$ $(i = 1, \ldots, t)$, we add to E' all the edges of E having as endpoints nodes in s_{v_i}. Notice that, since the sets in S cover all the nodes in V, each edge in the set $E \setminus \mathrm{E}'$ connects two different connected components of G'. Let $E'' \subseteq E \setminus E'$ be a set of edges containing for each couple of connected components of G' one edge in $E \setminus E'$ reconnecting them, if it exists. We now compute a spanning tree $T = (V, E_T)$ of the graph $(V, E' \cup E'')$, that by construction is a connected graph. The solution for the instance I of the `RPP/`∞`/req` problem is obtained by selecting as paths (between all the couples of nodes) the simple ones in T, and by placing regenerators at the following locations: t locations are the vertices $v_1, \ldots, v_t$; moreover, for each edge $e = (u, v) \in E_T \cap E''$, we select as locations both u and v. See Figure 4 for an example of an association between a solution of the `Set Cover` problem and the corresponding one of the `RPP/`∞`/req` problem.

Since in G' there are at most t connected components, in the spanning tree T $|E_t \cap E''| \leq t - 1$; therefore, the number of locations is at most $t + 2(t - 1) = 3t - 2$. Finally, it is easy

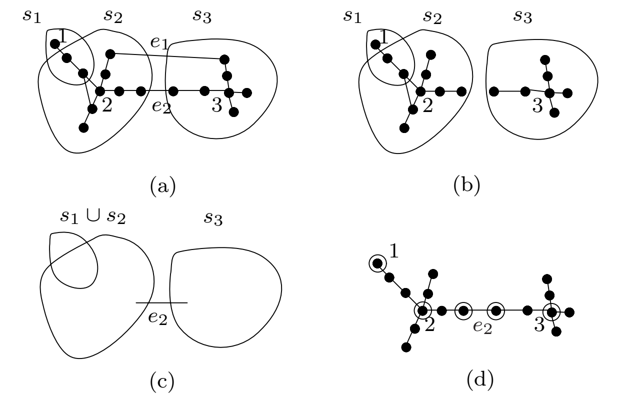

Figure 4: (a) The graph G with an associated solution $\{s_1, s_2, s_3\}$ of the corresponding instance of the `Set Cover` problem, for the case $d = 2$. (b) The graph G' associated to G. (c) The selected edge of E''. (d) The solution for the `RPP/∞/req` problem (regenerators are located at the circled nodes).

to check that such a solution is a feasible one since in any path of T there is a regenerator at least every d edges. □

2.4 Problem `RPP/k/req`

In this section we study the `RPP/k/req` problem; we prove that is NP-Complete to find a feasible solution for the problem if $k = 1$ and $d = 1$.

Theorem 2.9. *Given an instance of the* `RPP/k/req`*, the problem of deciding whether there exists a feasible solution is NP-Complete when $k = 1$ and $d = 1$.*

Proof. Since checking if a solution for the `RPP/k/req` problem is feasible is easily doable in polynomial time, the problem belongs to NP.

In order to prove its NP-Completeness, we provide a polynomial reduction from the `Disjoint paths` problem, known to be NP-Complete (see [8]).

`Disjoint connecting paths`
`Instance:` A Graph $G = (V, E)$, a collection of x disjoint vertex pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_x, t_x)$
`Question:` Does G contain x mutually vertex-disjoint paths $P_1, P_2, \ldots, P_x$ such that P_i joins s_i and t_i for $i = 1, \ldots, x$?

Given an instance of the `Disjoint connecting paths` problem, $G = (V, E)$, $(s_1, t_1), (s_2, t_2), \ldots, (s_x, t_x)$, we construct an instance of `RPP/k/req` as follows: the network is the same and the entry $a_{i,j}$ of the *demand matrix* $A_{n\times n}$ is equal to 1 if and only if for some $z = 1, \ldots, x$, $i = s_z$ and $j = t_z$. Putting $d = 1$ and $k = 1$ easily follows that a solution is feasible for the `RPP/k/req` problem if and only if the answer to the `Disjoint connecting paths` problem is *yes*.

In fact, if the answer to the `Disjoint connecting paths` problem is *yes*, a feasible solution for the `RPP/k/req` problem can be obtained by routing each request relative to a 1-entry $a_{i,j}$ of the *demand matrix* (corresponding to the vertex pair (s_z, t_z)) on the path P_z. Since the paths $P_1, P_2, \ldots, P_x$ are mutually vertex-disjoint, it is possible to put a regenerator at each intermediate node of each path, thus obtaining a feasible solution for the `RPP/k/req` problem.

Conversely, given a feasible solution for the `RPP/k/req` problem, since when $d = 1$ a regenerator is needed at each intermediate node of the paths used for the routing of the requests, by recalling that $k = 1$ it follows that such paths are vertex-disjoint. □

Since it is difficult even to find a feasible solution for the considered problem, the following corollary holds.

Corollary 2.10. *The* `RPP/k/req` *problem is not approximable in polynomial time, i.e.* `RPP/k/req` $\notin$ `Exp–Apx` *unless* $P = NP$.

3. THE NON-SIMPLE PATHS CASE

In this section we consider the case in which the routing can be done using non-simple paths. Notice that in such a case a lot of location on which regenerators have to be put can be saved. For instance, consider the network in Figure 5; there are n requests from a_i to b_i for $i = 1, \ldots, n$. If $d = 3$ and $k \geq n$, by routing all such requests on the non-simple path of length 6 crossing node r, regenerators have to be located only at node r. In fact, regenerators are electro-optical converters, and therefore a path could be composed by two or more lightpaths crossing a same edge and using different colors. Conversely, if only simple paths are considered for the routing, n locations (say $v_1, \ldots, v_n$) are needed.

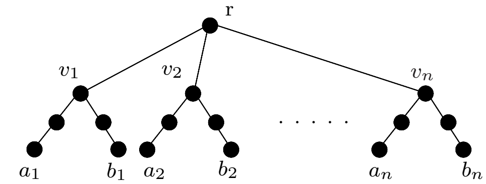

Figure 5: An instance in which the use of non simple paths is helpful.

All the (negative) results about the hardness and the inapproximability of the simple path case still hold for this (more general) case. The extension is trivial for the proofs relative to the problems in which the paths are given (Theorems 2.3 and 2.5 and Corollary 2.6). Moreover, it is easy to check that the proofs of Theorems 2.7 and 2.9 and Corollary 2.10 still hold since, being $d = 1$, any solution using non-simple paths can be converted in a better one using only simple paths.

Therefore, in the following of this section we focus on the positive algorithmic results of `RPP/∞/rt` and `RPP/∞/req` problems.

3.1 Problem `RPP/∞/rt`

The algorithms of Section 2.1.1 for tree and ring networks work when the given routing contains only simple paths. Nevertheless, it is easy to check that, since we are dealing with tree and ring topologies, every instance I containing non simple paths can be converted in another instance I' having only simple paths. In fact, let us consider a non simple paths and let v be the last vertex of it such that the path till v is simple; since two lightpaths sharing an edge must use different colors, a color conversion is needed at node v and we split the initial path in two paths, the first one being simple; v is called a *split-node*. Such a process can be iterated until the initial non simple paths is split in simple paths. Moreover, since we are interested in minimizing the number of locations where regenerators are placed, in order to obtain I', for each split-node v we split at v all the paths crossing it. The output of the (optimal) algorithm for the initial instance I with non-simple paths is finally given by the union of the set of split-nodes (where regenerators are needed since color conversions have to be performed) and the set given in output by the algorithm on the instance I' containing only simple paths.

Concerning general network topologies, the algorithm of Theorem 2.4 can be extended, and the following theorem holds.

THEOREM 3.1. *Let Δ be the maximum number of times that a given path of the instance goes through a same node. It is possible to find a solution for* RPP/∞/rt *problem in which non-simple paths are allowed with approximation ratio $O(\log m + \log d + \log \Delta)$.*

3.2 Problem RPP/∞/req

The RPP/∞/req problem has unexpected similarities with the power consumption minimization problem in ad-hoc wireless networks. In some sense, a regenerator at a node corresponds to a wireless station at that node whose range assignment to transmit all the nodes at distance at most d costs 1. If non simple paths are allowed, thanks to such similarities and by exploiting ideas in [3], it is possible to provide an approximation algorithm asymptotically matching the inapproximability result $\Omega(\log m)$.

THEOREM 3.2. *For any demand matrix it is possible to find a solution of the* RPP/∞/req *problem in which non-simple paths are allowed with approximation ratio $3.22 \log m$.*

PROOF. In order to provide the claimed $3.22 \log m$-approximation algorithm we transform an instance I of the RPP/∞/req problem in an instance I' of the Node Weighted Steiner Forest problem such that a ρ-approximating solution for I' induces a 2ρ-approximation for I.

The Node Weighted Steiner Forest is defined as follows.

`Node Weighted Steiner Forest`

`Input:` An undirected graph $G' = (V', E')$ with a node cost function $c : V' \to R^+$ and a set of nodes $D \subseteq V'$ partitioned into p disjoint sets $D_1, \ldots, D_p$.

`Output:` A forest subgraph H of G' such that any two nodes belonging to the same set D_l, $1 \le l \le p$, are connected by a path in H.

`Measure of a solution:` total cost of H, that is the sum of the costs of its nodes.

`Objective:` Minimizing the measure of the returned solution.

Given the demand matrix A and the graph G corresponding to the instance I of RPP/∞/req, I' is constructed as follows. G' contains a 0-node z_i and a 1-node o_i for every $i \in V$ of G, of costs 0 and 1, respectively. There is an edge between z_i and o_i for every i, $1 \le i \le n$. Moreover, every o_i is connected to every z_j such that j is at distance at most d from i in G.

Sets $D_1, \ldots, D_p$ correspond to the connected components of the *demand graph* (namely, the graph in which the node set is the same of G, and there is an edge between i and j if there is a request between i and j in A, that is $A_{i,j} > 0$). Then, each D_l is the set of the 0-nodes z_i of all the nodes i contained in a distinguished connected component of the demand graph.

Every solution of I induces a solution of I' having the same cost. Such a solution contains all the 0-nodes which correspond to endpoints of requests, plus all the 0- and 1-nodes corresponding to the nodes of G in which I puts a regenerator. Clearly, this solution is feasible for I', because, since in I for every request (i, j) there is a path from i to j containing a sequence of regenerators $t_1, \ldots, t_h$ placed at most every d nodes, in I' there is a path from z_i to z_j, whose edges in I' in the order are $\{z_i, o_i\}$, $\{o_i, z_{t_1}\}$, $\{z_{t_1}, o_{t_1}\}$, $\{o_{t_1}, z_{t_2}\}$, ... $\{o_j, z_j\}$. As a consequence $opt(I') \le opt(I)$.

Conversely, consider any solution H of I' and the corresponding solution of the instance I of RPP/∞/req in which there is a regenerator in every node whose corresponding 1-node belongs to H. In such a solution, the routing is obtained by associating to each edge in G' between a 0-node z_i and a 1-node o_j a lightpath in G (of length at most d) between i and j. Notice that the concatenation of such lightpath may result in a non-simple path. Unfortunately, I' might be unfeasible, since in the path in G from i to j induced by the path from z_i to z_j in H, we are guaranteed of the existence of a regenerator at most every $2d$ nodes. This can happen if such a path in H does not alternate horizontal edges, that is between 0- and 1-nodes with the same index, and vertical ones, between 0- and 1-nodes with different indices. In fact, for each subpath corresponding to an edge of H, a regenerator is guaranteed to be placed only at the endpoint corresponding to its 1-node. However, feasibility can be recovered by at most doubling the number of regenerators as follows. For each subtree T corresponding to a connected component of the Steiner forest H, select a leaf as source, and whenever there are two consecutive vertical edges $\{o_i, z_t\}, \{z_t, o_j\}$ in a path from such source towards another leaf, add a regenerator at node t corresponding to the middle of the two vertical edges. Thus, charging this regenerator to o_j, since o_j has only one father z_t, we have that at the end of the process every 1-node has been charged with at most one regenerator. The new solution is feasible because the path connecting a given request between two nodes i and j is obtained concatenating the subpaths of length at most d in G corresponding to the single edges in

H from z_i to z_j, and such subpaths have a regenerator at both their endpoints, except at i and j.

In conclusion, the final number of regenerators is at most twice the number of 1-nodes in H, that is it has been at most doubled, and the obtained solution is feasible. Therefore, given any ρ-approximate solution for I' of cost m, we get a solution for I of cost at most $2m \leq 2\rho \cdot opt(I') \leq 2\rho \cdot opt(I)$. The theorem then follows by the $1.61 \ln |D|$-approximating algorithm presented in [9], by observing that $m \leq |D|$. □

Notice that the above approximation ratio is actually logarithmic in the number of nodes being endpoints of at least one request. In the all-to-all case, such a number is about $\sqrt{m}$, thus giving an approximation ratio roughly equal to $1.61 \ln m$.

4. DISCUSSION AND OPEN PROBLEMS

In this paper we have presented basic fundamental results concerning the problem of minimizing the number of locations for placing of generators in optical networks. We considered four cases, depending on whether or not there is a bound on the number of regenerators placed in a single node, and depending on whether or not the routing is given. Moreover, we deal both with the case in which only simple path are allowed for the routing, and with the case in which the paths can be non-simple.

The main open problem is solving the $\texttt{RPP}/\infty/\texttt{req}$ problem in the general case when only simple paths are allowed.

This is the first study of these problems, and it suggests many possible extensions, such as considering the objective function of minimizing the total number of regenerators, adding some constraint on the number of colors, solving the on-line version of any of these problems and dealing with specific network topologies. Finally, it would be worthy of considering the general case where each edge e has a weight $w(e)$ (we assumed $w(e) = 1$ for every edge e), and the constraint is that the signal never travels a path whose weight (that is the sum of weights of its edges) is greater than d.

5. REFERENCES

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela and M. Protasi. Complexity and Approximation, Combinatorial Optimization Problems and Their Approximability Properties. *Springer-verlag*, 1999.

[2] C. A. Brackett, A. S. Acampora, J. Sweitzer, G. Tangonan, M. T. Smith, W. Lennon, K.-C. Wang, R.H. Hobbs. A scalable multiwavelength multihop optical network: a proposal forresearch on all-optical networks. *Journal of Lightwave Technology*, 11(5), pp. 736-753, 1993.

[3] I. Caragiannis, C. Kaklamanis and P. Kanellopoulos. Energy-Efficient Wireless Network Design. *Theory of Comput. Syst.*, 39(5), pp. 593-617, 2006.

[4] S. Chen and S. Raghavan. The Regenerator Location Problem. In *Proceedings of the International Network Optimization Conference (INOC 2007)*.

[5] V. Chvatal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4(3), pp. 233-235, 1979.

[6] T. Erlebach and S. Stefanakos. Wavelength Conversion in All-Optical Networks with Shortest-Path Routing. *Algorithmica*, 43(1-2), pp. 43-61, 2005.

[7] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4), pp. 634-652, 1998.

[8] M. Garey and D. S. Johnson. Computers and Intractability, A Guide to the Theory of NP-Completeness. *Freeman*, 1979.

[9] S. Guha and S. Khuller. Improved Methods for Approximating Node Weighted Steiner Trees and Connected Dominating Sets. *Infomation and Computation*, 150(1), pp. 57-74, 1999.

[10] J. M. Kleinberg and A. Kumar. Wavelength Conversion in Optical Networks. *Journal of Algorithms*, 38(1), pp. 25-50, 2001.

[11] S. W. Kim and S. W. Seo. Regenerator placement algorithms for connection establishment in all-optical networks. *IEEE Proceedings Communications*, 148(1), pp. 25-30, 2001.

[12] S. Pachnicke, T. Paschenda and P. M. Krummrich. Physical Impairment Based Regenerator Placement and Routing in Translucent Optical Networks. *Proceedings of the Optical Fiber communication/National Fiber Optic Engineers Conference, (OFC/NFOEC 2008)*, pp. 1-3, 2008.

[13] B. Ramamurthy, H. Feng, D. Datta, J. P. Heritage and B. Mukherjee. Transparent vs. opaque vs. translucent wavelength-routed optical networks. *Proceedings of the Optical Fiber Communication Conference and the International Conference on Integrated Optics and Optical Fiber Communication, (OFC/IOOC 99)*, pp. 59-61, 1999.

[14] C. V. Saradhi and S. Zaks. Placement of regenerators in optical networks. *In preparation*.

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

[16] G.T. Wilfong and P. Winkler. Ring Routing and Wavelength Translation. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA 1998)*, pp. 333-341, 1998.

[17] X. Yang and B. Ramamurthyn. Sparse Regeneration in Translucent Wavelength-Routed Optical Networks: Architecture, Network Design and Wavelength Routing. *Photonic Network Communications*, 10(1), pp. 39-53, 2005.

[18] X. Yang and B. Ramamurthy. Dynamic routing in translucent WDM optical networks. *Proceedings of the IEEE International Conference on Communications (ICC 2002)*, pp. 2796-2802, 2002.