Optimizing Regenerator Cost in Traffic Grooming[⋆] (Extended Abstract)

Michele Flammini[1], Gianpiero Monaco[2], Luca Moscardelli[3], Mordechai Shalom[4], and Shmuel Zaks[5]

[1] Dipartmento di Informatica, Università degli Studi dell'Aquila, L'Aquila, Italy
flammini@di.univaq.it
[2] Mascotte Project, INRIAI3S(CNRS/UNSA), Sophia Antipolis, France
gianpiero.monaco@sophia.inria.fr
[3] Dipartmento di Scienze, Università degli Studi di Chieti-Pescara, Pescara, Italy
moscardelli@sci.unich.it
[4] Tel Hai Academic College, Upper Galilee, 12210, Israel
cmshalom@telhai.ac.il
[5] Department of Computer Science, Technion, Haifa, Israel
zaks@cs.technion.ac.il

Abstract. In optical networks regenerators have to be placed on lightpaths in order to regenerate the signal. In addition, grooming enables the use of the same regenerator by several lightpaths. In this work we consider the problem of minimizing the number of regenerators used in traffic grooming in optical networks. We deal with the case in which a regenerator has to be placed at every internal node of each lightpath. Up to g (the grooming factor) lightpaths can use the same regenerator. Starting from the 4-approximation algorithm of [7] that solves this problem for a path topology, we provide an approximation algorithm with the same approximation ratio for 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.

Keywords: Optical Networks, Wavelength Division Multiplexing(WDM), Regenerators, Traffic Grooming, Tree Networks.

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

[⋆] 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.

hundred 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 *total* number of regenerators is an important cost parameter to be minimized. 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 [6], 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 travelling 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 viewed as assigning wavelengths to these paths so that at most g of them using the same wavelength (g being the *grooming factor*) can share one edge. Thus, all paths (i.e. connections) that get the same color and form a connected subgraph correspond to grooming of these connections into one lightpath.

In this work we concentrate on the special case $d = 1$ and general g. It is expected that the techniques presented in our work will be carried out to similar studies for higher values of d.

1.1 Related Work

Various variants of regenerator placement problems were studied in [1,4,5,10,12,13,15,16]. Most of these results concentrate in heuristics and simulations and do not consider traffic grooming.

In [6] 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 [11] uses the same cost measure but still does not consider traffic grooming.

The problem is shown to be NP-hard in other contexts such as fiber minimization in [14] and is also implied by the proof of a similar result in [8].

When the underlying graph is a path the problem is equivalent to a machine scheduling problem studied in [7]. Several approximation algorithms are presented in this work for this scheduling problem and its special cases.

1.2 Our Contribution

In this work we consider the traffic grooming problem to minimize the number of regenerators used. We consider only the case $d = 1$, i.e. the case that a regenerator has to be placed at every internal node of every lightpath. Our starting point is a 4-approximation algorithm of [7] that solves a closely related problem for a path topology. We prove that the same algorithm can be used for our problem and show that 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, where L is the maximum load (i.e. number of paths that share a common edge) in the input.

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 toplogies. We summarize the results and suggest open research directions in Section 5. Due to the lack of space, some proofs and figures have been removed. For a full version of the paper see [9].

2 Preliminaries

2.1 Definitions and Problem Statement

An instance of the *Regenerators Grooming Problem* is a triple $(G, \mathcal{P}, g)$ where $G = (V, E)$ is a graph modeling the optical network, $\mathcal{P}$ is a set of simple paths in G and g is a positive integer, namely 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, $\mathcal{P}^w_\lambda$ is the subset of paths from $\mathcal{P}$ colored λ by w, i.e. $\mathcal{P}^w_\lambda \overset{def}{=} \{P \in \mathcal{P} | w(P) = \lambda\}$. When there is no ambiguity on the coloring w under consideration, we omit the superscript w and use $\mathcal{P}_\lambda$.

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 $load(\mathcal{P}, e) \overset{def}{=} |\mathcal{P}_e|$ and $load(\mathcal{P}) \overset{def}{=} \max_{e \in E} load(\mathcal{P}, e)$. A *valid* coloring (or wavelength assignment) w of $(G, \mathcal{P}, g)$ 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 $load(\mathcal{P}^w_\lambda) \leq g$.

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 $int(P) \overset{def}{=} |INT(P)|$. For a set $\mathcal{P}$ of paths we define

$$SPAN(\mathcal{P}) \overset{def}{=} \bigcup_{P \in \mathcal{P}} INT(P),$$

$$span(\mathcal{P}) \overset{def}{=} |SPAN(\mathcal{P})|,$$

$$len(\mathcal{P}) \overset{def}{=} \sum_{P \in \mathcal{P}} int(P).$$

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 2. Due to technological constraints, paths using a same wavelength and going through a 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 $span(\mathcal{P}^w_\lambda)$; in fact, at each node being an intermediate node of some path in $\mathcal{P}^w_\lambda$ a regenerator operating at this wavelength is needed.

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

Input: An instance $(G, \mathcal{P}, g)$, where $G = (V, E)$ is a network, $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ is a set of simple paths in G, and g is the grooming factor.

Output: A valid coloring $w : \mathcal{P} \mapsto \mathbb{N}$ of the paths such that, for every λ, $\mathcal{P}_\lambda$ is an NSI (*no splitting condition*).

Measure: The cost of a solution is given by the total number of regenerators $REG^w \overset{def}{=} \sum_\lambda span(\mathcal{P}^w_\lambda)$.

Objective: 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.

2.3 Path and Ring Networks

Now we focus on ring and path networks. We adapt Theorem 2.1 in [7] to our problem and generalize it to the case of ring networks. Specifically, we show that the *FirstFit* algorithm presented in [7] 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 [7] 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., $int(P_1) \geq int(P_2) \geq \ldots \geq$
   $int(P_n)$.
2: Consider the paths by the above order: assign to the next path, $P_j$, the first possible
   color $\lambda$ 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 $w(P_j) \leftarrow \lambda$.
```

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

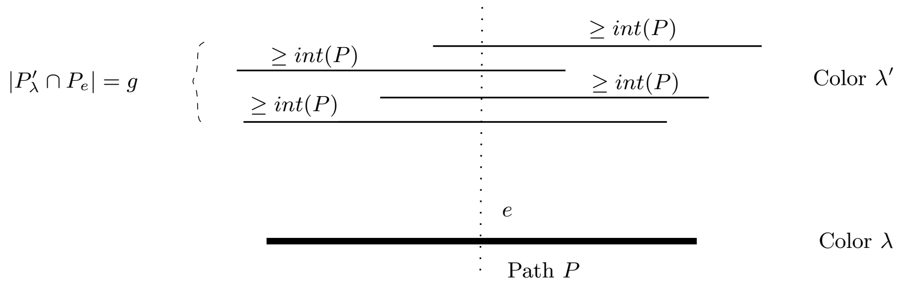


Fig. 1. Basic observation

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

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

Lemma 2. *For any $\lambda > 1$, $len(\mathcal{P}_{\lambda-1}) \geq \frac{g}{3} span(\mathcal{P}_\lambda)$.*

Proof. For every path $P \in \mathcal{P}_\lambda$, we choose arbitrarily an edge e of P among the edges whose existence is guaranteed by Lemma 1. Let $b(P) \overset{def}{=} \mathcal{P}_{\lambda-1} \cap \mathcal{P}_e$ 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. $\overline{\mathcal{P}} \overset{def}{=} \cup_{P \in \mathcal{P}_\lambda} b(P)$. Clearly $\overline{\mathcal{P}} \subseteq \mathcal{P}_{\lambda-1}$.

Now, we consider a blocking path $P' \in \overline{\mathcal{P}}$ (see Figure 3 in [9]). Consider the set of all paths in $\mathcal{P}_\lambda$ *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,

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

We are now ready to prove the following theorem providing an upper bound to the approximation ratio of the *FirstFit* algorithm. The proof exploits arguments similar of the one of the corresponding theorem in [7].

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

The following lemma and its proof follow from a similar claim in [7].

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

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 $G, \mathcal{P}, \infty$ 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 $NSI_1, NSI_2, ...$ such that $\sum_\lambda span(NSI_\lambda)$ is minimized.

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

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 NSI and NSI' are said to be *compatible* if their union is also an NSI. We denote this fact as $NSI \sim NSI'$. Otherwise they are said to be *incompatible* and denoted as $NSI \nsim NSI'$.
- The overlap of two NSIs NSI and NSI' is $OV(NSI, NSI') \overset{def}{=} SPAN(NSI) \cap SPAN(NSI')$ and $ov(NSI, NSI') \overset{def}{=} |OV(NSI, NSI')|$.
- Two NSIs NSI and NSI' are overlapping if $ov(NSI, NSI') > 0$.
- An NSI is said to be *connected* if the union of its paths (as sets of edges) induces a connected graph.
- We say that $NSI \sqsubseteq NSI'$ if $\cup_{P \in NSI} P \subseteq \cup_{P \in NSI'} P$

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 $(G, \mathcal{P}, \infty)$ 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. □

Lemma 5. *At any given point of the execution of Algorithm GreedyMatch the sets NSI_i are connected NSIs.*

Proof. The sets NSI_i are trivially connected at the beginning of the algorithm. Moreover, since a new NSI is constructed by unifying two compatible and overlapping NSIs, they are connected also at any point of the execution of Algorithm $GreedyMatch$. □

Lemma 6. *At any given point of the execution of Algorithm GreedyMatch, after step 1, consider the partition $\{NSI_1, NSI_2, ...\}$. There is an optimal solution $\{NSI_1^*, NSI_2^*, ...\}$ such that every NSI_i is a subset of some NSI_i^*, or in other words the partition given by the algorithm is a refinement of the partition given by some optimal solution.*

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

```
 1: $\forall P_i \in \mathcal{P}, NSI_i \leftarrow \{P_i\}$    ▷ Every path constitutes a connected NSI.
 2: while there exist $NSI_i, NSI_j$ such that $NSI_i \sqsubseteq NSI_j$ do  ▷ Eliminate inclusions
 3:     $NSI_j \leftarrow NSI_j \cup NSI_i$
 4:     $NSI_i \leftarrow \emptyset$
 5: end while
 6: while there exist two compatible $NSI_i, NSI_j$ such that $ov(NSI_i, NSI_j) > 0$ do
 7:     Find two compatible NSIs $NSI_i, NSI_j$ maximizing $ov(NSI_i, NSI_j)$
 8:     $NSI_j \leftarrow NSI_j \cup NSI_i$
 9:     $NSI_i \leftarrow \emptyset$
10:     Eliminate inclusions (as in Steps 3-6)  ▷ We will prove in Lemma 7 that this is unnecessary
11: end while
```

Proof. Without loss of generality we can assume that all the NSIs in an optimal solution of $(G, \mathcal{P}, \infty)$ are connected, because if we have a disconnected NSI NSI we can replace NSI with a connected NSI for each connected component of NSI. The claim is obviously true immediately after step 1 of the algorithm. Assume by contradiction that the claim is false and consider the first time during the execution of the algorithm that it becomes false. This can happen only after execution of step 8. NSI_i and NSI_j are overlapping and compatible, because they are chosen by the algorithm in step 7. They are also connected by Lemma 5. As the condition was true prior to the execution of step 8, there is some optimal solution $S^* = \{NSI_1^*, NSI_2^*, \ldots\}$ such that $NSI_i \subseteq NSI_i^*$ and $NSI_j \subseteq NSI_j^*$. Therefore $NSI_i^* \supset NSI_i$ and $NSI_j^* \supset NSI_j$ are overlapping. Also, by Lemma 4, $NSI_i^* \nsim NSI_j^*$.

As there are no inclusions, the $OV(NSI_i, NSI_j)$ is a proper subset of both $SPAN(NSI_i)$ and $SPAN(NSI_j)$ (see Figure 4 in [9]). Let $a, b, c, d \in V$ be four distinct nodes of the tree such that $SPAN(NSI_i)$ (resp. $SPAN(NSI_j)$) is the path between b and d (resp. a and c). Then $OV(NSI_i, NSI_j)$ is the path between b and c. The partition $\{NSI_1, \ldots\}$ is a refinement of the partition $\{NSI_1^*, \ldots\}$. Let $NSI_i^* = NSI_i \uplus NSI_{i_1} \uplus NSI_{i_2} \uplus \ldots$. We observe that for none of these sets $SPAN(NSI_{i_k})$ can intersect with both $a-b$ and $c-d$, because this would imply that $OV(NSI_i, NSI_j) \subsetneq OV(NSI_i, NSI_{i_k})$, a contradiction to the way NSI_i and NSI_j are chosen by the algorithm. Given this observation we partition the set NSI_i^* into three sets NSI_i, NSI_{ii} and NSI_{ij} such that the sets NSI_{i_k} spanning at least one edge of $c-d$ (resp. $a-b$) are in NSI_{ii} (resp. NSI_{ij}), the rest are divided arbitrarily. We do the same for NSI_j^*.

$NSI_i, NSI_{ii}, NSI_{ij}$ are pairwise compatible, because they make part of NSI_i^*, and so are NSI_j, NSI_{jj}, and NSI_{ji}. Moreover $NSI_{ij} \sim NSI_{ji}$ and $NSI_{ii} \sim NSI_{jj}$, because the underlying graph is a tree and thus they can overlap only in the path $b-c$ in which there can not 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 NSI_i and NSI_j are contained in the same set of S'^*, a contradiction to the assumption that the condition became false.

Assume $NSI_{ji} \sim NSI_i$:

- $NSI_{ji} \sim NSI_{ii}$: In this case we can move NSI_{ii} and NSI_i into NSI_j^* without increasing the cost of the solution.
- $NSI_{ji} \nsim NSI_{ii}$: In this case the node with induced degree more than 2 is necessarily beyond (in Figure 4 in [9], at the right of) the node d, proving that $SPAN(NSI_{ji})$ contains the path $c - d$. Therefore we can move NSI_i into NSI_j^* without increasing the cost of the solution.

After handling the case $NSI_{ij} \sim NSI_j$ symmetrically, it remains to solve the case $NSI_{ji} \nsim NSI_i$ and $NSI_{ij} \nsim NSI_j$. If NSI_{ji} and NSI_{ij} are overlapping than we can repartition these six sets into two sets $NSI_{ij} \cup NSI_{ji}$ and $NSI_i \cup NSI_j \cup NSI_{ii} \cup NSI_{jj}$ without increasing the cost. Otherwise we build three sets NSI_{ij}, NSI_{ji} and $NSI_i \cup NSI_j \cup NSI_{ii} \cup NSI_{jj}$ without increasing the cost. □

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

Theorem 3. *When Algorithm GreedyMatch ends, the solution* $\{NSI_1, NSI_2, ...\}$ *is optimal.*

The following lemma shows that step 10 is redundant, and therefore can be removed from algorithm *GreedyMatch*.

Lemma 7. *When the Algorithm GreedyMatch reaches step 10, there are no inclusions.*

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 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 $NSI_1, NSI_2, ...$ using algorithm $\mathcal{A}$ computed on the corresponding $(G, \mathcal{P}, \infty)$ instance.
2: For each $i$, let $G(NSI_i)$ be the graph induced by the paths of $NSI_i$. Split $NSI_i$ into sets $\mathcal{P}_{i,1}, \mathcal{P}_{i,2}, ...$ by solving the instance FirstFit$(G(NSI_i), NSI_i, g)$.
3: Assign each one of the sets $\mathcal{P}_{i,j}$ a distinct color $\lambda_{i,j}$.
```

Lemma 8. *Given any* $g \geq 1$*, if Algorithm* $\mathcal{A}$ *is a* ρ*-approximation algorithm for instance* $(G, \mathcal{P}, \infty)$*, 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 NSI_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 $(SPAN(NSI_i), NSI_i, g)$ and any color $\lambda_{i,j}$ we have $span(\mathcal{P}_{i,j+1}) \leq \frac{3}{g} len(\mathcal{P}_{i,j})$.

Therefore

$$\sum_{i\geq 1, j\geq 2} span(\mathcal{P}_{i,j}) \leq \frac{3}{g} \sum_{i,j\geq 1} len(\mathcal{P}_{i,j})$$

$$= \frac{3}{g} \sum_{i\geq 1} len(NSI_i) = \frac{3}{g} \sum len(\mathcal{P}) \leq 3 \cdot OPT(G, \mathcal{P}, g).$$

On the other hand

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

Combining we get

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

□

3.3 The Approximation Algorithm for Tree Networks

By combining Theorem 3 with Lemma 8, we get the following theorem.

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

The following lemma and its proof exploit arguments similar to the ones used in lemma 3.

Lemma 9. *For any $\epsilon > 0$, there are infinitely many instances $(G, \mathcal{P}, g)$ having infinitely many input sizes, such that $Combined(GreedyMatch, (G, \mathcal{P}, g)) > (3 - \epsilon) \cdot OPT(G, \mathcal{P}, g)$, where G is a tree network.*

By combining Theorem 4 with Lemma 9 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 $(G, \mathcal{P}, \infty)$ 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 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 $v_{1,1}, v_{1,2}, v_{2,1}, v_{2,2}, ..., v_{i,1}, v_{i,2}, ...$, one for each endpoint of a path $P_i \in \mathcal{P}$. There is an edge between two nodes $v_{i,k}, v_{j,k'} (k, k' \in \{1, 2\})$ 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 $f : E' \to \mathbb{N}$ 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 the edges belonging to it.

Lemma 10. *For every solution w of $(G, \mathcal{P}, \infty)$ in any graph, there is a matching $M(w)$ of $EIG(G, \mathcal{P})$.*

Proof. Consider an NSI of a solution w. By definition, all the nodes of its paths, in particular their endpoints are in some subgraph of G with maximum degree 2. This subgraph is the union of some paths and cycles of G.

Let us first consider a path Q of this graph. We choose some arbitrary direction of Q, and number the paths of the NSI as $P_1, P_2, ..., P_l$ according to the order, in the chosen direction of their starting nodes. Let w.l.o.g. these nodes be $v_{1,1}, v_{2,1}, ...$ As the paths are inclusion-free, the order of the ending nodes in this direction is the same, namely $v_{1,2}, v_{2,2}, ...$ (upper part of Figure 2). 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 $(v_{1,2}, v_{2,1}), (v_{2,2}, v_{3,1}), ..., (v_{l-1,2}, v_{l,1})$ constitute a matching of $EIG(G, \mathcal{P})$ with $l - 1$ edges.

Now we consider a cycle C of this graph. This case is similar to the previous case (consult the lower part of the figure), except that in this case $(v_{l,2}, v_{1,1})$ is also an edge of $EIG(G, \mathcal{P})$, and $(v_{1,2}, v_{2,1}), (v_{2,2}, v_{3,1}), ..., (v_{l-1,2}, v_{l,1}), (v_{l,2}, v_{1,1})$ constitute a matching of $EIG(G, \mathcal{P})$ with l edges. □

Lemma 11. *$REG^w = len(\mathcal{P}) - f(M(w))$.*

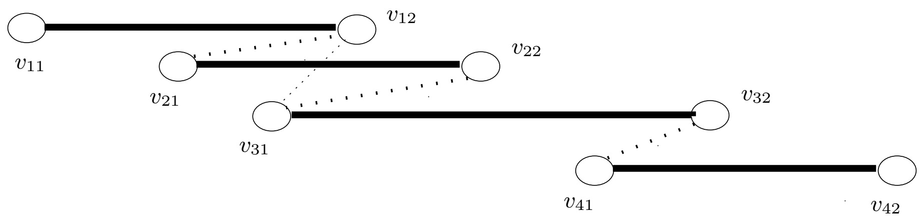
Proof. As in the proof of the previous lemma, consider an NSI that induces a path of G. Let without loss of generality this NSI be $\{P_1, P_2, ..., P_l\}$ in the chosen direction of the path, and assume that the endpoints of each path are indexed 1 and 2 in this direction. Then

$$\begin{aligned} span(NSI) &= int(P_1) + (int(P_2) - f((v_{1,2}, v_{2,1}))) + \dots \\ &\quad + (int(P_l) - f((v_{l-1,2}, v_{l,1}))) = len(NSI) - f(M(w) \cap NSI). \end{aligned}$$

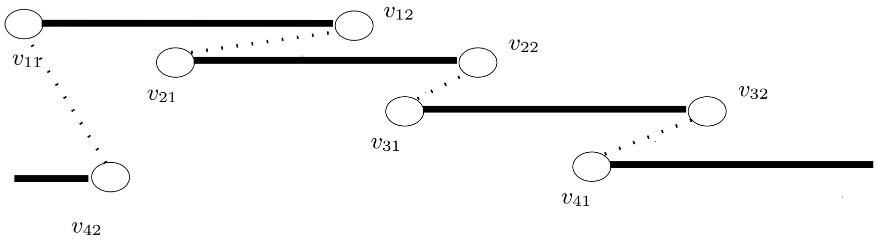
The same result holds, similarly, for an NSI that induces a cycle. Summing up over all the NSIs, we get

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

□



A path Q induced by an NSI



A cycle C induced by an NSI

Fig. 2. Correspondence between feasible solutions of $(G, \mathcal{P}, \infty)$ and the matchings of $EIG(G, \mathcal{P})$

Lemma 12. *If G is a ring or tree, then to every matching M of $EIG(G, \mathcal{P})$ corresponds a solution $w(M)$ of the $(G, \mathcal{P}, \infty)$ instance.*

Proof. Consider a matching M of $EIG(G, \mathcal{P})$. Consider also the *auxiliary* edges (not belonging to the edges of $EIG(G, \mathcal{P})$) $M' = \{(v_{i,1}, v_{i,2})\}$ of $EIG(G, \mathcal{P})$. 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 $M \cup M'$ can be partitioned into (alternating) paths and cycles. Note that a path 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.

We first consider a path of $M \cup M'$. It is of the form $v_{i_1,k_1} - v_{i_1,k'_1} - v_{i_2,k_2} - v_{i_2,k'_2} - ... - v_{i_l,k_l}, v_{i_l,k'_l}$, where $k_i \neq k'_i$ for all $i = 1, \ldots, l$. This corresponds to the sequence of paths $P_{i_1}, ..., P_{i_l}$ of $\mathcal{P}$. Since $\mathcal{P}$ is inclusion-free, these paths constitute an NSI of $\mathcal{P}$. The case of the cycle is similar.

By coloring the paths of each such component with a different color, we get the desired coloring $w(M)$. $\square$

It is worth noticing that as an immediate consequence of lemmata 10, 11 and 12 the following lemma, providing another optimal algorithm for the case in which G is a tree and $g = \infty$, holds.

The following lemma is an immediate consequence of lemmata 10, 11 and 12.

Lemma 13. *If G is a tree, then algorithm $MaxMatch$ runs in polynomial time and provides an optimal solution for any instance $(G, \mathcal{P}, \infty)$.*

Algorithm 4. $MaxMatch(G, \mathcal{P}, \infty)$

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

4.2 Algorithm for General Networks

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

Theorem 6. *The problem for $(G, \mathcal{P}, \infty)$ 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 [2]).

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 $V'_{G'} = \{v'^1_{G'}, v'^2_{G'}, \ldots, v'^{n'}_{G'}\}$ and $E'_{G'} = \{e'^1_{G'}, e'^2_{G'}, \ldots, e'^{3q}_{G'}\}$ (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 $(G, \mathcal{P}, \infty)$ of the Regenerators Grooming Problem. $G = (V_1 \cup V_2, E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5)$, where $V_1 = \{a_i, b_i, c_i | i = 1, \ldots, n'\}$, $V_2 = \{d_{j,k}, e_{j,k}, f_{j,k} | j = 1, \ldots, 3q \wedge k = 1, \ldots, 3q+1\}$ and the edge sets are defined as follows. $E_1 = \{(a_i, b_i), (b_i, c_i) | i = 1, \ldots, n'\}$, $E_2 = \{(d_{j,k}, e_{j,k}), (e_{j,k}, f_{j,k}) | j = 1, \ldots, 3q \wedge k = 1, \ldots, 3q+1\}$ and $E_3 = \{(f_{j,k}, d_{j,k+1}) | j, k = 1, \ldots, 3q\}$; moreover, for each edge $e'^j \in E'_{G'}$, connecting nodes $v'^i_{G'}$ and $v'^{i'}_{G'}$ $(i < i')$ we add to E_4 edges $(a_i, d_{j,1})$, $(c_i, d_{j,1})$, $(a_{i'}, f_{j,3q+1})$ and $(c_{i'}, f_{j,3q+1})$. Finally, $E_5 = \{(d_{j,k}, e_{j',k}), (f_{j',k}, d_{j,k+1}) | j, k = 1, \ldots, 3q \wedge 1 \leq j' < j\}$.

Now we are ready to define the paths of the instance. For each edge $e'^j \in E'_{G'}$, connecting nodes $v'^i_{G'}$ and $v'^{i'}_{G'}$ $(i < i')$, we add the following couple of paths (see Figure 5 in [9]): the *top* path $[a_i, b_i, c_i, d_{j,1}, e_{g(j,1),1}, f_{g(j,1),1}, d_{j,2}, e_{g(j,2),2}, f_{g(j,2),2}, \ldots, d_{j,3q}, e_{g(j,3q),3q}, f_{g(j,3q),3q}, d_{j,3q+1}, e_{j,3q+1}, f_{j,3q+1}, a_{i'}, b_{i'}, c_{i'}]$ and the *bottom* path $[c_i, b_i, a_i, d_{j,1}, e_{g(j,1),1}, f_{g(j,1),1}, d_{j,2}, e_{g(j,2),2}, f_{g(j,2),2}, \ldots, d_{j,3q}, e_{g(j,3q),3q}, f_{g(j,3q),3q}, d_{j,3q+1}, e_{j,3q+1}, f_{j,3q+1}, c_{i'}, b_{i'}, a_{i'}]$, where $g(j,k)$ is j if edges e'^j and e'^k are not consecutive in G', and is the minimum between j and k otherwise.

Notice that the top path and the bottom path relative to each edge $e'^j \in E'_{G'}$ cannot be put in a same NSI since otherwise nodes $d_{j,1}$ and $f_{j,3q+1}$ would have degree 3 (*Property* 1).

Moreover, 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 a same NSI since otherwise nodes $e_{j,k}$ and $f_{j,k}$ would have degree 3 (*Property* 2).

Finally, it can be easily verified that the only nodes in which it is possible to save regenerators are the b nodes of V_1 (*Property* 3).

In order to prove the claim, it is sufficient to prove *(i)* that if the answer to the $TRIPART$ problem is YES, then there exists a solution of the constructed $(G, \mathcal{P}, \infty)$ instance in which it is possible to save $6q$ regenerators and, conversely, *(ii)* that if it is possible to save $6q$ regenerators in the constructed $(G, \mathcal{P}, \infty)$ 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 $v'^{i}_{G'}$, $v'^{i'}_{G'}$ and $v'^{i''}_{G'}$ $(i < i' < i'')$ induces 6 paths in $\mathcal{P}$ that can be rearranged in 2 NSIs as follows (see Figure 6 in [9]): the top paths corresponding to edges $(v'^{i}_{G'}, v'^{i'}_{G'})$ and $(v'^{i'}_{G'}, v'^{i''}_{G'})$ and the bottom path corresponding to edge $(v'^{i}_{G'}, v'^{i''}_{G'})$ belong to an NSI, while the the bottom paths corresponding to edges $(v'^{i}_{G'}, v'^{i'}_{G'})$ and $(v'^{i'}_{G'}, v'^{i''}_{G'})$ and the top path corresponding to edge $(v'^{i}_{G'}, v'^{i''}_{G'})$ 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 a same NSI, and moreover, by Property 1, the two paths corresponding to a same edge cannot be put in a same NSI. Therefore, regenerators can be saved only by putting in a same NSI the 2 paths corresponding to consecutive edges of G' or the 3 paths corresponding to a triangle in G'. In the first case, a regenerator is saved ($\frac{1}{2}$ regenerator 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 a same triangle of G', q triangles have to be in G' and the claim follows. □

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

Lemma 14. *For every matching M of $EIG(G, \mathcal{P})$ we can find in polynomial time a matching $\overline{M} \subseteq M$ such that $f(\overline{M}) \geq f(M)/2$ and there is a solution $w(\overline{M})$ of the $(G, \mathcal{P}, \infty)$ instance.*

Proof. We start as in the proof of Lemma 12. Consider a path or cycle of $M \cup M'$. Let $\{e_1, e_2, ...\}$ be the edges of M in this path. We obtain a matching $M_o \subseteq M$ (resp. M_e) by removing the edges with odd (resp. even) indices in this paths. This breaks the paths into sub-paths of length three, in other words into paths containing exactly one edge of M_o (resp. M_e), which in turn corresponds to a sequence of two paths P_{i_1}, P_{i_2} of $\mathcal{P}$. These paths constitute an NSI of $\mathcal{P}$. Clearly $\max\{f(M_o), f(M_e)\} \geq f(M)/2$, thus either M_o or M_e is the claimed matching $\overline{M}$. □

The following lemma, relates the approximation ratio of a solution with respect to the maximum matching problem to the one of the corresponding solution for our problem.

Algorithm 5. $MatchAndCut(G, \mathcal{P}, \infty)$

```
1: Construct the weighted endpoint intersection graph $EIG(G,\mathcal{P})$ of $G$ and $\mathcal{P}$ with
   the weight function $f$.
2: Calculate the maximum weighted matching $MM$ of $EIG(G,\mathcal{P})$ with weights $f$.
3: Calculate the matching $\overline{MM}$ of $EIG(G,\mathcal{P})$ as described in proof of Lemma 14.
4: Return $w(\overline{MM})$.
```

Lemma 15. *If a matching M is a ρ-approximation to the maximum matching of $EIG(G,\mathcal{P})$ for some $\rho \geq 1$ and $w(M)$ exists, then $w(M)$ is a $(1/\rho + (1-1/\rho)\,g)$-approximation for the $(G,\mathcal{P},g)$ instance.*

Lemma 16. *Algorithm $MatchAndCut$ runs in polynomial time and constitutes a $\left(\frac{1+load(\mathcal{P})}{2}\right)$-approximation for any $(G,\mathcal{P},\infty)$ instance.*

Proof. Clearly $g = \infty$ is equivalent to $g = load(\mathcal{P})$. By Lemma 14 $\overline{MM}$ is a 2-approximation to the maximum matching of $EIG(G,\mathcal{P})$. Substituting $\rho = 2$ and $g = load(\mathcal{P})$ in Lemma 15 we get $\rho' = (1+load(\mathcal{P}))/2$ as the approximation ratio of $MatchAndCut$. □

Combining Lemma 16 with Lemma 8, we finally obtain the following theorem.

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

5 Conclusion and Future Work

In this paper we have studied an optimization problem in Optical Networks, that minimizes the use of regenerators when traffic grooming is exploited. We have considered the case in which a regenerator has to be placed at every internal node of every lightpath, and at most g lightpaths can use the same regenerator. Starting from the 4-approximation algorithm of [7] that solves a closely related problem for a path topology, we have shown that it has the same approximation ratio for the ring topology. We have presented a greedy 4-approximation algorithm for tree networks that uses the mentioned algorithm as a subroutine. We have also introduced a new technique using matchings that can be used to obtain approximation algorithms for other topologies, and have used this technique for general topology to get an $\left(\frac{7+load(\mathcal{P})}{2}\right)$-approximation.

A 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 $d > 1$, i.e. the case that regenerators do not have to be present at every node, or more involved cost functions taking into account other switching parameters (e.g., the ADMs - Add-Drop-Multiplexers - used at the endpoints of the lightpath). Finally, studying the on-line version of the problem is an intriguing future research direction.

References

1. Chen, S., Ljubic, I., Raghavan, S.: The regenerator location problem. Networks 55(3), 205–220 (2010)
2. Dor, D., Tarsi, M.: Graph decomposition is np-complete: A complete proof of holyer's conjecture. SIAM Journal on Computing 26(4), 1166–1187 (1997)
3. Edmonds, J.: Paths, trees, and flowers. Canad. J. Math. 17, 449–467 (1965)
4. Fedrizzi, R., Galimberti, G.M., Gerstel, O., Martinelli, G., Salvadori, E., Saradhi, C.V., Tanzi, A., Zanardi, A.: A Framework for Regenerator Site Selection Based on Multiple Paths. In: Prooceedings of IEEE/OSA Conference on Optical Fiber Communications (OFC) (to appear, 2010)
5. Fedrizzi, R., Galimberti, G.M., Gerstel, O., Martinelli, G., Salvadori, E., Saradhi, C.V., Tanzi, A., Zanardi, A.: 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) (to appear, 2010)
6. Flammini, M., Marchetti-Spaccamela, A., Monaco, G., Moscardelli, L., Zaks, S.: On the complexity of the regenerator placement problem in optical networks. IEEE/ACM Transactions on Networking (to appear, 2010)
7. Flammini, M., Monaco, G., Moscardelli, L., Shachnai, H., Shalom, M., Tamir, T., Zaks, S.: Minimizing total busy time in parallel scheduling with application to optical networks. Theoretical Computer Science 411(40-42), 3553–3562 (2010)
8. Flammini, M., Monaco, G., Moscardelli, L., Shalom, M., Zaks, S.: Approximating the traffic grooming problem with respect to adms and oadms. In: Luque, E., Margalef, T., Benítez, D. (eds.) Euro-Par 2008. LNCS, vol. 5168, pp. 920–929. Springer, Heidelberg (2008)
9. Flammini, M., Monaco, G., Moscardelli, L., Shalom, M., Zaks, S.: Optimizing regenerator cost in traffic grooming. Technical report, Faculty of Computer Science, Technion (September 2010), `http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2010/CS/CS-2010-16`
10. Kim, S.W., Seo, S.W.: Regenerator placement algorithms for connection establishment in all-optical networks. IEE Proceedings Communications 148(1), 25–30 (2001)
11. Mertzios, G.B., Sau, I., Shalom, M., Zaks, S.: Placing regenerator in optical networks: New model, hardness results and algorithms. In: Gavoille, C. (ed.) ICALP 2010, Part II. LNCS, vol. 6199, pp. 333–344. Springer, Heidelberg (2010)
12. Pachnicke, S., Paschenda, T., Krummrich, P.M.: Physical Impairment Based Regenerator Placement and Routing in Translucent Optical Networks. In: Optical Fiber Communication Conference and Exposition and The National Fiber Optic Engineers Conference (Optical Society of America, paper OWA2) (2008)
13. Sriram, K., Griffith, D., Su, R., Golmie, N.: Static vs. dynamic regenerator assignment in optical switches: models and cost trade-offs. In: Workshop on High Performance Switching and Routing (HPSR), pp. 151–155 (2004)
14. Winkler, P., Zhang, L.: Wavelength assignment and generalized interval graph coloring. In: SODA, pp. 830–831 (2003)
15. Yang, X., Ramamurthy, B.: Dynamic routing in translucent WDM optical networks. In: Proceedings of the IEEE International Conference on Communications (ICC), pp. 955–971 (2002)
16. Yang, X., Ramamurthy, B.: Sparse Regeneration in Translucent Wavelength-Routed Optical Networks: Architecture, Network Design and Wavelength Routing. Photonic Network Communications 10(1), 39–53 (2005)