



Optimizing regenerator cost in traffic grooming[☆]

Michele Flammini^a, Gianpiero Monaco^a, Luca Moscardelli^{b,*}, Mordechai Shalom^c, Shmuel Zaks^d

^a Department of Computer Science, University of L'Aquila Via Vetoio, Loc. Coppito, 67100 L'Aquila, Italy

^b Department of Science, University of Chieti-Pescara, Viale Pindaro 42, 65127 Pescara, Italy

^c TelHai Academic College, Upper Galilee, 12210, Israel

^d Department of Computer Science, Technion, Haifa, Israel

ARTICLE INFO

Article history:

Received 2 December 2010

Received in revised form 30 August 2011

Accepted 20 September 2011

Communicated by R. Klasing

Keywords:

Optical networks

Wavelength division multiplexing (WDM)

Regenerators

Traffic grooming

Tree networks

ABSTRACT

In optical networks, regenerators have to be placed on lightpaths every d consecutive nodes in order to regenerate the signal. In addition, grooming enables the use of the same regenerator by several lightpaths. Up to g (the grooming factor) lightpaths can use the same regenerator. In this work we consider the problem of minimizing the number of regenerators used in traffic grooming in optical networks. Starting from the 4-approximation algorithm of Flammini et al. (2010) [10] for $d = 1$ and a path topology, we provide an approximation algorithm with the same approximation ratio for $d = 1$ and the ring and tree topologies. We present also a technique based on matching that leads to the same approximation ratio in tree topology and can be used to obtain approximation algorithms in other topologies. We provide an approximation algorithm for general topology that uses this technique. Finally, all the results are extended to the case of general d .

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

In modern optical networks, high-speed signals are sent through optical fibers using WDM (Wavelength Division Multiplexing) technology. Currently deployed networks carry around 80 wavelengths per fiber, whereas networks with a few hundreds of wavelengths per fiber are being used in testbeds. The decrease in the energy of the signal with the traveled distance raises the requirement of optical amplifiers at every (almost) fixed distance. However, optical amplifiers introduce noise into the signal, thus after a certain number of amplifications, the optical signal needs to be regenerated. In the current technology, the signal is regenerated by first using a ROADM (Reconfigurable Optical Add-Drop Multiplexer) to extract a set of wavelengths 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.

Nowadays the cost of a regenerator is considerably higher than the cost of an ROADM. Moreover, as described above, the regenerator cost is per wavelength, as opposed to ROADM cost that is paid once per several wavelengths. Therefore the

[☆] A preliminary version of this paper appeared in the Proceedings of the 14th International Conference On Principles Of Distributed Systems (Opodis), 2010. This work was partially supported by the Israel Science Foundation grant No. 1249/08, by British Council Grant UKTELHAI09, and by the PRIN 2008 research project COGENT (COmputational and Game-theoretic aspects of uncoordinated NeTworks), funded by the Italian Ministry of University and Research.

* Corresponding author. Tel.: +39 0862433189; fax: +39 0862433131.

E-mail addresses: flammini@di.univaq.it (M. Flammini), gianpiero.monaco@di.univaq.it (G. Monaco), moscardelli@di.univaq.it, moscardelli@sci.unich.it (L. Moscardelli), cmshalom@telhai.ac.il (M. Shalom), zaks@cs.technion.ac.il (S. Zaks).

total number of regenerators is an important cost parameter to be minimized [14]. 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 part of the cost is the set-up of new optical nodes, or when the equipment to be placed at each node is the same for all nodes. In this work we consider the total number of regenerators as the cost function.

A logical path formed by a signal traveling from its source to its destination using a unique wavelength is termed a *lightpath*. Let d be the maximum number of hops a lightpath can make without meeting a regenerator. Then, for each lightpath ℓ , we need to place one regenerator every d consecutive vertices in ℓ , to get an optimal solution. However the problem becomes harder when the *traffic grooming* comes into the picture.

Traffic grooming: The network usually supports traffic that is at rates which are lower than the full wavelength capacity, and therefore the network operator has to be able to put together (= groom) low-capacity connections into the high capacity lightpaths. In graph-theoretic terms, we associate a path in the graph with each connection, and the problem can be viewed as assigning wavelengths to these paths so that at most g of them use the same wavelength (g being the *grooming factor*) can share one edge. Thus, all paths (i.e., connections) that get the same color (i.e., the same wavelength) and form a connected subgraph correspond to grooming of these connections into one lightpath.

An ADM (Add-Drop Multiplexer) is a hardware component operating at a node of the optical network, able to insert sub-wavelength signals into an existing lightpath, and also to drop sub-wavelength signals from a lightpath. Other sub-wavelength signals (i.e. those that travel through this node) are routed by an optical router, without being converted to electrical form and vice versa. Such a router will route all such signals to a certain incident edge of the node. If such signals have to be routed to distinct incident edges, then they can be dropped from their source lightpath by one ADM and then added to another lightpath by another ADM, however this introduces costly O-E-O conversions which in turn cause delays. There are two different models that can be considered, depending on whether these delays are acceptable. In this work, we consider the case that such delays are unacceptable, and therefore O-E-O conversions are not used in intermediate nodes. Consequently we introduce the *no-splitting condition* in our problem definition. Actually the option of using O-E-O conversions exists also in the case that all the signals continue to the same destination edge. This option might be useful to reduce potential ADM costs in other nodes. We refer to works on both models in the following subsection.

1.1. Related work

Various variants of regenerator placement problems were studied in [4,7,8,13,15,16,18,19]. Most of these results concentrate in heuristics and simulations and do not consider traffic grooming.

In [9] theoretical results (upper bounds and lower bounds) are presented for some variants of this problem. This work considers the number of regenerator locations (as opposed to the total number of regenerators) as the cost measure, and does not consider traffic grooming. On the other hand, [14] uses the same cost measure of this paper but still does not consider traffic grooming.

The problem we study is shown to be NP-hard in other contexts such as fiber minimization in [17] and its NP-hardness is also implied by the proof of a similar result in [11] holding even for path topology and $g = 2$.

When the underlying graph is a path the problem is equivalent to a machine scheduling problem studied in [10]. In particular, the path topology corresponds to a timeline in which every node is an instant; each path corresponds to a job starting at the instant corresponding to its leftmost node and ending to the instant corresponding to its rightmost node; g is the maximum number of jobs a processor can execute at the same time. Such a scheduling problem aims at minimizing the sum of the computation time of all the machines, provided that an unbounded number of machines is available. Several approximation algorithms are presented in [10] for this scheduling problem and its special cases.

In the literature, two different scenarios have been studied, depending on whether or not it is allowed to split the paths in order, for instance, to reduce the number of used wavelengths or the cost of hardware components. In particular, [1,3] assume that no splitting is allowed, while [2] allows to split paths and [12] considers both scenarios.

1.2. Our contribution

In this work we consider the traffic grooming problem to minimize the number of regenerators used. We first consider the case $d = 1$, i.e., the case that a regenerator has to be placed at every internal node of every lightpath, and then we extend all the results to general d . Our starting point is a 4-approximation algorithm of [10] that solves a closely related problem for a path topology and $d = 1$. We prove that the same algorithm can be used for our problem and show that, always for $d = 1$, it has the same approximation ratio not only for path topology, but also for ring topology. We present a greedy 4-approximation algorithm for tree networks. We also show a general technique using matchings that can lead to approximation algorithms in other topologies. We use this technique and show an $\lfloor \frac{L+7}{2} \rfloor$ -approximation algorithm for general topology and $d = 1$, where L is the maximum load (i.e., number of paths that share a common edge) in the input. It is worth noticing that in this paper we do not consider the hardware cost due to the ADMs, as also [4,7,8,13] do.

In Section 2 we present preliminary results and definitions, including the above mentioned algorithm for path networks and extension of its analysis to the case of ring topology. In Section 3 we present an algorithm with the same performance for tree topology. In Section 4 we present the matching technique and its use for general topologies. In all the above mentioned

sections we always consider the case $d = 1$; in Section 5, apart from summarizing the results and suggesting open research directions, we extend all the results to general d .

2. Preliminaries

2.1. Definitions and problem statement

Until Section 5, in which the results will be extended to any d , we deal with the case $d = 1$.

We will consider instances (G, \mathcal{P}, g) where $G = (V, E)$ is a graph modeling the optical network, \mathcal{P} is a set of simple paths in G , $g \in \mathbb{N}^+ \cup \{\infty\}$ is the grooming factor.

A coloring (or wavelength assignment) of (G, \mathcal{P}) is a function $w : \mathcal{P} \mapsto \mathbb{N}$. For a coloring w and color λ , \mathcal{P}_λ^w is the subset of paths from \mathcal{P} colored λ by w , i.e., $\mathcal{P}_\lambda^w \stackrel{\text{def}}{=} \{P \in \mathcal{P} | w(P) = \lambda\}$. When there is no ambiguity on the coloring w under consideration, we omit the superscript w and use \mathcal{P}_λ .

For a node v , \mathcal{P}_v denotes the subset of paths of \mathcal{P} having v as an intermediate node, and similarly for an edge e , \mathcal{P}_e denotes the subset of paths of \mathcal{P} using the edge e . For every $e \in E$ we define $load(\mathcal{P}, e) \stackrel{\text{def}}{=} |\mathcal{P}_e|$ and $load(\mathcal{P}) \stackrel{\text{def}}{=} \max_{e \in E} load(\mathcal{P}, e)$. A valid coloring (or wavelength assignment) w of (G, \mathcal{P}, g, d) is a coloring of \mathcal{P} in which for any edge e at most g paths using e are colored with the same color, i.e., for every color λ we have $load(\mathcal{P}_\lambda^w) \leq g$.

We denote by $INT(P)$ the set of intermediate nodes, i.e., of all the nodes not being endpoints, of a path P in G , and $int(P) \stackrel{\text{def}}{=} |INT(P)|$. For a set \mathcal{P} of paths we define

$$\begin{aligned} SPAN(\mathcal{P}) &\stackrel{\text{def}}{=} \bigcup_{P \in \mathcal{P}} INT(P), \\ span(\mathcal{P}) &\stackrel{\text{def}}{=} |SPAN(\mathcal{P})|, \\ len(\mathcal{P}) &\stackrel{\text{def}}{=} \sum_{P \in \mathcal{P}} int(P). \end{aligned}$$

A set of paths is called a *no-split instance* or shortly an NSI if the union of its paths (as sets of edges) induces a graph of maximum degree at most 2. Since in this paper we assume (as [1,3,12] do) that splitting paths is not allowed, paths using the same wavelength and going through the same edge of the network can be routed only to another unique edge, and therefore every set of paths with the same color has to be an NSI.

The number of regenerators operating at wavelength λ is $span(\mathcal{P}_\lambda^w)$; in fact, at each node being an intermediate node of some path in \mathcal{P}_λ^w a regenerator operating at this wavelength is needed.

We are now ready to give a formal definition of our problem.

TOTAL REGENERATORS WITH GROOMING (TRG)

Input: A triple (G, \mathcal{P}, g) , where $G = (V, E)$ is a graph, $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$ is a set of simple paths in G , and g is an integer, namely the grooming factor.

Output: A valid coloring $w : \mathcal{P} \mapsto \mathbb{N}$ of the paths such that, for every λ , \mathcal{P}_λ is an NSI (we will refer to the latter condition as the *no splitting condition* thorough this work).

Objective: The cost of a solution is given by the total number of regenerators $REG^w \stackrel{\text{def}}{=} \sum_\lambda span(\mathcal{P}_\lambda^w)$. The goal is to minimize the total number of regenerators REG^w .

$OPT(G, \mathcal{P}, g)$ denotes the cost of any optimal coloring and $ALG(G, \mathcal{P}, g)$ denotes the cost of the coloring returned by some algorithm ALG on instance (G, \mathcal{P}, g) . As the cost function depends only on the partition of the paths induced by the coloring, with some abuse of notation, a coloring w denotes also the equivalence class of colorings that induce the same partition as w .

2.2. Lower bounds

We have the following trivial lower bounds for the cost of any coloring w , in particular for an optimal coloring.

- The grooming bound:

$$REG^w \geq \frac{len(\mathcal{P})}{g}.$$

- The span bound:

$$REG^w \geq span(\mathcal{P}).$$

The grooming bound holds because a regenerator can be used by a maximum of g intermediate nodes of paths. The span bound holds because at least one regenerator is needed on any node that is an intermediate node of some path.

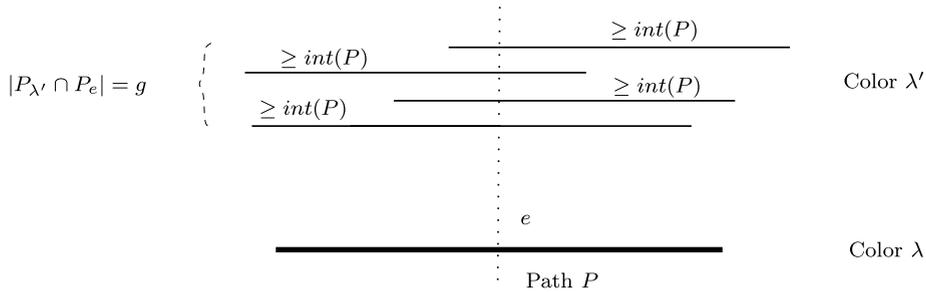


Fig. 1. Basic observation.

2.3. Path and ring networks

In this subsection we focus on ring and path networks. We adapt Theorem 2.1 in [10] to our problem and generalize it to the case of ring networks. Specifically, we show that the *FirstFit* algorithm presented in [10] is a 4-approximation algorithm for our problem. The proof goes along the same lines, and we bring it here for sake of completeness; the main difference is in Lemma 2, whose proof required modifications of the proof of the corresponding claim in [10] in order to assure correctness for the case of ring topology.

Notice that when G is a ring or a path, all subsets of \mathcal{P} constitute an NSI.

Algorithm *FirstFit* colors the paths greedily by considering them one after the other, from longest to shortest. Each path is assigned the lowest possible color for it.

Algorithm 1 *FirstFit*(G, \mathcal{P}, g) with G being a path or a ring

- 1: Sort the paths in non-increasing order of length, i.e., $int(P_1) \geq int(P_2) \geq \dots \geq int(P_n)$.
- 2: Consider the paths by the above order and, for any path $P_j, j \in \{1, \dots, n\}$, assign to P_j the first possible color λ that will not violate the load condition. Namely, find the minimum value $\lambda \geq 1$ such that, for every edge e of P_j , $load(\mathcal{P}_\lambda, e) \leq g - 1$ and set $w(P_j) \leftarrow \lambda$.

The upper bound proof is based on the observation stated in the following lemma, and depicted in Fig. 1.

Lemma 1. Let w be the coloring returned by *FirstFit*. Let P be a path colored λ , i.e., $P \in \mathcal{P}_\lambda^w$, for some $\lambda \geq 2$. Then for any $\lambda' < \lambda$, (a) there is an edge $e \in P$ such that $load(\mathcal{P}_{\lambda'}^w, e) = g$, (b) each path $P' \in \mathcal{P}_{\lambda'} \cap \mathcal{P}_e$ is no shorter than P .

Proof. In order to prove property (a), assume by contradiction that for some $\lambda' < \lambda$ and every edge $e \in P$, $load(\mathcal{P}_{\lambda'}^w, e) \leq g - 1$. Since the algorithm assigns colors to paths incrementally (i.e., it never un-colors paths), this condition should have held at the point *FirstFit* considered P for coloring. In this case, P should have been colored at most λ' by *FirstFit*, a contradiction to the fact that P is colored λ .

Property (b) follows from property (a) and the fact that the paths are considered by the algorithm in a non-increasing order of their lengths. \square

We use the property stated in Lemma 1 in order to show the following claim, which will be crucial in order to prove the desired result.

Lemma 2. Let $W \geq 1$ be the number of colors used by w ; for any $1 < \lambda \leq W$, $len(\mathcal{P}_{\lambda-1}) \geq \frac{g}{3} span(\mathcal{P}_\lambda)$.

Proof. For every path $P \in \mathcal{P}_\lambda$, we fix $\lambda' = \lambda - 1$ and choose arbitrarily an edge e of P among those whose existence is guaranteed by Lemma 1. Let $b(P) \stackrel{def}{=} \mathcal{P}_{\lambda-1} \cap \mathcal{P}_e$ be the blocking paths of P . By Lemma 1, $|b(P)| = g$ and $len(b(P)) \geq g \cdot int(P)$. Let $\overline{\mathcal{P}}$ be the set of all blocking paths defined as above, i.e., $\overline{\mathcal{P}} \stackrel{def}{=} \cup_{P \in \mathcal{P}_\lambda} b(P)$. Clearly $\overline{\mathcal{P}} \subseteq \mathcal{P}_{\lambda-1}$.

Now, we consider a blocking path $P' \in \overline{\mathcal{P}}$ (see Fig. 2). Consider the set of all paths in \mathcal{P}_λ blocked by P' . With a little abuse of notation we denote them by $b^{-1}(P')$. Consider a node $v \in SPAN(b^{-1}(P'))$. It is in some path $P'' \in \mathcal{P}_\lambda$ which is no longer than P' and intersects with P' , therefore there exists an intermediate node of P' which is at distance to v at most $int(P'') \leq int(P')$. As G is a path or a cycle the number of such nodes v is at most $3 \cdot int(P')$. We conclude that $span(b^{-1}(P')) \leq 3 \cdot int(P')$. Summing up for all the paths in $\overline{\mathcal{P}}$ we get

$$\sum_{P' \in \overline{\mathcal{P}}} span(b^{-1}(P')) \leq 3 \sum_{P' \in \overline{\mathcal{P}}} int(P') = 3 \cdot len(\overline{\mathcal{P}}).$$

Consider a node $v \in SPAN(\mathcal{P}_\lambda)$. It is an intermediate node of at least one path $P \in \mathcal{P}_\lambda$, which in turn is blocked by at least g paths of $\overline{\mathcal{P}}$. Therefore $v \in SPAN(b^{-1}(P'))$ for at least g paths P' of $\overline{\mathcal{P}}$, in other words v contributes at least g to the sum in the left hand side above. Thus we have $\sum_{P' \in \overline{\mathcal{P}}} span(b^{-1}(P')) \geq g \cdot span(\mathcal{P}_\lambda)$. Therefore,

$$3 \cdot len(\mathcal{P}_{\lambda-1}) \geq 3 \cdot len(\overline{\mathcal{P}}) \geq \sum_{P' \in \overline{\mathcal{P}}} span(b^{-1}(P')) \geq g \cdot span(\mathcal{P}_\lambda). \quad \square$$

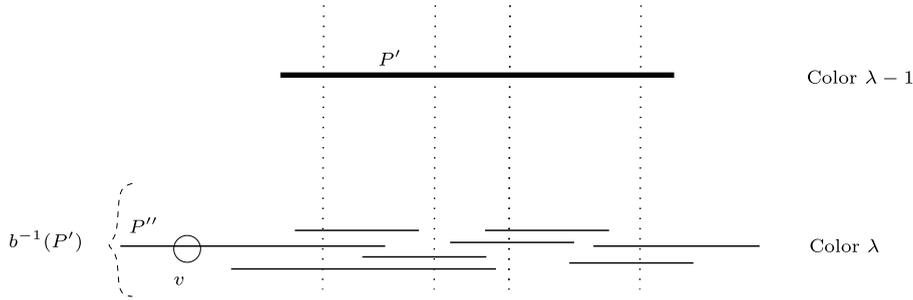


Fig. 2. Analysis of a path P' colored $\lambda - 1$.

The following theorem, providing an upper bound to the approximation ratio of the *FirstFit* algorithm, and Lemma 3 follow from similar claims in [10].

Theorem 1. *If G is a path or a ring, then for any instance (G, \mathcal{P}, g) , $\text{FirstFit}(G, \mathcal{P}, g) \leq 4 \cdot \text{OPT}(G, \mathcal{P}, g)$.*

Proof. Combining the span bound and Lemma 2 we can now complete the analysis of the algorithm.

Let $W \geq 1$ be the number of colors used by w , then $\text{FirstFit}(G, \mathcal{P}, g) = \sum_{\lambda=1}^W \text{span}(\mathcal{P}_\lambda^w)$. Moreover

$$\begin{aligned} \sum_{\lambda=2}^W \text{span}(\mathcal{P}_\lambda) &= \sum_{\lambda=1}^{W-1} \text{span}(\mathcal{P}_{\lambda+1}) \leq \frac{3}{g} \sum_{\lambda=1}^{W-1} \text{len}(\mathcal{P}_\lambda) \\ &\leq \frac{3}{g} \sum_{\lambda=1}^W \text{len}(\mathcal{P}_\lambda) = \frac{3}{g} \text{len}(\mathcal{P}) \leq 3 \cdot \text{OPT}(G, \mathcal{P}, g), \end{aligned}$$

where the last inequality follows from the grooming bound.

Using the span bound, we have $\text{span}(\mathcal{P}_1) \leq \text{span}(\mathcal{P}) \leq \text{OPT}(G, \mathcal{P}, g)$. Therefore, $\text{FirstFit}(G, \mathcal{P}, g) \leq 4 \cdot \text{OPT}(G, \mathcal{P}, g)$. \square

Lemma 3. *For any $\epsilon > 0$, there are infinitely many instances (G, \mathcal{P}, g) having arbitrarily large input sizes, such that $\text{FirstFit}(G, \mathcal{P}, g) > (3 - \epsilon) \cdot \text{OPT}(G, \mathcal{P}, g)$.*

Proof. Consider the instance (G, \mathcal{P}, g) depicted in Fig. 3 where m is a non-zero integer. For this instance an optimal solution uses one regenerator in each one of the nodes $1, \dots, m$, one regenerator in each node $2m+1, \dots, 3m$, and $g-1$ regenerators in the nodes $m+1, \dots, 2m$, for a total cost of $m(g+1)$. To see that this is an optimal solution, observe that this solution achieves the grooming bound tightly. In contrast as all the paths have the same length, they might be sorted in any order by the Step 1 of Algorithm *FirstFit*. If they happen to be sorted in the order shown in the figure from top to bottom, then *FirstFit* will use g regenerators in $1, \dots, 3m$ for a total cost of $3mg = 3 \cdot \frac{g}{g+1} \text{OPT}(G, \mathcal{P}, g)$. Choosing g sufficiently large we get $\text{FirstFit}(G, \mathcal{P}, g) > (3 - \epsilon) \text{OPT}(G, \mathcal{P}, g)$. \square

Combining Theorem 1 and Lemma 3, we finally get the following theorem.

Theorem 2. *The approximation ratio of FirstFit is between 3 and 4 in ring and path networks.*

3. Tree networks

In this section we present an optimal algorithm *GreedyMatch* for the case where the graph G is a tree and $g = \infty$. Combining this algorithm and algorithm *FirstFit* described in the previous section we obtain a 4-approximation algorithm for tree networks and any value of g .

3.1. G, \mathcal{P}, ∞ instances

We first consider the special case of $g = \infty$, that will be useful in order to provide an approximation algorithm for general g . When $g = \infty$, any solution is a valid coloring. It remains to satisfy the no splitting condition. Therefore the problem becomes to partition \mathcal{P} into no-split instances $\mathcal{N}_1, \mathcal{N}_2, \dots$ such that $\sum_{\lambda} \text{span}(\mathcal{N}_\lambda)$ is minimized.

Note that the span (lower) bound holds in this special case, i.e., $\text{OPT}(G, \mathcal{P}, \infty) \geq \text{span}(\mathcal{P})$.

Since $g = \infty$, we can assume that there is no path $P \in \mathcal{P}$ completely included in another path $P' \in \mathcal{P}$, because in this case we could remove P from the input. In any solution of the remaining instance P can be added to the NSI containing P' without increasing the cost.

We introduce some additional notation.

- Two NSIs \mathcal{N} and \mathcal{N}' are said to be *compatible* if their union is also an NSI. We denote this fact as $\mathcal{N} \sim \mathcal{N}'$. Otherwise they are said to be *incompatible* and denoted as $\mathcal{N} \not\sim \mathcal{N}'$.
- The overlap of two NSIs \mathcal{N} and \mathcal{N}' is $OV(\mathcal{N}, \mathcal{N}') \stackrel{\text{def}}{=} \text{SPAN}(\mathcal{N}) \cap \text{SPAN}(\mathcal{N}')$ and $ov(\mathcal{N}, \mathcal{N}') \stackrel{\text{def}}{=} |OV(\mathcal{N}, \mathcal{N}')|$.
- Two NSIs \mathcal{N} and \mathcal{N}' are *overlapping* if $ov(\mathcal{N}, \mathcal{N}') > 0$.

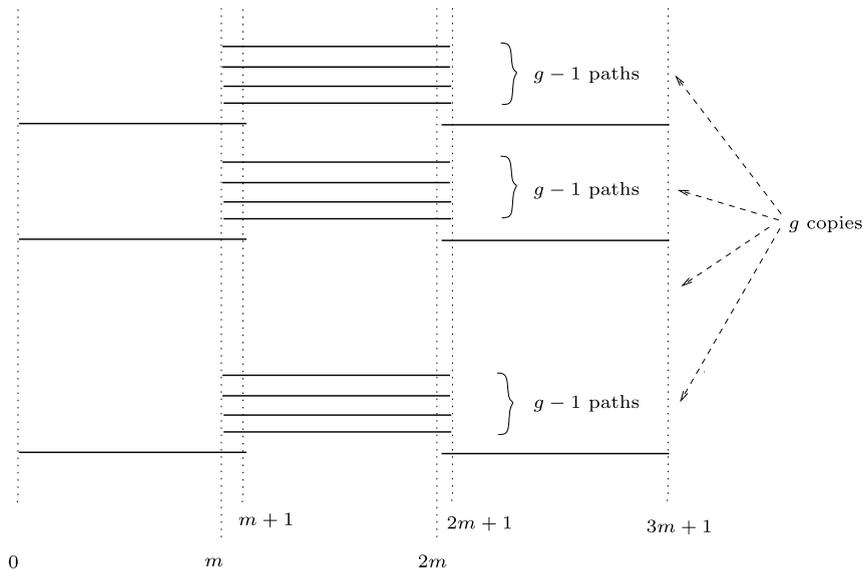


Fig. 3. Instance for the proof of the lower bound.

- An NSI is said to be *connected* if the union of its paths (as sets of edges) induces a connected graph.
- We say that $\mathcal{N} \sqsubseteq \mathcal{N}'$ if $\cup_{P \in \mathcal{N}} P \subseteq \cup_{P \in \mathcal{N}'} P$.

Consider algorithm *GreedyMatch*; the following lemmata are needed for proving [Theorem 3](#), in which it is shown that such an algorithm is optimal.

Lemma 4. *Every two NSIs in an optimal solution of (G, \mathcal{P}, ∞) are either non-overlapping or incompatible.*

Proof. Assume, by contradiction that there are two NSIs that are both compatible and overlapping. Then they can be joined to form one NSI, and decrease the cost of the solution by the size of their overlap. \square

Algorithm 2 *GreedyMatch* (G, \mathcal{P}, ∞) , G being a tree

- | | |
|--|---|
| <ol style="list-style-type: none"> 1: $\forall P_i \in \mathcal{P}, \mathcal{N}_i \leftarrow \{P_i\}$ 2: while there exist $\mathcal{N}_i, \mathcal{N}_j$ such that $\mathcal{N}_i \sqsubseteq \mathcal{N}_j$ do 3: $\mathcal{N}_j \leftarrow \mathcal{N}_j \cup \mathcal{N}_i$ 4: $\mathcal{N}_i \leftarrow \emptyset$ 5: end while 6: while there exist two compatible and overlapping NSIs do 7: Find two compatible NSIs $\mathcal{N}_i, \mathcal{N}_j$ maximizing $ov(\mathcal{N}_i, \mathcal{N}_j)$ 8: $\mathcal{N}_j \leftarrow \mathcal{N}_j \cup \mathcal{N}_i$ 9: end while | <ul style="list-style-type: none"> \triangleright Every path constitutes a connected NSI. \triangleright Eliminate inclusions |
|--|---|
-

We note that at the beginning of the algorithm each sets \mathcal{N}_i consists of a single path, therefore connected. Moreover, since a new NSI is constructed by unifying two compatible and overlapping NSIs, this condition continues to hold at any point of the execution of Algorithm. Therefore, the following observation holds:

Observation 1. *At any given point of the execution of Algorithm *GreedyMatch* the sets \mathcal{N}_i are connected.*

Lemma 5. *When the Algorithm *GreedyMatch* reaches Step 7, there are no inclusions.*

Proof. Inclusions are eliminated at the beginning of the algorithm. Therefore the claim is correct for the first time the algorithm reaches Step 7. Consider the first time that the claim is falsified, i.e., there is an NSI included in another. This can only happen at Step 8 of the previous iteration. In this step the algorithm unifies two NSIs $\mathcal{N}_i, \mathcal{N}_j$. Assume that there is some \mathcal{N}_k such that $\mathcal{N}_k \sqsubseteq (\mathcal{N}_i \cup \mathcal{N}_j)$. As there were no inclusions before this step $\mathcal{N}_k \not\sqsubseteq \mathcal{N}_i$ and $\mathcal{N}_k \not\sqsubseteq \mathcal{N}_j$. Then $OV(\mathcal{N}_i, \mathcal{N}_j)$ is a proper subset of $SPAN(\mathcal{N}_k)$. In other words $OV(\mathcal{N}_i, \mathcal{N}_j)$ is a proper subset of $OV(\mathcal{N}_i, \mathcal{N}_k)$. A contradiction to the maximality of $ov(\mathcal{N}_i, \mathcal{N}_j)$. \square

Lemma 6. *At any given point of the execution of Algorithm *GreedyMatch*, after Step 1, consider the partition $\{\mathcal{N}_1, \mathcal{N}_2, \dots\}$. There is an optimal solution $\{\mathcal{N}_1^*, \mathcal{N}_2^*, \dots\}$ such that every \mathcal{N}_i is a subset of some \mathcal{N}_i^* , or in other words the partition given by the algorithm is a refinement of the partition given by some optimal solution.*

Proof. Without loss of generality we can assume that all the NSIs in an optimal solution of (G, \mathcal{P}, ∞) are connected, because if we have a disconnected NSI \mathcal{N} we can replace \mathcal{N} with a connected NSI for each connected component of \mathcal{N} . The claim is obviously true immediately after Step 1 of the algorithm. Assume by contradiction that the claim is false and consider

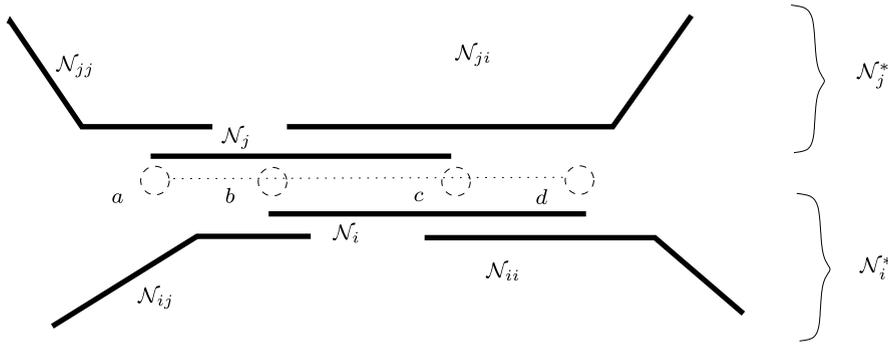


Fig. 4. Proof of Lemma 6.

the first time during the execution of the algorithm that it becomes false. This can happen only after execution of Step 8. \mathcal{N}_i and \mathcal{N}_j are overlapping and compatible, because they are chosen by the algorithm in Step 7. They are also connected by Observation 1. As the condition was true prior to the execution of Step 8, there is some optimal solution $S^* = \{\mathcal{N}_1^*, \mathcal{N}_2^*, \dots\}$ such that $\mathcal{N}_i \subseteq \mathcal{N}_i^*$ and $\mathcal{N}_j \subseteq \mathcal{N}_j^*$. Therefore $\mathcal{N}_i^* \supset \mathcal{N}_i$ and $\mathcal{N}_j^* \supset \mathcal{N}_j$ are overlapping. Also, by Lemma 4, $\mathcal{N}_i^* \approx \mathcal{N}_j^*$.

As, by Lemma 5 there are no inclusions, the $OV(\mathcal{N}_i, \mathcal{N}_j)$ is a proper subset of both $SPAN(\mathcal{N}_i)$ and $SPAN(\mathcal{N}_j)$ (see Fig. 4). Let $a, b, c, d \in V$ be four distinct nodes of the tree such that $SPAN(\mathcal{N}_i)$ (resp. $SPAN(\mathcal{N}_j)$) is the path between b and d (resp. a and c). Then $OV(\mathcal{N}_i, \mathcal{N}_j)$ is the path between b and c . The partition $\{\mathcal{N}_1, \dots\}$ is a refinement of the partition $\{\mathcal{N}_1^*, \dots\}$. Let $\mathcal{N}_i^* = \mathcal{N}_i \uplus \mathcal{N}_{i1} \uplus \mathcal{N}_{i2} \uplus \dots$. We observe that for none of these sets $SPAN(\mathcal{N}_{ik})$ can intersect with both $a - b$ and $c - d$, because this would imply that $OV(\mathcal{N}_i, \mathcal{N}_j) \subsetneq OV(\mathcal{N}_i, \mathcal{N}_{ik})$, a contradiction to the way \mathcal{N}_i and \mathcal{N}_j are chosen by the algorithm. Given this observation we partition the set \mathcal{N}_i^* into three sets $\mathcal{N}_i, \mathcal{N}_{ii}$ and \mathcal{N}_{ij} such that the sets \mathcal{N}_{ik} spanning at least one edge of $c - d$ (resp. $a - b$) are in \mathcal{N}_{ii} (resp. \mathcal{N}_{ij}), the rest are divided arbitrarily. We do the same for \mathcal{N}_j^* .

$\mathcal{N}_i, \mathcal{N}_{ii}, \mathcal{N}_{ij}$ are pairwise compatible, because they make part of \mathcal{N}_i^* , and so are $\mathcal{N}_j, \mathcal{N}_{jj},$ and \mathcal{N}_{ji} . Moreover $\mathcal{N}_{ij} \sim \mathcal{N}_{ji}$ and $\mathcal{N}_{ii} \sim \mathcal{N}_{jj}$, because the underlying graph is a tree and thus they can overlap only in the path $b - c$ in which there cannot exist nodes with induced degree 3 or more.

We conclude the proof by case analysis. For each case we show how an optimal solution S^{*^*} can be built from S^* such that \mathcal{N}_i and \mathcal{N}_j are contained in the same set of S^{*^*} , a contradiction to the assumption that the condition became false.

Assume $\mathcal{N}_{ji} \sim \mathcal{N}_i$:

- $\mathcal{N}_{ji} \sim \mathcal{N}_{ii}$: In this case we can move \mathcal{N}_{ii} and \mathcal{N}_i into \mathcal{N}_j^* without increasing the cost of the solution.
- $\mathcal{N}_{ji} \approx \mathcal{N}_{ii}$: In this case there exists a node belonging to $SPAN(\mathcal{N}_{ji}) \cap SPAN(\mathcal{N}_{ii})$ with induced degree more than 2, and such a node is necessarily beyond (in Fig. 4, at the right of) the node d , proving that $SPAN(\mathcal{N}_{ji})$ contains the path $c - d$. Therefore we can move \mathcal{N}_i into \mathcal{N}_j^* without increasing the cost of the solution.

After handling the case $\mathcal{N}_{ij} \sim \mathcal{N}_j$ symmetrically, it remains to handle the case $\mathcal{N}_{ji} \approx \mathcal{N}_i$ and $\mathcal{N}_{ij} \approx \mathcal{N}_j$. If \mathcal{N}_{ji} and \mathcal{N}_{ij} are overlapping then we can repartition these six sets into two sets $\mathcal{N}_{ij} \cup \mathcal{N}_{ji}$ and $\mathcal{N}_i \cup \mathcal{N}_j \cup \mathcal{N}_{ii} \cup \mathcal{N}_{jj}$ without increasing the cost. Otherwise we build three sets $\mathcal{N}_{ij}, \mathcal{N}_{ji}$ and $\mathcal{N}_i \cup \mathcal{N}_j \cup \mathcal{N}_{ii} \cup \mathcal{N}_{jj}$ without increasing the cost. \square

We are now able to prove that Algorithm GreedyMatch is optimal.

Theorem 3. When Algorithm GreedyMatch ends, the solution $\{\mathcal{N}_1, \mathcal{N}_2, \dots\}$ is optimal.

Proof. By Lemma 6, the solution $\{\mathcal{N}_1, \dots\}$ is a refinement of some optimal solution $\{\mathcal{N}_1^*, \mathcal{N}_2^*, \dots\}$. We claim that these partitions are equal. Assume, by contradiction that there is some set \mathcal{N}_i^* containing at least two of the sets \mathcal{N}_i . These sets are pairwise non-overlapping because they are compatible and the algorithm stopped. Therefore \mathcal{N}_i^* is not connected, a contradiction. \square

3.2. An approximation algorithm scheme for any graph G and any value of g

We propose the algorithm scheme $Combined(\mathcal{A}, (G, \mathcal{P}, g))$ for general graphs and any value of g , depending on a generic Algorithm \mathcal{A} working for the specific case in which $g = \infty$.

Algorithm 3 $Combined(\mathcal{A}, (G, \mathcal{P}, g))$

- 1: Partition \mathcal{P} into NSIs $\mathcal{N}_1, \mathcal{N}_2, \dots$ using algorithm \mathcal{A} computed on the corresponding (G, \mathcal{P}, ∞) instance.
- 2: For each i , let $G(\mathcal{N}_i)$ be the graph induced by the paths of \mathcal{N}_i . Split \mathcal{N}_i into sets $\mathcal{P}_{i,1}, \mathcal{P}_{i,2}, \dots$ by solving the instance $FirstFit(G(\mathcal{N}_i), \mathcal{N}_i, g)$.
- 3: Assign each one of the sets $\mathcal{P}_{i,j}$ a distinct color $\lambda_{i,j}$.

Lemma 7. Given any $g \geq 1$, if Algorithm \mathcal{A} is a ρ -approximation algorithm for instance (G, \mathcal{P}, ∞) , then Algorithm $Combined(\mathcal{A}, (G, \mathcal{P}, g))$ is a $(\rho + 3)$ -approximation algorithm for instance (G, \mathcal{P}, g) .

Proof. In order to prove the correctness of the algorithm, it is sufficient to notice that every \mathcal{N}_i is a no-split instance, thus satisfies the no splitting condition. Therefore any subset $\mathcal{P}_{i,j}$ of it also satisfies the no splitting condition. Moreover, by the correctness of *FirstFit* the output is a valid coloring.

By [Lemma 2](#), for any instance $(SPAN(\mathcal{N}_i), \mathcal{N}_i, g)$ and any color $\lambda_{i,j}$ we have $span(\mathcal{P}_{i,j+1}) \leq \frac{3}{g} len(\mathcal{P}_{i,j})$.

Therefore

$$\begin{aligned} \sum_{i \geq 1, j \geq 2} span(\mathcal{P}_{i,j}) &\leq \frac{3}{g} \sum_{i,j \geq 1} len(\mathcal{P}_{i,j}) \\ &= \frac{3}{g} \sum_{i \geq 1} len(\mathcal{N}_i) = \frac{3}{g} \sum len(\mathcal{P}) \leq 3 \cdot OPT(G, \mathcal{P}, g). \end{aligned}$$

On the other hand

$$\sum_{i \geq 1} span(\mathcal{P}_{i,1}) \leq \sum_{i \geq 1} span(\mathcal{N}_i) \leq \rho \cdot OPT(G, \mathcal{P}, \infty) \leq \rho \cdot OPT(G, \mathcal{P}, g).$$

Combining we get

$$Combined(\mathcal{A}, (G, \mathcal{P}, g)) = \sum_{i,j \geq 1} span(\mathcal{P}_{i,j}) \leq (\rho + 3) \cdot OPT(G, \mathcal{P}, g). \quad \square$$

3.3. The approximation algorithm for tree networks

By combining [Theorem 3](#) with [Lemma 7](#), we get the following theorem.

Theorem 4. Given any $g \geq 1$, Algorithm $Combined(GreedyMatch, (G, \mathcal{P}, g))$ is a 4-approximation algorithm when G is a tree network.

The following lemma and its proof exploit arguments similar to the ones used in [Lemma 3](#).

Lemma 8. For any $\epsilon > 0$, there are infinitely many instances (G, \mathcal{P}, g) having arbitrarily large input sizes, such that $Combined(GreedyMatch, (G, \mathcal{P}, g)) > (3 - \epsilon) \cdot OPT(G, \mathcal{P}, g)$, where G is a tree network.

Proof. Starting from the instance (G, \mathcal{P}, g) depicted in [Fig. 3](#), we consider a slightly different scenario in which the *leftmost* paths start from node 0 and end at node $m + 2$ (instead of $m + 1$); the *central* paths start from node m and end at node $2m + 2$ (instead of $2m + 1$) and finally the *rightmost* paths start from node $2m - 1$ (instead of $2m$) and end at node $3m + 1$. Note that all the paths have the same length. It is easy to see that in this case *GreedyMatchAlgorithm* produces as input one NSI containing all the paths. Following the same arguments of [Lemma 3](#), we argue that the optimal solution uses one regenerator in each one of the nodes $1, \dots, m + 1$, one regenerator in each node $2m, \dots, 3m$, and $g - 1$ regenerators in the nodes $m + 1, \dots, 2m + 1$, for a total cost of $m(g + 1) + g + 1$. A solution of the *FirstFit* Algorithm (if it sorts paths in the order shown in [Fig. 3](#) from top to bottom) will use g regenerators in the nodes $1, \dots, 3m$ for a total cost of $3mg = \frac{g}{(g+1) + (\frac{g+1}{m})} OPT(G, \mathcal{P}, g)$. Choosing g sufficiently large and m much greater than g we get $Combined(GreedyMatch, (G, \mathcal{P}, g)) > (3 - \epsilon) OPT(G, \mathcal{P}, g)$. \square

By combining [Theorem 4](#) with [Lemma 8](#) we get the following theorem.

Theorem 5. The approximation ratio of $Combined(GreedyMatch)$ is between 3 and 4 in tree networks.

4. Beyond tree networks: a matching technique

In this section we present a new technique to approximate (G, \mathcal{P}, ∞) instances in any topology. In particular, we show a general technique able to reduce an instance of the general network to instances of ring and path networks. Using such a technique, and exploiting a reduction of the problem to an instance of the Maximum Weighted Matching on an auxiliary graph, we present an approximation algorithm for the general topology.

4.1. The endpoint intersection graph

In order to describe the matching technique, we need to define the *edge-weighted endpoint intersection graph* $EIG(G, \mathcal{P}) = (V', E')$ of G and \mathcal{P} . V' contains $2|\mathcal{P}|$ nodes $v_{1,1}, v_{1,2}, v_{2,1}, v_{2,2}, \dots, v_{i,1}, v_{i,2}, \dots$, one for each endpoint of a path $P_i \in \mathcal{P}$. There is an edge between two nodes $v_{i,k}, v_{j,k'} (k, k' \in \{1, 2\} \text{ and } i \neq j)$ if $P_i \cup P_j$ is either a path or a ring and $P_i \cap P_j$ contains a path in G with endpoints $v_{i,k}$ and $v_{j,k'}$.

The weight function $f : E' \rightarrow \mathbb{N}$ is defined as follows: $f(\{v_{i,k}, v_{j,k'}\})$ is the length of the path between $v_{i,k}$ and $v_{j,k'}$ in the intersection, minus one. As usual, the weight of a set of edges is defined as the sum of the weights of its edges.

Definition 1. Given a solution w of (G, \mathcal{P}, ∞) , the *endpoint matching* $EM(w)$ of w is defined as a matching of $EIG(G, \mathcal{P})$ that is constructed as follows in polynomial time.

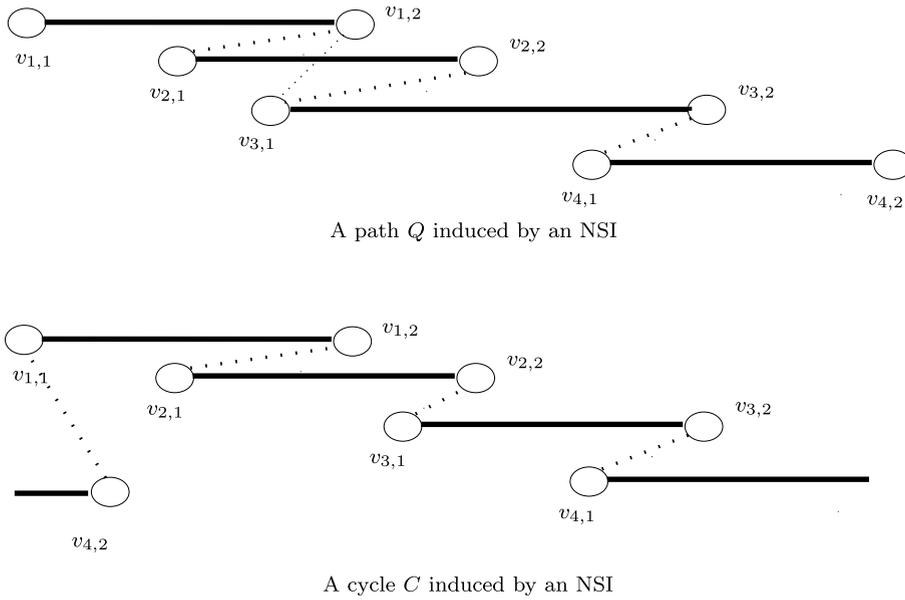


Fig. 5. Correspondence between feasible solutions of (G, \mathcal{P}, ∞) and the matchings of $EIG(G, \mathcal{P})$.

Let w be a solution of (G, \mathcal{P}, ∞) and \mathcal{N} an NSI of w . Assume without loss of generality that \mathcal{N} is connected. By definition, $\cup \mathcal{N}$ is a connected subgraph H of G with maximum degree at most 2. Thus H is either a path or a cycle. Orient the edges of H arbitrarily in some common direction. Let the paths of \mathcal{N} be numbered as P_1, P_2, \dots, P_l according to the order, in the chosen direction, of their starting nodes. Let without loss of generality these nodes be $v_{1,1}, v_{2,1}, \dots$. As the paths are inclusion-free, the order of the ending nodes in this direction is the same, namely $v_{1,2}, v_{2,2}, \dots$.

We distinguish between two cases:

- H is a path (upper part of Fig. 5): For every two consecutive paths P_i, P_{i+1} , their intersection is the segment of the path between $v_{i,2}$ and $v_{i+1,1}$. Therefore $\{v_{i,2}, v_{i+1,1}\}$ is an edge of $EIG(G, \mathcal{P})$, and the edges $\{v_{1,2}, v_{2,1}\}, \{v_{2,2}, v_{3,1}\}, \dots, \{v_{l-1,2}, v_{l,1}\}$ constitute a matching of $EIG(G, \mathcal{P})$ with $l - 1$ edges.
- H is a cycle (lower part of Fig. 5): This case is similar to the case of a path, except that in this case $\{v_{l,2}, v_{1,1}\}$ is also an edge of $EIG(G, \mathcal{P})$, and $\{v_{1,2}, v_{2,1}\}, \{v_{2,2}, v_{3,1}\}, \dots, \{v_{l-1,2}, v_{l,1}\}, \{v_{l,2}, v_{1,1}\}$ constitutes a matching of $EIG(G, \mathcal{P})$ with l edges.

Lemma 9. For every solution w , $REG^w = len(\mathcal{P}) - f(EM(w))$.

Proof. Consider a connected NSI \mathcal{N} of w that induces a path of G . Let without loss of generality $\mathcal{N} = \{P_1, P_2, \dots, P_l\}$ in the chosen direction of the path, and assume that the endpoints of each path are indexed by 1 and 2 in this direction. Letting $END_{\mathcal{N}}$ be the set of all the endpoints of the paths in \mathcal{N} , we obtain

$$\begin{aligned} span(\mathcal{N}) &= int(P_1) + (int(P_2) - f((v_{1,2}, v_{2,1}))) + \dots + (int(P_l) - f((v_{l-1,2}, v_{l,1}))) \\ &= len(\mathcal{N}) - f(\{(u, v) \in EM(w) | u, v \in END_{\mathcal{N}}\}). \end{aligned}$$

The same result holds, similarly, when \mathcal{N} induces a cycle. Summing up over all the NSIs of w , we get

$$\begin{aligned} REG^w &= \sum_{\lambda} span(\mathcal{N}_{\lambda}) = \sum_{\lambda} len(\mathcal{N}_{\lambda}) - \sum_{\lambda} f(M(w) \cap \mathcal{N}_{\lambda}) \\ &= len(\mathcal{P}) - f(EM(w)). \quad \square \end{aligned}$$

Lemma 10. EM is a one-to-one function from the set of solutions of an (G, \mathcal{P}, ∞) instance of TRG to the set of matchings of $EIG(G, \mathcal{P})$. Moreover if G is a ring or tree, then EM is onto. The inverse (partial) function EM^{-1} can be computed in polynomial time.

Proof. Consider a matching M of $EIG(G, \mathcal{P})$. Consider also the auxiliary edges (not belonging to the edge set of $EIG(G, \mathcal{P})$) $M' = \{\{v_{i,1}, v_{i,2}\} | P_i \in \mathcal{P}\}$. Every node has degree at most 1 with respect to the edges in M , and degree exactly 1 with respect to the edges in M' , thus degree at most 2 with respect to the edges in $M \cup M'$. Therefore the connected components of the graph G' induced by the edge set $M \cup M'$ are (alternating) paths and cycles. Note that a path in G' ends with an auxiliary edge in M' (because a node with degree 1 has its only incident edge in M'), thus has odd length; the cycles have even length. For each connected component C of G' we pick a distinct color $\lambda(C)$. First assume that C is a path. Then C is of the form $v_{i_1, k_1} - v_{i_1, k'_1} - v_{i_2, k_2} - v_{i_2, k'_2} - \dots - v_{i_l, k_l}, v_{i_l, k'_l}$, where $k_i \neq k'_i$ for all $i = 1, \dots, l$. This corresponds to the sequence of paths $\mathcal{N}(C) = P_{i_1}, \dots, P_{i_l}$ of \mathcal{P} . We color these paths so that $w(P_{i_1}) = \dots = w(P_{i_l}) = \lambda(C)$. The case of the cycle is treated similarly.

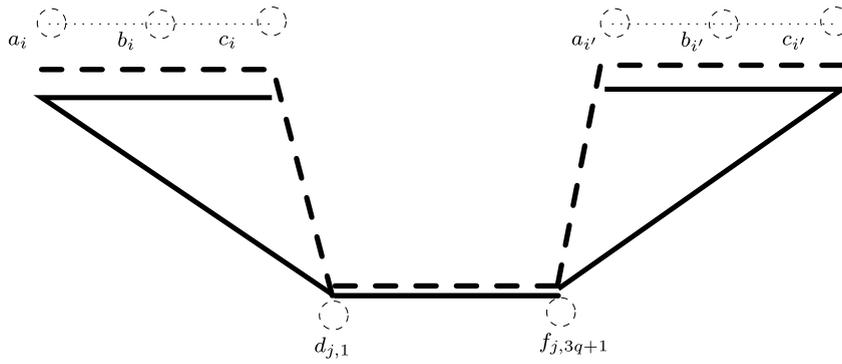


Fig. 6. The top (in dashed line) and the bottom (in solid line) paths corresponding to edge $e^j \in E'_G$, connecting nodes v_G^i and $v_G^{i'}$ ($i < i'$).

The connectedness of a component C implies that $\mathcal{N}(C)$ is connected. If for every connected component C , $\mathcal{N}(C)$ constitutes an NSI then w is a solution of (G, \mathcal{P}, ∞) . Note that $EM(w) = M$. Moreover w is the only solution having this property. Thus EM is one-to-one and $EM^{-1}(M) = w$.

If G is a ring, then every subset of \mathcal{P} and in particular every $\mathcal{N}(C)$ is an NSI, therefore w is a solution of (G, \mathcal{P}, ∞) , thus EM is onto. If G is a tree then for every $\mathcal{N}(C) = P_{i_1}, \dots, P_{i_l}$, using the fact that $\mathcal{N}(C)$ is inclusion free, one can show by induction on l that $\mathcal{N}(C)$ is an NSI. We conclude in the same way, that EM is onto. \square

Since the problem of finding a maximum matching is polynomial time solvable [6], it is worth noticing that, as an immediate consequence of Lemmas 9 and 10, the following lemma provides another polynomial time algorithm finding an optimal solution for instances where G is a tree and $g = \infty$.

Algorithm 4 *MaxMatch*(G, \mathcal{P}, ∞)

- 1: Construct the weighted endpoint intersection graph $EIG(G, \mathcal{P})$ of G and \mathcal{P} with the weight function f .
- 2: Calculate a maximum weighted matching MM of $EIG(G, \mathcal{P})$ with weights f .
- 3: Return $EM^{-1}(MM)$.

Lemma 11. *If G is a tree, then algorithm MaxMatch runs in polynomial time and provides an optimal solution for any instance (G, \mathcal{P}, ∞) .*

4.2. Algorithm for general networks

Unfortunately, as shown in the following theorem, the problem for (G, \mathcal{P}, ∞) instances, with G being a general network, is NP-hard. Therefore, an approximation algorithm for solving it has to be provided.

Theorem 6. *The problem TRG for (G, \mathcal{P}, ∞) instances, with G being a general network, is NP-hard.*

Proof. In order to prove the NP-hardness, we provide a polynomial reduction from the TRIPART problem, known to be NP-complete (see [5]).

An instance of the TRIPART problem is a simple graph $G' = (V'_G, E'_G)$. The question is whether or not there is a partition of E'_G into triangles. Let $V'_G = \{v_G^1, v_G^2, \dots, v_G^{n'}\}$ and $E'_G = \{e_G^1, e_G^2, \dots, e_G^{3q}\}$ (note that if $|E'_G|$ is not a multiple of 3, a partition does not exist and the answer is obviously NO).

From the above instance $G' = (V'_G, E'_G)$ of TRIPART we build the following instance (G, \mathcal{P}, ∞) of the TRG problem. $G = (V_1 \cup V_2, E)$, where $V_1 = \{a_i, b_i, c_i | i = 1, \dots, n'\}$, $V_2 = \{d_{j,k}, e_{j,k}, f_{j,k} | j = 1, \dots, 3q \wedge k = 1, \dots, 3q + 1\}$ and the edge set E is the minimal one containing all the paths of the instance, that are defined as follows.

For each edge $e^j \in E'_G$, connecting nodes v_G^i and $v_G^{i'}$ ($i < i'$), we add the following pair of paths (see Fig. 6): the top path $[a_i, b_i, c_i, d_{j,1}, e_{g(j,1),h(j,1)}, f_{g(j,1),h(j,1)}, d_{j,2}, e_{g(j,2),h(j,2)}, f_{g(j,2),h(j,2)}, \dots, d_{j,3q}, e_{g(j,3q),h(j,3q)}, f_{g(j,3q),h(j,3q)}, d_{j,3q+1}, e_{j,3q+1}, f_{j,3q+1}, a_{i'}, b_{i'}, c_{i'}]$ and the bottom path $[c_i, b_i, a_i, d_{j,1}, e_{g(j,1),h(j,1)}, f_{g(j,1),h(j,1)}, d_{j,2}, e_{g(j,2),h(j,2)}, f_{g(j,2),h(j,2)}, \dots, d_{j,3q}, e_{g(j,3q),h(j,3q)}, f_{g(j,3q),h(j,3q)}, d_{j,3q+1}, e_{j,3q+1}, f_{j,3q+1}, c_{i'}, b_{i'}, a_{i'}]$, where $g(j, k)$ is j if edges e^j and e^k are consecutive in G' , and is the minimum between j and k otherwise, and $h(j, k)$ is k if edges e^j and e^k are consecutive in G' , and is the maximum between j and k otherwise.

The following properties hold:

- Property 1. The top path and the bottom path relative to each edge $e^j \in E'_G$ cannot be put in the same NSI, since otherwise nodes $d_{j,1}$ and $f_{j,3q+1}$ would have degree 3.
- Property 2. Any two (bottom or top) paths relative to non-consecutive edges e^j and e^k ($j < k$) of G' (overlapping on edge $\{e_{j,k}, f_{j,k}\}$) cannot be put in the same NSI, since otherwise nodes $e_{j,k}$ and $f_{j,k}$ would have degree 3.
- Property 3. As it can be easily verified, the only nodes in which it is possible to save regenerators are the b nodes of V_1 .

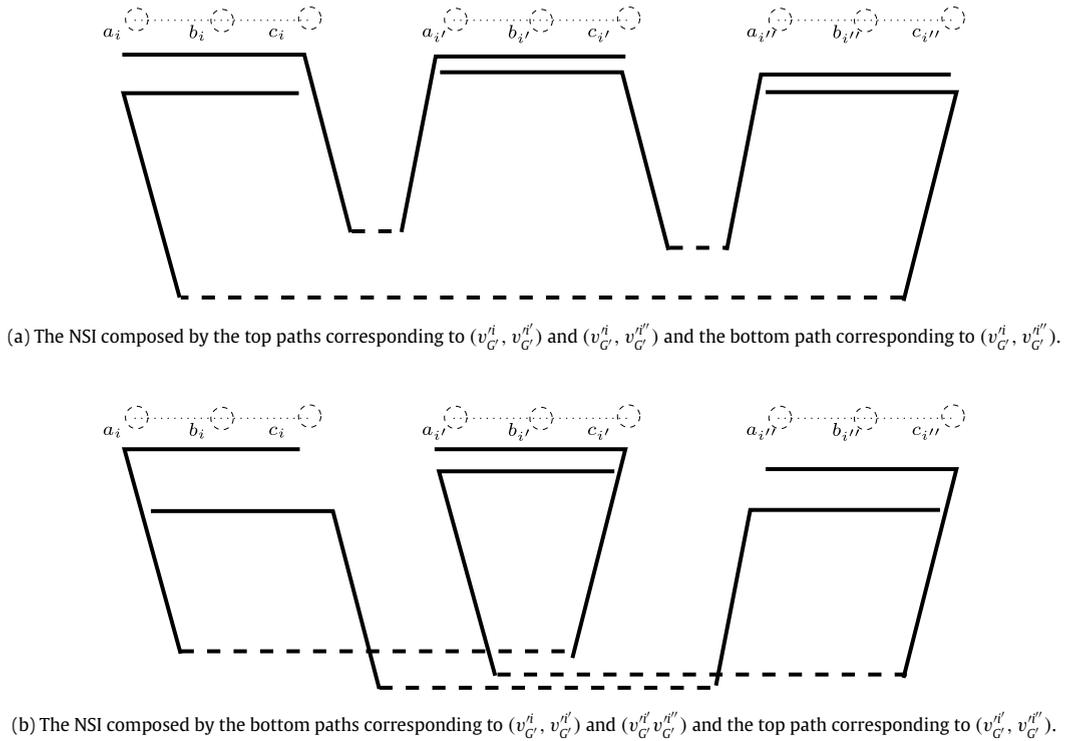


Fig. 7. The two NSIs corresponding to triangle in G' with vertices v_G^i, v_G^i and v_G^i ($i < i' < i''$).

In order to prove the theorem, it is sufficient to prove that (i) if the answer to the *TRIPART* problem is YES, then there exists a solution of the constructed (G, \mathcal{P}, ∞) instance in which it is possible to save $6q$ regenerators and, conversely, (ii) if it is possible to save $6q$ regenerators in the constructed (G, \mathcal{P}, ∞) instance, then the answer to the *TRIPART* problem is YES.

In order to prove (i), it is sufficient to notice that a triangle in G' with vertices v_G^i, v_G^i and v_G^i ($i < i' < i''$) induces 6 paths in \mathcal{P} that can be rearranged in 2 NSIs as follows (see Fig. 7): the top paths corresponding to edges $\{v_G^i, v_G^i\}$ and $\{v_G^i, v_G^i\}$ and the bottom path corresponding to edge $\{v_G^i, v_G^i\}$ belong to an NSI, while the bottom paths corresponding to edges $\{v_G^i, v_G^i\}$ and $\{v_G^i, v_G^i\}$ and the top path corresponding to edge $\{v_G^i, v_G^i\}$ belong to the other NSI. Therefore, in such paths 6 regenerators (one per path, at nodes $b_i, b_{i'}, b_{i''}$) are saved. Since when the *TRIPART* problem is YES E' can be partitioned in q triangles, $6q$ regenerators are saved in total.

It remains to prove (ii). First of all, Property 3 ensures that regenerators can be only saved at b nodes. By Property 2, only paths corresponding to edges of G' sharing a node can be put in the same NSI, and moreover, by Property 1, the two paths corresponding to the same edge cannot be put in the same NSI. Therefore, regenerators can be saved only by putting in the same NSI either x paths corresponding to edges of G' sharing a same node, or the 3 paths corresponding to a triangle in G' . In the first case, $x - 1$ regenerators are saved ($\frac{x-1}{x}$ regenerators per path), whereas in the second case 3 regenerators are saved (1 regenerator per path). Since in \mathcal{P} there are $6q$ paths, if it is possible to save $6q$ regenerators, then all the savings have to be due to $2q$ NSIs each containing 3 paths and in which 1 regenerator per path is saved; therefore, since at most 2 different NSIs correspond to the same triangle of G' , q triangles have to be in G' and the claim follows. \square

The following lemma provides an approximation algorithm for the (G, \mathcal{P}, ∞) problem in general networks.

Lemma 12. For every matching M of $EIG(G, \mathcal{P})$ we can find in polynomial time a matching $\bar{M} \subseteq M$ such that $f(\bar{M}) \geq f(M)/2$ and $EM^{-1}(\bar{M})$ is defined.

Proof. We start as in the proof of Lemma 10. For any path and any cycle of $M \cup M'$, let $\{e_1, e_2, \dots\}$ be the edges of M in the considered path (resp. cycle). We obtain a matching $M_0 \subseteq M$ (resp. $M_E \subseteq M$) by removing the edges with odd (resp. even) indices in $\{e_1, e_2, \dots\}$. This breaks such a path (resp. cycle) into sub-paths of length at most three, in other words into paths containing exactly one edge of M_0 (resp. M_E), which in turn corresponds to a sequence of at most two paths of \mathcal{P} . Assume without loss of generality $f(M_0) \geq f(M_E)$. We claim that $\bar{M} = M_0$ satisfies the claim. Clearly $f(M_0) \geq f(M)/2$. It remains to show that each connected component of $M \cup M_0$ corresponds to an NSI. If such a component consists of one path, than it is an NSI, otherwise it consists of two paths with at least one edge of $EIG(G, \mathcal{P})$ between their endpoints, therefore these two paths are compatible thus constitute an NSI. \square

Algorithm 5 *MatchAndCut*(G, \mathcal{P}, ∞)

- 1: Construct the weighted endpoint intersection graph $EIG(G, \mathcal{P})$ of G and \mathcal{P} with the weight function f .
- 2: Calculate a maximum weighted matching M^* of $EIG(G, \mathcal{P})$ with weights f .
- 3: Calculate the matching \bar{M} of $EIG(G, \mathcal{P})$ as described in proof of Lemma 12.
- 4: Return $EM^{-1}(\bar{M})$.

The following lemma relates the approximation ratio of a solution w for a (G, \mathcal{P}, ∞) instance of TRG to the approximation ratio of $EM(w)$ with respect to the maximum weighted matching problem.

Lemma 13. *Let M be a ρ -approximation to the maximum weighted matching of $EIG(G, \mathcal{P})$ for some $\rho \geq 1$. If $EM^{-1}(M)$ is defined, then $w = EM^{-1}(M)$ is a $(1/\rho + (1 - 1/\rho) \text{load}(\mathcal{P}))$ -approximation for the (G, \mathcal{P}, ∞) instance.*

Proof. Let M^* be a maximum weighted matching of $EIG(G, \mathcal{P})$, and let w^* be an optimal solution of (G, \mathcal{P}, ∞) . Let $\rho' = 1/\rho$, and let M be a matching satisfying our assumption, i.e., $M = EM(w)$ and $f(M) \geq \rho' \cdot f(M^*)$. We have

$$\begin{aligned} REG^w - REG^* &= f(EM(w^*)) - f(EM(w)) = f(EM(w^*)) - f(M) \\ &\leq f(EM(w^*)) - \rho' \cdot f(M^*) \leq (1 - \rho')f(EM(w^*)) \\ &= (1 - \rho')(\text{len}(\mathcal{P}) - REG^*) \\ &= (1 - \rho')\text{len}(\mathcal{P}) + (\rho' - 1)REG^*. \end{aligned}$$

Therefore,

$$\begin{aligned} REG^w &\leq (1 - \rho')\text{len}(\mathcal{P}) + \rho' REG^* \\ &\leq (1 - \rho')\text{load}(\mathcal{P}) \cdot REG^* + \rho' \cdot REG^* \\ &= (\rho' + (1 - \rho')\text{load}(\mathcal{P})) REG^* \end{aligned}$$

where the second inequality follows because a regenerator can be exploited by at most $\text{load}(\mathcal{P})$ paths and therefore $REG^* \geq \frac{\text{len}(\mathcal{P})}{\text{load}(\mathcal{P})}$. \square

Theorem 7. *Combined(MatchAndCut, (G, \mathcal{P}, g)) is a $\left(\frac{7+\text{load}(\mathcal{P})}{2}\right)$ -approximation algorithm for (G, \mathcal{P}, g) instance of TRG.*

Proof. By Lemma 12, \bar{M} is a 2-approximation to the maximum matching¹ of $EIG(G, \mathcal{P})$. Substituting $\rho = 2$ in Lemma 13, we get $\rho'' = (1 + \text{load}(\mathcal{P}))/2$ as the approximation ratio of Algorithm *MatchAndCut*. Finally, by Lemma 7, we obtain the claim. \square

5. Extensions and conclusion

In this paper we have studied an optimization problem in optical networks, that minimizes the use of regenerators when traffic grooming is exploited. Up to here, we have considered the case in which a regenerator has to be placed at every internal node of every lightpath ($d = 1$). Now we focus on the more general problem (G, \mathcal{P}, g, d) that is defined as the problem (G, \mathcal{P}, g) with the difference that $d > 1$ is the maximum number of hops a path can make without meeting a regenerator.

The following theorem allows to extend the results of the previous sections to the general case of any $d > 1$.

Theorem 8. *Given a polynomial time ρ -approximation algorithm \mathcal{A} for (G, \mathcal{P}, g) , for any $d > 1$, it is possible to obtain a polynomial time algorithm \mathcal{A}' guaranteeing a $4 \cdot \rho$ -approximation for (G, \mathcal{P}, g, d) .*

Proof. Given a coloring w , we denote by $\mathcal{N}_1^w, \mathcal{N}_2^w, \dots$ the NSIs induced by w , and by $\mathcal{N}_1^*, \mathcal{N}_2^*, \dots$ the NSIs induced by an optimal coloring w^* of (G, \mathcal{P}, g) .

Recall that for the case $d = 1$, the number of regenerators associated with a coloring w is $REG^w = \sum_i \text{span}(\mathcal{N}_i)$. As w^* is optimal, then $\sum_i \text{span}(\mathcal{N}_i^*) \leq \sum_i \text{span}(\mathcal{N}_i^w)$ for any feasible coloring w .

Let \tilde{w} the coloring returned by algorithm \mathcal{A} ; then $\sum_i \text{span}(\mathcal{N}_i^{\tilde{w}}) \leq \rho \sum_i \text{span}(\mathcal{N}_i^*) \leq \rho \sum_i \text{span}(\mathcal{N}_i^w)$ for any feasible coloring w .

We now turn our attention to an instance of the (G, \mathcal{P}, g, d) problem in which we can discard all the paths of length at most d , because they do not need any regenerator. Notice that a solution for (G, \mathcal{P}, g, d) is not completely characterized by a coloring w (as in the $d = 1$ case), because it also has to specify where to put regenerators.

Let $G(\mathcal{N}) = (V(\mathcal{N}), E(\mathcal{N}))$ be the graph corresponding to the NSI \mathcal{N} , i.e., such that $V(\mathcal{N}) = \bigcup_{P \in \mathcal{N}} \bigcup_{\{u, v\} \in P} \{u, v\}$ and $E(\mathcal{N}) = \bigcup_{P \in \mathcal{N}} \bigcup_{\{u, v\} \in P} \{\{u, v\}\}$. Without loss of generality, we consider NSIs \mathcal{N} verifying the following property: the subgraph of $G(\mathcal{N})$ induced by the nodes in $\text{SPAN}(\mathcal{N})$ is connected. Notice that given an NSI \mathcal{N} not satisfying such a

¹ It is worth noticing that we cannot use a maximum matching M^* of $EIG(G, \mathcal{P})$ because it could be the case that $EM^{-1}(M^*)$ does not exist.

property, we can easily split \mathcal{N} into two or more NSIs $\mathcal{N}_1, \mathcal{N}_2, \dots$ satisfying it such that $span(\mathcal{N}) = \sum_i span(\mathcal{N}_i)$. Note that $|E(\mathcal{N})| \geq span(\mathcal{N})$.

In order to guarantee that $d > 1$ is the maximum number of hops a path traverses without meeting a regenerator, it must be that any edge in $E(\mathcal{N})$ is at distance at most d from a regenerator (we consider the edges incident to a node at distance 1 from it). Since a regenerator serves at most $2d$ edges (i.e., at most d edges on each direction) it follows that at least $\left\lceil \frac{|E(\mathcal{N})|}{2d} \right\rceil$ regenerators are needed for \mathcal{N} . Consider an optimal solution for (G, \mathcal{P}, g, d) associated with a coloring w^{**} with NSIs $\mathcal{N}_1^{**}, \mathcal{N}_2^{**}, \dots$. By the above argument, the number REG^{**} of regenerators needed by w^{**} is $REG^{**} \geq \sum_i \left\lceil \frac{|E(\mathcal{N}_i^{**})|}{2d} \right\rceil \geq \sum_i \left\lceil \frac{span(\mathcal{N}_i^{**})}{2d} \right\rceil \geq \frac{1}{2d} \sum_i span(\mathcal{N}_i^{**})$.

On the other hand, starting from \bar{w} , \mathcal{A}' builds a solution for (G, \mathcal{P}, g, d) in the following way: for any NSI $\mathcal{N}_i^{\bar{w}}$ it puts a regenerator every d nodes in $SPAN(\mathcal{N}_i^{\bar{w}})$, clearly satisfying the regenerator constraint. Since we have discarded all the paths with length at most d , for any NSI $\mathcal{N}_i^{\bar{w}}$ we have that $span(\mathcal{N}_i^{\bar{w}}) \geq d$. Therefore, the number $REG^{\bar{w}}$ used by \mathcal{A}' is $REG^{\bar{w}} = \sum_i \left\lceil \frac{span(\mathcal{N}_i^{\bar{w}})}{d} \right\rceil \leq \sum_i \left(\frac{span(\mathcal{N}_i^{\bar{w}})}{d} + 1 \right) \leq \frac{2}{d} \sum_i span(\mathcal{N}_i^{\bar{w}}) \leq \frac{2\rho}{d} \sum_i span(\mathcal{N}_i^{**}) \leq 4\rho \cdot REG^{**}$. \square

By exploiting Theorem 8 we can extend the results of Algorithms 1 and 3 to the case of $d > 1$ and thus obtain, for problem (G, \mathcal{P}, g, d) , 16-approximation algorithms for ring and tree topologies.

The main open problem is that of finding a better approximation algorithm for general topologies.

Another natural open problem is to discover the exact approximability of the problem. The problem is NP-complete already for $g = 2$ and networks with path topology. In this paper we have shown that the problem is in APX in tree networks. Determining whether the problem is in PTAS for these topologies and for particular cases is an open problem.

It would be also interesting to extend our result by considering more involved cost functions taking into account other switching parameters (e.g., the ADMs – Add-Drop-Multiplexers – used at the endpoints of the lightpath) or the possibility of splitting paths. Finally, studying the on-line version of the problem is an intriguing future research direction.

References

- [1] G. Călinescu, O. Frieder, P.-J. Wan, Minimizing electronic line terminals for automatic ring protection in general WDM optical networks, *IEEE Journal of Selected Area on Communications* 20 (1) (2002) 183–189.
- [2] G. Călinescu, P.-J. Wan, Splittable traffic partition in WDM/sonet rings to minimize sonet adms, *Theoretical Computer Science* 276 (1-2) (2002) 33–50.
- [3] G. Călinescu, P.-J. Wan, Traffic partition in WDM/sonet rings to minimize sonet adms, *Journal of Combinatorial Optimization* 6 (4) (2002) 425–453.
- [4] S. Chen, S. Raghavan, The Regenerator Location Problem, *Networks* 55 (3) (2010) 205–220.
- [5] D. Dor, M. Tarsi, Graph decomposition is Np-complete: a complete proof of Holyer's conjecture, *SIAM Journal on Computing* 26 (4) (1997) 166–1187.
- [6] J. Edmonds, Paths, trees, and flowers, *Canadian Journal of Mathematics* 17 (1965) 449–467.
- [7] R. Fedrizzi, G.M. Galimberti, O. Gerstel, G. Martinelli, E. Salvadori, C.V. Saradhi, A. Tanzi, A. Zanardi, A framework for regenerator site selection based on multiple paths, in: *Proceedings of IEEE/OSA Conference on Optical Fiber Communications, OFC, 2010*.
- [8] R. Fedrizzi, G.M. Galimberti, O. Gerstel, G. Martinelli, E. Salvadori, C.V. Saradhi, A. Tanzi, 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*.
- [9] M. Flammini, A. Marchetti-Spaccamela, G. Monaco, L. Moscardelli, S. Zaks, On the complexity of the regenerator placement problem in optical networks, *IEEE/ACM Transactions on Networking* 19 (2) (2011) 498–511.
- [10] M. Flammini, G. Monaco, L. Moscardelli, H. Shachnai, M. Shalom, T. Tamir, S. Zaks, Minimizing total busy time in parallel scheduling with application to optical networks, *Theoretical Computer Science* 411 (40–42) (2010) 3553–3562.
- [11] M. Flammini, G. Monaco, L. Moscardelli, M. Shalom, S. Zaks, Approximating the traffic grooming problem with respect to adms and oadms, in: *Proceedings of the 14th International European Conference on Parallel and Distributed Computing, EURO-PAR, 2008*.
- [12] O. Gerstel, P. Lin, G. Sasaki, Wavelength assignment in a WDM ring to minimize cost of embedded sonet rings, in: *Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, 1998*.
- [13] S.W. Kim, S.W. Seo, Regenerator placement algorithms for connection establishment in all-optical networks, *Communications, IEE Proceedings* 148 (1) (2001) 25–30.
- [14] G.B. Mertzios, I. Sau, M. Shalom, S. Zaks, Placing regenerators in optical networks to satisfy multiple sets of requests, in: *Proceedings of the 37th International Colloquium on Automata, Languages and Programming, ICALP, 2010, (Best Paper Award)*.
- [15] S. Pachnicke, T. Paschenda, P.M. Krummrich, Physical impairment based regenerator placement and routing in translucent optical networks, in: *Proceedings of Optical Fiber Communication Conference and Exposition and The National Fiber Optic Engineers Conference (Optical Society of America (OWA2)), 2008*.
- [16] K. Sriram, D. Griffith, R. Su, N. Golmie, Static vs. dynamic regenerator assignment in optical switches: models and cost trade-offs, in: *Proceedings of Workshop on High Performance Switching and Routing, HPSR, 2004*.
- [17] P. Winkler, L. Zhang, Wavelength assignment and generalized interval graph coloring, in: *Proceedings of ACM-SIAM Symposium on Discrete Algorithms, SODA, 2003*.
- [18] X. Yang, B. Ramamurthy, Dynamic routing in translucent WDM optical networks, in: *Proceedings of the IEEE International Conference on Communications, ICC, 2002*.
- [19] X. Yang, B. Ramamurthy, Sparse regeneration in translucent wavelength-routed optical networks: architecture, network design and wavelength routing, *Photonic Network Communications* 10 (1) (2005) 39–53.