Competitive Analysis of Online Traffic Grooming in WDM Rings

Karyn Benson, Benjamin Birnbaum, Esteban Molina-Estolano, and Ran Libeskind-Hadas, *Member, IEEE*

***Abstract*—This paper addresses the problem of traffic grooming in wavelength-division multiplexing (WDM) rings where connection requests arrive online. Each request specifies a pair of nodes that wish to communicate and also the desired bandwidth of this connection. If the request is to be satisfied, it must be allocated to one or more wavelengths with sufficient remaining capacity. We consider three distinct profit models specifying the profit associated with satisfying a connection request. We give results on offline and online algorithms for each of the three profit models. We use the paradigm of competitive analysis to theoretically analyze the quality of our online algorithms. Finally, experimental results are given to provide insight into the performance of these algorithms in practice.**

***Index Terms*—Competitive analysis, online algorithms, optical networks, wavelength-division multiplexing (WDM) rings.**

I. Introduction

WAVELENGTH-DIVISION multiplexing (WDM) partitions the capacity of an optical fiber into separate channels that may be transmitted simultaneously over distinct wavelengths. However, traffic demands are frequently much smaller than the capacity of even a single wavelength. In order to maximize utilization of the network, a large number of traffic streams can be multiplexed or "groomed" onto a smaller number of wavelengths.

Rings are among the most widely deployed and studied WDM network topologies. We consider rings in which each node has some number of add–drop multiplexers (ADMs), each tuned to a particular wavelength. A node with an ADM for a particular wavelength can initiate ("add") traffic onto that wavelength and can receive ("drop") traffic on that wavelength. Nodes that do not contain an ADM for a given wavelength can allow traffic on that wavelength to pass through but can neither send nor receive traffic on that wavelength.

Substantial work has been reported on traffic grooming for WDM rings for traffic patterns that are known in advance [1], [2]–[4]. The traffic pattern may be static, and described by a single traffic matrix, or dynamic, and described by a sequence of traffic matrices. Typically, the objective is to minimize the number of ADMs or other equipment costs. These traffic grooming problems are generally *NP*-complete [1], [5] and are often solved optimally by integer linear programming (albeit in worst-case exponential time) or sub-optimally by heuristics.

Recently, several authors have considered online traffic grooming problems, in which requests must be groomed as they arrive [3], [6]–[8]. These studies propose online grooming heuristics with the objective of minimizing blocking probability and related measures. The heuristics are generally evaluated empirically using simulation studies.

In this paper, we study a family of online traffic grooming problems in WDM rings via *competitive analysis*. Competitive analysis provides a theoretical guarantee on the worst-case quality of solutions found by an online algorithm. To the best of our knowledge, this paper presents the first competitive analyses of online algorithms for traffic grooming problems.

A request comprises a pair of nodes, (i, j), and an integral demand, d, for bandwidth. A full-duplex connection between nodes i and j must be established. Thus, the bandwidth allocated for a connection from i to j has equal bandwidth allocated from j to i. We assume that this full-duplex connection uses the same wavelength resources along the entirety of the ring. This situation applies both to unidirectional rings as well as to bidirectional rings employing dedicated path protection [9]. In the latter case, the connection from i to j comprises a primary path from i to j and a second backup path in the opposite direction. Similarly, the connection from j to i comprises a pair of paths in opposite directions. Since this full-duplex connection typically uses the same wavelength resources in both directions [9], the problem can be reduced to that of consuming the full resources in a unidirectional ring. Therefore, without loss of generality, we henceforth restrict our attention to unidirectional rings.

We consider three *profit models*. In the *partial demand–profit model*, we receive profit equal to the bandwidth provided to the request, which may be less than or equal to the actual demand of the request. In the *AON demand–profit model* we receive profit equal to the demand if the entire demand of the request is satisfied and otherwise receive no profit for the request. In the *AON unit-profit model* we receive a single unit of profit if the entire demand can be satisfied and otherwise receive no profit for the request.

For each of the three profit models we first examine the complexity of the offline problem and then investigate the online problem. For the partial demand–profit model we show that the offline problem can be solved optimally in polynomial time. For the online version of the problem we give an algorithm and its competitive analysis. For the AON demand–profit and unit-

Manuscript received April 10, 2006; revised January 15, 2007; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Qiao. This work was supported by the National Science Foundation under Grant CNS-0451293.

K. Benson is with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218 USA.

B. Birnbaum is with the Department of Computer Science and Engineering, University of Washington, Seattle, WA 98105-2350 USA.

E. Molina-Estolano is with the Department of Computer Science, University of California, Santa Cruz, CA 95064 USA.

R. Libeskind-Hadas is with the Department of Computer Science, Harvey Mudd College, Claremont, CA 91711 USA (e-mail: hadas@cs.hmc.edu).

Digital Object Identifier 10.1109/TNET.2007.901065

profit models we show that the offline versions are *NP*-complete and give bounds on competitive analyses for the online versions.

Competitive analysis has been applied to some wavelength and routing problems in WDM networks without traffic grooming. For example, Bartal and Leonardi [10] and Awerbuch *et al.* [11] examine the problem of online routing and wavelength assignment with the objective of minimizing the number of wavelengths used under the assumption that each request requires an entire wavelength. Law and Siu [12] study online routing and wavelength assignment for single-hub rings in which all traffic emanates from a single node in the ring. Tuchscherer [13] examines online call-admission problems for optical networks. Li, Qiao, and Xu [14] use competitive analysis to analyze a number of scheduling algorithms for optical burst switching networks. In earlier related work motivated by ATM networks, Awerbuch *et al.* [15], [16] examine online routing for requests with variable bandwidth demands and variable durations in both arbitrary topologies and trees. These problems differ from WDM problems in that they do not involve wavelengths.

The remainder of this paper is organized as follows: In Section II we give precise formulations of the problems under study. In Sections III–V we give results on offline and online algorithms for the three profit models under consideration. In Section VI we present experimental results, and we conclude in Section VII.

II. Preliminaries and Problem Formulations

In this section, we formally describe the problems studied in this paper. Consider a unidirectional ring with some number of nodes. For simplicity of exposition, we assume that each link between adjacent nodes comprises a single fiber with a specified number of wavelengths, although our results immediately generalize to the case of multiple fibers. Each node has some number of ADMs, with each ADM tuned to a specific wavelength. Each wavelength has some maximum capacity C, but we assume that some wavelengths may have preexisting traffic and thus the wavelengths can have different amounts of available capacity.

A request sequence, denoted $R = [r_1, \ldots, r_n]$, is an ordered set of full-duplex connection requests, each comprising a pair of nodes (i, j) and an integer demand d for bandwidth. A request (i, j) requires a connection from i to j and from j to i. Since the ring is unidirectional, this request is satisfied by selecting some set of wavelengths and allocating a total of d bandwidth (or less, depending on the profit model as described below) on these wavelengths around the entire ring. A wavelength can be only used by connection (i, j) if both nodes i and j have an ADM tuned to that wavelength. We assume that the durations of the requests are effectively infinitely long. A more realistic model could include finite durations for each request, but this problem is clearly very difficult even in the case that there is only one wavelength [15], [16]. We believe that this is an interesting and challenging direction for future research. Finally, we assume that *bifurcation* of traffic is permitted: the bandwidth demand of a request may be split over multiple wavelengths [5].

We consider three distinct profit models. In the *partial demand–profit model*, the full demand of a request need not be satisfied. If a request is for demand d, then any integral portion of this demand p, where $0 \leq p \leq d$, may be satisfied with a resulting profit of p. In the *all-or-nothing (AON) demand–profit model*, a request with demand d results in profit d if the entirety of the request is satisfied and profit 0 otherwise. In the *AON unit-profit model*, a request with demand d results in profit 1 if the entirety of the request is satisfied and profit 0 otherwise.

For each profit model, we begin by studying the computational complexity of the offline problem, in which the entire request sequence is known *a priori*. We then consider the online problem, in which the requests arrive one-by-one. In the online case, when a request arrives we must decide whether or not to satisfy it. If we choose to satisfy the request, we must allocate bandwidth for it on one or more wavelengths. Once established, connections cannot be revoked or altered.

We now formally define the notion of competitiveness of online algorithms [17]. Consider a given profit model and an online algorithm ALG. Let $\mathrm{OPT}(R)$ denote the maximum possible profit that can be obtained by an offline algorithm for request sequence R and let $\mathrm{ALG}(R)$ be the profit obtained by ALG. Algorithm ALG is said to be *c-competitive* if $\mathrm{ALG}(R) \geq c \times \mathrm{OPT}(R)$ for every finite request sequence R. The *competitive ratio* of ALG is defined to be the largest value of c for which ALG is c-competitive.

We distinguish between two types of online algorithms for the problems under consideration. An online algorithm is said to be *fair* if it always satisfies a request when wavelength capacity exists for the request. In the partial demand–profit model, this means that available bandwidth cannot be withheld from a request. In the AON models, this means that if the entirety of a request's demand is available it must be allocated to the request. Thus, a fair online algorithm does not decide which requests to satisfy but rather how to satisfy the given requests. In contrast, an online algorithm is said to be *unfair* if it may decide to withhold resources from a request even if they are available.

Since the set of fair online algorithms is a subset of the set of unfair online algorithms, an unfair algorithm may achieve a better competitive ratio than a fair algorithm[1]. However, for the partial demand–profit model we show that any unfair algorithm has a corresponding fair algorithm that satisfies at least as much demand. For the two AON profit models, unfair algorithms may achieve more profit than unfair algorithms. However, as proved in [13], unfair algorithms for AON profit models cannot achieve "good" competitive ratios.

We now formalize our problems using bipartite graphs and a generalized definition of matchings in such graphs. Let $G = (R \cup W, E)$ be a bipartite graph where the vertices in R represent the requests and the vertices in W represent the wavelengths. An edge (r, w) represents the fact that request r can use wavelength w because the nodes involved in request r have ADMs for wavelength w. Thus, wavelength w can be used to satisfy some or all of the traffic demand of this request. Recall that since the connection is full-duplex and the ring is unidirectional, any allocation of bandwidth for request r on wavelength w is made around the entire ring. Let $\mathbb{Z}^+$ denote the set of positive integers and let $\mathbb{Z}^*$ denote the set of nonnegative integers. The function *demand*: $R \rightarrow \mathbb{Z}^+$ associates a bandwidth demand with each

[1]We note that there is no notion of fairness or unfairness for offline algorithms. An offline algorithm is presented with all of the requests *a priori* and may choose to satisfy only a subset of the requests.

request. The function $\text{capacity} : W \rightarrow \mathbb{Z}^+$ associates a bandwidth capacity with each wavelength. An instance of a dynamic grooming problem is a triplet ($G = (R \cup W, E)$, *demand, capacity*).

Next, we define a generalized matching as an assignment of demand from requests to wavelengths such that each request r receives bandwidth not exceeding its demand and each wavelength w is assigned bandwidth not exceeding its capacity:

Definition 1: A *generalized matching* for ($G = (R \cup W, E)$, *demand, capacity*) is a function $M : E \rightarrow \mathbb{Z}^*$ such that

$$\sum_{\forall w \in W \mid (r,w) \in E} M(r,w) \leq \text{demand}(r), \quad \forall r \in R$$
$$\sum_{\forall r \in R \mid (r,w) \in E} M(r,w) \leq \text{capacity}(w), \quad \forall w \in W.$$

An AON generalized matching is one in which every request has either all or none of its demand satisfied:

Definition 2: An *all-or-nothing* (AON) *generalized matching* for ($G = (R \cup W, E)$, *demand, capacity*) is a generalized matching M with the additional constraint: $\sum_{\forall w \in W \mid (r,w) \in E} M(r,w)$ is equal to either *demand*(r) or $0, \forall r \in R$.

Finally, we formalize the offline problems under study in this paper using generalized matchings. Each offline problem has a corresponding online problem.

Partial Demand–Profit Problem:
Given ($G = (R \cup W, E)$, *demand, capacity*) find a generalized matching M maximizing $\sum_{(r,w) \in E} M(r,w)$.
AON Demand–Profit Problem:
Given ($G = (R \cup W, E)$, *demand, capacity*) find an AON generalized matching M maximizing $\sum_{(r,w) \in E} M(r,w)$.
AON Unit-Profit Problem:
Given ($G = (R \cup W, E)$, *demand, capacity*) find an AON generalized matching M maximizing $\sum_{r \in R} \text{satisfied}(r)$ where *satisfied*: $R \rightarrow \{0, 1\}$ is defined by

$$\text{satisfied}(r) = \begin{cases} 1, & \text{if } \sum_{(r,w) \in E} M(r,w) = \text{demand}(r) \\ 0, & \text{otherwise.} \end{cases}$$

III. Partial Demand–Profit

In this section, we examine the partial demand–profit problem. We show that the offline problem can be solved in polynomial time and then give an online algorithm and its competitive analysis.

A. Offline Problem

We begin by showing that the offline partial demand–profit problem can be solved by an efficient algorithm.

Theorem 1: The offline partial demand–profit problem can be solved in polynomial time.

Proof: The problem is reduced to network flow. Given ($G = (R \cup W, E)$, *demand, capacity*) we construct a corresponding network flow problem as follows: We introduce a source vertex s and a sink vertex t. An edge (s, r) with flow capacity *demand*(r) is introduced for each $r \in R$ and edge (w, t) with flow capacity *capacity*(w) is introduced for each $w \in W$. The edges in E are each given capacity ∞. It is easily seen that a maximum flow in the constructed network gives a generalized matching that maximizes $\sum_{\forall (r,w) \in E} M(r,w)$. The problem can now be solved in polynomial time by any of a number of network flow algorithms [18]. ■

B. Online Problem

Recall that in the online version of the partial demand–profit problem, requests are received one at a time, and the algorithm must commit to an assignment of demand to capacity as each request arrives. A *fair* online algorithm for this problem is one that will satisfy as much demand of a request as the available capacity permits. In general, an unfair online algorithm may generate more profit than a fair algorithm and thus achieve a better competitive ratio. However, the following lemma states that in the partial demand–profit model, for every unfair algorithm there exists a fair algorithm whose competitive ratio is at least as large. Therefore, without loss of generality, we restrict our attention to fair online algorithms for this model.

Lemma 2: Let U be an unfair online algorithm for the partial demand–profit problem. There exists a fair algorithm F whose competitive ratio is at least as large as that of U.

Proof: We construct a fair algorithm from an unfair algorithm U iteratively as follows: We begin by constructing an algorithm F that behaves identically to U unless U chooses to satisfy less demand than possible for a particular request. Let r denote the first such request, let w denote a wavelength on which some demand $d \geq 1$ of request r could have been satisfied but U chose not to satisfy this demand. Then F augments this solution by satisfying demand d of request r on wavelength w. Afterwards, F behaves identically to U. If U uses some residual capacity of wavelength w for subsequent requests, then F emulates this but may fail to satisfy as much as d of this demand due to the allocation of capacity given to the earlier request r. However, F has already obtained d more profit than U for request r and loses at most d profit for future requests allocated by U to wavelength w. Thus, F is at least as profitable as U. Repeating this process ensures that F satisfies as much of the demand of each request as possible and is thus fair. Moreover, since each transformation of the algorithm does not decrease the profit, the resulting fair algorithm is at least as profitable as U. ■

It is interesting to note that an analogous result does not apply for the AON models. In the proof of Lemma 2, we exploited a key property of the partial demand–profit model: Any portion of a satisfied request contributes that amount to the total profit. In the AON models this property does not hold. Indeed, it is straightforward to construct cases where unfair algorithms outperform fair algorithms in the AON models.

In the remainder of this section, we represent a request of demand d as d consecutive requests of unit demand[2]. In the event that all wavelengths have the same capacity, b, the problem reduces to the so-called *b-matching problem* studied by Kalyanasundaram and Pruhs [19]. An algorithm called BALANCE is proposed in [19] that simply assigns each unit of demand to the available wavelength with the most unused capacity, breaking ties arbitrarily. BALANCE is shown to achieve a competitive ratio of $\text{bal}(b) = 1 - \frac{1}{(1+1/b)^b}$, which approaches $1 - \frac{1}{e} \approx 0.632$ as b

[2]This assumption is valid since competitive analysis is defined with respect to the worst-case request sequence. Since traffic bifurcation is permitted in our model, it is easily verified that d consecutive identical unit demand requests are at least as "bad" for an online algorithm as a single request of demand d.

approaches infinity. It is also shown that no deterministic online algorithm can have a competitive ratio better than $\text{bal}(b)$ [19].

Because wavelengths may have preexisting traffic, this paper is concerned with the case of *heterogeneous capacity*, in which the wavelengths can have different amounts of usable capacity. Mehta *et al.* [20] and Buchbinder *et al.* [21] study a generalized version of the heterogeneous matching problem in the context of the Google AdWords market. They provide elegant deterministic and randomized online algorithms but their methods of analysis rely on the capacities being effectively infinite. Therefore, those results are not applicable to the model under consideration in this paper, where wavelengths are assumed to have finite capacities.

In this section, we prove an upper bound on the competitive ratio of any deterministic online algorithm and provide an online algorithm with competitive analysis that generalizes BALANCE in the heterogeneous case.

We begin with some notation that will be used throughout this section. Let $\{b_1, \ldots, b_r\}$ be a set of distinct integers, where $b_1 < b_2 < \cdots < b_r$, representing the distinct initial capacities of the wavelengths, so that each wavelength has a capacity selected from this set. We define the *capacity fraction* of b_i to be the fraction of the total capacity in the network that is contained in wavelengths of capacity b_i. We denote this fraction by q_i. More precisely, if there are k_i wavelengths with capacity $b_i, 1 \leq i \leq r$, then $q_i = \frac{k_i b_i}{\sum_{j=1}^{r} k_j b_j}$. We define the *capacity-weighted average* to be $\sum_{i=1}^{r} q_i bal(b_i)$. Since competitive analysis examines the worst-case behavior of the online algorithm with respect to an optimal offline algorithm, we assume that in every request sequence, the optimal offline algorithm can satisfy every request and that the total demand of all of the requests is equal to the total capacity in the network. The following theorem states that no deterministic online algorithm can have a competitive ratio better than the capacity-weighted average.

Theorem 3: For a given set of initial wavelength capacities $\{b_1, \ldots b_r\}$ with capacity fractions $\{q_1, \ldots, q_r\}$ the competitive ratio of any online algorithm cannot exceed the capacity-weighted average.

Proof: We construct an adversary that forces any deterministic online algorithm to have competitive ratio no better than the capacity-weighted average. The adversary that we construct for the heterogeneous case relies on the adversary in [19] for the homogeneous case.

By definition, each q_i can be expressed as $\frac{x_i}{y}$ where x_i is an integer multiple of b_i and $y = \sum_{i=1}^{r} x_i$. Let $\beta = \prod_{i=1}^{r} b_i(b_i + 1)^{b_i}$. For each $1 \leq i \leq r$, we construct a set of $\frac{\beta x_i}{b_i}$ wavelengths of capacity b_i. Let k_i denote $\frac{\beta x_i}{b_i}$. Note that k_i is an integer since x_i is, by definition an integer multiple of b_i. Note also that the capacity fraction of b_i under this construction is

$$\frac{\frac{\beta x_i}{b_i} b_i}{\sum_{j=1}^{r} \frac{\beta x_j}{b_j} b_j} = \frac{x_i}{\sum_{j=1}^{r} x_j} = q_i$$

and therefore the construction is consistent with the given set of capacity fractions.

The total capacity of all wavelengths is βy and we now construct a request sequence of βy requests of unit demand. The request sequence is partitioned into r groups with sizes $\beta x_i, 1 \leq i \leq r$. Group i makes requests only to wavelengths with capacity b_i. The requests in group i are further subdivided into $\frac{\beta}{b_i(b_i+1)^{b_i}}$ identical blocks of size $b_i(b_i + 1)^{b_i}$. Each block uses the adversarial construction in [19] which permits an optimal offline algorithm to satisfy all of the requests in the block but forces any deterministic online algorithm to satisfy a fraction of at most $bal(b_i)$ of the requests for each block.

Therefore, the competitive ratio of any algorithm is at most

$$\frac{\beta \sum_{i=1}^{r} bal(b_i) x_i}{\beta y} = \sum_{i=1}^{r} q_i bal(b_i).$$

■

We now give an online algorithm and its competitive analysis for the heterogeneous case that generalizes the bounds known for BALANCE. The competitive ratio that we give here is not as good as the capacity-weighted average bound established in Theorem 3. It remains open whether there exists an algorithm whose competitive ratio is equal to the capacity weighted-average.

There are several ways to generalize BALANCE. We explore two versions in this paper: BALANCE-LU, which assigns requests to the wavelength with the least used capacity and BALANCE-MR, which assigns requests to the wavelength with the most remaining capacity. Note that in the homogeneous case, in which all wavelengths have the same capacity, these two algorithms are the same as BALANCE. However, in the heterogeneous case, these algorithms are different. Both algorithms break ties arbitrarily, although we explore some tie-breaking heuristics in Section 6.

In the Appendix, we prove competitive ratios for both BALANCE-LU and BALANCE-MR. We summarize these results in Theorem 4.

Theorem 4: Let the set of wavelength capacities be $\{b_1, \ldots, b_r\}$ with capacity fractions $\{q_1, \ldots, q_r\}$. Define $b_0 = 0$. For $1 \leq j \leq r$, let $\alpha_j = 1 + \frac{1}{b_j}$ and let $\beta_j = \alpha_j^{b_j - b_{j-1}}$. Then the competitive ratio of BALANCE-LU is at least

$$1 - \frac{1 + \sum_{k=1}^{r-1} q_k \left(\frac{b_r}{b_k}\left(1 - \alpha_r^{b_k}\right) + \alpha_r^{b_r} - 1\right)}{\alpha_r^{b_r}} \quad (1)$$

and the competitive ratio of BALANCE-MR is at least

$$1 - \frac{1}{\prod_{k=1}^{r} \beta_k} \times \left(\prod_{k=2}^{r} \beta_k + \sum_{k=2}^{r} q_k \times \left(\sum_{\ell=1}^{k-1} \left(\prod_{m=\ell+1}^{r} \beta_m\right)(\beta_\ell - 1)\left(1 - \frac{b_\ell}{b_k}\right) - \prod_{\ell=2}^{r} \beta_\ell + \prod_{\ell=k+1}^{r} \beta_\ell\right)\right) \quad (2)$$

Proof: See Appendix. ■

It can be easily verified that for the extreme case that all wavelengths have capacity b_r, that is $(q_1, \ldots, q_{r-1}, q_r) = (0, \ldots, 0, 1)$, (1) reduces to $1 - \frac{1}{\alpha_r^{b_r}}$, which matches the result of [19] for a homogenous set of wavelengths with capacity b_r. Similarly, for $(q_1, q_2, \ldots, q_r) = (1, 0, \ldots, 0)$, (2) reduces to $1 - \frac{1}{\alpha_1^{b_1}}$, which matches the result of [19] for a homogenous set of wavelengths with capacity b_1. Thus, both (1) and (2) generalize the result of [19].

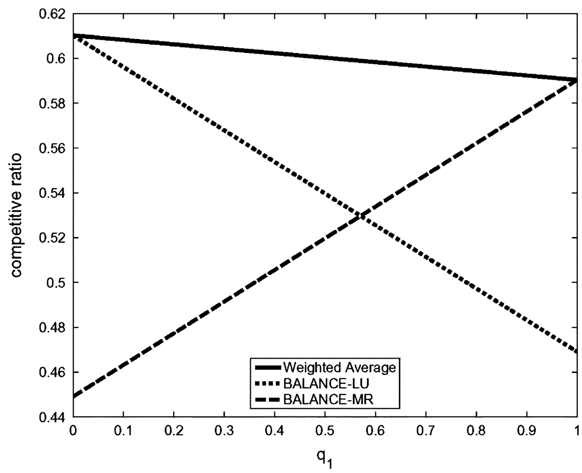


Fig. 1. Competitive ratios for BALANCE-LU and BALANCE-MR when $b_1 = 4$ and $b_2 = 8$ as a function of q_1.

As a concrete example, Fig. 1 plots the competitive ratios that Theorem 4 provides for $b_1 = 4$ and $b_2 = 8$ as a function of q_1. It also plots the capacity-weighted average as a function of q_1, which by Theorem 3 is an upper bound on how good an online algorithm can be. The plot looks qualitatively similar for any two values of b_1 and b_2.

From Fig. 1 it is apparent that for small values of q_1, our lower bound on the competitive ratio of BALANCE-LU is higher, and for large values of q_1, our lower bound on the competitive ratio of BALANCE-MR is higher. This suggests algorithm BALANCE-HYBRID, which takes advantage of the fact that in our application the values of the q_i are known *a priori*. BALANCE-HYBRID uses (1) and (2) to obtain lower bounds on the competitive ratios for each algorithm for a particular vector of q_i values. BALANCE-HYBRID then proceeds as either BALANCE-LU or BALANCE-MR, depending on which gives the higher competitive ratio. In Fig. 1, the performance of BALANCE-HYBRID is the upper envelope of the performance of BALANCE-LU and BALANCE-MR.

Note that in Fig. 1, there are points on the curves for both BALANCE-LU and BALANCE-MR in which the competitive ratio is below what it would be if the wavelengths had only capacity b_1 or only capacity b_2. Although (1) and (2) are not known to be tight for all capacity fractions q_i, it is easily verified that for both BALANCE-LU and BALANCE-MR there exist instances of the problem for a given $\{b_1, b_2\}$ in which the performance of the online algorithm is indeed lower than the minimum of $\mathrm{bal}(b_1)$ and $\mathrm{bal}(b_2)$.

Because the analysis of Theorem 4 is not known to be tight (except at the endpoints), an interesting direction for future research is to try to improve the bounds of Theorem 4 or prove tightness. It would also be interesting to investigate other algorithms to see if competitive ratios closer to the capacity-weighted average can be obtained. We conjecture that better algorithms than BALANCE-LU and BALANCE-MR exist, but finding such an algorithm is a challenging open problem. Theorem 4 provides an algorithm, BALANCE-HYBRID, with the best known performance guarantees for the partial demand–profit model.

IV. AON DEMAND–PROFIT

Recall that in the AON demand–profit model, a connection must have either all or none of its demand satisfied and a satisfied request provides a profit equal to its demand. In this section, we first prove that the offline version of this problem is *NP*-complete. We then provide lower bounds on the competitive ratio of any fair online algorithm.

A. The Offline Problem

To show NP-completeness, we first state the decision version of the AON demand–profit problem. The AON DEMAND–PROFIT DECISION PROBLEM (AONDP) is defined as follows: Given ($G = (R \cup W, E)$, *demand, capacity*) and an integer $K \in \mathbb{Z}^+$, determine if there exists an AON generalized matching M such that $\sum_{(r,w)\in E} M(r,w) \geq K$. We reduce from SUBSET SUM, which is defined as follows: Given a finite set $A = \{a_1, \ldots, a_n\}$, size $s(a_i) \in \mathbb{Z}^+$ for each $a_i \in A$, and a positive integer B, determine if there exists a subset $A' \subseteq A$ such that $\sum_{a_i \in A'} s(a_i) = B$. The SUBSET SUM problem is known to be NP-complete [22].

Theorem 5: AONDP is NP-complete.

Proof: Clearly AONDP is in the class NP since a certificate consisting of an AON generalized matching can be easily verified in polynomial time. We show that AONDP is *NP*-hard by a reduction from SUBSET SUM.

Consider an instance of SUBSET SUM defined by $A = \{a_1, \ldots, a_n\}, s : A \rightarrow \mathbb{Z}^+, B \in \mathbb{Z}^+$. We construct an instance of AONDP in which $R = \{r_1, \ldots, r_n\}$, there is only one wavelength, denoted w, and $E = \{(r_i, w) \mid r_i \in R\}$. For each $r_i \in R$, *demand*$(r_i) = s(a_i)$. Finally, we let *capacity*$(w) = B$ and let $K = B$. Clearly, this reduction can be performed in time polynomial in the size of the SUBSET SUM problem instance.

We now show that the answer to the given instance of SUBSET SUM is "yes" if and only if the answer to the constructed instance of AONDP is "yes." Suppose that the answer to the given instance of SUBSET SUM is "yes." Then an AON generalized matching M can be constructed such that $M(r_i, w) = \mathrm{demand}(r_i)$ if $a_i \in A'$ and $M(r_i, w) = 0$ otherwise. The profit is equal to $\sum_{r_i \in R} M(r_i, w) = \sum_{a_i \in A'} s(a_i) = B = K$, and thus the answer to the constructed instance of AONDP is also "yes."

Conversely, suppose that the answer to the constructed instance of AONDP is "yes." Then $\sum_{r_i \in R} M(r_i, w) \geq K = B$. By the definition of a generalized matching, $\sum_{r_i \in R} M(r_i, w) \leq \mathrm{capacity}(w) = K = B$. Therefore, $\sum_{r_i \in R} M(r_i, w) = B$. Because M is an AON generalized matching, either $M(r_i, w) = \mathrm{demand}(r_i) = s(a_i)$ or $M(r_i, w) = 0$. Let R' be the set of requests such that $M(r_i, w) = s(a_i)$ and let $A' = \{a_i \mid r_i \in R'\}$. Then $\sum_{a_i \in A'} s(a_i) = \sum_{r_i \in R'} M(r_i, w) = B$, and the answer to SUBSET SUM is also "yes." ■

B. Online Problem

We now establish a lower bound on the competitiveness of any fair online algorithm, under the assumption that the maximum request demand is less than or equal to the initial minimum wavelength capacity. Although this assumption reduces the generality of the result, in practice the wavelength capacities are frequently much larger than the request demands, in some cases by two or more orders of magnitude [23]. Thus, the residual capacity prior to the invocation of the online algorithm will in many cases satisfy this assumption.

Theorem 6: Given $(G=(R\cup W,E)$, *demand, capacity*), let $r_{\max}=\max_{r\in R}\mathrm{demand}(r)$, $w_{\max}=\max_{w\in W}\mathrm{capacity}(w)$, and $w_{\min}=\min_{w\in W}\mathrm{capacity}(w)$. If $r_{\max}\leq w_{\min}$, then the competitive ratio of any fair online algorithm is at least

$$\frac{w_{\min}-r_{\max}+1}{w_{\max}+w_{\min}-r_{\max}+1}.$$

Proof: Let $O\subseteq R$ be the set of requests satisfied in some optimal offline solution and let M_O be an optimal AON matching for all requests in O. Let $F\subseteq R$ be the set of requests satisfied by a fair online algorithm and let M_F be the AON matching for all requests in F obtained by the fair online algorithm. If $R'\subseteq R$ is a subset of the requests, define $N(R')$, the *neighborhood* of R', to be the set of vertices in W adjacent to vertices in R'. Define $f: W\rightarrow\mathbb{Z}^*$ to be the total bandwidth assigned to wavelength w by the fair online algorithm. That is, $f(w)=\sum_{(r,w)\in E}M_F(r,w)$. Let $\mathrm{OPT}=\sum_{r\in O}\mathrm{demand}(r)$ be the profit obtained by M_O and $\mathrm{FAIR}=\sum_{r\in F}\mathrm{demand}(r)$ be the profit obtained by M_F. Note that $\mathrm{FAIR}=\sum_{w\in W}f(w)$.

Consider the set of requests $O-F$ and its neighborhood $N(O-F)$. For each wavelength $w\in N(O-F)$, we must have that $f(w)\geq w_{\min}-r_{\max}+1$. If this were not true, at most $w_{\min}-r_{\max}$ bandwidth was assigned to w by the online algorithm. Since every request has demand at most $r_{\max}$ and w has capacity at least $w_{\min}$, a request in $O-F$ could have been satisfied by the online algorithm, contradicting the assumption that the algorithm is fair. Therefore

$$(w_{\min}-r_{\max}+1)|N(O-F)|\leq\sum_{w\in N(O-F)}f(w)\leq\mathrm{FAIR}. \quad (3)$$

Because the optimal solution assigns each request in $O-F$ to wavelengths in $N(O-F)$,

$$\sum_{r\in O-F}\mathrm{demand}(r)\leq w_{\max}|N(O-F)|. \quad (4)$$

Combining (3) and (4) yields

$$\mathrm{FAIR}\geq\frac{w_{\min}-r_{\max}+1}{w_{\max}}\sum_{r\in O-F}\mathrm{demand}(r).$$

Since $O-F=O-(O\cap F)$, we rewrite $\sum_{r\in O-F}\mathrm{demand}(r)$ as $\mathrm{OPT}-\sum_{r\in O\cap F}\mathrm{demand}(r)$ and thus

$$\begin{aligned}\mathrm{FAIR}&\geq\frac{w_{\min}-r_{\max}+1}{w_{\max}}\left(\mathrm{OPT}-\sum_{r\in O\cap F}\mathrm{demand}(r)\right)\\&\geq\frac{w_{\min}-r_{\max}+1}{w_{\max}}(\mathrm{OPT}-\mathrm{FAIR})\end{aligned}$$

which implies our competitive ratio bound:

$$\frac{\mathrm{FAIR}}{\mathrm{OPT}}\geq\frac{w_{\min}-r_{\max}+1}{w_{\max}+w_{\min}-r_{\max}+1}. \qquad \blacksquare$$

V. AON Unit-Profit

Recall that in the AON unit-profit model, a connection must have either all or none of its demand satisfied and a satisfied request provides a profit of 1. In this section, we prove that the offline version of this problem is *NP*-complete and give nearly tight bounds on the performance of any fair online algorithm.

A. Offline Problem

To show NP-completeness, we first state the decision version of the AON unit-profit problem. The AON Unit-Profit Decision Problem (AONUP) is defined as follows: Given $(G=(R\cup W,E)$, *demand, capacity*) and an integer $K\in\mathbb{Z}^+$ determine if there exists an AON generalized matching M such that $\sum_{r\in R}\mathrm{satisfied}(r)\geq K$, where $\mathrm{satisfied}: R\rightarrow\{0,1\}$ is defined by

$$\mathrm{satisfied}(r)=\begin{cases}1, & \text{if } \sum_{(r,w)\in E}M(r,w)=\mathrm{demand}(r)\\ 0, & \text{otherwise.}\end{cases}$$

We reduce from 3SAT [22], which is defined as follows: Given a set of boolean variables V and a set of clauses C where each clause is the disjunction of exactly three literals over the variables in V, determine if there exists a valuation of the variables such that each clause is satisfied.

Theorem 7: AONUP is *NP*-complete.

Proof: Clearly AONUP is in the class *NP* since a certificate consisting of an AON generalized matching can be easily verified in polynomial time. We show that AONUP is *NP*-hard by a reduction from 3SAT. For a given instance of 3SAT, let V and C denote the sets of variables and clauses, respectively. For a literal ℓ over V, let $\mathrm{complement}(\ell)$ denote the number of occurrences of the negation of ℓ in the 3SAT instance. We construct an instance of AONUP as follows: For each clause $c\in C$, we introduce a corresponding wavelength c in W with $\mathrm{capacity}(c)=1$. For each literal $\ell_i^c\in c, 1\leq i\leq 3$, we introduce a corresponding request $\ell_i^c\in R$ with $\mathrm{demand}(\ell_i^c)=\mathrm{complement}(\ell_i^c)+1$. In addition, we introduce edges $(\ell_i^c,c)\in E, 1\leq i\leq 3$. We refer to these three requests, three edges, and wavelength as the "clause gadget" for c. Next, for every pair of requests $r_1,r_2\in R$ corresponding to complementary literals, we introduce a new wavelength w with $\mathrm{capacity}(w)=1$ and introduce edges $(r_1,w),(r_2,w)\in E$. Finally, we let $K=|C|$. Clearly, this reduction takes time polynomial in the size of the 3SAT instance.

Now we show that the answer to the constructed instance of AONUP is "yes" if and only if the given instance of 3SAT is satisfiable. Suppose that we are given a satisfying assignment for the 3SAT is instance. For each clause c, we select a true literal $\ell_i^c\in c, 1\leq i\leq 3$. The corresponding request ℓ_i^c in the clause gadget for c has $\mathrm{demand}(\ell_i^c)=\mathrm{complement}(\ell_i^c)+1$ and is incident upon wavelength c with capacity 1 and $\mathrm{complement}(\ell_i^c)$ additional wavelengths with capacity 1. We construct an AON generalized matching using all of these edges and thus satisfying request ℓ_i^c. Since the satisfying assignment for the 3SAT instance does not assign a variable and its negation the same value, no two selected requests in this AON generalized matching share a wavelength and thus $|C|=K$ requests are satisfied.

Conversely, assume we have an AON generalized matching that satisfies $|C|=K$ requests. By construction, a request ℓ_i^c corresponding to literal $i, 1\leq i\leq 3$, in clause c has $\mathrm{demand}(\ell_i^c)=\mathrm{complement}(\ell_i^c)+1$ and is adjacent to exactly $\mathrm{complement}(\ell_i^c)+1$ wavelengths with capacity 1. Moreover, two requests in the same clause gadget share a common wavelength. Therefore, at most one request per clause gadget can be satisfied. The AON generalized matching must, therefore, satisfy exactly one request from each clause and cannot satisfy any two requests corresponding to complementary literals since such requests are also adjacent to a common wavelength. Thus, this matching corresponds to a valuation satisfying the 3SAT instance. ∎

B. Online Problem

In this section, we provide upper and lower bounds on the competitive ratios of any fair online algorithm for the AON unit-profit problem. The assumption required for Theorem 6, namely that the maximum demand of all requests is less than or equal to the minimum initial capacity of all wavelengths, is no longer required here.

Theorem 8: Let $r_{\max} = \max_{r \in R} \operatorname{demand}(r)$. Every fair online algorithm for the AON unit-profit problem achieves a competitive ratio of $\frac{1}{r_{\max}+1}$. Moreover, no fair online algorithm can achieve a competitive ratio better than $\frac{1}{r_{\max}}$.

Proof: First, we give an adversarial argument to show that no fair online algorithm may guarantee a competitive ratio greater than $1/r_{\max}$. The adversary creates a single wavelength of capacity $kr_{\max}$ where k is some positive integer. The adversary first makes k requests, each with demand $r_{\max}$ followed by $kr_{\max}$ requests with demand 1. Any fair online algorithm must satisfy the first k requests at which point the wavelength is saturated and none of the subsequent requests can be satisfied, resulting in a profit of k. An optimal offline algorithm would not satisfy any of the first k requests and would then satisfy the remaining $kr_{\max}$ requests for a profit of $kr_{\max}$. This establishes an upper bound of $\frac{1}{r_{\max}}$ for the competitive ratio of any fair online algorithm.

Next, we establish the lower bound on the competitive ratio of any online algorithm as follows: Let F be the set of requests satisfied by an arbitrary fair online algorithm and let M_F denote the corresponding AON matching. Let O be the set of requests satisfied in some optimal solution and let M_O be a corresponding AON matching. We now describe a procedure to transform M_F to M_O, and bound how many new requests are satisfied after the procedure. First, all requests in $F - O$ are removed and the generalized matching M_F is updated by increasing the residual capacity of the wavelengths by the demands of these requests. Let $N(F - O)$ denote the set of wavelengths used by the requests in a set $F - O$ and let δ denote the total demand of these requests, which is also the total increase in the residual capacity upon the removal of these requests.

Next, for each request in $r \in O - F$ and $w \in N(F - O)$ such that $(r, w) \in E$, we increase the value of generalized matching $M_F(r, w)$ to the largest value less than or equal to $M_O(r, w)$ that does not violate the capacity constraint on wavelength w. Let M'_F denote this matching. Since the online algorithm was unable to satisfy the requests in $O - F$, any request in $O - F$ that is now satisfied must use at least one unit of the δ capacity introduced in the previous step. Therefore, this step increases the profit by at most $\delta \leq |F - O| r_{\max}$.

We now transform M'_F into M_O by re-matching the requests in $F \cap O$ to be identical to M_O and, finally, making any changes to the matching for the requests in $O - F$ to result in matching M_O. The number of requests in $O - F$ that were previously unsatisfied but became satisfied in this step cannot exceed $|F \cap O| r_{\max}$ since only the requests in $F \cap O$ were rematched and each such request uses capacity at most $r_{\max}$. Therefore, the competitive ratio is

$$\frac{|F|}{|O|} \geq \frac{|F|}{|F| + |F - O| r_{\max} + |F \cap O| r_{\max}} = \frac{|F|}{|F|(1 + r_{\max})} = \frac{1}{r_{\max} + 1}.$$

VI. Experimental Results

In this section, we describe experimental results for all three models. For the partial demand–profit model, we implemented both the BALANCE-MR and the BALANCE-LU online algorithms. (Although our BALANCE-HYBRID algorithm uses both of these algorithms, we examined BALANCE-MR and BALANCE-LU individually in the experiments.)

In our first set of experiments, wavelengths had capacities selected at random uniformly between 4 and 8. In the second set of experiments, wavelengths had capacities ranging uniformly between 4 and 32. The total capacity of all wavelengths was set to be as close too 1000 as possible.

After the wavelengths and their capacities were established, requests were generated using the following technique: Each generated request had demand uniformly selected between 1 and 4. Each time a request of demand d was created, d unused units of capacity in the wavelengths were chosen randomly, and edges were created from the request to each of the wavelengths containing these units of capacity. The units of capacity were then marked as used. Requests were generated until the total demand was at least as large as the total capacity, at which point the demand of the last request was set so that the total demand was exactly equal to the total capacity. In this way, the optimal solution to the problem instance was known by construction.

At this point, each problem instance had requests with a total demand equal to the total capacity of the wavelengths and a set of edges that ensured a solution that satisfied the total demand. Now, for each request and wavelength that did not already have an edge between them, we added an edge with probability given by a parameter ρ, which we varied form 0.01 to 1. Finally, the requests were permuted at random to construct a request sequence for the online algorithm. Note that as ρ goes to 1, there is an edge between every request and every wavelength, making the problem trivially solvable by greedy assignment. Thus, any fair algorithm is optimal for $\rho = 1$. The problems instances are evidently more difficult when ρ is relatively small.

For each problem instance, we computed the profit found by each candidate algorithm and divided this by the optimal profit. We then computed the average of these ratios over 1000 random problem instances.

Fig. 2(a) shows the results for the partial demand–profit model. The top graph shows the average ratio obtained for the case that wavelength capacities ranged from 4 to 8 and the bottom graph shows the averages when the wavelength capacities ranged from 4 to 32. The averages are plotted as a function of the value of the extra edge probability parameter ρ. Note that BALANCE-LU does not stipulate how ties should be broken when there are two possible wavelengths that tie

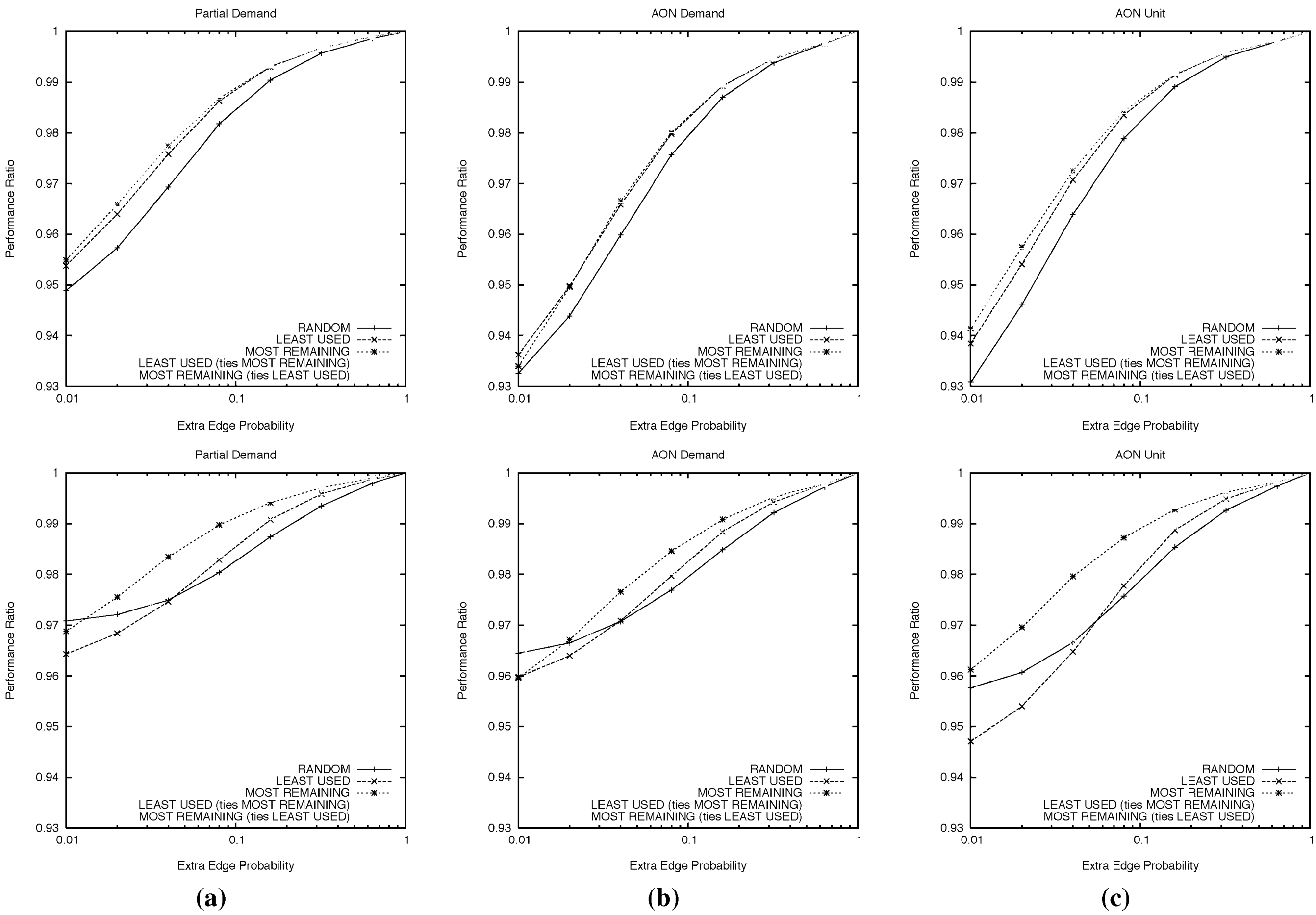


Fig. 2. Ratios for algorithms for three profit models: (a) partial demand–profit model, (b) AON demand profit model, (c) AON unit profit model. The top row is for the case of wavelength capacities selected at uniform between 4 and 8. The bottom row is for wavelength capacities between 4 and 32.

for the least capacity used but have different total capacities. The BALANCE-LU algorithm breaks such ties arbitrarily. By introducing a heuristic tie-breaking rule, it may be possible to obtain better results. For this reason, we implemented a variant of BALANCE-LU which breaks ties by choosing a wavelength with the most capacity remaining. Similarly, we implemented a variant of BALANCE-MR which breaks ties by choosing a wavelength with the least capacity used.

Our algorithms were also compared to a simple fair randomized algorithm, RANDOM, similar to one analyzed by Karp *et al.* [24]. RANDOM considers each unit of demand separately. For each unit, it chooses a random wavelength w from the set of adjacent wavelengths that have at least one unit of available capacity and then assigns the unit to w. It does this until either the entire request is matched, or there are no adjacent wavelengths with available capacity.

Next, we experimented with fair online algorithms for the AON demand–profit [Fig. 2(b)] and unit-profit [Fig. 2(c)] models, generating request sequences in a similar fashion to that described above for the partial demand–profit model. We implemented algorithms similar to BALANCE-MR and BALANCE-LU. The "most remaining" policy works as follows: When a request of demand d arrives, the demand is partitioned into unit pieces. Each unit is assigned to the available wavelength with most capacity remaining. This process is repeated until the entire request is satisfied or it is determined that there does not exist enough capacity to satisfy the request, in which case the demand for this request is retracted and the request is not satisfied. The "least used" policy is analogous. Both of these policies are fair and thus their competitive ratios are governed by the analyses in Sections IV and V.

It is notable that the "most remaining" policy breaking ties with "least used" was generally the best algorithm in all three profit models. These results also show that the actual ratios obtained by the algorithms under study generally perform much better than the worst-case bounds obtained by competitive analysis. This phenomena has been widely observed for many online problems [17]. The competitive analysis is important in establishing theoretical guarantees and algorithms that satisfy these guarantees often behave much better in practice.

VII. CONCLUSION

In this paper, we have studied an online dynamic traffic grooming problem in unidirectional WDM rings. We have considered three distinct profit models. For the partial demand–profit model, we proposed the BALANCE-HYBRID algorithm and provided a competitive analysis. For the two AON models, we gave bounds on the competitive ratios of fair algorithms. We have also evaluated a number of algorithms experimentally and demonstrated that their performance is generally much better than the worst-case analysis provided by our theoretical competitive analyses.

There are numerous interesting problems for future research. First, we conjecture that there exists an online algorithm for the partial demand–profit model with a better competitive ratio than BALANCE-HYBRID. We believe that some of the techniques used in the competitive analysis of BALANCE-HYBRID are quite general and may be useful in deriving a more competitive algorithm. In this study we have assumed that all requests have effectively infinite duration. Extending these results to apply to connection requests of finite duration is an important problem. We have presented two different AON profit models. It is possible that a single general AON model could contain both of these models as special cases and that results could be obtained for this model. Finally, extending this work to the case of bidirectional rings without path protection, and more generally to arbitrary topologies, is an interesting and challenging area for future work.

APPENDIX
PROOFS

In this appendix, we prove the competitive ratio bounds of Theorem 4. We use slightly different terminology in the Appendix to be consistent with previous work on BALANCE [19], [20]. We describe each wavelength as a *server site* comprised of a number of *servers*, which correspond to units of capacity in our original formulation. Furthermore, we define an *s-server site* to be a server site (wavelength) having s servers (units of capacity).

We restate the theorem here as two lemmas, one stating the competitive ratio bound for BALANCE-LU and the other stating the competitive ratio bound for BALANCE-MR. Suppose that the sizes of the server sites (corresponding to the capacities of the wavelengths) are chosen from $B = \{b_1, \ldots, b_r\}$, where $b_1 < \cdots < b_r$. For $1 \le j \le r$, let q_j be the fraction of servers at server sites with capacity b_j. Define $b_0 = 0$. Finally, for $1 \le j \le r$, let $\alpha_j = 1 + \frac{1}{b_j}$, and let $\beta_j = \alpha_j^{b_j - b_{j-1}}$.

Lemma 9: The competitive ratio of BALANCE-LU is at least

$$1 - \frac{1 + \sum_{k=1}^{r-1} q_k \left(\frac{b_r}{b_k} \left(1 - \alpha_r^{b_k}\right) + \alpha_r^{b_r} - 1 \right)}{\alpha_r^{b_r}}.$$

Lemma 10: The competitive ratio of BALANCE-MR is at least

$$1 - \frac{1}{\prod_{k=1}^{r} \beta_k} \times \left(\prod_{k=2}^{r} \beta_k + \sum_{k=2}^{r} q_k \times \left(\sum_{\ell=1}^{k-1} \left(\prod_{m=\ell+1}^{r} \beta_m \right) (\beta_\ell - 1) \left(1 - \frac{b_\ell}{b_k}\right) - \prod_{\ell=2}^{r} \beta_\ell + \prod_{\ell=k+1}^{r} \beta_\ell \right) \right).$$

Least Used: To prove Lemma 9, we define the following the notation. For $1 \le i \le b_r$ and $1 \le j \le r$, let $R_i^{b_j}$ be the set of requests assigned by BALANCE-LU to the ith used server of a b_j-server site. Note that by definition $R_i^{b_j} = \emptyset$ for $i > b_j$. Let $S_i^{b_j}$ be the set of b_j-server sites with *at least* i servers used by BALANCE-LU when the algorithm is complete. Again, $S_i^{b_j} = \emptyset$ for $i > b_j$. Let $R_i = \bigcup_{j=1}^{r} R_i^{b_j}$ and $S_i = \bigcup_{j=1}^{r} S_i^{b_j}$. Define R_{b_r+1} to be the set of requests satisfied by the optimal offline algorithm but not by BALANCE-LU. Finally, for $1 \le i \le b_r + 1$, let $X_i = \bigcup_{l=i}^{b_r+1} R_l$. Since we assume that an optimal solution can assign every request to a server, the ratio of the performance of BALANCE-LU to that of the optimal offline algorithm is

$$\frac{|X_1| - |X_{b_r+1}|}{|X_1|} = 1 - \frac{|X_{b_r+1}|}{|X_1|}.$$

We begin the proof of Lemma 9 with some preliminary lemmas derived from the definitions provided above.

Lemma 11: For $1 \le j \le r, |R_1^{b_j}| \le \frac{q_j}{b_j} |X_1|$.

Proof: The number of b_j-server sites is $\frac{q_j}{b_j}|X_1|$, and for each b_j-server site there can be at most one request that BALANCE-LU satisfied first at that site. ■

Lemma 12: For $1 \le j \le r$ and $1 \le i \le b_j - 1, |R_i^{b_j}| \ge |R_{i+1}^{b_j}|$.

Proof: Every request in $R_{i+1}^{b_j}$ must have been routed to a different server site by BALANCE-LU. Each b_j-server site with a request routed to it from $R_{i+1}^{b_j}$ must also have a request routed to it from $R_i^{b_j}$. Thus for each request in $R_{i+1}^{b_j}$, we can associate a unique request in $R_i^{b_j}$, and the lemma follows. ■

The following lemma uses the behavior of BALANCE-LU to bound $|X_i|$ based on $|X_{i+1}|$. Eventually, we will use this bound inductively to bound $|X_1|$ by $|X_{b_r+1}|$.

Lemma 13: For $0 \le j \le r-1$ and $b_j + 1 \le i \le b_{j+1}$

$$|X_i| \ge \alpha_r |X_{i+1}| + \sum_{k=j+1}^{r-1} \left(1 - \frac{b_k}{b_r}\right) \left| R_i^{b_k} \right| - \sum_{k=1}^{j} \frac{b_k}{b_r} \left| R_{b_k}^{b_k} \right|.$$

Proof: Consider a request x in X_{i+1}. Either BALANCE-LU assigned x to a server site or it did not. Suppose first that BALANCE-LU assigned x to server site y. By definition, y had at least i servers used when BALANCE-LU assigned x to it. Consider any server site s that has an edge from x. If $s = y$, then clearly $s \in S_i$. If $s \ne y$, then BALANCE-LU preferred y over s. By the behavior of BALANCE-LU, either s had at least as many servers used as y did when x was assigned or s did not have any unused servers when x was assigned. Thus $s \in (\bigcup_{k=1}^{j} S_{b_k}^{b_k}) \bigcup S_i$.

Now suppose that BALANCE-LU did not assign x to any server site. Then, by definition, $x \in X_{b_r+1}$, which implies that $i = b_r$ and $j = r-1$. Consider any server site s that has an edge from x. Since BALANCE-LU always assigns a request if possible, s cannot have any unused servers. Thus $s \in \bigcup_{k=1}^{r} S_{b_k}^{b_k} = (\bigcup_{k=1}^{r-1} S_{b_k}^{b_k}) \bigcup S_{b_r}^{b_r} = (\bigcup_{k=1}^{r-1} S_{b_k}^{b_k}) \bigcup S_{b_r} = (\bigcup_{k=1}^{j} S_{b_k}^{b_k}) \bigcup S_i$.

Let $D = (\bigcup_{k=1}^{j} S_{b_k}^{b_k}) \bigcup S_i$. We have just shown that D contains every server site adjacent to a request in X_{i+1}. In an optimal solution, each request in X_{i+1} must be matched to a distinct server in D. Therefore, the number of servers in D must be at least as large as $|X_{i+1}|$. Finally, note that for each server site $t \in S_i^{b_k}$, there must be a unique request in $R_i^{b_k}$ that BALANCE-LU assigned to t. Thus, for $1 \le k \le r$ and $1 \le i \le b_k, |R_i^{b_k}| \ge |S_i^{b_k}|$. Putting this together, we have that

$$|X_{i+1}| \le \sum_{k=1}^{j} b_k \left| S_{b_k}^{b_k} \right| + \sum_{k=j+1}^{r} b_k \left| S_i^{b_k} \right| \quad (5)$$

$$\le \sum_{k=1}^{j} b_k \left| R_{b_k}^{b_k} \right| + \sum_{k=j+1}^{r} b_k \left| R_i^{b_k} \right|. \quad (6)$$

By the definitions of R_i and X_i, we have that for $b_j+1 \le i \le b_{j+1}$,

$$\left|R_i^{b_r}\right| = |R_i| - \sum_{k=j+1}^{r-1} \left|R_i^{b_k}\right| = |X_i| - |X_{i+1}| - \sum_{k=j+1}^{r-1} \left|R_i^{b_k}\right| \quad (7)$$

Combining inequality (6) with (7) yields

$$|X_{i+1}| \le \sum_{k=1}^{j} b_k \left|R_{b_k}^{b_k}\right| + b_r \left(|X_i| - |X_{i+1}| - \sum_{k=j+1}^{r-1} \left|R_i^{b_k}\right|\right) + \sum_{k=j+1}^{r-1} b_k \left|R_i^{b_k}\right| \quad (8)$$

which can be rearranged to yield the statement of the lemma. ■

In the proof of Lemma 15, we will need the following technical lemma.

Lemma 14: For integers i, b_j, b_k, b_r such that $1 \le i \le b_j \le b_k < b_r$

$$(b_r - b_k)\left(\alpha_r^{b_k - b_j + i - 1} - 1\right) - \alpha_r^{b_k - b_j + i - 1} b_k \left(\alpha_r^{b_r - b_k} - 1\right) < 0$$

where $\alpha_r = 1 + (1/b_r)$.

Proof: Let $a = b_j - i + 1, b = b_k$, and $c = b_r$. Note that $1 \le a \le b < c$. By rearranging terms and using the new notation, we can rewrite the expression as

$$c\left(1 + \frac{1}{c}\right)^{b-a} - b\left(1 + \frac{1}{c}\right)^{c-a} + b - c$$

By the binomial theorem

$$c\left(1 + \frac{1}{c}\right)^{b-a} - b\left(1 + \frac{1}{c}\right)^{c-a} + b - c \quad (9)$$

$$= c\sum_{n=0}^{b-a} \binom{b-a}{n} \left(\frac{1}{c}\right)^n - b\sum_{n=0}^{c-a} \binom{c-a}{n} \left(\frac{1}{c}\right)^n \quad (10)$$

$$+ b - c \quad (11)$$

$$= c\sum_{n=1}^{b-a} \binom{b-a}{n} \left(\frac{1}{c}\right)^n - b\sum_{n=1}^{c-a} \binom{c-a}{n} \left(\frac{1}{c}\right)^n \quad (12)$$

$$= \sum_{n=1}^{b-a} \left(c\binom{b-a}{n} - b\binom{c-a}{n}\right) \left(\frac{1}{c}\right)^n - b\sum_{n=b-a+1}^{c-a} \binom{c-a}{n} \left(\frac{1}{c}\right)^n \quad (13)$$

$$< \sum_{n=1}^{b-a} \left(c\binom{b-a}{n} - b\binom{c-a}{n}\right) \left(\frac{1}{c}\right)^n. \quad (14)$$

For $1 \le n \le b-a$ and $1 \le a \le b < c$, it can be shown by a simple combinatorial argument or induction on n that

$$c\binom{b-a}{n} - b\binom{c-a}{n} < 0$$

which implies the statement of the lemma. ■

Lemma 15 applies Lemma 13 inductively to yield a bound on $|X_1|$ based on $|X_{b_r+1}|$, which can be applied to prove Lemma 9.

Lemma 15: For $0 \le j \le r$

$$|X_{b_j+1}| \ge \alpha_r^{b_r - b_j} |X_{b_r+1}| + \sum_{k=j+1}^{r-1} \left((b_r - b_k)\left(\alpha_r^{b_k - b_j} - 1\right) - \alpha_r^{b_k - b_j} b_k \left(\alpha_r^{b_r - b_k} - 1\right)\right) \left|R_{b_j+1}^{b_k}\right| - \sum_{k=1}^{j} b_k \left(\alpha_r^{b_r - b_j} - 1\right) \left|R_{b_k}^{b_k}\right|.$$

Proof: By induction on $r - j$. When $r - j = 0$, this inequality reduces to $|X_{b_r+1}| \ge |X_{b_r+1}|$, which is trivially true. Now, assuming that it is true for $r - (j+1)$, we prove that it is true for $r - j$. By Lemma 13

$$|X_{b_{j+1}}| \ge \alpha_r |X_{b_{j+1}+1}| + \sum_{k=j+1}^{r-1} \left(1 - \frac{b_k}{b_r}\right) \left|R_{b_{j+1}}^{b_k}\right| - \sum_{k=1}^{j} \frac{b_k}{b_r} \left|R_{b_k}^{b_k}\right|. \quad (15)$$

Combining (15) with the inductive hypothesis yields

$$|X_{b_{j+1}}| \ge \alpha_r^{b_r - b_{j+1} + 1} |X_{b_r+1}| + \alpha_r \sum_{k=j+2}^{r-1} \left((b_r - b_k)\left(\alpha_r^{b_k - b_{j+1}} - 1\right) - \alpha_r^{b_k - b_{j+1}} b_k \left(\alpha_r^{b_r - b_k} - 1\right)\right) \left|R_{b_{j+1}+1}^{b_k}\right| - \alpha_r \sum_{k=1}^{j+1} b_k \left(\alpha_r^{b_r - b_{j+1}} - 1\right) \left|R_{b_k}^{b_k}\right| + \sum_{k=j+1}^{r-1} \left(1 - \frac{b_k}{b_r}\right) \left|R_{b_{j+1}}^{b_k}\right| - \sum_{k=1}^{j} \frac{b_k}{b_r} \left|R_{b_k}^{b_k}\right|. \quad (16)$$

By Lemma 14, the quantity inside the first summation of (16) is negative, so we can apply Lemma 12 to replace $|R_{b_{j+1}+1}^{b_k}|$ by $|R_{b_{j+1}}^{b_k}|$ in (16). We do this and pull the $k = j+1$ terms out of the following summations:

$$|X_{b_{j+1}}| \ge \alpha_r^{b_r - b_{j+1} + 1} |X_{b_r+1}| + \alpha_r \sum_{k=j+2}^{r-1} \left((b_r - b_k)\left(\alpha_r^{b_k - b_{j+1}} - 1\right) - \alpha_r^{b_k - b_{j+1}} b_k \left(\alpha_r^{b_r - b_k} - 1\right)\right) \left|R_{b_{j+1}}^{b_k}\right| - \alpha_r b_{j+1} \left(\alpha_r^{b_r - b_{j+1}} - 1\right) \left|R_{b_{j+1}}^{b_{j+1}}\right| - \alpha_r \sum_{k=1}^{j} b_k \left(\alpha_r^{b_r - b_{j+1}} - 1\right) \left|R_{b_k}^{b_k}\right| + \left(1 - \frac{b_{j+1}}{b_r}\right) \left|R_{b_{j+1}}^{b_{j+1}}\right| + \sum_{k=j+2}^{r-1} \left(1 - \frac{b_k}{b_r}\right) \left|R_{b_{j+1}}^{b_k}\right| - \sum_{k=1}^{j} \frac{b_k}{b_r} \left|R_{b_k}^{b_k}\right|. \quad (17)$$

Noting that $\alpha_r - 1 = \frac{1}{b_r}$, this inequality can be rearranged to yield

$$\begin{aligned}|X_{b_{j+1}}| &\geq \alpha_r^{b_r-b_{j+1}+1}|X_{b_r+1}| + \sum_{k=j+2}^{r-1} \left((b_r-b_k)\left(\alpha_r^{b_k-b_{j+1}+1}-1\right)\right.\\ &\quad \left.-\alpha_r^{b_k-b_{j+1}+1}b_k\left(\alpha_r^{b_r-b_k}-1\right)\right)\left|R_{b_{j+1}}^{b_k}\right|\\ &\quad + \left((\alpha_r-1)(b_r-b_{j+1}) - \alpha_r b_{j+1}\left(\alpha_r^{b_r-b_{j+1}}-1\right)\right)\left|R_{b_{j+1}}^{b_{j+1}}\right|\\ &\quad - \sum_{k=1}^{j} b_k\left(\alpha_r^{b_r-b_{j+1}+1}-1\right)\left|R_{b_k}^{b_k}\right| \qquad (18)\\ &= \alpha_r^{b_r-b_{j+1}+1}|X_{b_r+1}| + \sum_{k=j+1}^{r-1}\left((b_r-b_k)\left(\alpha_r^{b_k-b_{j+1}+1}-1\right)\right.\\ &\quad \left.-\alpha_r^{b_k-b_{j+1}+1}b_k\left(\alpha_r^{b_r-b_k}-1\right)\right)\left|R_{b_{j+1}}^{b_k}\right|\\ &\quad - \sum_{k=1}^{j} b_k\left(\alpha_r^{b_r-b_{j+1}+1}-1\right)\left|R_{b_k}^{b_k}\right|. \qquad (19)\end{aligned}$$

We now prove by induction on i that for $1 \leq i \leq b_{j+1} - b_j$

$$\begin{aligned}|X_{b_{j+1}+1-i}| &\geq \alpha_r^{b_r-b_{j+1}+i}|X_{b_r+1}| + \sum_{k=j+1}^{r-1}\left((b_r-b_k)\left(\alpha_r^{b_k-b_{j+1}+i}-1\right)\right.\\ &\quad \left.-\alpha_r^{b_k-b_{j+1}+i}b_k\left(\alpha_r^{b_r-b_k}-1\right)\right)\left|R_{b_{j+1}+1-i}^{b_k}\right|\\ &\quad - \sum_{k=1}^{j} b_k\left(\alpha_r^{b_r-b_{j+1}+i}-1\right)\left|R_{b_k}^{b_k}\right|. \qquad (20)\end{aligned}$$

We have just proven the base case, and plugging in $i = b_{j+1} - b_j$ yields the statement of this lemma, so it remains to prove the inductive case. Suppose that this statement is true for $i-1$. Then combining Lemma 13 with the inductive hypothesis yields

$$\begin{aligned}&|X_{b_{j+1}+1-i}| \geq\\ &\alpha_r^{b_r-b_{j+1}+i}|X_{b_r+1}| + \alpha_r\sum_{k=j+1}^{r-1}\left((b_r-b_k)\left(\alpha_r^{b_k-b_{j+1}+i-1}-1\right)\right.\\ &\left.-\alpha_r^{b_k-b_{j+1}+i-1}b_k\left(\alpha_r^{b_r-b_k}-1\right)\right)\left|R_{b_{j+1}+2-i}^{b_k}\right|\\ &- \alpha_r\sum_{k=1}^{j} b_k\left(\alpha_r^{b_r-b_{j+1}+i-1}-1\right)\left|R_{b_k}^{b_k}\right|\\ &+ \sum_{k=j+1}^{r-1}\left(1-\frac{b_k}{b_r}\right)\left|R_{b_{j+1}+1-i}^{b_k}\right| - \sum_{k=1}^{j}\frac{b_k}{b_r}\left|R_{b_k}^{b_k}\right|. \qquad (21)\end{aligned}$$

By Lemma 14, the quantity inside the first sum is negative, so by Lemma 12, we can replace $|R_{b_{j+1}+2-i}^{b_k}|$ with $|R_{b_{j+1}+1-i}^{b_k}|$ in (21). Doing so, and rearranging terms, yields (20), concluding the proof of the lemma. ■

From Lemma 15, we derive Lemma 9 as follows. For $j = 0$, Lemma 15 states (with some rearrangement)

$$\begin{aligned}&|X_1|\\ &\geq \alpha_r^{b_r}|X_{b_r+1}| + \sum_{k=1}^{r-1}\left(\left(b_r\left(\alpha_r^{b_k}-1\right) + b_k\left(1-\alpha_r^{b_r}\right)\right)\left|R_1^{b_k}\right|\right. \qquad (22)\end{aligned}$$

By an argument very similar to the argument used to prove Lemma 14, it can be shown that the quantity inside the summation in (22) is negative. Thus, we can apply Lemma 11 to replace $|R_1^{b_k}|$ by $\frac{q_k}{b_k}|X_1|$ in (22). Doing so, and rearranging terms, yields

$$\frac{|X_{b_r+1}|}{|X_1|} \leq \frac{1+\sum_{k=1}^{r-1} q_k\left(\frac{b_r}{b_k}\left(1-\alpha_r^{b_k}\right)+\alpha_r^{b_r}-1\right)}{\alpha_r^{b_r}}. \qquad (23)$$

Therefore,

$$\frac{|X_1|-|X_{b_r+1}|}{|X_1|} \geq 1 - \frac{1+\sum_{k=1}^{r-1} q_k\left(\frac{b_r}{b_k}\left(1-\alpha_r^{b_k}\right)+\alpha_r^{b_r}-1\right)}{\alpha_r^{b_r}} \qquad (24)$$

and the proof of Lemma 9 is complete.

A. *Most Remaining*

We now prove Lemma 10. For $1 \leq j \leq r$ and $1 \leq i \leq b_r$, let $P_i^{b_j}$ be the set of requests assigned by BALANCE-MR to a b_j-server site that had exactly i unused servers before the requests were assigned. Note that by definition $P_i^{b_j} = \emptyset$ for $i > b_j$. For $1 \leq j \leq r$ and $0 \leq i \leq b_r$, let $Q_i^{b_j}$ be the set of b_j-server sites that have *at most* i unused servers after BALANCE-MR has processed all of the requests. Note that $Q_i^{b_j} = Q_{b_j}^{b_j}$ for $i \geq b_j$. Let $P_i = \bigcup_{j=1}^{r} P_i^{b_j}$ and $Q_i = \bigcup_{j=1}^{r} Q_i^{b_j}$. Define P_0 to be the set of requests satisfied by the optimal offline algorithm but not by BALANCE-MR. Finally, for $0 \leq i \leq b_r$, let $Y_i = \bigcup_{l=0}^{i} P_l$. With this notation, the ratio of the performance of BALANCE-MR to that of the optimal offline algorithm is

$$\frac{|Y_{b_r}| - |Y_0|}{|Y_{b_r}|} = 1 - \frac{|Y_0|}{|Y_{b_r}|}$$

Lemma 16 follows from the definitions provided above.

Lemma 16: For $1 \leq j \leq r$ and $1 \leq i \leq b_j, |P_i^{b_j}| \leq (q_j)/(b_j)|Y_{b_r}|$.

Proof: The number of b_j-server sites is $(q_j)/(b_j)|Y_{b_r}|$, and for all i, no server site can have more than one request assigned to it that was satisfied when there were exactly i unused servers. ■

The following lemma is derived from the behavior of BALANCE-MR and forms the basis of the proof.

Lemma 17: For $1 \leq j \leq r$ and $b_{j-1} + 1 \leq i \leq b_j$,

$$\begin{aligned}|Y_i| \geq{}& \alpha_j|Y_{i-1}| - \sum_{k=j+1}^{r}\left(\frac{b_k}{b_j}-1\right)\frac{q_k}{b_k}|Y_{b_r}|\\ &- \frac{1}{b_j}\left(1-\sum_{k=j}^{r} q_k\right)|Y_{b_r}|\end{aligned}.$$

Proof: Consider a request $y \in Y_{i-1}$. Either BALANCE-MR assigned y to a server site or it did not. First, suppose that BALANCE-MR assigned y to server site q. By the definition of Y_{i-1}, q had no more than $i-1$ unused servers before y was routed. Therefore, y can only have edges to server sites in Q_{i-1} If not, then when BALANCE-MR assigned y to q, it could have assigned y to a server site with more than $i-1$ unused servers, which contradicts the fact that BALANCE-MR chooses the server site with the most remaining servers.

Now suppose that BALANCE-MR did not assign y to any server site. Then, by definition, $y \in Y_0$, which implies that $i = 1$ and $j = 1$. Since BALANCE-MR always assigns a request if possible, y can only have edges to server sites with zero unused servers. Thus, y can only have edges to server sites in $Q_0 = Q_{i-1}$.

In either case, requests in Y_{i-1} can only have edges to server sites in Q_{i-1}. In an optimal solution, then, each request in Y_{i-1} must be routed to a distinct server in Q_{i-1}. Therefore, the number of servers in Q_{i-1} must be at least as large as the number of requests in Y_{i-1}. Recall that $Q_{i-1} = \bigcup_{k=1}^{r} Q_{i-1}^{b_k}$, and for $b_{j-1}+1 \le i \le b_j, \bigcup_{k=1}^{j-1} Q_{i-1}^{b_k} = \bigcup_{k=1}^{j-1} Q_{b_k}^{b_k}$. Therefore, $Q_{i-1} = (\bigcup_{k=1}^{j-1} Q_{b_k}^{b_k}) \cup (\bigcup_{k=j}^{r} Q_{i-1}^{b_k})$, and the number of servers in Q_{i-1} is $\sum_{k=1}^{j-1} b_k |Q_{b_k}^{b_k}| + \sum_{k=j}^{r} b_k |Q_{i-1}^{b_k}|$.

Thus we have that

$$|Y_{i-1}| \le \sum_{k=1}^{j-1} b_k \left|Q_{b_k}^{b_k}\right| + \sum_{k=j}^{r} b_k \left|Q_{i-1}^{b_k}\right|. \quad (25)$$

For $j \le k \le r$, there must be a unique request in $P_i^{b_k}$ for each server site in $Q_{i-1}^{b_k}$. Thus for $j \le k \le r$

$$\left|Q_{i-1}^{b_k}\right| \le \left|P_i^{b_k}\right| \quad (26)$$

The total number of servers at b_k-server sites is $b_k |Q_{b_k}^{b_k}|$. Thus, by definition, $b_k |Q_{b_k}^{b_k}| = q_k |Y_{b_r}|$. Using this fact and inequality (26), we write inequality (25) as

$$\begin{aligned}|Y_{i-1}| &\le \sum_{k=1}^{j-1} q_k |Y_{b_r}| + \sum_{k=j}^{r} b_k \left|P_i^{b_k}\right| \\ &= \left(1 - \sum_{k=j}^{r} q_k\right) |Y_{b_r}| + \sum_{k=j}^{r} b_k \left|P_i^{b_k}\right|. \quad (27)\end{aligned}$$

By definition, $|P_i^{b_j}| = |P_i| - \sum_{k=1}^{j-1} |P_i^{b_k}| - \sum_{k=j+1}^{r} |P_i^{b_k}|$. But for $b_{j-1}+1 \le i \le b_j, \sum_{k=1}^{j-1} |P_i^{b_k}| = 0$. Thus $|P_i^{b_j}| = |P_i| - \sum_{k=j+1}^{r} |P_i^{b_k}| = |Y_i| - |Y_{i-1}| - \sum_{k=j+1}^{r} |P_i^{b_k}|$. Therefore, we can rewrite (27) as

$$|Y_{i-1}| \le \left(1 - \sum_{k=j}^{r} q_k\right) |Y_{b_r}| + b_j \left(|Y_i| - |Y_{i-1}| - \sum_{k=j+1}^{r} |P_i^{b_k}|\right) + \sum_{k=j+1}^{r} b_k |P_i^{b_k}|. \quad (28)$$

This can be rearranged to yield

$$|Y_i| \ge \alpha_j |Y_{i-1}| - \sum_{k=j+1}^{r} \left(\frac{b_k}{b_j} - 1\right) \left|P_i^{b_k}\right| - \frac{1}{b_j} \left(1 - \sum_{k=j}^{r} q_k\right) |Y_{b_r}|. \quad (29)$$

Applying Lemma 16 allows us to substitute $(q_k)/(b_k)|Y_{b_r}|$ for $|P_i^{b_k}|$ in the above inequality to yield the statement of the lemma. ■

The following lemma applies Lemma 17 inductively to provide a bound on $|Y_{b_j}|$ based on $|Y_{b_{j-1}}|$. Recall that for $1 \le j \le r$, we have defined β_j to be equal to $\alpha_j^{b_j - b_{j-1}}$.

Lemma 18: For $1 \le j \le r$,

$$|Y_{b_j}| \ge \beta_j |Y_{b_{j-1}}| - (\beta_j - 1) \sum_{k=j+1}^{r} \left(1 - \frac{b_j}{b_k}\right) q_k |Y_{b_r}| - (\beta_j - 1) \left(1 - \sum_{k=j}^{r} q_k\right) |Y_{b_r}|.$$

Proof: We prove that for $0 \le i \le b_j - b_{j-1}$,

$$|Y_{b_{j-1}+i}| \ge \alpha_j^i |Y_{b_{j-1}}| - \left(\alpha_j^i - 1\right) \sum_{k=j+1}^{r} \left(1 - \frac{b_j}{b_k}\right) q_k |Y_{b_r}| - \left(\alpha_j^i - 1\right) \left(1 - \sum_{k=j}^{r} q_k\right) |Y_{b_r}| \quad (30)$$

from which the statement of the lemma follows by substituting $b_j - b_{j-1}$ for i. The proof proceeds by induction on i. For $i = 0$, inequality (30) is trivially true. Assuming that inequality (30) is true for $i - 1$, we show that it is true for i. By Lemma 17

$$|Y_{b_{j-1}+i}| \ge \alpha_j |Y_{b_{j-1}+i-1}| - \sum_{k=j+1}^{r} \left(\frac{b_k}{b_j} - 1\right) \frac{q_k}{b_k} |Y_{b_r}| - \frac{1}{b_j} \left(1 - \sum_{k=j}^{r} q_k\right) |Y_{b_r}|. \quad (31)$$

By the inductive hypothesis

$$\begin{aligned}&|Y_{b_{j-1}+i}| \ge \\ &\alpha_j \left(\alpha_j^{i-1} |Y_{b_{j-1}}| - \left(\alpha_j^{i-1} - 1\right) \sum_{k=j+1}^{r} \left(1 - \frac{b_j}{b_k}\right) q_k |Y_{b_r}| - \left(\alpha_j^{i-1} - 1\right) \left(1 - \sum_{k=j}^{r} q_k\right) |Y_{b_r}|\right) \\ &- \sum_{k=j+1}^{r} \left(\frac{b_k}{b_j} - 1\right) \frac{q_k}{b_k} |Y_{b_r}| - \frac{1}{b_j} \left(1 - \sum_{k=j}^{r} q_k\right) |Y_{b_r}|. \quad (32)\end{aligned}$$

By noting that $\frac{1}{b_j} = \alpha_j - 1$, inequality (32) can be rearranged to yield (30), thus proving the lemma. ■

Finally, the next lemma applies Lemma 18 inductively to provide a bound on $|Y_{b_j}|$ based on $|Y_0|$. As will be shown, Lemma 10 follows easily from the result of this lemma.

Lemma 19: For $1 \le j \le r$

$$\begin{aligned}|Y_{b_j}| &\ge \left(\prod_{k=1}^{j} \beta_k\right) |Y_0| \sum_{k=2}^{r} \left(-\sum_{l=1}^{\min(j,k-1)} \left(\prod_{m=l+1}^{j} \beta_m\right) (\beta_l - 1) \times \left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{j} \beta_l - \prod_{l=k+1}^{j} \beta_l\right) q_k |Y_{b_r}| \\ &\quad - \left(\prod_{k=2}^{j} \beta_k - 1\right) |Y_{b_r}|.\end{aligned}$$

Proof: The proof proceeds by induction on j. For $j = 1$, the statement of the lemma becomes

$$|Y_{b_1}| \ge \beta_1 |Y_0| - \sum_{k=2}^{r} (\beta_1 - 1) \left(1 - \frac{b_1}{b_k}\right) q_k |Y_{b_r}|. \quad (33)$$

This is true because it is the statement of Lemma 18 for $j = 1$ (since $\sum_{k=1}^{r} q_k = 1$). Now, assuming the lemma is true for $j-1$, we prove that it is true for j. Combining Lemma 18 with the inductive hypothesis yields

$$\begin{aligned}
&|Y_{b_j}| \\
&\geq \beta_j \Bigg(\Bigg(\prod_{k=1}^{j-1} \beta_k \Bigg) |Y_0| + \sum_{k=2}^{r} \Bigg(- \sum_{l=1}^{\min(j-1,k-1)} \Bigg(\prod_{m=l+1}^{j-1} \beta_m \Bigg) \\
&\times (\beta_l - 1) \left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{j-1} \beta_l - \prod_{l=k+1}^{j-1} \beta_l \Bigg) \\
&\times q_k |Y_{b_r}| - \Bigg(\prod_{k=2}^{j-1} \beta_k - 1 \Bigg) |Y_{b_r}| \Bigg) \\
&\quad - (\beta_j - 1) \sum_{k=j+1}^{r} \left(1 - \frac{b_j}{b_k}\right) q_k |Y_{b_r}| \\
&- (\beta_j - 1) \Bigg(1 - \sum_{k=j}^{r} q_k \Bigg) |Y_{b_r}|. \qquad (34)
\end{aligned}$$

We distribute the β_j term and rearrange the sums to group terms by q_k:

$$\begin{aligned}
|Y_{b_j}| \geq &\Bigg(\prod_{k=1}^{j} \beta_k \Bigg) |Y_0| \\
&+ \sum_{k=2}^{r} \Bigg(- \sum_{l=1}^{\min(j-1,k-1)} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) (\beta_l - 1) \\
&\times \left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{j} \beta_l - \beta_j \prod_{l=k+1}^{j-1} \beta_l \Bigg) \\
&\times q_k |Y_{b_r}| - \Bigg(\prod_{k=2}^{j} \beta_k - \beta_j \Bigg) |Y_{b_r}| \\
&+ \sum_{k=j+1}^{r} \left(-(\beta_j - 1) \left(1 - \frac{b_j}{b_k}\right) (\beta_j - 1) \right) \\
&+ q_k |Y_{b_r}| + (\beta_j - 1) q_j |Y_{b_r}| - (\beta_j - 1) |Y_{b_r}|. \qquad (35)
\end{aligned}$$

Separating the first sum into the ranges $2 \leq k \leq j-1, k = j$, and $j+1 \leq k \leq r$, we get

$$\begin{aligned}
&|Y_{b_j}| \geq \\
&\Bigg(\prod_{k=1}^{j} \beta_k \Bigg) |Y_0| + \sum_{k=2}^{j-1} \Bigg(- \sum_{l=1}^{k-1} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) (\beta_l - 1) \\
&\times \left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{j} \beta_l - \prod_{l=k+1}^{j} \beta_l \Bigg) q_k |Y_{b_r}| \qquad (36) \\
&+ \Bigg(- \sum_{l=1}^{j-1} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) (\beta_l - 1) \left(1 - \frac{b_l}{b_k}\right) \\
&+ \prod_{l=2}^{j} \beta_l - \beta_j \Bigg) q_j |Y_{b_r}| \\
&+ \sum_{k=j+1}^{r} \Bigg(- \sum_{l=1}^{j-1} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) (\beta_l - 1) \\
&\times \left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{j} \beta_l - \beta_j \Bigg) q_k |Y_{b_r}| \\
&- \Bigg(\prod_{k=2}^{j} \beta_k - \beta_j \Bigg) |Y_{b_r}| \\
&+ \sum_{k=j+1}^{r} \left(-(\beta_j - 1) \left(1 - \frac{b_j}{b_k}\right) + (\beta_j - 1) \right) \\
&\times q_k |Y_{b_r}| + (\beta_j - 1) q_j |Y_{b_r}| - (\beta_j - 1) |Y_{b_r}|. \qquad (37)
\end{aligned}$$

We can now combine the q_k terms to get

$$\begin{aligned}
|Y_{b_j}| \geq &\Bigg(\prod_{k=1}^{j} \beta_k \Bigg) |Y_0| + \sum_{k=2}^{j-1} \Bigg(- \sum_{l=1}^{k-1} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) \\
&\times (\beta_l - 1) \left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{j} \beta_l \\
&- \prod_{l=k+1}^{j} \beta_l \Bigg) q_k |Y_{b_r}| + \Bigg(- \sum_{l=1}^{j-1} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) \\
&\times (\beta_l - 1) \left(1 - \frac{b_l}{b_j}\right) + \prod_{l=2}^{j} \beta_l - 1 \Bigg) q_j |Y_{b_r}| \\
&+ \sum_{k=j+1}^{r} \Bigg(- \sum_{l=1}^{j-1} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) (\beta_l - 1) \\
&\times \left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{j} \beta_l - 1 - (\beta_j - 1) \\
&\times \left(1 - \frac{b_j}{b_k}\right) \Bigg) q_k |Y_{b_r}| - \Bigg(\prod_{k=2}^{j} \beta_k - 1 \Bigg) |Y_{b_r}| \qquad (38) \\
= &\Bigg(\prod_{k=1}^{j} \beta_k \Bigg) |Y_0| + \sum_{k=2}^{j-1} \Bigg(- \sum_{l=1}^{k-1} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) \\
&\times (\beta_l - 1) \left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{j} \beta_l - \prod_{l=k+1}^{j} \beta_l \Bigg) q_k |Y_{b_r}| \\
&+ \Bigg(- \sum_{l=1}^{j-1} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) (\beta_l - 1) \left(1 - \frac{b_l}{b_j}\right) \\
&+ \prod_{l=2}^{j} \beta_l - \prod_{l=j+1}^{j} \beta_l \Bigg) q_j |Y_{b_r}| \\
&+ \sum_{k=j+1}^{r} \Bigg(- \sum_{l=1}^{j} \Bigg(\prod_{m=l+1}^{j} \beta_m \Bigg) (\beta_l - 1) \left(1 - \frac{b_l}{b_k}\right) \\
&+ \prod_{l=2}^{j} \beta_l - \prod_{l=k+1}^{j} \beta_l \Bigg) q_k |Y_{b_r}| - \Bigg(\prod_{k=2}^{j} \beta_k - 1 \Bigg) |Y_{b_r}|. \\
&\qquad (39)
\end{aligned}$$

The outermost sums in inequality (39) can now be recombined to form the statement of the lemma. ■

We are now ready to prove Lemma 10. By Lemma 19

$$|Y_{b_r}| \geq \Bigg(\prod_{k=1}^{r} \beta_k \Bigg) |Y_0| + \sum_{k=2}^{r} \Bigg(- \sum_{l=1}^{k-1} \Bigg(\prod_{m=l+1}^{r} \beta_m \Bigg)$$

$$\times(\beta_l - 1)\left(1 - \frac{b_l}{b_k}\right) + \prod_{l=2}^{r}\beta_l - \prod_{l=k+1}^{r}\beta_l\Bigg)$$
$$\times\, q_k|Y_{b_r}| - \left(\prod_{k=2}^{r}\beta_k - 1\right)|Y_{b_r}|.$$

Therefore, $\frac{|Y_0|}{|Y_{b_r}|}$ is less than or equal to

$$\frac{1}{\prod_{k=1}^{r}\beta_k}\left(\prod_{k=2}^{r}\beta_k + \sum_{k=2}^{r} q_k\right.$$
$$\times\left(\sum_{l=1}^{k-1}\left(\prod_{m=l+1}^{r}\beta_m\right)(\beta_l - 1)\left(1 - \frac{b_l}{b_k}\right)\right.$$
$$\left.\left. - \prod_{l=2}^{r}\beta_l + \prod_{l=k+1}^{r}\beta_l\right)\right).$$

which implies Lemma 10, recalling that the ratio of the performance of BALANCE-MR to that of the optimal offline algorithm is $1 - \frac{|Y_0|}{|Y_{b_r}|}$.

ACKNOWLEDGMENT

The authors thank Prof. S. Irani for conversations related to the competitive analyses of the online algorithms presented, Prof. K. Pruhs and A. Mehta for important references and suggestions, and the four anonymous reviewers who provided many suggestions that helped to significantly improve this paper.

REFERENCES

[1] A. L. Chiu and E. H. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *J. Lightw. Technol.*, vol. 18, no. 1, pp. 2–12, Jan. 2000.

[2] R. Dutta and G. N. Rouskas, "On optimal traffic grooming in WDM rings," in *Proc. ACM SIGMETRICS*, 2001, pp. 164–174.

[3] J.-Q. Hu, "Traffic grooming in wavelength-division-multiplexing ring networks: A linear programming solution," *J. Opt. Netw.*, vol. 1, no. 11, pp. 397–408, Oct. 2002.

[4] C. Zhao and J. Q. Hu, "Traffic grooming for WDM rings with dynamic traffic," manuscript, 2003.

[5] R. Dutta, S. Huang, and G. N. Rouskas, "Traffic grooming in path, star, and tree networks: Complexity bounds and algorithms," in *Proc. ACM SIGMETRICS 2003*, Jun. 2003, pp. 298–299.

[6] F. Farahmand, X. Huang, and J. P. Jue, "Efficient online traffic grooming algorithms in WDM mesh networks with drop-and-continue node architecture," in *Proc. 1st Annu. Int. Conf. Broadband Networks (BROADNETS 2004)*, Oct. 2004, pp. 180–189.

[7] S. Huang and R. Dutta, "Research problems in dynamic traffic grooming in optical networks," in *Proc. 1st Int. Workshop on Traffic Grooming*, , San Jose, CA, Oct. 2004.

[8] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "Dynamic traffic grooming in WDM mesh networks using a novel graph model," *Opt. Netw. Mag.*, vol. 4, no. 3, pp. 65–75, May/Jun. 2003.

[9] G. Maier, A. Pattavina, S. D. Patre, and M. Martinelli, "Optical network survivability: Protection techniques in the WDM layer," *Photon. Netw. Commun.*, vol. 4, no. 3/4, pp. 251–269, 2002.

[10] Y. Bartal and S. Leonardi, "On-line routing in all-optical networks," *Theoretical Comput. Sci.* vol. 221, no. 1–2, pp. 19–39, 1999 [Online]. Available: citeseer.ist.psu.edu/bartal97line.html

[11] B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosen, "On-line competitive algorithms for call admission in optical networks," *Algorithmica* vol. 31, no. 1, pp. 29–43, 2001 [Online]. Available: citeseer.ist.psu.edu/article/awerbuch96line.html

[12] C. Law and K.-Y. Siu, "On-line routing and wavelength assignment in single-hub WDM rings," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 2111–2122, Oct. 2000.

[13] A. Tuchscherer, "Dynamical configuration of transparent optical telecommunication networks," in *Oper. Res. Proc.*, 2004, pp. 25–32.

[14] J. Li, C. Qiao, and D. Xu, "Maximizing throughput for optical burst switching networks," in *Proc. IEEE INFOCOM*, 2004, pp. 1853–1863.

[15] B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosen, "Competitive non-preemptive call control," in *SODA: ACM-SIAM Symp. Discrete Algorithms (A Conf. on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1994 [Online]. Available: citeseer.ist.psu.edu/376929.html

[16] B. Awerbuch, Y. Azar, and S. A. Plotkin, "Throughput-competitive on-line routing," in *IEEE Symp. Found. Comput. Sci.*, 1993, pp. 32–40 [Online]. Available: citeseer.ist.psu.edu/article/awerbuch93throughputcompetitive.html

[17] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.

[19] B. Kalyanasundaram and K. Pruhs, "An optimal deterministic algorithm for online b-matching," *Theoretical Comput. Sci.*, vol. 233, no. 1, pp. 319–325, Feb. 2000.

[20] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, "AdWords and generalized on-line matching," in *Proc. IEEE Symp. Foundations of Computer Science*, Oct. 2005.

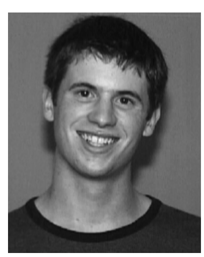
[21] N. Buchbinder, K. Jain, and J. Naor, "Online primal-dual algorithms for maximizing ad-auction revenue," manuscript, 2006.

[22] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 2000.

[23] S. Ramasubramanian and A. Somani, "Analysis of optical networks with heterogeneous grooming architectures," *IEEE/ACM Trans. Networking*, vol. 12, no. 5, pp. 931–943, Oct. 2004.

[24] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, "An optimal algorithm for on-line bipartite matching," in *Proc. 22nd ACM Symp. Theory of Computing (STOC)*, Baltimore, MD, May 1990, pp. 352–358.

Karyn Benson received the B.A. degree in computer science from Wellesley College, Wellesley, MA, and is currently doing graduate work in information security at The Johns Hopkins University, Baltimore, MD.



Benjamin Birnbaum received the B.S. degree in computer science from Washington University in St. Louis and is currently working toward the doctorate degree in computer science at the University of Washington in Seattle, where he is an NSF graduate fellow.

His interests are in algorithm design and analysis.

Esteban Molina-Estolano received the B.S. degree in computer science from Harvey Mudd College, Claremont, CA, and is currently working toward the doctorate degree in computer science at the University of California, Santa Cruz.

Ran Libeskind-Hadas (M'93) received the A.B. degree in applied mathematics from Harvard University, Cambridge, MA, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign.

He currently holds the Joseph B. Platt Endowed Chair in Computer Science at Harvey Mudd College, Claremont, CA. His interests are in optical networking and algorithm design and analysis.