Theoretical Computer Science 410 (2009) 3751–3760

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Hardness and approximation of traffic grooming[☆,☆☆]

Omid Amini[a], Stéphane Pérennes[b], Ignasi Sau[b,c,*]

[a] Max-Planck-Institut für Informatik, Saarbrücken, Germany

[b] Mascotte joint Project of I3S (CNRS/UNSA) and INRIA - Sophia-Antipolis, France

[c] Graph Theory and Combinatorics Group at Applied Mathematics IV Department of UPC - Barcelona, Spain

ARTICLE INFO

Article history:
Received 28 November 2008
Received in revised form 21 April 2009
Accepted 28 April 2009
Communicated by D. Peleg

Keywords:
Traffic grooming
Optical networks
SONET ADM
Approximation algorithms
APX-hardness
PTAS

ABSTRACT

Traffic grooming is a central problem in optical networks. It refers to packing low rate signals into higher speed streams, in order to improve bandwidth utilization and reduce network cost. In WDM networks, the most accepted criterion is to minimize the number of electronic terminations, namely the number of SONET Add–Drop Multiplexers (ADMs). In this article we focus on ring and path topologies. On the one hand, we provide an inapproximability result for TRAFFIC GROOMING for fixed values of the grooming factor g, answering affirmatively the conjecture of Chow and Lin [T. Chow, P. Lin, The ring grooming problem, Networks 44 (2004), 194–202]. More precisely, we prove that RING TRAFFIC GROOMING for fixed $g \geq 1$ and PATH TRAFFIC GROOMING for fixed $g \geq 2$ are APX-complete. That is, they do not accept a PTAS unless P = NP. Both results rely on the fact that finding the maximum number of edge-disjoint triangles in a tripartite graph (and more generally cycles of length $2g+1$ in a $(2g+1)$-partite graph of girth $2g+1$) is APX-complete.

On the other hand, we provide a polynomial-time approximation algorithm for RING and PATH TRAFFIC GROOMING, based on a greedy cover algorithm, with an approximation ratio independent of g. Namely, the approximation guarantee is $\mathcal{O}(n^{1/3}\log^2 n)$ for any $g \geq 1$, n being the size of the network. This is useful in practical applications, since in backbone networks the grooming factor is usually greater than the network size. Finally, we improve this approximation ratio under some extra assumptions about the request graph.

1. Introduction

Background and Problem Definition

Optical wavelength division multiplexing (WDM) is today the most promising technology to accommodate the explosive growth of Internet and telecommunication traffic in wide-area, metro-area, and local-area networks. Using WDM, the potential bandwidth of 50 THz of a fiber can be divided into multiple non-overlapping wavelength or frequency channels. Since currently the commercially available optical fibers can support over a hundred frequency channels, such a channel has

☆ This work has been partially supported by European project IST FET AEOLUS, Ministerio de Educación y Ciencia of Spain, European Regional Development Fund under project TEC2005-03575, Catalan Research Council under project 2005SGR00256, PACA region of France and COST action 293 GRAAL, and has been done in the context of the CRC CORSO with France Telecom.

☆☆ A preliminary conference version of this work appeared in [O. Amini, S. Pérennes, I. Sau, Hardness and approximation of traffic grooming, in: the 18th International Symposium on Algorithms and Computation, ISAAC, in: LNCS, vol. 4835, 2007, pp. 561–573].

* Corresponding author at: Mascotte joint Project of I3S (CNRS/UNSA) and INRIA - Sophia-Antipolis, France. Tel.: +34 626423588; fax: +33 4 92 38 79 71.
E-mail addresses: amini@mpi-inf.mpg.de (O. Amini), Stephane.Perennes@sophia.inria.fr (S. Pérennes), ignasi@ma4.upc.edu (I. Sau).

0304-3975/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2009.04.028

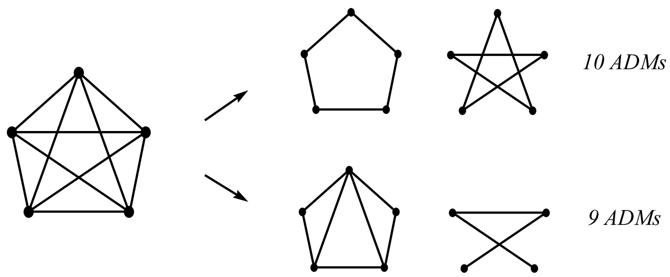

Fig. 1. Two valid partitions of K_5 when $g = 2$, using different number of ADMs.

over one gigabit-per-second transmission speed. However, the network is usually required to support traffic connections at rates that are lower than the full wavelength capacity. In order to save equipment cost and improve network performance, it turns out to be very important to aggregate the multiple low-speed traffic connections, namely *requests*, into higher speed streams. Traffic grooming is the term used to carry out this aggregation, while optimizing the equipment cost. In WDM optical networks the most accepted criterion is to minimize the number of electronic terminations, which is unanimously considered as the dominant cost, rather than the number of wavelengths.

SONET ring is the most widely used optical network infrastructure today. In these networks, a communication between a pair of nodes is done via a *lightpath*, and each lightpath uses an Add–Drop Multiplexer (*ADM*), i.e. an electronic termination, at each of its two endpoints. If each request uses $1/g$ of the capacity of a wavelength, g is said to be the *grooming factor*. The problem is equivalent to assigning a wavelength to each request in such a way that for any wavelength and any link of the network, there can be at most g requests using this link on this wavelength. The aim is to minimize the total number of ADMs. In the graph-theoretical approach that we use, the set of requests is modeled by a graph R, and each vertex in the subgraph of R corresponding to a wavelength represents an ADM. The problem, in the case where the communication network is a ring, can be formally stated as follows:

Ring Traffic Grooming

Input: A cycle C_n on n vertices (network), a graph R (set of requests) on vertices of C_n, and a grooming factor g.

Output: Find for each edge $r = \{x, y\}$ of R, a path $P(r)$ in C_n between x and y, and a partition of the edges of R into subgraphs R_ω, $1 \leq \omega \leq W$, such that for each edge e in $E(C_n)$ and for all ω, the number of paths $P(r)$ using e, r being an edge of R_ω, is at most g.

Objective: Minimize $\sum_{\omega=1}^{W} |V(R_\omega)|$.

The number of paths $P(r)$ using an edge $e \in E(C_n)$ in a given subgraph R_ω is known as the *load* of e in R_ω. That is, the load of the edges in any subgraph of the partition of $E(R)$ can be at most g. The statement of Path Traffic Grooming is analogous, replacing *cycle* C_n with *path* P_n. To fix ideas, consider a ring on five nodes and the complete graph of Fig. 1 as request graph, and let $g = 2$. We exhibit two valid solutions of the problem, both using two subgraphs (i.e. two wavelengths). The lower solution is better because it uses 9 vertices instead of 10.

Previous work and our contribution

The notion of traffic grooming was introduced in [16] for the ring topology. Since then, traffic grooming has been widely studied in the literature (cf. [12,24,28] for some surveys). The problem has been proved to be NP-hard for ring networks and general g [7]. Many heuristics have been done [11], but exact solutions have been found only for certain values of g and for the uniform all-to-all traffic case in unidirectional ring [4], bidirectional ring [5], and path topologies [3]. Recently exact solutions have been also given in the unidirectional ring when the request graph has bounded degree [25]. On the other hand, there was no result on the inapproximability of the problem for fixed $g \geq 1$. In [8] the authors conjecture that Traffic Grooming is Max SNP-hard (or equivalently, Apx-hard, modulo PTAS-reductions) for any fixed value of the grooming factor. We answer affirmatively to this question in Theorem 3.1, providing the first hardness result for the Ring Traffic Grooming problem for fixed values of the grooming factor g.

Considering g as part of the input, in [19] it was proved that Path Traffic Grooming does not accept a constant-factor approximation unless P = NP. For fixed values of g, Path Traffic Grooming was proved to be in P for $g = 1$ [3], but the complexity for fixed $g \geq 2$ has been an open question for a while. Recently, it has been proved in [26] that Path Traffic Grooming for fixed $g > 1$ is NP-hard for *bounded number of wavelengths*. Our method permits us to improve this result in Section 3, by proving the Apx-hardness of Path Traffic Grooming for any fixed $g > 1$ and *unbounded* number of wavelengths. In particular, this extends the NP-hardness result of [26] to the case where the number of wavelengths is not bounded.

The main ingredient of our approach is the proof of the Apx-completeness (given in Section 2) of the problem of finding the maximum number of edge-disjoint triangles in a tripartite graph with bounded degree B: Maximum B-Bounded Edge Covering by Triangles (MECT-B for short). The proof is obtained by L-reduction from Maximum Bounded Covering by 3-Sets, which was proved to be MAX SNP-complete in [21]. A simple modification of this technique permits us to prove the Apx-completeness of finding the maximum number of edge-disjoint odd cycles of given length in a graph. This latter claim is then used to extend our results to arbitrary values of g, see Sections 2 and 3.

The design of approximation algorithms for Traffic Grooming is the topic of the second part of this paper. We present the results for the ring topology, but the same algorithm works also for the path topology. As we show in Section 3, it is trivial to obtain a $\mathcal{O}(\sqrt{g})$-approximation with running time polynomial in g and n. For $g = 1$, the best algorithm in rings achieves an approximation ratio of 10/7 [13]. For general g, the best approximation algorithm [15] achieves an approximation factor of $\mathcal{O}(\log g)$, but the problem is that the running time is exponential in g (that is, n^g). Since in practical applications SONET WDM rings are widely used as backbone optical networks [12,24], the grooming factor is usually greater than the size of the network, i.e. $g \geq n$. For those networks, the running time of this algorithm becomes exponential in n. Thus, it turns out to be important to find good approximation algorithms with running time polynomial in both n and g. In Section 4 we provide such an approximation algorithm, considering g as part of the input. Our algorithm finds a solution of Ring Traffic Grooming that approximates the optimal value within a factor $\mathcal{O}(n^{1/3}\log^2 n)$ for any $g \geq 1$. To the best of our knowledge, this is the first polynomial-time approximation algorithm for the Ring Traffic Grooming problem with an approximation ratio which does not depend on g. Although the performance of this algorithm seems not to be very good at first sight, in fact we conjecture that for the general instance of the problem it is not possible to get rid of a factor n^δ, for some constant $\delta > 0$. Finally, we show that the general scheme of the algorithm yields a $\mathcal{O}(\log^2 n)$-approximation if the request graph excludes a fixed graph as minor, for example if R is planar or of bounded genus. The main theoretical contribution of the second part of this paper is to relate the Traffic Grooming problem to the Dense k-Subgraph problem [14]. We conclude by proposing some further research directions to better understand the complexity of Traffic Grooming.

Notation. We use standard graph theoretical terminology (cf. for instance [10]), and we consider simple undirected graphs without loops or multiple edges. Given a graph $G = (V, E)$, and the edge between the vertices u and v is denoted $\{u, v\}$. A graph on n vertices is called *complete* if it contains an edge between each pair of vertices, and is denoted K_n. The complete graph on three vertices is known as the *triangle*. The *path* on n vertices $v_0, \ldots, v_{n-1}$ with the $n-1$ edges $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{n-2}, v_{n-1}\}$ is denoted P_n. The *cycle* on n vertices obtained from P_n by adding the edge $\{v_{n-1}, v_0\}$ is denoted C_n. A graph G is k-partite if $V(G)$ can be partitioned into k classes $V_0, \ldots, V_{k-1}$ such that there are only edges between classes V_i and V_j with $i \neq j$. The 2-partite (resp. 3-partite) graphs are known as *bipartite* (resp. *tripartite*). The *density* ρ of a graph $G = (V, E)$ is its edges-to-vertices ratio, that is $\rho(G) = \frac{|E(G)|}{|V(G)|}$.

2. Apx-completeness of MECT-*B*

In complexity theory, the class Apx (Approximable) stands for all NP-hard optimization problems that can be approximated within a constant factor. The subclass PTAS (Polynomial-Time Approximation Scheme) contains the problems that can be approximated in polynomial time within a ratio $1 + \varepsilon$ for *all* constants $\varepsilon > 0$. Intuitively, these problems are the easiest ones among all NP-hard problems. For example, the Traveling Salesman Problem in the Euclidean plane accepts a PTAS. Since, assuming P $\neq$ NP, there is a strict inclusion of PTAS into Apx (for instance, Minimum Vertex Cover $\in$ Apx \ PTAS), an Apx-hardness result for a problem implies the non-existence of a PTAS.

The problem of finding the maximum number of node or edge-disjoint cycles in an undirected graph G has several applications, for instance in computational biology [2]. It is often the case that both the maximum degree of G and the length of the cycles to be found are bounded by a constant. In this section we are interested in the following problem:

Maximum B-Bounded Edge Covering by Triangles (MECT-B)

Input: An undirected graph G with maximum degree at most B.

Objective: Find the maximum number of edge-disjoint triangles in G.

MECT-B is long known to be NP-hard [18], and the Apx-hardness when requiring node-disjoint triangles was proved in [21]. Following the ideas of [21], in [6] it was proved that MECT-5 is Apx-hard for general graphs and NP-hard for planar graphs. Finally, in [23] MECT-B was studied from a parameterized view, considering the number of edge-disjoint triangles as the parameter. Namely, is was proved that MECT-B is *Fixed Parameter Tractable* (FPT) by achieving a linear kernel.

In this article we prove that MECT-B remains Apx-hard for tripartite graphs. For convenience, we prove the Max SNP-hardness of MECT-B, which is known to be the same as the Apx-hardness modulo PTAS-reductions. MECT-B is trivially in Apx, since a simple greedy algorithm provides a 3-approximation. The best approximation guarantee for MECT-B is a $(3/2 + \varepsilon)$-approximation algorithm for any $\varepsilon > 0$ [20]. We need to introduce two problems to be used in the proof of Theorem 2.1: Maximum Bounded Covering by 3-Sets (Max 3SC-B for short): Given a collection of 3-subsets of a given set, each element appearing in at most B subsets, find the maximum number of disjoint subsets; and Maximum Bounded Independent Set (Indep. Set-B for short): Given a graph of maximum degree $\leq B$, find a maximum independent set.

Theorem 2.1. MECT-B, $B \geq 10$, *is* Apx-*complete for tripartite graphs.*

Proof. L-reduction from Max 3SC-B and L-reduction to Indep. Set-B.
We define h : MECT-B $\rightarrow$ Indep. Set - $(3/2(B-2))$ as follows: given a graph G as instance I of MECT-B, we define the following instance $h(I)$ of Indep. Set - (3/2(B-2)): the graph $h(G)$ contains a node v_T for every triangle T in G. There is an edge $\{v_{T_0}, v_{T_1}\}$ in $h(G)$ if and only if T_0 and T_1 share an edge in G. Given a solution A of $h(I)$, we define a solution $S_h(A)$ of I by taking the triangles corresponding to nodes in A. It is easily verified that (h, S_h) is an L-reduction.

We define f : Max 3SC-B $\rightarrow$ MECT-(3B+1) in the following way: suppose that we are given as instance I, a collection C of 3-element subsets of a set X such that every element of X belongs to at most B members of C. The problem for I consists in

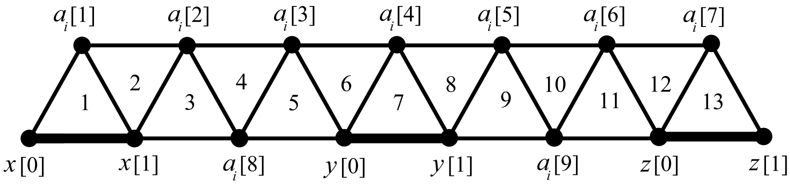

Fig. 2. Gadget G_i used in the reduction of the proof of Theorem 2.1.

finding the maximal number $OPT(I)$ of disjoint subsets in C. We construct an instance $f(I)$ of MECT-(3B + 1), i.e. we construct a graph $G = (V, E)$ in which we ask for the maximum number $OPT(f(I))$ of edge-disjoint triangles. Let $C = \{c_1, \ldots, c_r\}$, with $|c_i| = 3$. The local replacement f substitutes for each element $c_i = \{x, y, z\} \in C$, the graph $G_i = (V_i, E_i)$ depicted in Fig. 2.

To avoid confusion, note by t any element in c_i, i.e. $t \in \{x, y, z\}$. Note that, for each element t, the nodes $t[0]$ and $t[1]$, and the edge $\{t[0], t[1]\}$ (corresponding to the thick edges in Fig. 2) appear only once in G, regardless of the number of occurrences of t. On the other hand, we add 9 new vertices $a_i[j]$, $1 \le j \le 9$ for each subset c_i, $1 \le i \le |C|$. More precisely, $G = (V, E) = \cup_{i=1}^{|C|} G_i$, where $V = \bigcup_{t \in X} \{t[i] : i = 0, 1\} \cup \bigcup_{i=1}^{|C|} \{a_i[j] : 1 \le j \le 9\}$ and $E = \bigcup_{i=1}^{|C|} E_i$.

Given a solution A of $f(I)$ of size s_2, we modify it in polynomial time to another equal or better solution A' in the following way: in each G_i, if the three triangles covering the edges $\{x[0], x[1]\}$, $\{y[0], y[1]\}$, and $\{z[0], z[1]\}$ (numbered 1, 7, 13 in Fig. 2) belong to A, we choose the seven *odd* triangles of G_i to belong to A'. If not, we take the six *even* triangles. Let $s'_2 \ge s_2$ be the size of A'. Then, we define a solution $S_f(A)$ of I by choosing the subset c_i to be in $S_f(A)$ if and only if A' contains exactly 7 triangles in G_i. We claim that the pair (f, S_f) is an L-reduction: in each G_i there are 13 different triangles, numbered from 1 to 13 in Fig. 2. The only way to choose 7 edge-disjoint triangles in G_i is by taking all the *odd* triangles, and thus by covering the three edges $\{x[0], x[1]\}$, $\{y[0], y[1]\}$, and $\{z[0], z[1]\}$. All other choices of triangles yield at most 6 edge-disjoint triangles. The key observation is that we are able to choose 7 triangles exactly $OPT(I)$ times. Indeed, each time we choose 7 triangles we cover the edges corresponding to 3 elements of c_i, and since the number of disjoint c_i's in C is $OPT(I)$, this can be done exactly $OPT(I)$ times. On the other hand, one can easily see that $OPT(I) \ge \frac{|C|}{3B}$. Hence:

$$\begin{aligned} OPT(f(I)) &= 7 \cdot OPT(I) + 6(|C| - OPT(I)) \le OPT(I) + 18B \cdot OPT(I) \\ &= (18B + 1)OPT(I). \end{aligned}$$

To conclude, note that if the solution $S_f(A)$ of I has size s_1, we have $OPT(I) - s_1 \le OPT(f(I)) - s_2$. To see this, we observe that $OPT(f(I)) = 6r + OPT(I)$, and also $s'_2 = 6r + s_1$, and so $OPT(f(I)) - OPT(I) = s_1 - s'_2 \le s_1 - s_2$.

Both (f, S_f) and (h, S_h) are L-reductions and MAX 3SC-B, $B \ge 3$ and Indep. Set-B, $B \ge 5$ are Max SNP-complete [21]. Thus, MECT-B, $B \ge 10$ is Max SNP-complete. Finally, note that the graph $G = (V, E)$ used in the proof is tripartite, where the vertex sets V_0, V_1, V_2 defining the tripartition are:

$$V_0 = \bigcup_{t \in X}^{|X|} t[0] \cup \bigcup_{i=1}^{|C|} \{a_i[2], a_i[5]\}, \qquad V_1 = \bigcup_{i=1}^{|C|} \{a_i[j] : j = 1, 4, 7, 8, 9\},$$

$$V_2 = \bigcup_{i=1}^{|X|} t[1] \cup \bigcup_{t \in X}^{|C|} \{a_i[3], a_i[6]\}. \quad \square$$

The proof of the Apx-hardness of MECT-B of Theorem 2.1 can be extended to obtain the Apx-completeness of the problem of finding the maximum number of edge-disjoint cycles of length $2g + 1$ for any fixed $g \ge 1$, as stated in the following theorem.

Theorem 2.2. *Let $\mathcal{G}$ be the class of* $(2g + 1)$*-partite graphs G of girth* $2g + 1$*, consisting of* $(2g + 1)$ *parts $A_0, \ldots, A_{2g}$ such that the only edges are between A_i and A_{i+1}* $(\mathrm{mod}\ 2g + 1)$*, $i = 0, \ldots, 2g$, and such that all the graphs induced by $V(G) \setminus A_i$ in G, for all $i = 0, \ldots, 2g$, form a forest. Then the problem of finding the maximum number of edge-disjoint C_{2g+1}'s is* Apx*-complete in $\mathcal{G}$.*

Proof. First, note that a greedy algorithm provides a constant-factor approximation with factor $2g + 1$, so the problem is in Apx. Consider the gadget of the proof of Theorem 2.1 (see Fig. 2). We modify this gadget in such a way that the same proof holds for C_{2g+1}'s instead of C_3's (triangles), and such that all the conditions of the theorem are verified. Given $g > 1$, we add a chain of $4g + 1$ triangles between any pair of triangles corresponding to *thick* edges (that is, between the edges corresponding to elements of X). Then we add $g - 1$ inner points to all the edges going from up to down in the triangles. An example if shown in Fig. 3.

It is easily seen that the graph built in this way is $(2g + 1)$-partite. Indeed, it admits a partition into $(2g + 1)$ parts, numbered $0, \ldots, 2g$, which consist of enumerating the vertices cyclically. Let $A_0, \ldots, A_{2g}$ be the different parts. In such a $(2g + 1)$-partition, for any element $t \in X$, the vertex $t[0]$ belongs to A_0, and the vertex $t[1]$ belongs to A_{2g}. We need this property to ensure the consistency of our gadget when an element appears in more that one subset. Note that the graphs induced by $V(G) \setminus A_i$ in G, for all $i = 0, \ldots, 2g$, form a forest. At this point, one can rewrite the proof of Theorem 2.1 to obtain the result, just by changing the multiplicative constants. $\square$

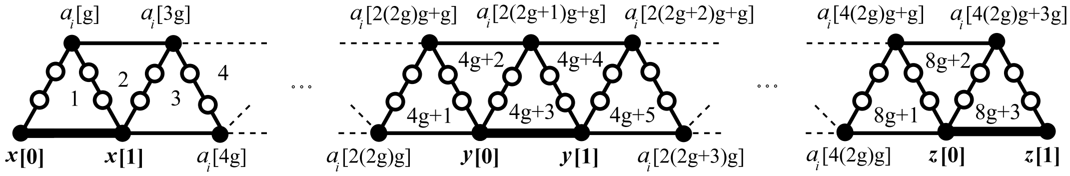

Fig. 3. Adding $g - 1$ inner points (depicted as ○ in the figure) to prove the APX-completeness of finding edge-disjoint C_{2g+1}'s.

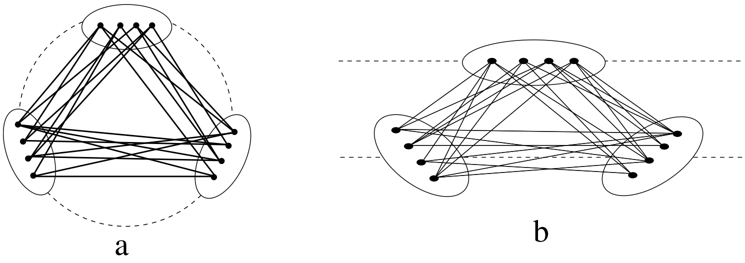

Fig. 4. Tripartite request graphs used in Lemma 3.2: (a) in the ring for $g = 1$; (b) in the path for $g = 2$.

3. APX-completeness of TRAFFIC GROOMING

In this section we prove the hardness results for RING TRAFFIC GROOMING and PATH TRAFFIC GROOMING. First we prove that RING TRAFFIC GROOMING belongs to APX when g is fixed (i.e., not part of the input). The same result holds for PATH TRAFFIC GROOMING.

Lemma 3.1. RING TRAFFIC GROOMING *belongs to* APX *for any fixed* $g \geq 1$.

Proof. To see that RING TRAFFIC GROOMING is in APX for any fixed $g \geq 1$, we have to find a constant-factor approximation algorithm. We use the fact that the best possible density ρ^* of any subgraph involved in the partition of the request graph in the ring is $\mathcal{O}(\sqrt{g})$, given by a complete graph inducing load g in the edges of the ring (it is clear that no graph has greater density than the complete graph). We prove that the cost A of any solution $R_1, \ldots, R_W$ is in the interval $[\frac{|E(R)|}{\rho^*}, 2|E(R)|]$. This clearly implies that any solution has cost at most $2\rho^* = \mathcal{O}(\sqrt{g})$ times the optimal cost. To see this, note that each edge of R contributes at most twice to the cost, so $A \leq 2|E(R)|$. On the other hand, we have

$$A = \sum_{\omega=1}^{W} |V(R_\omega)| = \sum_{\omega=1}^{W} \frac{|E(R_\omega)|}{\rho(R_\omega)} \geq \sum_{\omega=1}^{W} \frac{|E(R_\omega)|}{\rho^*} = \frac{|E(R)|}{\rho^*}.$$

Thus, a $\mathcal{O}(\sqrt{g})$-approximation is obtained just by taking *any* partition of the request graph. □

Since we will deal with tripartite graphs in the proof of Theorem 3.1, we need first a technical lemma concerning the structure of the optimal solutions of RING TRAFFIC GROOMING in tripartite request graphs.

Lemma 3.2. *Let R be a tripartite instance graph of* RING TRAFFIC GROOMING *for $g = 1$ such that the vertices belonging to the same class of the tripartition are placed consecutively in the ring, and let t^* be the maximum number of edge-disjoint triangles in R. If there exists a partition of $E(R)$ into triangles and P_4's which uses exactly t^* triangles, then this partition is optimal. The same property holds for* PATH TRAFFIC GROOMING *and $g = 2$.*

Proof. We focus first on RING TRAFFIC GROOMING. Let t^* the maximum number of edge-disjoint triangles of a partition of $E(R)$. When R is tripartite and $g = 1$, it is clear that the only possible subgraphs that can be involved in a partition of $E(R)$ are K_3, P_2, P_3, and P_4 (see Fig. 4a). Since these three paths have density at most 3/4 (attained by the P_4), the cost A_t of any solution using t triangles satisfies

$$A_t \geq t + 4 \cdot \frac{|E(R)| - 3t}{3} = \frac{4}{3}|E(R)| - 3t \geq \frac{4}{3}|E(R)| - 3t^*. \quad (1)$$

Note that the above bound does not depend on t, and therefore holds for any solution. A partition as stated in the conditions of the lemma attains this lower bound, hence it is optimal. The same argument applies to the path and $g = 2$ (see Fig. 4b), since the same subgraphs are involved in any partition. □

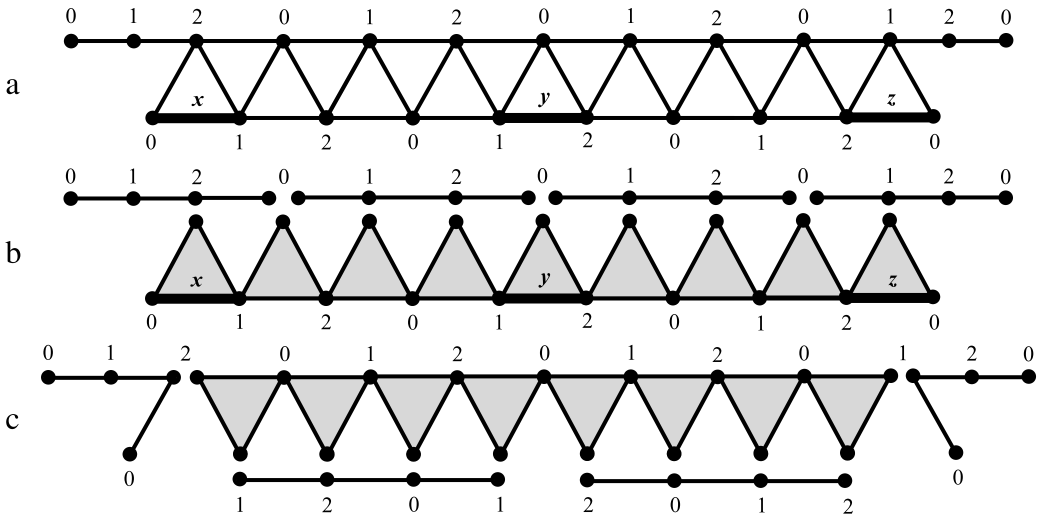

Fig. 5. Request graphs used in the proof of Theorem 3.1: (a) gadget G_i corresponding to the set $c_i = \{x, y, z\}$. The labels of the vertices indicate the tripartition; (b) partition into 9 K_3's and 4 P_4's *with* the edges x, y, z ; (c) partition into 8 K_3's and 4 P_4's *without* the edges x, y, z.

We are ready to state the main result of this section.

Theorem 3.1. Ring Traffic Grooming *is* Apx*-complete for fixed* $g = 1$*, even if the request graph has degree bounded by a constant* $B \geq 10$*. Thus, it does not accept a* PTAS *unless* $\mathrm{P} = \mathrm{NP}$.

Proof. The problem is in Apx by Lemma 3.1. To prove the Apx-hardness, we consider the family of request graphs $\mathcal{R}$ defined as follows.

Mimic the proof of Theorem 2.1 replacing the gadget of Fig. 2 with the gadget of Fig. 5a. With slight abuse of notation, the edge corresponding to an element x is also denoted x. It is easy to check that the same proof carries over with these new gadgets, and therefore the problem of finding the maximum number of edge-disjoint triangles in this class $\mathcal{R}$ of graphs is Apx-hard. Note that all the graphs built in this way are also tripartite, as shown in Fig. 5a.

Håstad proved [17] that Maximum Bounded Covering by 3-Sets is Apx-hard even when restricted to instances for which we know that there exists a collection of mutually disjoint 3-subsets covering *all* the elements in the set. Therefore, we can assume without loss of generality that any optimal solution of MECT-B in a graph $R \in \mathcal{R}$ corresponds to a collection of mutually disjoint 3-subsets covering *all* the elements in the set. Hence, such an optimal solution of MECT-B restricted to each gadget G_i corresponding to the set $c_i = \{x, y, z\}$ satisfies:

(i) either it contains the three edges x, y, z corresponding to the elements in the set $c_i = \{x, y, z\}$; or
(ii) it contains *none* of the edges x, y, z.

Thinking of the graphs $R \in \mathcal{R}$ as instances of Ring Traffic Grooming, the key observation is that:

- in case (i), the gadget G_i can be partitioned into 9 K_3's and 4 P_4's (see Fig. 5b);
- in case (ii), the gadget $G_i - \{x, y, z\}$ can be partitioned into 8 K_3's and 4 P_4's (see Fig. 5c).

It is easy to see that such a partition uses the maximum number of edge-disjoint triangles in the tripartite graph R, and only K_3's and P_4's are involved. By Lemma 3.2, this partition is an optimal solution of Ring Traffic Grooming for $g = 1$ in R. Let OPT be the number of vertices of such an optimal solution in R, and let t^* be the number of triangles in an optimal solution in R. (We simply write OPT and t^* instead of $OPT(R)$ and $t^*(R)$, respectively.) It is clear that

$$|E(R)| \leq OPT \leq 2|E(R)|. \tag{2}$$

We have seen in the proof of Lemma 3.2 that the cost A_t of any solution using t triangles satisfies $A_t \geq \frac{4}{3}|E(R)| - 3t$. We can also write

$$OPT = \frac{4}{3}|E(R)| - 3t^*. \tag{3}$$

Since MECT-B is Apx-hard in $\mathcal{R}$, there exists a constant $\varepsilon_0 > 0$ such that, unless $\mathrm{P} = \mathrm{NP}$, one cannot find in polynomial time more than $(1 - \varepsilon_0)t^*$ triangles in an arbitrary graph $R \in \mathcal{R}$. Therefore, the cost A of any solution of Ring Traffic Grooming that can be found in polynomial time satisfies

$$A \geq \frac{4}{3}|E(R)| - 3(1 - \varepsilon_0)t^* = OPT + 3\varepsilon_0 t^*, \tag{4}$$

where we have used Eq. (3). On the other hand, from Eq. (3) and using Eq. (2) twice we get

$$t^* = \frac{4}{9}|E(R)| - \frac{OPT}{3} \geq \frac{4}{9}|E(R)| - \frac{|E(R)|}{3} = \frac{|E(R)|}{9} \geq \frac{OPT}{18}. \quad (5)$$

Combining Eqs. (4) and (5) yields that the cost A of any solution satisfies

$$A \geq OPT + 3\varepsilon_0 \frac{OPT}{18} = \left(1 + \frac{\varepsilon_0}{6}\right) OPT = (1 + \varepsilon_1)OPT,$$

with $\varepsilon_1 = \varepsilon_0/6 > 0$. Therefore, unless P = NP, Ring Traffic Grooming does not accept a PTAS for fixed $g = 1$. □

As expected, the result can be generalized to any $g \geq 1$.

Theorem 3.2. Ring Traffic Grooming *is* Apx-*complete for any fixed* $g \geq 1$, *even if the request graph has degree bounded by a constant* $B \geq 10$. *Thus, it does not accept a* PTAS *unless* P = NP.

Proof. Sketch. The case $g = 1$ has been proved in Theorem 3.1, so assume henceforth that $g > 1$. The problem is in Apx for any $g \geq 1$ by Lemma 3.1. To prove the Apx-hardness, take a $(2g + 1)$-partite graph as request graph, in such way that each cycle makes at least g tours around the center of the ring. At this point we can reduce the grooming problem to the problem of finding a maximum number of cycles of length $2g + 1$ in this graph (as in the case $g = 1$). This later problem is also Apx-complete, see Theorem 2.2. The details follow.

Let G be a graph satisfying the conditions of Theorem 2.2: G is a $(2g + 1)$-partite graph, consisting of $2g + 1$ parts $A_0, \ldots, A_{2g}$ such that the only edges are between A_i and A_{i+1} (mod $2g + 1$), $i = 0, \ldots, 2g$, and such that the graph induced between two consecutive parts of G forms a forest (or more generally, a graph of girth at least $g + 1$). In order to simplify the presentation, suppose that this graph can be partitioned into C_{2g+1}'s.

Let $c_0, \ldots, c_{2g}$ be a permutation of the vertices of the cycle C_{2g+1}, such that the polygon $(c_0, \ldots, c_{2g})$ makes g tours around the center (for $g = 1$ take the triangle; for g arbitrary, let $c_i = \exp(\frac{2ig\pi}{2g+1})$). Now replace each vertex c_i with an interval consisting of vertices of A_i. In this cyclic representation of the graph G, each cycle makes at least g tours around the origin. To see this, recall that the only possible edges are between A_i and A_{i+1} (mod $2g + 1$), $i = 0, \ldots, 2g$, and also the graph induced between two consecutive parts forms a forest. This implies that every cycle should intersect each A_i at least once, and so this cycle makes at least g tours around the origin, as the original cycle $\{c_0, \ldots, c_{2g}\}$ does so.

Each cycle used in the solution should be of length exactly $2g+1$, there is no cycle of smaller length, and longer cycles use each edge more than g times, as they make more than g tours around the origin. Then the problem is reduced to finding edge-disjoint cycles of length $2g + 1$, which is also Apx-hard by Theorem 2.2. The proof of Theorem 3.1 can now be reproduced to obtain the same result for any g, replacing the factor $\frac{4}{3}$ for $g = 1$ (because the path with greatest density in any solution for $g = 1$ is P_4) with a factor $\frac{2g+2}{2g+1}$ for a general g (because the path with greatest density in any solution for general g is P_{2g+2}). Hence, Ring Traffic Grooming is Apx-complete even for bounded number of requests per node $B \geq 10$. □

These ideas can be naturally extended to prove the Apx-completeness of Path Traffic Grooming for any fixed $g \geq 2$.

Theorem 3.3. Path Traffic Grooming *is* Apx-*complete for any fixed* $g \geq 2$. *Thus, it does not accept a* PTAS *unless* P = NP.

Proof. Again, the result holds even for bounded number B of requests per node, $B \geq 10$. We prove the result for $g = 2$, proceeding for $g > 2$ as in the proof of Theorem 3.2. Consider the family of request graphs $\mathcal{R}$ defined in the proof of Theorem 3.1, and place the three partition classes consecutively on the path one after the other, as shown in Fig. 4b. Since each triangle induces load 2 (that is), minimizing the number of ADMs corresponds to finding the maximum number of edge-disjoint triangles. Therefore, the problem does not accept a PTAS unless P = NP. □

4. Approximating ring traffic grooming

We are now interested in finding good approximation algorithms considering g as part of the input. As we saw in Section 3, obtaining a $\mathcal{O}(\sqrt{g})$-approximation is trivial. Since in practical applications SONET WDM rings are widely used as backbone optical networks [12,24], the grooming factor is usually greater than the size of the network, i.e., $g \geq n$. Thus, it turns out to be important to find approximation algorithms with an approximation ratio not depending on g. A general approximation algorithm with this property is the main result of this section. It provides in the worst case a $\mathcal{O}(n^{1/3} \log^2 n)$-approximation. We describe it for the ring topology, but exactly the same arguments provide an algorithm for the path. The main idea is to greedily find subgraphs with high density using approximation algorithms for the Dense k-Subgraph problem, which is defined as follows: given a graph G and an integer k, find an induced subgraph $H \subseteq G$ on k vertices with the greatest density among all subgraphs on k vertices. In [14] the authors provide a polynomial-time algorithm with approximation ratio $2n^{1/3}$. To simplify the presentation, suppose that $n = 2^t$ for some $t > 0$ (otherwise, introduce dummy vertices on the ring until getting size $n' = 2^t$, with $n' < 2n$):

Algorithm $\mathcal{A}$:

(1) Divide the request set into $\log n$ classes, such that in each class C_i the length of the requests lies in the interval $[2^i, 2^{i+1})$, $i = 0, \ldots, \log n - 1$. For each class C_i, the ring can be divided into intervals of length 2^i such that the only requests are between consecutive intervals. In this way we obtain $\frac{n}{2^i}$ subproblems for each class: each one consists in finding an optimal solution in a bipartite graph of size $2 \cdot 2^i$. More precisely, each subproblem can be formulated as:
Bipartite Traffic Grooming
Input: A bipartite graph R, and a grooming factor g.
Output: Partition of the edges of R into subgraphs R_ω with at most g edges, $1 \leq \omega \leq W$.
Objective: Minimize $\sum_{\omega=1}^{W} |V(R_\omega)|$.
Find a solution to each Bipartite Traffic Grooming subproblem independently using step **(2)**, and output the union of all solutions.

(2) To find a solution to each Bipartite Traffic Grooming subproblem in a bipartite graph R, proceed greedily (until all edges are covered) by finding at step j a subgraph R_j of $G \setminus (R_1 \cup \cdots \cup R_{j-1})$ with at most g edges in the following way: For each $k = 2, \ldots, 2g$ find a subgraph B_k of $R \setminus (R_1 \cup \cdots \cup R_{j-1})$ using the algorithm of [14] for the Dense k-Subgraph problem.
- If for some k^*, $|E(B_{k^*})| > g$, and $|E(B_j)| \leq g$ for all $j < k^*$, remove $|E(B_{k^*})| - g$ arbitrary edges from B_{k^*} and output the densest graph among $B_2, \ldots, B_{k^*-1}, B_{k^*}$.
- Otherwise, output the densest graph among $B_2, \ldots, B_{2g}$.

Let OPT be the optimal solution of Ring Traffic Grooming, and let OPT_1 be the cost of the solution obtained by solving *optimally* all the subproblems generated by step **(1)** of Algorithm $\mathcal{A}$. We prove a lemma before stating Theorem 4.1.

Lemma 4.1. *Let β be a given positive real number. Suppose that there exists an algorithm that finds in any bipartite graph R on at most n vertices, a subgraph with at most g edges which has density at least $1/\beta$ times the density of the densest subgraph with at most g edges. Then in the greedy procedure of step* **(2)** *of Algorithm $\mathcal{A}$, one obtains a solution of cost OPT_2 such that $OPT_2 \leq \mathcal{O}(\log n) \cdot \beta \cdot OPT_1$.*

Proof. Let m be the number of edges of the request graph R, and let $R_1, R_2, \ldots, R_r$ be the subgraphs generated in order by the above algorithm. We will prove that $\sum |V(R_i)| \leq \log(m) \cdot \beta \cdot OPT_1$. To prove this, we first enumerate the edges of R in order of appearance in R_i's: all the edges in R_1 will be enumerated $e_1, \ldots, e_{g_1}$ ($g_1 = |E(R_1)| \leq g$), all the edges in R_2 will be enumerated $e_{g_1+1}, \ldots, e_{g_1+g_2}$ ($g_2 = |E(R_2)| \leq g$), and so on. Let ρ_i be the density of the subgraph R_i, i.e. $\rho_i = \frac{|E(R_i)|}{|V(R_i)|}$, and $\Sigma = \sum |V(R_i)|$ the total cost of the solution. For every edge $e_j \in R_i$, we define $c(e_j) = \frac{1}{\rho_i}$. We claim that $\sum_j c(e_j) = \Sigma$. To prove this equality just note that $\sum_{e_j \in E(R_i)} c(e_j) = \frac{|E(R_i)|}{\rho_i} = |V(R_i)|$, and so $\sum_j c(e_j) = \sum_i |V(R_i)| = \Sigma$. Let us define R'_i to be the union of $R_i, R_{i+1}, \ldots, R_r$. We define ρ'_i to be the density of the densest subgraph of R'_i containing at most g edges. Let us take an optimal solution for R'_i, i.e. a decomposition of R'_i into subgraphs $A_1, \ldots, A_s$ such that $\sum_{k=1}^{s} |V(A_k)|$ is minimum. Let $\overline{\rho}_1, \ldots, \overline{\rho}_s$ be the density of these subgraphs. We have:

- $\forall k \leq s,\ \ \overline{\rho}_k \leq \rho'_i$: because each A_k is a subgraph of R'_i containing at most g edges, and ρ'_i is the density of the densest subgraph with at most g edges in R'_i.
- $\rho'_i \leq \beta \rho_i$: because we suppose that we can find an approximation of ρ'_i up to a factor $1/\beta$.

This implies that $\frac{1}{\overline{\rho}_k} \geq \frac{1}{\beta \rho_i}$, and so

$$\sum_k |V(A_k)| = \sum_k \frac{|E(A_k)|}{\overline{\rho}_k} \geq \sum_k \frac{|E(A_k)|}{\beta \rho_i} = \frac{|E(R'_i)|}{\beta \rho_i}.$$

But an optimal solution for R provides a solution for R'_i of cost at least the optimal solution for R'_i, i.e. $\sum_k |V(A_k)| \leq OPT_1$. Using this in the above inequality we get $\frac{1}{\rho_i} \leq \frac{\beta \cdot OPT_1}{|E(R'_i)|}$, and so for an edge $e_j \in R_i$ we have $c(e_j) = \frac{1}{\rho_i} \leq \frac{\beta \cdot OPT_1}{|E(R'_i)|} \leq \frac{\beta \cdot OPT_1}{m-j+1}$, and this proves that

$$\Sigma = \sum_j c(e_j) \leq \beta \cdot \left(\sum_j \frac{1}{m-j+1} \right) \cdot OPT_1 \leq \beta \cdot \log(m) \cdot OPT_1 \leq 2\beta \cdot \log(n) \cdot OPT_1. \quad \square$$

Theorem 4.1. *$\mathcal{A}$ is a polynomial-time approximation algorithm that approximates* Ring Traffic Grooming *within a factor $\mathcal{O}(n^{1/3} \log^2 n)$ for any $g \geq 1$.*

Proof. Algorithm $\mathcal{A}$ returns a valid solution of Ring Traffic Grooming, because each request is contained in some bipartite graph, and no request is counted twice. The runtime is polynomial in both n and g, because we run at most $2g - 1$ times the algorithm of [14] for each subproblem, and there are $n(\sum_{i=0}^{t-1} \frac{1}{2^i}) - 1 = 2n - 3$ subproblems. We prove the approximation guarantee:

- We claim that $OPT_1 \leq 2\log n \cdot OPT$. Indeed, let $\overline{c}_i$ be the optimal cost of the subset of requests of length in the interval $[2^i, 2^{i+1})$, $i = 0, \ldots, \log(n) - 1$. It is clear that $\overline{c}_i \leq OPT$ for each i, and thus $\sum_{i=0}^{\log n-1} \overline{c}_i \leq \log n \cdot OPT$. Finally, $OPT_1 \leq 2\sum_{i=0}^{\log n-1} \overline{c}_i$, because each vertex is taken into account in two subproblems.
- The greedy procedure described in step **(2)** of Algorithm $\mathcal{A}$ outputs a graph whose density is at least $\frac{1}{2n^{1/3}}$ times the greatest density (with at most g edges) of the updated request graph. To see that, note that the optimal density is achieved by a subgraph on at most $2g$ vertices (it would be the case of g disjoint edges). Then, for each value of k, the algorithm of [14] finds a $2n^{1/3}$-approximation of the maximum number of edges of an induced subgraph on k vertices.[1] Thus, if we take the densest subgraph among $B_2, \ldots, B_{2g}$ (removing edges if necessary) we also obtain a $2n^{1/3}$-approximation of the greatest density of a subgraph with at most g edges. Let ρ_k be the density of B_k before removing edges. The explicit formula of the greatest density ρ that we output in step **(2)** of Algorithm $\mathcal{A}$ is:

$$\rho := \max_{k\in\{2,\ldots,2g\}} \min\left(\rho_k, \frac{g}{k}\right).$$

  The above formula justifies that the algorithm stops the search at $k = k^*$. Summarizing, we can use $\beta = 2n^{1/3}$ in Lemma 4.1.
- By combining the remarks above and Lemma 4.1 we obtain that the cost A returned by Algorithm $\mathcal{A}$ satisfies $A \leq 2n^{1/3} \cdot OPT_2 \leq 4n^{1/3}\log n \cdot OPT_1 \leq 8n^{1/3}\log^2 n \cdot OPT$. □

We can improve the approximation ratio of the algorithm if all the requests have short length compared to the length of the ring. This situation is usual in practical applications since nodes may want to communicate only with their nearest neighbors. Let $f(n)$ be any function of n. If all the requests have length at most $f(n)$, then the above algorithm provides an approximation ratio of $\mathcal{O}(f(n)^{1/3}\log^2 n)$. Indeed, in step **(2)** of Algorithm $\mathcal{A}$, we have to find dense subgraphs in bipartite graphs of size at most $2f(n)$, hence the factor $2n^{1/3}$ can be replaced with $2(2f(n))^{1/3}$.

Remark that all the instances of Dense k-Subgraph problem in our algorithm are bipartite. Using the results of [27], it is possible to obtain a better approximation ratio when the request graph is bipartite and satisfies some uniform density conditions.

Corollary 4.1. *If the request graph R is such that in any large enough subgraph $H \subseteq R$, a densest subgraph $(A \cup B, E)$ satisfies $|A|, |B| = \mathcal{O}(\sqrt{g})$ and $|E| = \Omega(g)$, then for any constant $\varepsilon > 0$ there exists a polynomial-time algorithm for* Ring Traffic Grooming *with approximation ratio $\mathcal{O}(n^{\varepsilon}\log^2 n)$.*

To end this section, it is interesting to mention that the results of [9] show that the density can be approximated within a constant factor two in the class of graphs excluding a fixed graph H as minor. Thus, if the request graph R is H-minor free (for instance if R is planar, or of bounded genus, etc.), Algorithm $\mathcal{A}$ achieves an approximation factor of $\mathcal{O}(\log^2 n)$.

5. Conclusions and further research

This article dealt with Traffic Grooming, a central problem in WDM optical networks. The contribution of this work can be divided into two main parts: on the one hand, we stated inapproximability results for Ring Traffic Grooming and Path Traffic Grooming for fixed values of g. More precisely, we proved that Ring Traffic Grooming is Apx-complete for fixed $g \geq 1$, and that Path Traffic Grooming is Apx-complete for fixed $g \geq 2$. In other works, we ruled out the existence of a PTAS for fixed values of g. To prove these results we reduced Ring Traffic Grooming for $g = 1$ to the problem of finding the maximum number of edge-disjoint triangles in a graph of degree bounded by B (MECT-B for short). We proved that MECT-B is Apx-complete, and we generalized this reduction for Path Traffic Grooming and for all values of $g \geq 1$. On the other hand, we provided a polynomial-time approximation algorithm for Ring and Path Traffic Grooming with an approximation ratio not depending on g, considering g as part of the input.

A number of interesting questions remain open. First, when g is not part of the input, the non-existence of a PTAS blows the whistle to start the race of finding the best constant-factor approximation for each value of g, for both the ring ($g \geq 1$) and the path ($g \geq 2$). We did not focus on this issue in this article.

Secondly, when g is part of the input, it is a challenging open problem to close the complexity gap of Traffic Grooming, that is, to provide an approximation algorithm with an approximation ratio matching the corresponding inapproximability result. We are convinced that the inherent difficulty of the problem resides in finding dense subgraphs with bounded number of edges. This problem is strongly related to the problem of finding the densest subgraph with bounded number of vertices, which has been recently proved to have, essentially, the same difficulty as the Dense k-Subgraph problem [1]. The non-existence of a PTAS for the Dense k-Subgraph problem has been proved in [22] involving very technical proofs, and this is the best existing hardness result. A long-standing conjecture claims that there exists some constant $\varepsilon > 0$ such that finding a n^{ε}-approximation for Dense k-Subgraph is NP-hard [14]. As we proved in Section 4, an α-approximation for Dense k-Subgraph yields a $\mathcal{O}(\alpha\log^2 n)$-approximation for Ring Traffic Grooming. We suspect that a similar result in the other direction should also exist. Because of this, we conjecture that:

[1] In fact, the improved approximation ratio of the Dense k-Subgraph problem is $\mathcal{O}(n^{\delta})$ for some constant $\delta < 1/3$ [14]. Obviously, the same applies to our algorithm, replacing the exponent 1/3 with the same $\delta < 1/3$.

Conjecture 5.1. *There exists some constant $\delta > 0$, such that* Ring Traffic Grooming *is* NP-*hard to approximate in polynomial time within a factor $\mathcal{O}(n^{\delta})$ when the grooming factor g is part of the input.*

Acknowledgements

The authors thank the anonymous referees for their helpful comments that improved the presentation of the paper.

References

[1] R. Andersen, Finding large and small dense subgraphs, February 2007. Preprint available at: arXiv:cs/0702032v1.
[2] V. Bafna, P.A. Pevzner, Genome rearrangements and sorting by reversals, SIAM Journal on Computing 25 (1996) 272–289.
[3] J.-C. Bermond, L. Braud, D. Coudert, Traffic grooming on the path, Theoretical Computer Science 384 (2–3) (2007) 139–151.
[4] J.-C. Bermond, D. Coudert, Grooming, in: C.J. Colbourn, J.H. Dinitz (Eds.), The CRC Handbook of Combinatorial Designs, 2nd edition, in: Discrete Mathematics and Its Applications, vol. 42, CRC Press, 2006, pp. 493–496 (Chapter VI.27).
[5] J.-C. Bermond, D. Coudert, X. Muñoz, I. Sau, Traffic grooming in bidirectional WDM ring networks, in: IEEE-LEOS ICTON/COST 293 GRAAL, vol. 3, 2006, pp. 19–22.
[6] A. Caprara, R. Rizzi, Packing triangles in bounded degree graphs, Information Processing Letters 84 (4) (2002) 175–180.
[7] A. Chiu, E. Modiano, Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks, IEEE/OSA Journal of Lightwave Technology 18 (1) (2000) 2–12.
[8] T. Chow, P. Lin, The ring grooming problem, Networks 44 (2004) 194–202.
[9] E. Demaine, M.T. Hajiaghayi, K.C. Kawarabayashi, Algorithmic graph minor theory: Decomposition, approximation and coloring, in: 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS, 2005, pp. 637–646.
[10] R. Diestel, Graph Theory, Springer-Verlag, 2005.
[11] R. Dutta, G. Rouskas, On optimal traffic grooming in WDM rings, in; ACM Sigmetrics/Performance, 2001, pp. 164–174.
[12] R. Dutta, N. Rouskas, Traffic grooming in WDM networks: Past and future, IEEE Network 16 (6) (2002) 46–56.
[13] L. Epstein, A. Levin, Better bounds for minimizing SONET ADMs, in: Workshop on Approximation and On-line Algorithms, WAOA, in: LNCS, vol. 3351, 2004, pp. 281–294.
[14] U. Feige, D. Peleg, G. Kortsarz, The dense *k*-subgraph problem, Algorithmica 29 (3) (2001) 410–421.
[15] M. Flammini, L. Moscardelli, M. Shalom, S. Zaks, Approximating the traffic grooming problem, Journal of Discrete Algorithms 6 (3) (2008) 472–479.
[16] O. Gerstel, R. Ramaswami, G. Sasaki, Cost effective traffic grooming in WDM rings, in: IEEE INFOCOM, 1998, pp. 69–77.
[17] J. Håstad, Some optimal inapproximability results, Journal of the ACM 48 (4) (2001) 798–859.
[18] I. Holyer, The NP-completeness of some edge-partition problems, SIAM Journal on Computing 10 (4) (1981) 713–717.
[19] S. Huang, R. Dutta, G. Rouskas, Traffic grooming in path, star, and tree networks: Complexity, bounds, and algorithms, IEEE Journal on Selected Areas in Communications 24 (4) (2006) 66–82.
[20] C.A.J. Hurkens, A. Schrijver, On the size of systems of sets every *t* of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems, SIAM Journal on Discrete Mathematics 2 (1) (1989) 68–72.
[21] V. Kann, Maximum bounded 3-dimensional matching is MAX SNP-complete, Information Processing Letters 37 (1991) 27–35.
[22] S. Khot, Ruling out PTAS for graph min-bisection, dense *k*-subgraph, and bipartite clique, SIAM Journal on Computing 36 (4) (2006) 1025–1071.
[23] L. Mathieson, E. Prieto, P. Shaw, Packing edge disjoint triangles: A parameterized view, in: the International Workshop on Parameterized and Exact Computation, IWPEC, in: LNCS, vol. 3162, 2004, pp. 127–137.
[24] E. Modiano, P. Lin, Traffic grooming in WDM networks, IEEE Communications Magazine 39 (7) (2001) 124–129.
[25] X. Muñoz, I. Sau, Traffic grooming in unidirectional WDM rings with bounded degree request graph, in: The 34th International Workshop on Graph-Theoretic Concepts in Computer Science, WG, in: LNCS, vol. 5344, 2008, pp. 300–311.
[26] M. Shalom, W. Unger, S. Zaks, On the complexity of the traffic grooming problem in optical networks, in: Fun with Algorithms, 4th International Conference, in: LNCS, vol. 4475, 2007, pp. 262–271.
[27] A. Suzuki, T. Tokuyama, Dense subgraph problem revisited, in: Joint Workshop New Horizons in Computing and Statistical Mechanical Approach to Probabilistic Information Processing, 2005.
[28] K. Zhu, B. Mukherjee, A review of traffic grooming in WDM optical networks: Architectures and challenges, Optical Networks 4 (2) (2003) 55–64.