The Ring Grooming Problem

Timothy Y. Chow
250 Whitwell Street #2, Quincy, Massachusetts 02169

Philip J. Lin
11 June Lane, Newton, Massachusetts 02459

The problem of minimizing the number of bidirectional SONET rings required to support a given traffic demand has been studied by several researchers. Here we study the related *ring-grooming* problem of minimizing the number of add/drop locations instead of the number of rings; in a number of situations this is a better approximation to the true equipment cost. Our main result is a new lower bound for the case of uniform traffic. This allows us to prove that a certain simple algorithm for uniform traffic is, in fact, a constant-factor approximation algorithm, and it also demonstrates that known lower bounds for the general problem—in particular, the linear programming relaxation—are not within a constant factor of the optimum. We also show that our results for uniform traffic extend readily to the more practically important case of quasi-uniform traffic. Finally, we show that if the number of nodes on the ring is fixed, then ring grooming is solvable in polynomial time; however, whether ring grooming is fixed-parameter tractable is still an open question. © 2004 Wiley Periodicals, Inc. NETWORKS, Vol. 44(3), 194–202 2004

Keywords: ADM minimization; ring loading; BLSR; SONET over WDM; traffic grooming; parameterized complexity

1. INTRODUCTION

Many of today's telecommunication networks carry traffic on a configuration known as a *SONET BLSR* or a *SONET bidirectional ring.* A number of authors [2, 5, 11, 13, 17, 18] have studied the so-called *ring-loading problem,* which is the problem of minimizing the number of concentric (or "stacked") SONET bidirectional rings required to carry a given amount of traffic. Some good approximation algorithms have been obtained for this problem.

The ring-loading problem has the advantage of being easy to state and amenable to rigorous analysis, but has the drawback that its cost function gives only an approximation to the true cost of building a SONET ring network. In some situations, particularly in so-called "SONET over WDM" networks, a better approximation to the cost may be obtained by defining the cost of a SONET ring to be proportional to the number of its *add/drop points,* that is, locations on the ring where traffic terminates (i.e., begins or ends). The idea is that at add/drop points, costly electronic multiplexers (ADMs) must be installed, whereas elsewhere one can have an "optical passthrough" or "glassthrough" whose cost is relatively negligible.

The problem of minimizing ADMs in SONET rings, which is a special case of a more general problem known as *traffic grooming,* has also attracted the attention of numerous researchers [7, 12, 14–16, 19–23], but these articles have concentrated either on heuristics or ILP methods without any provable a priori performance bounds, or on proving bounds under special assumptions such as shortest path routing. (In the case of unidirectional rings, however, there has been some recent work on approximation algorithms [1, 9].) Part of the reason for this dearth of theoretical results is that the ring-grooming problem appears to be more difficult to analyze than the ring-loading problem. Even in the case of uniform traffic on a bidirectional ring, the answer to the following question was not known prior to the present article: Does there exist an α-approximation algorithm for some absolute constant α? Theorem 2 below answers this question affirmatively. In fact, we show that if the traffic is *K-quasi-uniform* (i.e., the largest traffic demand between any two nodes is at most K times the traffic between any other two nodes), then there is an α-approximation algorithm with α depending only on K.

We should add that our approximation algorithm is not of tremendous practical value, as its guaranteed approximation ratio is poor. More important than the algorithm itself is Theorem 1, a new lower bound that allows us to *prove* that our algorithm is in fact a constant-factor approximation algorithm. In certain cases, Theorem 1 is an order of magnitude better than previous bounds [$O(n^{3/2})$ vs. $O(n)$]. This has an important theoretical implication: If one wishes to prove that a given algorithm—either an existing algorithm

Received September 2003; accepted January 2004
Correspondence to: T. Chow
DOI 10.1002/net.20029
Published online in Wiley InterScience (www.interscience.wiley.com).

or a new algorithm—is a constant-factor approximation algorithm for the general case, then one must first improve the best known lower bounds.

2. STATEMENT OF PROBLEM

We first give an informal description, which we follow with a precise integer linear programming (ILP) formulation of the problem.

An instance of the ring grooming problem consists of positive integers $n \geq 2$ (the *ring size*) and c (the *capacity*), and a list $\{j_1, k_1\}, \{j_2, k_2\}, \ldots, \{j_s, k_s\}$ of unordered pairs of integers between 1 and n inclusive; these are the *traffic demands.* (For all i, we require $j_i \neq k_i$, but the list may contain the same pair of integers multiple times.) Let C_n be the cycle graph with n vertices, labeled 1 through n in clockwise order. On this cycle are "stacked" multiple rings, each with capacity c on each edge. The traffic must be routed on these rings, that is, for each traffic demand $\{j, k\}$ in the list, we must choose one of the rings and one of the two arcs on that ring between j and k (either the "minor arc" or the "major arc") for its route. To compute the cost of a routing, we consider each of the stacked rings in turn. If there exists any traffic terminating at the jth vertex of the ith ring, then an ADM must be placed on that ring at that vertex. (Recall that "terminating" means either beginning or ending, so in particular an ADM is required at both ends of a demand pair.) The goal is to find a routing with the minimum number of ADMs.

Now for the formal description. Given an instance, we first set d_{jk} equal to the number of times the pair $\{j, k\}$ occurs in the list of traffic demands. Note that $d_{jk} = d_{kj}$ and that $d_{jj} = 0$ for all j. We refer to (d_{jk}) as the *traffic matrix.* We then set

$$R = \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} d_{jk}.$$

(This should be thought of as an upper bound on the number of rings we will stack.) To formalize major arcs and minor arcs, we define the symbol δ_{jkl} for $1 \leq j < k \leq n$ and $1 \leq l \leq n$ by setting $\delta_{jkl} = 1$ if the arc in C_n between vertices j and k that contains the edge between vertices l and $l + 1$ (if $l = n$ then this means the edge between vertices n and 1) also contains the edge between vertices n and 1, and setting $\delta_{jkl} = 0$ otherwise.

The variables of our ILP are of two types: 0–1 variables x_{ij} for $1 \leq i \leq R$ and $1 \leq j \leq n$ (indicating whether any traffic terminates at the jth vertex of the ith ring) and nonnegative integer variables t^0_{ijk} and t^1_{ijk} for $1 \leq i \leq R$ and $1 \leq j < k \leq n$ (indicating the amount of traffic between vertices j and k on ring i on each of the two arcs). The constraints are

$$\sum_{i=1}^{R} (t^0_{ijk} + t^1_{ijk}) = d_{jk}, \quad \forall\, j, k(j < k) \qquad (1)$$

$$\sum_{j=1}^{n-1} \sum_{k=j+1}^{n} (\delta_{jkl} t^0_{ijk} + (1 - \delta_{jkl}) t^1_{ijk}) \leq c, \quad \forall\, i, l \qquad (2)$$

$$\sum_{k=1}^{j-1} (t^0_{ikj} + t^1_{ikj}) + \sum_{k=j+1}^{n} (t^0_{ijk} + t^1_{ijk}) \leq 2cx_{ij}, \quad \forall\, i, j \qquad (3)$$

$$0 \leq x_{ij} \leq 1; \quad t^0_{ijk} \geq 0; \quad t^1_{ijk} \geq 0 \quad \forall\, i, j, k \qquad (4)$$

Constraint (1) forces all the traffic to be routed, constraint (2) enforces the capacity constraint, and constraint (3) forces ADMs to be placed where traffic terminates. The goal is to minimize the objective function $\Sigma_{i,j}\, x_{ij}$ subject to these constraints; formally,

$$m := \min\left\{ \sum_{i,j} x_{ij} \;\middle|\; (x, t) \text{ satisfies constraints (1)–(4)} \right\}. \qquad (\dagger)$$

This completes the formal statement of the problem, but some additional comments are in order. Experts may notice that the version of the ring grooming problem that we consider here differs in some details from other versions in the literature. These variations are discussed in Section 7.

A trivial upper bound on m is $2R$, obtained by putting each unit of traffic on a separate ring. From this we see that if we choose any integer $R' > R$ and consider the ILP that is defined exactly as above except with R' in place of R, then the value of m will remain unchanged. In other words, there is some leeway in the definition of R; any sufficiently large value will do, but to be formal we need to fix a specific value.

The ring grooming problem is an optimization problem, and as usual it may be converted into a decision problem by introducing a bound M and asking if there exists a solution with $m \leq M$. Clearly, this decision problem is in NP. Note, however, that if the traffic demands had been specified not as a list but in the more succinct form of the traffic matrix (d_{jk}), then it would no longer be clear that ring grooming would be in NP, because then the number of rings required could potentially be exponential in the size of the input.

In the rest of the article we mostly use the informal language of rings and ADMs rather than the formal ILP description. So, for example, the reader should think of a variable number r of rings, all of which carry traffic, rather than a fixed number R of rings, some of which may be empty.

3. RING GROOMING COMPARED TO RING LOADING

In view of the similarity between ring grooming and ring loading, we might wonder whether the two problems are equivalent. Note first that the ring-loading problem as stated in [18] does not allow the demand between a given pair of nodes to be split between the major and minor arcs, whereas the ring grooming problem, at least as we have stated it, does. Hence, we should really compare ring grooming to the special case of ring loading treated in [5], which allows traffic splitting, and which can be solved in polynomial time.

As we shall see in the next section, even with traffic splitting allowed, ring grooming is NP-complete, so the two problems are manifestly not equivalent. To further highlight the difference, consider the following instance. Let $n = 9$, let $c = 1$, and let the list of traffic demands be

$$\{1, 2\}, \{1, 3\}, \{2, 3\}, \{4, 5\}, \{4, 6\}, \{5, 6\}, \{7, 8\}, \{7, 9\}, \{8, 9\}.$$

One can check that the optimal ring-grooming solution uses three rings: one with ADMs at vertices 1, 2, and 3, one with ADMs at 4, 5, and 6, and one with ADMs at 7, 8, and 9, for a total of nine ADMs. On the other hand, the optimal ring-loading solution uses 2 rings and 15 ADMs: one with ADMs at all nine vertices, and one with ADMs at vertices 1, 3, 4, 6, 7, and 9. In particular, the number of rings and the number of ADMs cannot be simultaneously minimized. This impossibility of simultaneous minimization is what we would expect from previous work on similar problems (e.g., [8]).

More complicated examples can easily be constructed, and it appears that none of the techniques used successfully for ring loading carry over readily to ring grooming.

4. RELATIONSHIP TO BIN PACKING

It is known [15, 16] that traffic grooming on rings is reducible from bin packing. However, strictly speaking, the NP-completeness of ring grooming as we have defined it is not explicitly proved in the literature, so we give a complete proof.

Proposition 1. *Ring grooming is NP-complete.*

Proof. Recall that an instance of bin packing consists of a positive integer B (the *bin size*) and a set $A = \{a_1, \ldots, a_N\}$ of positive integers (the sizes of the objects to be packed) such that $a_i \leq B$ for all i. The objective is to partition A into as few disjoint subsets as possible, subject to the constraint that the sum of the a_i in each subset is at most B. Now, it is well known [6] that bin packing is in fact *strongly* NP-complete, so we may (and will) assume that the a_i are given in *unary* rather than in *binary.*

Given an instance of bin packing, set $n = N + 1$ and $c = B$. For each j in the range $1 \leq j \leq n - 1$, add $2a_j$ copies of the pair $\{j, n\}$ to the list of traffic demands. This gives us an instance of ring grooming (whose size is polynomial in the size of the bin packing instance, because the a_j are given in unary). If m is the minimum number of ADMs for this instance, then we claim that the minimum number of bins for the original bin packing instance is $m - N$. This implies in particular that (the decision version of) ring grooming is NP-complete.

To prove the claim, we begin by observing that we may assume that the optimal solution to the ring grooming instance has the following property: If $j < n$, then there is only one ring with an ADM at vertex j. For suppose we have a solution S with r rings in which there is some vertex $j < n$ at which there is an ADM on rings $i_1, i_2, \ldots, i_s$ with $s \geq 2$. Then we can create a new solution S' by adding a new ring to S, putting ADMs at vertices j and n on this new ring, transferring all the traffic between vertices j and n from rings $i_1, i_2, \ldots, i_s$ to the new ring, and deleting the ADMs at vertex j from rings $i_1, i_2, \ldots, i_s$. This is always feasible, because the total traffic between vertices j and n is $2a_j$, which is at most $2B = 2c$ and, therefore, fits onto a single ring (we route a_j units on the major arc and a_j units on the minor arc). Moreover, because $s \geq 2$, when we pass from S to S' we add two ADMs and delete at least two ADMs, so S' contains no more ADMs than S.

For the remainder of the proof we restrict ourselves to configurations such that every ring has an ADM at vertex n (this is necessary because all traffic terminates at vertex n) and such that, for each $j < n$, exactly one ring has an ADM at vertex j. So if r is the number of rings, then the number of ADMs is $m = r + N$. Because N is given, minimizing m is equivalent to minimizing r.

All that remains is to establish a one-to-one correspondence between rings and bins. Specifically, we need to show that for any $J \subseteq \{1, 2, \ldots, N\}$, we have

$$\sum_{j \in J} a_j \leq B$$

if and only if all the demands $\{d_{jn} \mid j \in J\}$ can be routed on a ring with ADMs at vertex n and all vertices $j \in J$. (As always, (d_{jk}) is the traffic matrix.) Suppose first that $\sum_{j \in J} a_j \leq B = c$. Then for each $j \in J$, we need to route $d_{jn} = 2a_j$ units of traffic between vertices j and n. We do this by routing a_j units of the major arc and a_j units on the minor arc; the given inequality ensures that no edge's capacity is exceeded, and therefore, the proposed routing is feasible. Conversely, given any feasible routing, consider the "cut" that separates vertex n from the rest of the ring. There are two edges across this cut, each with capacity c, so the total amount of traffic across the cut is at most $2c$, i.e.,

$$2c \geq \sum_{j \in J} d_{jn} = \sum_{j \in J} 2a_j,$$

or $\Sigma_{j \in J} a_j \leq c = B$ as required. ■

If the bin size is fixed, then bin packing is solvable in polynomial time [6]. A similar result holds for ring grooming.

Proposition 2. *If n is fixed, then ring grooming is solvable in polynomial time.*

Proof. Create a graph whose vertices are vectors with $n(n-1)/2$ coordinates, with one coordinate corresponding to each pair $\{j, k\}$ of distinct integers from 1 to n, and with each coordinate taking on a value between zero and d_{jk}. Note that the number of such vectors is polynomial in the size of the instance. If u and v are two such vectors, then we draw an edge from u to v if the coordinates of $v - u$ are all nonnegative and if the traffic demands represented by $v - u$ all fit onto a single ring of capacity c. (Determining whether such an edge exists requires solving a ring-loading problem with traffic splitting, which as we remarked before can be solved in polynomial time [5].) The weight of such an edge, if it exists, equals the number of ADMs required to support the single ring with the demands $v - u$. Ring grooming now reduces to finding the minimum-weight path from the zero vector to the vector with coordinates d_{jk}. ■

Although theoretically polynomial-time, the algorithm of Proposition 2 is not practical for typical values of n and c (i.e., $n \leq 16$, and c in the hundreds). Intuitively, this is because the degree of the polynomial depends on n; in the terminology of parameterized complexity [4], Proposition 2 shows only that ring grooming is in XP and not necessarily in FPT (if n is the parameter). Michael Fellows (personal communication) has shown that if the number of ADMs is taken to be the parameter, then ring grooming is in FPT. Unfortunately, the number of ADMs tends to be much larger than the ring size, so it remains an interesting open problem whether ring grooming is in FPT if n is the parameter.

5. LOWER BOUNDS

The simplest lower bound on m is the *LP bound,* obtained by relaxing the integer variables to rational (or real) variables. It is easy to show that the LP bound is given by the explicit formula

$$m \geq \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{d_{jk}}{2c}.$$

Unfortunately, although this bound is easy to compute, we shall see shortly that the integrality gap is unbounded (this is one reason why ring grooming appears to be hard to solve). A slightly better bound may be obtained by observing that $\Sigma_k d_{jk}$ is the total amount of traffic terminating at vertex j and that each ADM can handle at most $2c$ units of traffic. Because the number of ADMs is an integer, this yields

$$m \geq \sum_{j=1}^{n} \left\lceil \sum_{k=1}^{n} \frac{d_{jk}}{2c} \right\rceil.$$

We call this the *add/drop lower bound.* Notice that it is slightly better than rounding up the LP bound to the nearest integer, because we round up at each vertex j separately before summing over j.

It is tempting to wonder if there is any sense in which the ceiling signs may be pushed inside the inner sum as well. This cannot be done naïvely, but in [7] the authors give a more delicate argument along these lines that yields a lower bound that is sometimes better than the add/drop lower bound. To state this bound we need some notation. Define q_{jk} and r_{jk} to be the quotient and remainder when d_{jk} is divided by c, that is, choose q_{jk} and r_{jk} such that

$$d_{jk} = cq_{jk} + r_{jk} \quad \text{with } 0 \leq r_{jk} < c.$$

Next, order the d_{jk} with $j < k$ in such a way that their corresponding remainders r_{jk} decrease monotonically. For simplicity, we write D_p, Q_p, and R_p, respectively, for d_{jk}, q_{jk}, and r_{jk}, where p runs from 1 to $n(n-1)/2$ and the labeling is chosen so that

$$R_p \geq R_{p'} \quad \text{whenever } p < p'.$$

Proposition 3 [7]. *Suppose we are given an instance of the ring grooming problem. With the notation above, let P be the smallest integer such that*

$$\left(\sum_{p=1}^{n(n-1)/2} cQ_p \right) + \left(\sum_{p=1}^{P} \frac{R_p + c}{2} \right) \geq \sum_{p=1}^{n(n-1)/2} D_p.$$

Then regardless of the routing, the minimum number m of ADMs as defined in (†) must satisfy

$$m \geq P + \sum_{p=1}^{N(N-1)/2} Q_p.$$

The add/drop lower bound and the lower bound of Proposition 3 are the best known lower bounds in general. However, for K-quasi-uniform traffic, a better lower bound is possible, and this is the first main theorem of this article. Recall from the introduction that "K-quasi-uniform" means that $d_{jk}/d_{j'k'} \leq K$ for all $j \neq k$, $j' \neq k'$. In particular, $d_{jk} \neq 0$ for all $j \neq k$. Conversely, if $d_{jk} \neq 0$ for all $j \neq k$, then the traffic is K-quasi-uniform for some $K \geq 1$. Uniform traffic corresponds to the case $K = 1$. Because bidirectional rings are often used in the core of a network, where every node typically has traffic to every other node, quasi-uniform traffic is not a bad approximation to reality. However, we shall see below that our results give bounds that depends on K, so we are still far from satisfactorily solving arbitrary instances of ring grooming.

We need some further terminology and notation. The *bandwidth* consumed by a traffic demand is the capacity of the demand multiplied by the number of edges that it traverses from end to end. For instance, in the example given in Section 3, the solution using shortest path routing consumes 12 units of bandwidth, while the nine-ADM solution consumes 27 units of bandwidth.

Our results for quasi-uniform traffic are really just easy corollaries of the case of uniform traffic, so let us now focus attention on uniform traffic. Then all the d_{jk} with $j \neq k$ are equal, and we use the letter d to denote this common value. It is also convenient to define

$$f \overset{\text{def}}{=} \frac{d}{2c}.$$

Theorem 1. *With the notation above, the minimum number m of ADMs required for uniform traffic must, regardless of routing, satisfy*

$$m \geq \frac{(n^2-1)\sqrt{f}}{4}.$$

Proof. Let S be any feasible solution to the given instance. For each ADM A in the solution, let $R(A)$ denote the ring that A is on, and define $B(A)$ by

$$B(A) \overset{\text{def}}{=} \frac{\text{Total bandwidth of the traffic carried on } R(A)}{\text{Total number of ADMs on } R(A)}.$$

Now suppose that on a particular ring in S, there are x ADMs. Then there are $x(x-1)/2$ pairs of ADMs, and between each pair there are at most d units of traffic on his ring (because there are d units total of traffic between this pair). The circumference of the ring is n edges, so the total bandwidth on this ring is at most $dnx(x-1)/2$.

On the other hand, the total bandwidth on this ring is at most cn just by capacity constraints. Therefore, if $R(A)$ has x ADMs, then

$$B(A) \leq \frac{\min(dnx(x-1)/2, cn)}{x} = cn \cdot \min(f(x-1), 1/x)$$

$$\leq cn \cdot \min(fx, 1/x) \leq cn\sqrt{f}.$$

(The last inequality in this chain may be obtained, for example, by graphing the curves $y = fx$ and $y = 1/x$ and finding their point of intersection, or by noting that the minimum of two positive reals is less than or equal to their geometric mean.) This inequality is independent of x, so if we sum over all ADMs A in S we find that the total bandwidth B used by all rings satisfies

$$B \leq mcn\sqrt{f}.$$

To obtain a lower bound for B, note that shortest path routing uses the minimum amount of bandwidth. It is well known and easy to prove that

$$B \geq \frac{dn(n^2-1)}{8} = \frac{fcn(n^2-1)}{4}.$$

(If n is even, one can improve this bound to $dn^3/8$, but we ignore this.) Combining the lower and upper bounds for B and solving for m yields the theorem. ∎

To compare this bound with the add/drop lower bound and the lower bound of Proposition 3, consider the special case in which $n = 2c + 1$ (where c as always is the capacity) and $d = 1$. Then the add/drop lower bound is n and the bound of Proposition 3 is about $2n$, while the lower bound of Theorem 1 is about $\frac{1}{4}n^{3/2}$. Thus, for the small values of n that typically occur in practice, the bound of Theorem 1 is actually worse than the previous bounds; however, as explained in the introduction, it is a significant improvement theoretically.

Note that the proof of Theorem 1 never uses the integrality of the traffic variables t^0_{ijk} and t^1_{ijk}. The lower bound, therefore, also holds for the "semi-relaxation" of ring grooming in which these variables (but not the x_{ij} variables) are allowed to be arbitrary real numbers.

Corollary 1. *If an instance of ring grooming is K-quasi-uniform, with d being the largest value of the d_{jk}, then the minimum number m of ADMs must satisfy*

$$m \geq \frac{(n^2-1)\sqrt{d/(2cK)}}{4}.$$

Proof. Between any two nodes there are at least d/K units of traffic, and by Theorem 1, supporting just this subset of the traffic already requires at least $\frac{1}{4}(n^2 - 1)\sqrt{d/(2cK)}$ ADMs. ∎

6. APPROXIMATION ALGORITHM FOR *K*-QUASI-UNIFORM TRAFFIC

Again, we focus first on the case of uniform traffic. Given n, c, and d, we define $f = d/2c$ as before. Then our approximation algorithm, which we call Algorithm A, proceeds as follows. If $f \geq 1$ then simply create, for each pair of vertices j and k, a set of $\lceil f \rceil$ rings with just two ADMs—one at j and one at k—and route all the traffic between j and k on these rings (and then terminate the algorithm). Otherwise, let

$$M = \begin{cases} \left\lfloor \sqrt{2/f} \right\rfloor, & \text{if } 2 \le \left\lfloor \sqrt{2/f} \right\rfloor \le n; \\ 2, & \text{if } \left\lfloor \sqrt{2/f} \right\rfloor < 2; \\ n, & \text{if } \left\lfloor \sqrt{2/f} \right\rfloor > n. \end{cases}$$

Construct an $(n, M, 2)$-*covering design,* that is, a family of M-element subsets, called *blocks,* of $\{1, 2, \ldots, n\}$ such that every pair of integers (j, k) with $1 \le j < k \le n$ is contained in at least one of the blocks. Finally, take each block $\{v_1, v_2, \ldots, v_M\}$ in turn, and construct a ring with an ADM at each v_i, routing *all* the demands among these v_i on this ring (provided that these demands have not already been routed on a previous ring). If at the end of this process there are some ADMs at which no traffic terminates, then discard them, and terminate.

This completes the description of Algorithm A, but several steps require further elaboration and justification. First we need a fast method of constructing an $(n, M, 2)$-covering design. This is easy: Let $\mu = \lfloor M/2 \rfloor$, and let F be a family of μ-element subsets of $\{1, 2, \ldots, n\}$ whose union is the entire set. Assume further that F has the minimum possible size, that is, that it has $\lceil n/\mu \rceil$ members. We obtain an $(n, M, 2)$-covering design of size at most

$$\binom{\lceil n/\mu \rceil}{2}$$

by taking all pairs of members of F (if M is odd, then we must add an additional integer—it does not matter which one—to each such pair to make the size of each block exactly equal to M) and discarding any duplicates.

Next, we need to show that M is small enough for all the designated traffic to fit onto a single ring, and we also need to provide a fast method for computing an actual routing. This is accomplished by the following lemma and its proof.

Lemma 1. *Let $f = d/2c$ and suppose $f \le 1$ and*

$$\nu = \begin{cases} \left\lfloor \sqrt{2/f} \right\rfloor, & \text{if } 2 \le \left\lfloor \sqrt{2/f} \right\rfloor; \\ 2, & \text{if } \left\lfloor \sqrt{2/f} \right\rfloor < 2. \end{cases}$$

Then a single ring with ν nodes and capacity c can support uniform traffic of d units.

Proof. Except for the fact that traffic can be split, this is essentially a ring loading problem, so the reader may expect us to apply known results for ring loading. However, a naïve method suffices for our present purposes: For each pair of nodes, route $\lfloor d/2 \rfloor$ units of traffic on the major (longer) arc and $\lceil d/2 \rceil$ units of traffic on the minor arc. When both arcs are of equal length, either one may be designated the major arc. There are four cases: d may be even ($d = 2\delta$) or odd ($d = 2\delta + 1$), and independently, ν may be even ($\nu = 2\mu$) or odd ($\nu = 2\mu + 1$). In three of these cases it is easy to calculate the amount of traffic on the most heavily loaded link. The exceptional case is $\nu = 2\mu$ and $d = 2\delta + 1$, where the maximum load depends on exactly how the major/minor arcs are chosen for diametrically opposite pairs of vertices. In this case, we obtain an upper bound by observing that the worst case occurs when there is some edge of C_n that is systematically chosen to be the "major arc" in all these tie-breaking situations.

The table below summarizes the results of these calculations.

	$d = 2\delta$	$d = 2\delta + 1$
$\nu = 2\mu$	$\mu(2\mu - 1)\delta$	$\frac{1}{2}\mu[(\mu + 1)(\delta + 1) + (3\mu - 3)\delta]$
$\nu = 2\mu + 1$	$\mu(2\mu + 1)\delta$	$\frac{1}{2}\mu[(\mu + 1)(\delta + 1) + (3\mu + 1)\delta]$

(In the right-hand column, the two summands correspond to the minor arcs and major arcs, respectively, that contain the most heavily loaded link.) We need to prove that in every case, c is large enough to accommodate the most heavily loaded link. First, let us suppose that $\nu = 2$. Then the first row of the table applies, with $\mu = 1$; the two entries reduce to δ and $\delta + 1$, respectively, and in either case this is at most $\lceil d/2 \rceil$. Because $f \le 1$ and c is an integer, we have $c \ge \lceil d/2 \rceil$, which shows that there is enough capacity if $\nu = 2$.

If $\nu \ne 2$, then we must have $\nu \le \sqrt{2/f}$, which implies that $c \ge \nu^2 d/4$. To check that there is sufficient capacity, we therefore just need to check, for each of the four entries in the above table, that if we subtract it from $\nu^2 d/4$, the result is always nonnegative. If $\nu = 2\mu$ and $d = 2\delta + 1$ then

$$\frac{\nu^2 d}{4} - \frac{\mu[(\mu + 1)(\delta + 1) + (3\mu - 3)\delta]}{2} = \frac{\mu(\mu + 2\delta - 1)}{2},$$

which is always nonnegative (because, e.g., $\mu \ge 1$). In the other three cases, it is easily checked that the analogous calculation yields a polynomial in μ and δ with nonnegative coefficients, which is therefore always nonnegative. ■

Finally, we need to show that Algorithm A is indeed an approximation algorithm.

Theorem 2. *Algorithm A is a $12\sqrt{2}$-approximation algorithm.*

Proof. If $f \ge 1$, then Algorithm A uses $n(n - 1)\lceil f \rceil$ ADMs, and the LP bound is $n(n - 1)f$. Dividing the former by the latter yields $\lceil f \rceil/f$, which is at most 2 because $f \ge 1$. So this case is all right.

Now suppose $f < 1$. The number of ADMs used is at most

$$m = \binom{\lceil n/\mu \rceil}{2} M,$$

where $\mu = \lfloor M/2 \rfloor$. We need to show that m is at most $12\sqrt{2}$ times the minimum number of ADMs. We begin by noting that the case $M = n$ is easy: Algorithm A generates a single ring with n ADMs, which Lemma 1 tells us can handle all the demands, and because there is some traffic terminating at every vertex, this solution is optimal. Therefore, we assume for the rest of the proof that $M < n$, which implies in particular that $\sqrt{2/f} < M + 1$ (by definition of M).

Let q denote m divided by the lower bound of Theorem 1. Consider first the case $\mu = 1$. If $\mu = 1$ then $M \le 3$ and $\sqrt{2/f} < 4$ so

$$q = \frac{m}{\frac{1}{4}(n^2 - 1)\sqrt{f}} = \frac{2n(n-1)M}{(n+1)(n-1)\sqrt{f}} < \frac{12n\sqrt{2}}{n+1} < 12\sqrt{2},$$

as required.

So suppose that $\mu \ge 2$. Now $\lceil n/\mu \rceil \le (n + \mu - 1)/\mu$ and $M \le 2\mu + 1$ so

$$m \le \frac{1}{2}\left(\frac{n+\mu-1}{\mu}\right)\left(\frac{n-1}{\mu}\right)(2\mu + 1).$$

Dividing by $\frac{1}{4}(n^2 - 1)\sqrt{f}$ and using the inequality $\sqrt{2/f} < M + 1 \le 2\mu + 2$ yields

$$q \le 2\left(\frac{n+\mu-1}{n+1}\right)\frac{2\mu+1}{\mu^2\sqrt{f}}$$
$$\le 2\sqrt{2}\left(\frac{n+\mu-1}{n+1}\right)\frac{(2\mu+1)(\mu+1)}{\mu^2}.$$

Now $\mu - 1 < M/2 < n/2$ so

$$\frac{n+\mu-1}{n+1} < \frac{3n/2}{n+1} < \frac{3}{2}.$$

Also, $(2\mu + 1)(\mu + 1)/\mu^2$ is a decreasing function of μ so it attains its maximum when μ is as small as possible, that is, when $\mu = 2$. Therefore

$$q < 2\sqrt{2}\left(\frac{3}{2}\right)\frac{(2\cdot 2+1)(2+1)}{2^2} = \frac{45\sqrt{2}}{4} < 12\sqrt{2},$$

completing the proof. ■

Corollary 2. *There is an α-approximation algorithm for ring grooming with K-quasi-uniform traffic, where* $\alpha = \max(2K, 12\sqrt{2K})$.

Proof. Given an instance I of ring grooming with K-quasi-uniform traffic, we let d be the size of the largest d_{jk}, and we create an instance I' of ring grooming with uniform traffic by changing every d_{jk} (with $j \ne k$) to d. Then we apply Algorithm A to I' to obtain a solution that a fortiori yields a solution to I. Now we trace through the proof of Theorem 2. If $d \ge 2c$ then at most $n(n-1)\lceil d/2c \rceil$ ADMs are used, but between any two vertices we have at least d/K units of traffic, yielding a lower bound of $n(n-1)d/(2cK)$. Dividing the former by the latter gives a ratio of at most $2K$, because $d \ge 2c$.

Otherwise, $d < 2c$, and the argument in the proof of Theorem 2 for the case $M = n$ still applies to show that the solution is optimal in this case. If $d < 2c$ and $M < n$, then let q' be m divided by the lower bound of Corollary 1. Then $q' = q\sqrt{K}$, where q is defined in the proof of Theorem 2. That proof shows that $q < 12\sqrt{2}$, whence $q' < 12\sqrt{2K}$. ■

Although we are not proposing that Algorithm A be used in a practical implementation, it is of some interest to consider whether the constant $12\sqrt{2}$ can be improved. One possibility is to use a more sophisticated method for constructing $(n, M, 2)$-covering designs (see, e.g., [10]). For a nontrivial example of what a good design can accomplish, suppose that $n = 15$ and $c = d = 1$. The add/drop lower bound is 105. Algorithm A tells us to take $M = 2$, but suppose we take $M = 3$ instead and look for a (15, 3, 2)-covering design. One example of this is the solution to the famous *Kirkman schoolgirl problem* (see [3] or any standard reference on combinatorial designs). This yields an optimal solution with 35 rings and three ADMs per ring.

In general, however, we cannot expect to find such excellent designs fast. Moreover, a trivial lower bound on the size of an $(n, M, 2)$-covering design is $\binom{n}{2}/\binom{M}{2}$, and if we carry this through the proof of Theorem 2, we find that the maximum possible improvement in the guaranteed performance ratio is a factor of 2 or 3.

Another possible improvement is to use a better ring-loading method in Lemma 1. By considering a maximum cut, one can show that the largest improvement factor we can hope for here is $\sqrt{2}$. Achieving this is not easy, though; for example, it is certainly not possible to naïvely replace $\sqrt{2}$ with 2 in Lemma 1: If $c = 2$ and $d = 1$ then M can be at most three, whereas $\lfloor 2/\sqrt{f} \rfloor = 4$.

Finally, perhaps the bound of Theorem 1 can be improved. For the special case of $d = 1$ we have the following argument that gives a slightly better lower bound. The idea is that at any particular ADM, the t units of traffic exiting from a particular side of the ADM all have different destinations, say $v_1, v_2, \ldots, v_t$. Because the network is a ring, this means that the traffic destined for v_j must pass through the ADMs at v_i for all $i < j$, thus "wasting" some of the capacity of these ADMs. If we work through the details of this for the example given at the end of Section 5, we obtain a lower bound of about $\frac{1}{2}n^{3/2}$, a factor-of-2 improvement.

Unfortunately, we do not see how to generalize this argument to arbitrary d.

7. OPEN PROBLEMS AND RELATED QUESTIONS

Of course, the main open problem is to find a constant-factor approximation algorithm for the general ring grooming problem, or to prove that one of the algorithms in the literature is in fact a constant-factor approximation algorithm. As we mentioned above, it is also open whether ring grooming is fixed-parameter tractable when n is the parameter. It would be of practical interest if the answer turns out to be yes, because SONET standards do not permit more than 16 ADMs on a ring.

If a good approximation algorithm cannot be found, then an inapproximability result would be desirable. We suspect that ring grooming is MAXSNP-hard, but we cannot prove it. Ring grooming is reminiscent of several other well-known problems, for example, integer multicommodity flow, clique cover, and facility location, but we have been unable to construct an explicit reduction. Note that comparing Theorem 1 and the LP bound reveals a large integrality gap. We expect that our ILP formulation can be improved; for example, Michel Goemans (personal communication) has pointed out that the following constraints can be added.

$$t^0_{ijk} + t^1_{ijk} \leq d_{jk}x_{ij}, \quad \forall\, i, j, k(j < k) \tag{5a}$$

$$t^0_{ijk} + t^1_{ijk} \leq d_{jk}x_{ik}, \quad \forall\, i, j, k(j < k) \tag{5b}$$

Summing these inequalities over i and using constraint (1), we deduce that $\sum_i x_{ij} \geq 1$ whenever there is any traffic terminating at vertex j, even if the d_{jk} are all small and the x_{ij} are allowed to be arbitrary real numbers. Although this is an improvement, further constraints would be desirable. Unfortunately, they seem to be hard to find.

Our formulation of the ring grooming problem is not the only possible one; other authors have considered slightly different versions. We describe here some of the more important variations.

(1) *Unidirectional rings.* Unlike ring loading, ring grooming is nontrivial even for unidirectional rings, and as mentioned in the introduction, it has been studied by some of the authors listed in the references. Note that some service providers tend to use unidirectional rings in situations where there exists a *hub node* at which all traffic terminates, and standard bin packing algorithms work well here.

(2) *No time slot interchange.* In our formulation, we regard a routing as feasible provided that the total number of units of traffic on each edge of C_n does not exceed c. In actual SONET rings, the c units of capacity are c separate *time slots.* We have assumed that all our ADMs have *time slot interchange* capability, meaning that whenever a unit of traffic passes through an ADM, it can be switched to a different time slot if necessary. Some less expensive ADMs do not have this capability, in which case all traffic must choose a time slot and remain on the same time slot from source to destination. This creates some additional constraints that are akin to graph coloring.

(3) *Crossconnection between rings.* In our formulation, we require that traffic stay on the same ring from source to destination. In some networks, there are *digital cross-connects* installed in the central offices, which allow traffic to switch from one ring to another at a vertex provided that both rings have an ADM at that location. If we assume that crossconnection is available, then ring grooming becomes more similar to multicommodity flow, and perhaps some of the techniques carry over.

(4) *No traffic splitting.* As we have already remarked, our formulation allows traffic to be split between major and minor arcs, as long as all variables remain integers. One might wonder whether this assumption is realistic. Certainly if $d_{jk} > c$ then d_{jk} must at least be split across different rings, and if splitting across rings is permissible, then splitting between major and minor arcs is likely to be permissible too. However, what if $d_{jk} \leq c$? Typically, in real networks, traffic demands come in multiples of a few standard units, for example, the traffic between a particular pair of vertices might be a combination of several units of size 12 and several units of size 3. It would not be permissible to split a size-3 or size-12 unit, either across rings or between major and minor arcs, but, for example, the various units of size 3 could be routed independently. It would be interesting to study the effect of multiple unit sizes.

(5) *Unidirectional traffic and asymmetric routing.* We have assumed not only that our rings are bidirectional, but that our traffic is bidirectional as well, and moreover, that the traffic from j to k must be routed the same way as the traffic from k to j, only in reverse. These are reasonable assumptions for current networks, but some experts predict future increases in unidirectional traffic ($d_{jk} \neq d_{kj}$). Also, even if $d_{jk} = d_{kj}$, it is, in principle, possible to route the two directions independently.

(6) *Dynamic traffic.* In real networks the traffic matrix is usually not fixed, but grows with time. Many network operators have a *cap-and-grow* policy, meaning that existing traffic is not allowed to be rerouted when new traffic arrives. There is, therefore, a practical need for an *online algorithm* for ring grooming. Very little work has been done so far on this important but difficult problem.

Acknowledgments

The first author would like to thank Xingxing Yu and Michel Goemans for useful conversations and Oren Patashnik for pointing him in the right direction regarding covering designs. Most of the work in this article was done while the authors were at the Tellabs Research Center in Cambridge.

REFERENCES

[1] N. Brauner, Y. Crama, G. Finke, P. Lemaire, and C. Wynants, Approximation algorithms for SONET networks, Technical Report GEMME 2002, University of Liège, 2002.

[2] S. Cosares and I. Saniee, An optimization problem related to balancing loads on SONET rings, Telecom Systems 3 (1994), 165–181.

[3] H. Dörrie, 100 great problems of elementary mathematics: Their history and solution, Dover, New York, 1965.

[4] R.G. Downey and M.R. Fellows, Parameterized complexity, Springer, New York, 1999.

[5] A. Frank, T. Nishizeki, N. Saito, H. Suzuki, and E. Tardos, Algorithms for routing around a rectangle, Discrete Appl Math 40 (1992), 363–378.

[6] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W.H. Freeman, San Francisco, 1979.

[7] O. Gerstel, P. Lin, and G. Sasaki, Combined WDM and SONET network design, Proc IEEE INFOCOM '99, pp. 734–743.

[8] O. Gerstel, P. Lin, and G. Sasaki, Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings, Proc IEEE INFOCOM '98, pp. 94–101.

[9] O. Goldschmidt, D.S. Hochbaum, A. Levin, and E.V. Olinick, The SONET edge-partition problem, Networks 41 (2002), 13–23.

[10] D.M. Gordon, O. Patashnik, and G. Kuperberg, New constructions for covering designs, J Combin Designs 3 (1995), 269–284; http://www.ccrwest.org/cover.html.

[11] P. Harshavardhana, P.K. Johri, and R. Nagarajan, A note on weight-based load balancing on SONET rings, Telecom Systems 6 (1996), 237–239.

[12] J.-Q. Hu, Traffic grooming in wavelength-division-multiplexing ring networks: A linear programming solution, J Optical Networking 1 (2002), 397–408.

[13] S. Khanna, A polynomial time approximation scheme for the SONET ring loading problem, Bell Labs Tech J 2 (Spring 1997), 36–41.

[14] Y. Lee, H. Sherali, J. Han, and S. Kim, A branch-and-cut algorithm for solving an intra-ring synchronous optical network design problem, Networks 35 (2000), 223–232.

[15] X.-Y. Li, L.-W. Liu, P.-J. Wan, and O. Frieder, Practical traffic grooming scheme for single-hub SONET/WDM rings, Proc IEEE LCN '00, pp. 556–564.

[16] E.H. Modiano and A.L. Chiu, Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks, J Lightwave Tech 18 (2000), 2–12.

[17] Y.-S. Myung, H.-G. Kim, and D.-W. Tcha, Optimal load balancing on SONET bidirectional rings, Operat Res 45 (1997), 148–152.

[18] A. Schrijver, P. Seymour, and P. Winkler, The ring loading problem, SIAM J Discrete Math 11 (1998), 1–14.

[19] J.M. Simmons, E.L. Goldstein, and A.A.M. Saleh, Quantifying the benefit of wavelength add-drop in WDM rings with distance-independent and dependent traffic, J Lightwave Tech 17 (1999), 48–57.

[20] A. Sutter, F. Vanderbeck, and L. Wolsey, Optimal placement of add/drop multiplexers: Heuristic and exact algorithms, Operat Res 46 (1998), 719–728.

[21] P.-J. Wan, L.-W. Liu, and O. Frieder, Grooming of arbitrary traffic in SONET/WDM BLSRs, IEEE J Selected Areas Comm 18 (2000), 1995–2003.

[22] J. Wang, W. Cho, V. Vemuri, and V. Mukherjee, Improved approaches for cost-effective traffic grooming in WDM ring networks: Non-uniform traffic and bidirectional ring, Proc IEEE ICC '00, pp. 1295–1299.

[23] X. Zhang and C. Qiao, An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings, IEEE/ACM Trans Networking 8 (2000), 608–617.