Scheduling Hybrid WDM/TDM Passive Optical Networks With Nonzero Laser Tuning Time

Jingjing Zhang, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—Owing to the high bandwidth provisioning, hybrid wavelength division multiplexing/time division multiplexing (WDM/TDM) passive optical network (PON) is becoming an attractive future-proof access network solution. In hybrid WDM/TDM PON, tunable lasers are potential candidate light sources attributed to their multiwavelength provisioning capability and color-free property. Currently, the laser tuning time ranges from a few tens of nanoseconds to seconds, or even minutes, depending on the adopted technology. Different laser tuning time may introduce different network performance. To achieve small packet delay and ensure fairness, the schedule length for given optical network unit (ONU) requests is desired to be as short as possible. This paper illustrates contributions in four main aspects. First, we show that both preemptive and nonpreemptive scheduling problems with the objective of minimizing the schedule length are NP-hard when the laser tuning time is nonzero. Second, we present a heuristic preemptive scheduling algorithm with an approximation factor of at most 2 and a heuristic nonpreemptive scheduling algorithm with an approximation factor of at most $2-1/m$, where m is the number of wavelengths. Third, extensive simulations have been conducted, and simulation results show that our proposed algorithms, which consider laser tuning time, achieve significantly better performances as compared to algorithms that are directly derived from existing algorithms without considering laser tuning time. Fourth, since the scheduling in one cycle is related to that in the last cycle, we provide some discussions on the scheduling in multiple cycles.**

***Index Terms*—Dynamic bandwidth allocation, laser tuning time, multiprocessor scheduling, passive optical network (PON), wavelength division multiplexing (WDM).**

I. Introduction

Hybrid wavelength division multiplexing/time division multiplexing (WDM/TDM) passive optical network (PON) is becoming a promising solution for next-generation broadband optical access [1], [2]. Instead of using only one wavelength to provision bandwidth in upstream and downstream as TDM PON does, hybrid WDM/TDM PON increases the number of working wavelengths in each stream to exploit the high bandwidth of optical fibers. On the other hand, hybrid WDM/TDM PON bridges the gap between TDM PON and pure WDM PON and can be deployed by smoothly migrating from the currently deployed TDM PON [3]–[5].

Manuscript received February 08, 2010; revised August 26, 2010 and October 29, 2010; accepted November 04, 2010; approved by IEEE/ACM Transactions on Networking Editor M. Reisslein. Date of publication December 03, 2010; date of current version August 17, 2011.

The authors are with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: jz58@njit.edu; nirwan.ansari@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2010.2093150

In hybrid WDM/TDM PONs, an important optical device is the optical laser used for generating optical signals with multiple wavelengths. Depending on the wavelength generation capability, there are three major classes of lasers available for use—namely multiwavelength lasers, wavelength-specified lasers, and wavelength-tunable lasers [6]. A multiwavelength laser is able to generate multiple WDM wavelengths simultaneously, including multifrequency laser, gain-coupled distributed feedback laser diode (DFB LD) array, and chirped-pulse WDM. Multiwavelength lasers are usually used at the optical line terminal (OLT) to generate downstream traffic or seed optical network units (ONUs) with optical signals for their upstream data transmission [7], [8]. Instead of generating multiple wavelengths, a wavelength-specified laser, e.g., the common DFB, can only emit one specific wavelength. Wavelength-specified sources have been extensively used in broadband PON (BPON), Ethernet PON (EPON) [9], gigabit PON (GPON) [10], and next-generation access stage 1 (NGA1) [11]. However, with wavelength-specified lasers, no statistical gain can be exploited among ONUs that can support different wavelengths. Wavelength-tunable lasers are able to generate multiple wavelengths, but only one wavelength at a time [12], [13]. As compared to wavelength-fixed lasers, tunable lasers possess advantages in two major aspects. First, from the perspective of the MAC layer, the wavelength tunability of tunable lasers facilitates the statistical multiplexing of traffic from all ONUs, thus potentially yielding better system performance [14], [15]. Second, from the perspective of network operators, tunable lasers enable the color-free property of ONUs, which further facilitates the simplified inventory management, reduced sparing cost, and automated wavelength provisioning [4]. Owing to these advantages, wavelength-tunable lasers are promising light source generators for hybrid WDM/TDM PONs.

One typical example of hybrid WDM/TDM PON architectures employing tunable lasers is SUCCESS [3]. SUCCESS equips the OLT with tunable lasers to generate downstream data traffic and provides ONUs with optical continuous wave (CW) bursts for their upstream data transmission. The wavelength tunability of tunable lasers was exploited to provision high bandwidth and realize a smooth migration path from current TDM PONs to WDM PONs. Bock *et al.* [16] also proposed an architecture of using tunable lasers at the OLT. In addition to equipping the OLT with tunable lasers, the network equips ONUs with tunable lasers as well. Das *et al.* [17] proposed to equip each ONU with a tunable laser to facilitate a fully flexible dynamic bandwidth allocation in the upstream direction. In [18], we also described an architecture that equips ONUs with tunable lasers. With the focus on the laser tuning range, which is an important parameter of tunable lasers, we investigated the impact of the laser tuning range on the network capacity and designed WDM PONs by selecting lasers with proper tuning ranges to minimize the capital investment of the PON. In [14], we theoretically analyzed capacities of hybrid WDM/TDM PONs.

Currently, tunable lasers can be manufactured by various technologies such as mechanical, acoustooptic, or electrooptic tunability. Despite the many kinds of options, tunable lasers are still costly, thus inhibiting their wide deployment in networks. The important cost elements of tunable lasers include "nonoptical" specifications such as package dimension, output power, power variation over wavelength, and electrical power dissipation and "optical" specifications such as the tuning speed and the tuning range [13]. According to the adopted technology, the tuning time may range from a few tens of nanoseconds (electrooptic) to a few tens of milliseconds (mechanical), or even seconds or minutes. Generally, a higher tuning speed can yield a better system performance. Bock *et al.* [16] claimed that tuning times in the range of microseconds offer good network performance at data rates of 2.5 Gb/s. However, the higher the tuning speed, the more sophisticated the technology is needed, and consequently the higher the tunable laser cost.

In this paper, we focus on investigating the impact of laser tuning (wavelength switching) time on dynamic bandwidth allocation (DBA) algorithms, and consequently the system performance. The bandwidth allocation refers to either downstream or upstream bandwidth allocation depending on the placement and usage of tunable lasers in the network. Laser tuning time constitutes an important consideration factor in designing DBA algorithms. When the laser tuning time is infinity, lasers have to stay on the same wavelengths all the time, and requests from ONUs can only be scheduled on the wavelengths their respective corresponding lasers stayed. No statistical gain can be exploited among ONUs that can support different wavelengths. When the laser tuning time is zero, requests can be scheduled on any wavelength any time. Then, statistical gain among all requests can be exploited. When the laser tuning time is between zero and infinity, proper DBA algorithms are desired to exploit the statistical gain among requests to the best under the condition that lasers are given enough time to switch wavelengths. To the best of our knowledge, this is the first time that the DBA problem with the consideration of laser tuning time is reported.

In this paper, we map the DBA problem under the condition of nonzero laser tuning time into a multiprocessor scheduling problem, with wavelength channels as machines and requests from ONUs as jobs. We try to minimize the latest ONU request (job) completion time for the sake of small delay, fairness, and load balancing. We show that when the laser tuning time is nonzero, both preemptive and nonpreemptive scheduling problems with the objective of minimizing the latest ONU request (job) completion time are NP-hard. We then present heuristic preemptive and nonpreemptive scheduling algorithms to address the scheduling problems, respectively. Theoretical analyses show that the heuristic preemptive scheduling algorithm achieves an approximation ratio of at most 2, and the heuristic nonpreemptive scheduling algorithm achieves an approximation ratio of at most $2 - 1/m$, where m is the number of wavelengths. Simulation results show that our proposed algorithms with the consideration of the laser tuning time have made significant performance improvement as compared to previous algorithms without considering the laser tuning time. It is also shown that the preemptive scheduling scheme has some advantages over nonpreemptive scheduling in terms of average delay and average throughput when the number of wavelengths is large and the number of ONUs is small. The advantages diminish with the decrease of the number of wavelengths and the increase of the number of ONUs. Note that this paper assumes that all lasers have the same optical specifications including tuning time and tuning range for the color-free purpose.

The rest of the paper is organized as follows. Section II describes the media access control, the scheduling framework, and the scheduling policy in a hybrid WDM/TDM PON. Section III maps the DBA problem into a multiprocessor scheduling problem and presents the formal problem formulation. Section IV presents preemptive scheduling schemes in a single cycle, and Section V describes nonpreemptive scheduling schemes in a single cycle. Section VI discusses the scheduling problem in multiple DBA cycles. Section VII presents simulation results and analyses. Section VIII presents concluding remarks.

II. Media Access Control, Scheduling Framework, and Scheduling Policy

For the downstream transmission in hybrid WDM/TDM PONs, the downstream incoming packets are queued in buffers at the OLT upon arrivals. Then, the OLT determines the downstream bandwidth allocated to ONUs and sends the downstream packets out to ONUs. Different from the downstream transmission, the upstream transmission in hybrid WDM/TDM PONs needs a proper MAC protocol to avoid data collision among ONUs. For backward compatibility, the MAC-layer control protocol of hybrid WDM/TDM PONs inherits some characteristics from those of EPON and GPON, two major flavors of the existing TDM PONs. The data transmission processes of the two PONs are similar and can be generalized as follows: ONUs report their queue lengths and send their data packets to the OLT using time slots allocated by the OLT; the OLT collects queue requests, makes bandwidth allocation decisions, and then notifies ONUs when and on which channel they can transmit packets. Such a request–grant-based transmission mechanism we believe is highly likely to be adopted in hybrid WDM/TDM PONs for consistency [19]–[21]. Following the assumption of a request–grant-based MAC control mechanism, the OLT gathers most of the intelligence and control of the network, and its functions determine the performance of the network.

Formerly, McGarry *et al.* [22] introduced the concept of the scheduling framework and scheduling policy to address the issues on when and how the OLT performs DBA, respectively. Three scheduling frameworks were defined, i.e., online scheduling, offline scheduling, and just-in-time scheduling. Online scheduling refers to the operation that OLT determines bandwidth allocated to an ONU immediately after receiving this ONU's request. Offline scheduling refers to the operation that OLT performs DBA after receiving queue requests from all ONUs. Both online scheduling and offline scheduling have their advantages and disadvantages. Online scheduling enables ONUs to get immediate grants. However, the bandwidth allocation decision is made based on only one ONU's request. This may result in unfairness for other ONUs with upcoming requests. Offline scheduling achieves better fairness by making decisions based on the requests of all ONUs. However, it incurs delays for ONUs to receive grants and underutilizes the gap between the time that the OLT sends out a grant and the time that the OLT receives the report from the first ONU. To overcome the near-sight problem of online scheduling and the underutilization problem of offline scheduling, McGarry *et al.* [22] proposed just-in-time scheduling, where the OLT postpones the decision-making time until one channel is about to become idle. The decision-making time in just-in-time scheduling is later than that in online scheduling and is earlier than that in offline scheduling. These three scheduling frameworks show similar advantages and disadvantages when they are applied in downstream scheduling.

The scheduling policy addresses the problem of how to perform DBA. It involves two problems: wavelength assignment and time allocation. For wavelength assignment, the earliest-channel-available-first rule was proposed to be employed with the assumption that each ONU can support all wavelengths [19], [21]. To make the algorithm applicable to the case that ONUs may support only a subset of the wavelengths, McGarry *et al.* [20] modified the earliest-channel-available-first rule into next-available-supported-channel-first. In [22], McGarry *et al.* converted the wavelength assignment problem into a matching problem between wavelengths and ONUs and proposed weighted bipartite matching to solve the matching problem. In [23], McGarry *et al.* modeled the problem into a multiprocessor scheduling problem and proposed to use the longest-processing-time (LPT)-first rule to address the minimizing makespan problem for the case that ONUs can access all the wavelengths. When ONUs can access a limited set of wavelengths, they schedule ONUs with the least flexible job (LFJ) first rule. Meng *et al.* [24] studied the joint grant scheduling and wavelength assignment problem. They formulated it into a mixed integer linear programming (MILP) problem and employed tabu search to obtain the optimal solution. For the time allocation problem, the time allocated to ONUs usually is equal to its corresponding request when the online scheduling framework is adopted. In offline scheduling, Dhaini *et al.* [21] proposed three time allocation algorithms, whereby low-load ONUs can always have their requests satisfied and high-load ONUs share the excess bandwidth by using different methods.

This paper focuses on investigating the scheduling policy for offline scheduling and just-in-time scheduling frameworks. Our objective is to consider laser tuning time and propose proper scheduling policy to exploit the benefit introduced by laser tunability to the best. The proposed scheduling policy can be applied in both downstream and upstream scheduling where tunable lasers are used to generate optical signals. To the best of our knowledge, this is the first time that the scheduling problem with the consideration of laser tuning time is investigated in WDM PONs.

Formerly, scheduling schemes with the consideration of the laser tuning time have been proposed for WDM broadcast-and-select networks [25], where each network node is configured with one tunable transmitter (TT) or fixed transmitter (FT) and one tunable receiver (TR) or fixed receiver (FR). The scheduling problem under the "FT-TR" configuration is the dual problem of that under the "TT-FR" configuration. The scheduling problems under both configurations have received extensive research attention [26]–[29]. Consider the "TT-FR" configuration. Since the receiver at each node is fixed tuned, the traffic to be scheduled on each wavelength is determined. The scheduling problem is reduced to the problem of sequencing requests on each wavelength channel such that enough time durations are left for two requests originating from the same node and the schedule length is minimized. This is different from the DBA problem to be addressed in this paper, which involves both wavelength assignment and time allocation problems.

For the "TT-TR" configuration, Rouskas *et al.* [30] decomposed it into two subproblems: determine the wavelength tuned by each receiver, and determine the scheduling time of the request of each node pair. The latter problem is equivalent to the problem under the "TT-FR" and "FT-TR" configurations. The former problem is similar to the DBA problem in WDM PONs, which is going to be addressed in this paper. For the former problem, Baldine *et al.* [31] considered the variation of traffic patterns and tried to adjust the wavelengths tuned by receivers to accommodate the new traffic pattern such that the total number of retunings is minimized. In this way, a small total amount of time is spent in the tuning process, and thus high bandwidth utilization may be achieved since the wavelength channel may be idle during the tuning. In this paper, we assume that during the retuning time that a receiver retunes to a wavelength, the objective wavelength can be used by other receivers, and thus a higher and even 100% time utilization can be achieved. Constructing a schedule that can fully utilize these retuning time durations can be considered as the objective of this paper. As will be discussed next, the problem can be considered as one multiprocessor scheduling problem where jobs take nonnegligible time to switch wavelengths.

III. Modeling and Problem Formulation

The bandwidth allocation problem in hybrid WDM/TDM PONs in a single DBA cycle discussed in this paper can be described as the following: Given the laser tuning time, requests from n ONUs, the available time of m wavelength channels, and the wavelength initially tuned by each laser, construct a schedule of the minimum length such that all requests can be accommodated and lasers are given enough time to switch wavelengths.

We can map the bandwidth allocation problem in WDM/TDM PONs into a multiprocessor scheduling problem [32] with ONU requests as jobs and wavelength channels as machines. Jobs and machines in this particular problem possess their respective unique characteristics.

A. Wavelengths $\Rightarrow$ Machines

Assume data rates are the same on all wavelengths. Wavelength channels are modeled as parallel machines. Note that these wavelength channels may not be simultaneously available.

B. ONU Requests ⇒ Jobs

There are two options to model jobs. The first one is to model each queue request of an ONU as an individual job. However, owing to the laser on/off time, some guard time is needed between scheduling of jobs from different ONUs. To save the guard time, jobs from the same ONU should be scheduled consecutively, and thus can be grouped together as a single job for simplicity. In this paper, we regard the total requests of an ONU as a single job. Then, jobs possess two main properties. First, owing to the laser tuning time, a certain time gap is needed between the scheduling of jobs from the same ONU on different wavelength channels. Second, a job can be divided into subjobs, corresponding to requests of queues in the ONU, and each subjob can be further divided into subjobs, corresponding to requests of packets in the queue. In GPON with the allowance of packet fragmentation, scheduling of a packet can even be divided into scheduling of its partial packets, while in EPON without packet fragmentation, scheduling of a packet cannot be further divided. Therefore, jobs are preemptable in hybrid WDM/TDM GPON and preemptable in a certain degree in hybrid WDM/TDM EPON.

In multiprocessor scheduling, preemption enables jobs to be scheduled more flexibly, thus yielding better system performances as compared to nonpreemption. However, when preemption is allowed, jobs may be divided and scheduled in noncontinuous time periods, which incur some extra time gap for laser on/off. It is not easy to tell whether the extra cost introduced by the guard time outweighs the extra performance improvement introduced by flexibility. In this paper, we simply assume zero guard time for laser on/off and investigate and compare the performances of preemptive and nonpreemptive scheduling.

C. Scheduling Objective

In assigning wavelengths to ONUs, we try to minimize the latest job completion time among all requests and equalize the usage of all wavelength channels for two main reasons. First, assume one wavelength is more loaded than another one. ONUs assigned with the overloaded wavelength may experience longer waiting time than those using the other wavelength. Equalizing the wavelength usage can ensure fairness among ONUs. Second, in terms of the just-in-time scheduling framework, OLT makes bandwidth allocation decisions before any of the wavelengths becomes idle. If all wavelengths become idle simultaneously, the scheduler can collect the requests from most of the ONUs and thus make a fair decision. If one wavelength turns idle much earlier than the others, few requests arrive at the scheduler before the decision-making time. In the worst case, just-in-time scheduling may be degraded into online scheduling, which makes the decision for one ONU request only. This will result in unfairness and increase the frequency of calculating bandwidth allocation.

Therefore, for the sake of small delay, fairness, and load balancing, we consider minimizing the latest job completion time as the scheduling objective.

D. Formal Problem Formulation

To mathematically formulate our scheduling problem, we introduce the definition of a *DBA cycle*. A DBA cycle in hybrid WDM/TDM PON refers to the time difference between two consecutive DBA decision-making instances.

Formally, our problem of DBA in a single DBA cycle can be stated as follows.

Given:

1) n : The number of ONUs.
2) m : The number of wavelengths.
3) τ : The laser tuning time.
4) t : The decision making time of the current cycle.
5) $\mathbf{r} = \{r_1, r_2, \ldots, r_n\}$: The time durations of requests from n ONUs.
6) $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_n^{-1}\}$: The wavelengths tuned by lasers at respective ONUs at the decision making time t.
7) $\{C_1^{-1}, C_2^{-1}, \ldots, C_m^{-1}\}$: The latest job completion time on m wavelengths in the last cycle.

Define:

1) γ_w: The sum of all requests with $\lambda_i^{-1} = w$, i.e., $\gamma_w = \sum_{\{i \mid \lambda_i^{-1}=w\}} r_i$.
2) $S^p(\mathbf{r},\tau)$, $S^{\bar{p}}(\mathbf{r},\tau)$: The preemptive and nonpreemptive schedules with respect to requests $\mathbf{r}$ and the laser tuning time τ.
3) $C_w^{S,p}(\mathbf{r},\tau)$, $C_w^{S,\bar{p}}(\mathbf{r},\tau)$: The latest job completion time on wavelength w in schedule S^p and $S^{\bar{p}}$, respectively.
4) $C_{\max}^{S,p}(\mathbf{r},\tau)$, $C_{\max}^{S,\bar{p}}(\mathbf{r},\tau)$: The latest job completion time among all wavelengths in schedule S^p and $S^{\bar{p}}$, respectively. $C_{\max}^{S,p}(\mathbf{r},\tau) = \max_{w=1}^{m} C_w^{S,p}(\mathbf{r},\tau)$. $C_{\max}^{S,\bar{p}}(\mathbf{r},\tau) = \max_{w=1}^{m} C_w^{S,\bar{p}}(\mathbf{r},\tau)$.

Objective:

Find optimal preemptive schedule $\mathfrak{S}^{\mathfrak{p}}$ and nonpreemptive schedule $\mathfrak{S}^{\mathfrak{p}}$ such that $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau) \leq C_{\max}^{S,p}(\mathbf{r},\tau)$ for all other S^p, and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau) \leq C_{\max}^{S,\bar{p}}(\mathbf{r},\tau)$ for all other $S^{\bar{p}}$.

Subject to:

1) Each request is allocated with sufficient time duration to be transmitted.
2) Each laser is given sufficient time to switch wavelengths if necessarily.
3) One laser cannot transmit on multiple wavelengths simultaneously.

To describe our proposed schemes, we further introduce the following notations.

Denote $\alpha_{i,w}$ as the earliest time that wavelength w can be allocated to the request from ONU i.

- For the downstream transmission, lasers can get the time allocation decision from the decision maker as early as the decision making time t. Consider the tuning time for lasers to switch wavelengths; laser i can tune to wavelength w at time t if $\lambda_i^{-1} = w$ and time $t+\tau$ if $\lambda_i^{-1} \neq w$. On the other hand, the latest job completion time on m wavelengths are $\{C_1^{-1}, C_2^{-1}, \ldots, C_n^{-1}\}$ in the former cycle. Therefore

$$\alpha_{i,w} = \begin{cases} \max\left\{C_w^{-1}, t\right\}, & \text{if } \lambda_i^{-1} = w \\ \max\left\{C_w^{-1}, t+\tau\right\}, & \text{otherwise} \end{cases}.$$

- For the upstream transmission, laser i needs to wait for $\mathrm{RTT}_i/2$ time duration to receive the decision sent from the

OLT, and the upstream traffic needs another $\mathrm{RTT}_i/2$ time to arrive at the OLT. Considering both the laser tuning time and the latest job completion time in the last cycle

$$\alpha_{i,w} = \begin{cases} \max\left\{C_w^{-1}, t + \mathrm{RTT}_i\right\}, & \text{if } \lambda_i^{-1} = w \\ \max\left\{C_w^{-1}, t + \tau + \mathrm{RTT}_i\right\}, & \text{otherwise} \end{cases}.$$

For simplicity, we assume $\mathrm{RTT}_i = \mathrm{RTT}'_i, \forall i \neq i'$, in this paper. Then, for both upstream and downstream transmissions, $\alpha_{i,w} = \alpha_{i',w}$, $\forall i$ with $\lambda_i^{-1} \neq w$, and $\forall i'$ with $\lambda_{i'}^{-1} \neq w$. For notational convenience, we denote $\alpha_{i,w}$ for request i with $\lambda_i^{-1} = w$ as a^l, and denote $\alpha_{i,w}$ for request i with $\lambda_i^{-1} \neq w$ as a^u. $a^l \leq a^u \leq a^l + \tau$, i.e., $\alpha_{i,w} = \begin{cases} a_w^l, & \text{if } \lambda_i^{-1} = w \\ a_w^u, & \text{otherwise} \end{cases}$.

Next, we investigate the preemptive and nonpreemptive scheduling algorithms, respectively.

IV. Preemptive Scheduling in a Single Cycle

The problem is equivalent to the preemptive multiprocessor scheduling problem with the objective of minimizing makespan subject to the constraints that machines are nonsimultaneously available and jobs take nonnegligible time to switch machines.

When the laser tuning time $\tau = 0$, and all wavelengths channels are available from the same time, i.e., $a_w^l = a_{w'}^l, \forall w \neq w'$, the problem is equivalent to the $p|pmtn|C_{\max}$ multiprocessor scheduling problem [33], which can be easily solved. When $\tau = 0$ and wavelength channels are not simultaneously available, i.e., $\exists w \neq w', a_w^l \neq a_{w'}^l$, the problem can be solved by slightly modifying the algorithm for the $p|pmtn|C_{\max}$ problem.

When the laser tuning time $\tau = +\infty$, the request from ONU i can only be scheduled on the original wavelength λ_i^{-1} tuned by ONU i. The latest job completion time on wavelength w equals $a_w^l + \sum_{\{i\,|\,\lambda_i^{-1}=w\}} r_i$. Among all wavelengths, the latest job completion time $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, +\infty)$ equals $\max_{w=1}^{m}\left(a_w^l + \sum_{\{i\,|\,\lambda_i^{-1}=w\}} r_i\right)$. Therefore

$$C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, +\infty) = \max_{w=1}^{m}\left(\sum_{\{i\,|\,\lambda_i^{-1}=w\}} r_i + a_w^l\right).$$

When the laser tuning time τ is an arbitrary value, as far as we know, this is the first time that the problem is investigated. We show that the preemptive scheduling problem is NP-hard.

Theorem 1: When the laser tuning time τ is arbitrary, the preemptive scheduling problem with the objective of minimizing the latest request completion time is NP-hard.

Proof: Consider the following downstream traffic scheduling problem with $C_w^{-1} = \begin{cases} t + \tau, & \text{if } w = 1 \\ t, & \text{otherwise} \end{cases}$, $\lambda_i^{-1} = 1\ \forall i$, and $r_i \in [\ell - t - 2\tau, \ell - t - \tau], \forall i$. Then, after checking all i and w, $\alpha_{i,w} = t + \tau, \forall i, w$. Since $\alpha_{i,w} + r_i \leq \ell$ and $\alpha_{i,w} + r_i + \tau \geq \ell$, any request can be scheduled on any wavelength, but cannot be divided into parts and scheduled on multiple wavelengths. The time duration that can be allocated on any wavelength equals $(\ell - t - \tau)$. The problem of determining whether all requests can be scheduled before ℓ is equivalent to the problem of deciding whether all these given requests can be divided into m groups, in which the sum of requests in each group is no greater than $\ell - t - \tau$. The latter problem is equivalent to the bin packing problem, which is NP-hard [34]. Hence, the preemptive scheduling problem with the objective of minimizing the latest request completion time is NP-hard when the laser tuning time τ is arbitrary. ■

Because of the NP-hard property, we next propose heuristic algorithms to solve the problem.

A. Naive Preemptive Scheduling

The first preemptive scheduling algorithm we propose, referred to as *naive preemptive scheduling*, is based on the schedules constructed for $\tau = 0$ and $\tau = +\infty$, respectively.

Since lasers with smaller tuning time yield a smaller latest job completion time, for any given requests $\mathbf{r}$, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) \leq C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau + \epsilon), \forall \epsilon > 0$. Hence

$$C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, 0) \leq C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) \leq C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, +\infty).$$

The main idea of naive preemptive scheduling is to first construct a schedule $\mathfrak{S}^p(\mathbf{r}, 0)$ assuming that the laser tuning time is zero and a schedule $\mathfrak{S}^p(\mathbf{r}, +\infty)$ assuming that the laser tuning time is $+\infty$. Then, naive preemptive scheduling adjusts the schedule $\mathfrak{S}^p(\mathbf{r}, 0)$ to give all lasers enough time to switch wavelengths. If the schedule length is less than the length of $\mathfrak{S}^p(\mathbf{r}, +\infty)$, the adjusted schedule based on $\mathfrak{S}^p(\mathbf{r}, 0)$ is considered as the final schedule. Otherwise, $\mathfrak{S}^p(\mathbf{r}, +\infty)$ is considered as the final schedule. Algorithm 1 details the proposed naive preemptive scheduling.

Algorithm 1: Naive Preemptive Scheduling

1: $\ell = (\sum_i r_i + \sum_w a_w)/m$
2: **while** The smallest latest request completion time ℓ has not been found **do**
3: Index ONU requests such that $r_1 \geq r_2 \geq \cdots \geq r_n$
4: Index wavelengths such that $a_1^l \leq a_2^l \leq \cdots \leq a_m^l$
5: Select an ONU request and a wavelength
6: Schedule the request on the back of the wavelength. If the remaining time on a wavelength is not enough for the request, schedule the remaining unscheduled part of the request to another wavelength.
7: **if** Not all requests can be scheduled before ℓ **then**
8: Find a proper ℓ
9: **end if**
10: **end while**
11: Postpone the scheduling of all requests on a wavelength by τ
12: Postpone the scheduling of the last request on a wavelength by τ
13: If the length of the constructed schedule is longer than $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, +\infty)$, $\mathfrak{S}^p(\mathbf{r}, +\infty)$ is considered as the final schedule.

The part between Lines 2 and 10 in Algorithm 1 is to construct a schedule assuming that the laser tuning time is zero. For the scheduling algorithm with zero laser tuning time, the main idea is to try different number ℓ and decide whether there exists a schedule whose latest job completion time is ℓ. Finally, the schedule with the minimum latest job completion time can be obtained.

For a given ℓ, requests are first sorted in the descending order of their sizes, and wavelengths are sorted in the ascending order of their available time as described in Lines 3 and 4. The sorting is to make sure that large requests receive enough allocations of nonoverlapping time durations. Then, the time resource on a wavelength is assigned to requests one by one from the back of the time span until the time on that wavelength is used out. If the remaining time on a wavelength is not enough to satisfy a request, the unscheduled part will be moved to the next wavelength as described in Line 6.

To generate a feasible schedule that gives lasers sufficient time to switch wavelengths, we perform some further adjustments. The first step is to postpone the scheduling of all requests by τ as described in Line 11. The second step is to postpone the scheduling of the last request on each wavelength by τ as described in Line 12.

We then prove that the schedule produced by naive preemptive scheduling is a feasible schedule under the condition that the laser tuning time equals τ.

Lemma 1: The schedule produced by Algorithm 1 is a feasible schedule for the case that the laser tuning time equals τ.

Proof: Since all requests are postponed by time τ, the corresponding laser for request i is idle during $[a_w^l, a_w^l + \tau]$, and laser i is given enough time to schedule the first request. Besides, in the schedule $\mathfrak{S}^p(\mathbf{r}, 0)$, only requests scheduled at the beginning or the end of the time span of a wavelength may be preempted. In Line 12, the last scheduled request on each wavelength is postponed by τ, and hence lasers are given sufficient time to schedule the last scheduled request.

Therefore, in the schedule produced by Algorithm 1, all requests have been scheduled, and lasers are all given enough time to switch wavelengths. ■

We have the following theorem regarding $C_{\max}^{S,p}(\mathbf{r}, \tau)$ produced by naive preemptive scheduling.

Theorem 2: For a given $\mathbf{r}$, the schedule S produced by Algorithm 1 has the property that $C_{\max}^{S,p}(\mathbf{r}, \tau) = \min\{C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, 0) + 2\tau, C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, +\infty)\}$, and consequently the optimal schedule $\mathfrak{S}(\mathbf{r}, \tau)$ has the property that $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) \leq \min\{C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, 0) + 2\tau, C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, +\infty)\}$

Proof: The "while" loop between Lines 2 and 10 optimally solves the scheduling problem under the condition of zero laser tuning time. As compared to the schedule with zero laser tuning time, the latest job completion time on each wavelength in the final schedule is increased by 2τ. Thus, $C_{\max}^{S}(\mathbf{r}, \tau)$ obtained by naive preemptive scheduling equals $\min\{C_{\max}^{\mathfrak{S}}(\mathbf{r}, 0) + 2\tau, C_{\max}^{\mathfrak{S}}(\mathbf{r}, +\infty)\}$. Since $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) \leq C_{\max}^{S,p}(\mathbf{r}, \tau)$, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) \leq \min\{C_{\max}^{\mathfrak{S}}(\mathbf{r}, 0) + 2\tau, C_{\max}^{\mathfrak{S}}(\mathbf{r}, +\infty)\}$. ■

Computational Complexity: The complexities of the two ordering processes are $O(n\log(n))$ and $O(m\log(m))$, respectively. The complexity of the "for" loop in Algorithm 1 is $O(n)$. Lines 11–13 are all of complexity $O(n)$. Hence, the total complexity of Algorithm 1 is $O(n\log(n) + m\log(m))$.

The example with eight ONUs and three wavelengths as shown in Fig. 1 illustrates Algorithm 1. Fig. 1(a) shows the constructed schedule assuming that the laser tuning time equals zero. Request 1 is allocated with the time duration [6, 14] on wavelength 1. The remaining time duration on wavelength 1 is not enough to satisfy request 2. Part of request 2 is scheduled in time duration [0, 6] on wavelength 1, and the other part is scheduled in time duration [13, 14] on wavelength 2. Similarly, part of request 4 is scheduled on wavelength 2, and the other part is scheduled on wavelength 3. All the requests can be scheduled before time 14. Fig. 1(b) shows the final schedule after adjustment. When $\tau = 1.5$, the latest job completion time is increased from 14 to 17.

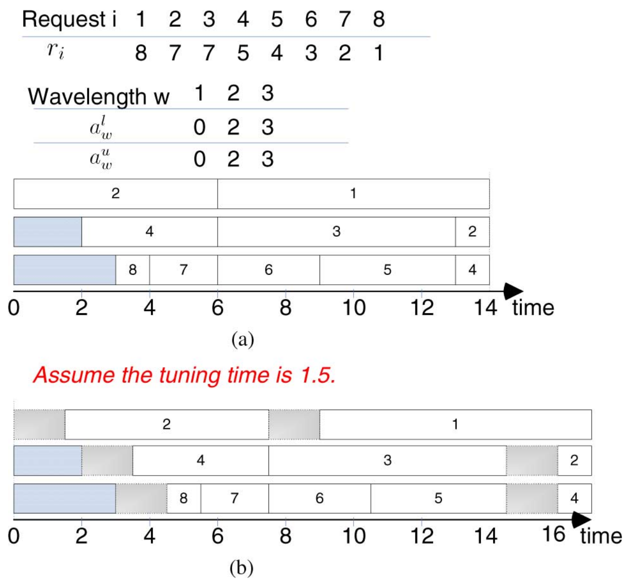


Fig. 1. One example of naive preemptive scheduling when $\tau = 1.5$. (a) Preemptive schedule when $\tau = 0$. (b) Preemptive schedule when $\tau = 1.5$.

B. Heuristic Preemptive Scheduling

In the schedule produced by Algorithm 1, there are two idle time gaps of duration τ on each wavelength. One is the time gap between a_w^l and $a_w^l + \tau$, and the other one is the time gap before the scheduling of the last request on the wavelength. To produce a schedule S with $C_{\max}^{S}(\mathbf{r}, \tau)$ smaller than $\min\{C_{\max}^{\mathfrak{S}}(\mathbf{r}, 0) + 2\tau, C_{\max}^{\mathfrak{S}}(\mathbf{r}, +\infty)\}$, these idle time gaps need to be filled to the best. To this end, we propose a heuristic preemptive scheduling algorithm as described in Algorithm 2.

Algorithm 2 constructs the schedule according to three basic rules.

- Since the time gap between a_w^l and a_w^u on wavelength w can only be filled by requests with $\lambda_i^{-1} = w$, Algorithm 2 fills this time gap with requests with $\lambda_i^{-1} = w$ to the best. Request i is scheduled on a wavelength other than its originally tuned wavelength λ_i^{-1} only if the gap filling between $a_{\lambda_i^{-1}}^l$ and $a_{\lambda_i^{-1}}^u$ is not affected after the scheduling.
- Similar to Algorithm 1, to guarantee that large requests receive enough bandwidth, the resource in the wavelength with the smallest a_w^l is allocated first, and the largest request is scheduled first.
- Similar to Algorithm 1, preemption is disallowed in the middle of the time span on a wavelength. If one request is preempted and scheduled during $[\mu, \nu]$ on a wavelength, this request cannot be scheduled during $\mu - \tau$ and $\nu + \tau$ on any wavelength, thus resulting in smaller chances of scheduling the remainder of the request in other wavelengths.

More specifically, Algorithm 2 divides the resource allocation process on a wavelength into two steps. The first step is to allocate the time duration between a_w^u and ℓ. The second step is to allocate the time duration between a_w^l and a_w^u. The time duration between a_w^u and ℓ on wavelength w can be allocated to any request, while the time duration between a_w^l and a_w^u on wavelength w can only be allocated to requests with $\lambda_i^{-1} \neq w$.

Algorithm 2: Heuristic Preemptive Scheduling

1: Initialize $x_w = \gamma_w, y_w = \ell$
2: Index wavelengths such that $a_1^l \leq a_2^l \leq \cdots \leq a_m^l$
3: **for** $w = 1 : m$ **do**
4:     */*Step 1: The allocation between a_w^u and ℓ*/*
5:     Index unscheduled requests such that $r_1 \geq r_2 \geq \cdots$
6:     **for** $i = 1 : n$ **do**
7:         **if** $r_i \leq y_w - a_w^u$ & $x_{\lambda_i^{-1}} - r_i \geq (a_{\lambda_i^{-1}}^u - a_{\lambda_i^{-1}}^l)$ **then**
8:             Allocate time duration $[y_w - r_i, y_w]$ to request i
9:             $y_w = y_w - r_i, x_{\lambda_i^{-1}} = x_{\lambda_i^{-1}} - r_i$
10:         **end if**
11:     **end for**
12:     Denote x_w and y_w as x_w^* and y_w^*, respectively.
13:     */*Step 2: The allocation between a_w^l and y_w^**/*
14:     Index unscheduled requests with $\lambda_i^{-1} = w$ such that $r_1 \geq r_2 \geq \cdots$
15:     $i = 1$
16:     **while** there is available time on wavelength w and there are unscheduled requests with $\lambda_i^{-1} = w$ **do**
17:         **if** $r_i \leq y_w - a_w^l$ **then**
18:             Allocate time $[y_w - r_i, y_w]$ on wavelength w to request i
19:         **else**
20:             Allocate time $[a_w^l, y_w]$ on wavelength w and time $[\ell-(r_i-y_w+a_w^l), \ell]$ on wavelength $w+1$ to request i
21:             $y_{w+1} = \ell - (r_i - y_w + a_w^l)$
22:         **end if**
23:         $i = i + 1$
24:     **end while**
25: **end for**

---

1) Step 1: Allocation Between a_w^u and ℓ: When allocating the resource between a_w^u and ℓ, the largest request is considered first. To avoid preemption in the middle of the schedule, a request is scheduled on wavelength w only if the remaining available time duration after a_w^u is enough to accommodate the request. On the other hand, since allocating request i to wavelength w will decrease the total traffic $\gamma_{\lambda_i^{-1}}$ that can be used to fill the gap between $a_{\lambda_i^{-1}}^l$ and $a_{\lambda_i^{-1}}^u$ on wavelength λ_i^{-1}, we allocate request i onto wavelength w only if the gap filling between $a_{\lambda_i^{-1}}^l$ and $a_{\lambda_i^{-1}}^u$ is not affected. The conditions are described in Line 7 of Algorithm 2.

We use x_w to track the total traffic that can fill the gap between a_w^l and a_w^u, and y_w to track the last available time stamp on wavelength w. x_w and y_w are initialized to be γ_w and ℓ, respectively. The time duration between a_w^l and y_w is the resource that is still available on wavelength w. Denote x_w and y_w after performing the first step as x_w^* and y_w^*, respectively. The first step tries to let y_w^* approach a_w^u without affecting the filling of the gap between a_w^l and $a_w^u, \forall w$.

2) Step 2: Allocation Between a_w^l and a_w^u: After performing the first step, the available time duration on wavelength w is actually between a_w^l and y_w^*. The second step of allocating the time between a_w^l and y_w^* considers the largest unscheduled request with $\lambda_i^{-1} = w$ first. If the remaining time is not enough to schedule the request, the remaining unscheduled part of the request is scheduled onto the next wavelength as described in Line 20.

It can be seen that if $x_w^* \geq y_w^* - a_w^l$, the time duration between a_w^l and y_w^* on wavelength w can be allocated, and there is no idle time gap on wavelength w. If $x_w^* < y_w^* - a_w^l$, the time duration between a_w^l and $y_w^* - x_w^*$ on wavelength w is idle. Algorithm 2 takes two measures to reduce the duration of the idle time gap. One is to let y_w^* approach a_w^u to the best, and the other one is to make sure that x_w^* is always above $a_w^u - a_w^l$ if $\gamma_w \geq a_w^u - a_w^l$.

3) Analysis: Theorem 3 describes the upper bound of $C_{\max}^{S,p}(\mathbf{r}, \tau)$ produced by Algorithm 2.

Theorem 3: For given $\mathbf{r}$ and τ, schedule S produced by Algorithm 2 has the property that $C_{\max}^{S,p}(\mathbf{r}, \tau) \leq C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) + \max_{i=1}^n r_i$, where $\mathfrak{S}$ is the optimal schedule with the minimum latest request scheduling time.

Proof: We prove this theorem by showing that when $\ell = C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) + \max_{i=1}^n r_i$, Algorithm 2 can schedule all requests before ℓ.

Denote gap_w and $\mathfrak{gap}_w$ as the duration of idle time on wavelength w in schedule S produced by Algorithm 2 and that in the optimal schedule $\mathfrak{S}$. As described, when $x_w^* \geq y_w^* - a_w^l$, there is no idle time on wavelength w. Otherwise, the time between a_w^l and $y_w^* - x_w^*$ on wavelength w is idle, and gap_w equals $y_w^* - x_w^* - a_w^l$. Also, we know from the above that $x_w^* \begin{cases} \geq a_w^u - a_w^l, & \text{if } \gamma_w \geq a_w^u - a_w^l \\ = \gamma_w, & \text{otherwise} \end{cases}$. Thus

$$\text{gap}_w \begin{cases} \leq y_w^* - a_w^u, & \text{if } \gamma_w \geq a_w^u - a_w^l \\ = y_w^* - \gamma_w - a_w^l, & \text{otherwise} \end{cases}.$$

On the other hand, in the optimal schedule $\mathfrak{S}$, when $\gamma_w < a_w^u - a_w^l$, there must be some idle time duration between a_w^l and a_w^u, and $\mathfrak{gap}_w = a_w^u - a_w^l - \gamma_w$. Accordingly, we can further obtain that $\text{gap}_w \leq \mathfrak{gap}_w + y_w^* - a_w^u$ for all w. In Step 1, we let y_w^* approach a_w^u to the best by trying every unscheduled job. It can be easily obtained that $y_w^* - a_w^u \leq \max_{i=1}^n r_i$ if there is still an unscheduled job that does not have to be scheduled on its originally tuned wavelength. Then

$$\text{gap}_w \leq \mathfrak{gap}_w + y_w^* - a_w^u \leq \mathfrak{gap}_w + \max_{i=1}^{n} r_i.$$

Therefore, all requests can be scheduled before $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) + \max_{i=1}^n r_i$ by using Algorithm 2. ■

Corollary 1: The approximation ratio of the heuristic preemptive scheduling algorithm is at most 2.

Proof: Based on Theorem 3, $C_{\max}^{S,p}(\mathbf{r}, \tau) \leq C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) + \max_{i=1}^n r_i$ On the other hand, $C_{\max}^{S,p}(\mathbf{r}, \tau) \leq \max_{i=1}^n r_i$. Therefore

$$C_{\max}^{S,p}(\mathbf{r}, \tau) / C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) \leq 2.$$

That is to say, the approximation ratio is at most 2. ■

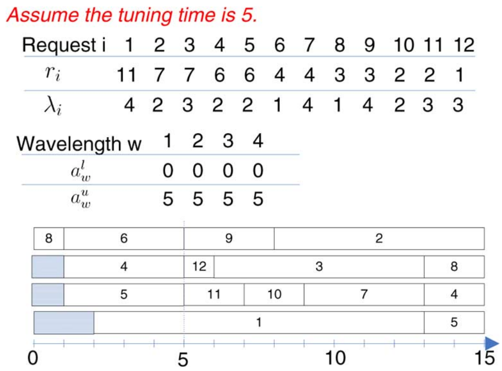


Fig. 2. One example of heuristic preemptive scheduling when $\tau = 5$.

Computational Complexity: The ordering process in Line 2 of Algorithm 2 has the complexity of $O(m\log(m))$. The "for" loop in Step 1 has the complexity $O(n)$. The total complexity of performing Step 1 is $O(mn)$. The complexity of Step 2 is $O(n\log(n))$. Hence, Algorithm 2 has the complexity of $O(mn + m\log(m) + n\log(n))$.

Taking the example with 12 ONUs and 4 wavelengths, Fig. 2 illustrates the algorithm. The laser tuning time $\tau = 5$. The decision making time $t = 0$. The latest job completion times on wavelengths in the last cycle are 0, 1, 1, and 2, respectively. Then, $a^l = \{0, 1, 1, 2\}$, $a^u = \{5, 5, 5, 5\}$, and $\ell = 15$. Request 1 with the largest size cannot be scheduled on wavelength 1 because the bandwidth from a_1^u to ℓ is not enough to satisfy request 1. Request 2 satisfies all conditions and is scheduled on wavelength 1. After scheduling Request 2, the remaining time duration between a_1^u and ℓ can only accommodate requests with sizes no greater than 3. Request 8 with size 3 is not scheduled between a_1^u and ℓ on wavelength 1 because the gap between a_1^l and a_1^u cannot be filled without request 8. After scheduling Request 9, the time between a_1^u and ℓ on wavelength 1 has been all allocated. The scheduling enters into the second step of allocating $[a_1^l, a_1^u]$. After scheduling Request 6, which is the largest among all requests with $\lambda_i^{-1} = 1$, the remaining time duration is not enough to schedule the next largest request, i.e., Request 8 with 3. Hence, Request 8 is divided into two parts, among which the first part of size 1 is scheduled on wavelength 1 and the second part of size 2 is scheduled on wavelength 2. After repeating this process, we find that all requests can be scheduled before $\ell = 15$.

V. Nonpreemptive Scheduling in a Single Cycle

When the laser tuning time $\tau = 0$, the nonpreemptive scheduling problem was proved NP-hard [33]. For the case that all wavelength channels are available from the same time, i.e., $a_w^l = a_{w'}^l, \forall w \neq w'$, the problem is equivalent to the $p||C_{\max}$ multiprocessor scheduling problem. LPT was shown to have $4/3 - 1/n$ approximation ratio, and the MULTIFIT algorithm is with a smaller approximation ratio of $72/61$ [35]. When the wavelength channels are nonsimultaneously available, Lee *et al.* [36] proposed modified LPT (MLPT) to achieve $4/3$ approximation ratio. Lin *et al.* [37] showed that $4/3$ is the exact bound for MLPT when the number of processors is greater than two, and the approximation ratio is $5/4$ when the number of processors equals to two. Chang *et al.* [38] showed that the approximation ratio of the MULTIFIT algorithm is $9/7 + 2^{-k}$, where k is the selected number of the major iterations in the MULTIFIT. This is the smallest one known so far.

When $0 < \tau < +\infty$, the problem is NP-hard since it is not easier than the problem under the case that $\tau = 0$. We then propose two heuristic algorithms to address it.

A. Naive Nonpreemptive Scheduling

As described in Algorithm 3, naive nonpreemptive scheduling is based on the algorithm proposed for the $\tau = 0$ case. In this paper, we use the MULTIFIT algorithm to construct the nonpreemptive schedule for the $\tau = 0$ case.

In Algorithm 3, schedule S with the assumption of zero laser tuning time is first constructed by using the MULTIFIT algorithm. Then, the scheduling of all requests is postponed by time τ to give lasers sufficient time to switch wavelengths.

Algorithm 3: Naive Nonpreemptive Scheduling

```
Construct schedule S by using MULTIFIT.
Postpone scheduling of all requests in S by time τ.
```

Since preemption is disallowed in schedule S, each request is scheduled on one wavelength only in schedule S. By postponing all requests in $S(\mathbf{r}, 0)$ with a time duration of τ, the time period between a_w^l and $a_w^l + \tau$ is idle, and hence lasers can have sufficient time to switch wavelengths.

From Algorithm 3, we can derive the following theorem regarding $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau)$.

Theorem 4: For a given $\mathbf{r}$, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) \leq C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, 0) + \tau$.

Proof: For a given $\mathbf{r}$, assume $\mathfrak{S}$ is the optimal schedule for the case that $\tau = 0$. If all requests in $\mathfrak{S}$ are postponed by τ time duration, the newly obtained schedule is a feasible schedule for the case that $\tau \neq 0$. Hence, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, \tau) \leq C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, 0) + \tau$. ■

B. Heuristic Nonpreemptive Scheduling

In the schedule produced by Algorithm 3, the time duration between a_w^l and $a_w^l + \tau$ is unoccupied. To generate a schedule with a smaller latest job completion time, we propose Algorithm 4 by filling idle time durations on all wavelengths. Algorithm 4 contains two steps. The first step is to allocate the time period between a_w^l and a_w^u on wavelength w for requests with $\lambda_i^{-1} = w$. Large jobs are given higher priorities over small jobs since they are not easy to switch wavelengths. The second step is to allocate the time period between a_u^l and ℓ. We apply the MULTIFIT algorithm directly in the second step.

Algorithm 4: Heuristic Nonpreemptive Scheduling

```
/*Step 1: Schedule between a_w^l and a_w^u*/
for w = 1 : m do
    Index requests with λ_i^{-1} = w such that
    r_1 ≥ r_2 ≥ r_3 ···.
    Allocate time to requests with λ_i^{-1} = w until the
    time exceeds a_w^u.
end for
```

6: /**Step 2: Schedule between* a_w^u *and* ℓ*/

7: Run the MULTIFIT algorithm to allocate bandwidth to the remaining unscheduled requests.

Theorem 5: The approximation ratio of Algorithm 4 is at most $2 - 1/m$.

Proof: Let job x be the last job assigned to all m wavelengths. The processing time of job x is r_x. Also, let s be the start time that job x is processed. Then, the latest job completion time equals $r_x + s$. Let $\beta_w = \max\{\tau - \sum_{\{i\,|\,\lambda_i^{-1}=w\}} r_i, 0\}$. β_w denotes the time duration between time α_w^l and α_w^u that cannot be filled anyway. It is easy to see that, with Algorithm 4, $s < (\sum_{i=1}^{x-1} r_i + \sum_{i=x+1}^{n} r_i + \sum_{w=1}^{m} (\alpha_w^l + \beta_w))/m$. $s + r_x/m < (\sum_{i=1}^{x-1} r_i + \sum_{i=x+1}^{n} r_i + \sum_{w=1}^{m} (\alpha_w^l + \beta_w))/m + r_x/m = (\sum_{i=1}^{n} r_i + \sum_{w=1}^{m} (\alpha_w^l + \beta_w))/m + (m-1)r_x/m$. On the other hand, the optimal value $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ must not be less than $(\sum_{i=1}^{n} r_i + \sum_{w=1}^{m} (\alpha_w^l + \beta_w))/m$. Therefore, $s + r_x < C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau) + (1-1/m)r_x$. Also, it is known that $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau) > r_x$. Consequently, we can obtain that $(s + r_x)/C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau) < 2 - 1/m$. Therefore, the approximation ratio is at most $2 - 1/m$. ■

Computational Complexity: The ordering process in Step 1 of Algorithm 4 has the complexity of $O(n\log(n))$. The allocation process in Step 1 has the complexity of $O(n)$. Hence, the complexity of Step 1 in Algorithm 4 is $O(n\log(n))$. Algorithm 3 has the complexity of $O(nm)$. Therefore, Algorithm 4 has the complexity of $O(n\log(n) + nm)$.

Taking the same example as shown in Fig. 2, Fig. 3 illustrates Algorithm 4. Assume the tuning time is 5 and $\ell = 13$. After the first step, jobs 2–4, 8, and 10 still use their last tuned wavelengths. After the second step, all the remaining jobs are successfully scheduled before time 13. Note that ℓ cannot be further decreased. The latest job completion time is thus 13.

Request i	1	2	3	4	5	6	7	8	9	10	11	12
r_i	1	4	3	7	6	2	1	7	2	8	3	4
λ_i	1	1	1	2	2	2	3	3	3	4	4	4

Wavelength w	1	2	3	4
a_w^l	0	0	0	0
a_w^u	5	5	5	5

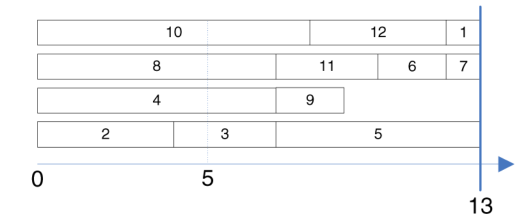


Fig. 3. One example of heuristic nonpreemptive scheduling when $0 < \tau < +\infty$.

VI. Discussions on the Scheduling in Multicycle Scenario

As discussed before, besides the decision making time t, the laser tuning time τ, and requests $\mathbf{r}$, the latest job completion time also depends on $\{C_1^{-1}, C_2^{-1}, \ldots, C_w^{-1}\}$ and $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$, which are determined by the schedule in the last cycle. We next discuss the relations between $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ and $\{C_1^{-1}, C_2^{-1}, \ldots, C_w^{-1}\}$, and $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$, respectively.

A. $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ *and* $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ *versus* $\{C_1^{-1}, C_2^{-1}, \ldots, C_m^{-1}\}$

To obtain small $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$, the earliest time $\alpha_{i,w}$ that wavelength w can be allocated to request i is desired to be small for all i, w; this consequently requires small $\{C_1^{-1}, C_2^{-1}, \ldots, C_m^{-1}\}$. Thus, minimizing the latest job completion time in the last cycle can help produce a small latest job completion time in the current cycle. On the other hand, as discussed in Section III-C, minimizing the latest job completion time in a cycle can achieve small delay, fairness, and load balancing of traffic in that particular cycle. Therefore, minimizing the latest job completion time yields good performances not only for traffic in the current cycle, but also for traffic in future cycles. The latest job completion time in each cycle should be minimized when considering the traffic in multiple cycles.

B. $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ *and* $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ *versus* $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$

$C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ are closely related to $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$. The following two lemmas can be derived.

Lemma 2: When $\tau = 0$, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ are independent of $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$.

Proof: When $\tau = 0$, $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_w^{-1}\}$ can be changed to any other $\{\lambda_1', \lambda_2', \ldots, \lambda_w'\}$ in no time at the decision-making time t. Therefore, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ are independent of $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$. ■

Lemma 2 states that, when $\tau = 0$, the scheduling in the current cycle does not depend on the scheduling in the last cycle and does not need to consider the scheduling in future cycles.

Lemma 3: When $\tau = +\infty$, if $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$ satisfies the condition that no other $\{\lambda_1', \lambda_2', \ldots, \lambda_m'\}$ can yield a smaller $\max_{w=1}^{m} (a_w^l + \gamma_w)$, $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$ can produce the smallest $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$.

Proof: As described, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty) = C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty) = \max_{w=1}^{m} (a_w^l + \gamma_w)$. If no other $\{\lambda_1', \lambda_2', \ldots, \lambda_m'\}$ yields a smaller $\max_{w=1}^{m} (a_w^l + \gamma_w)$ than $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$, $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$ can produce the smallest $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$. ■

Lemma 3 states that, when $\tau = +\infty$, the scheduling in the current cycle is closely related to $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$, which is determined in the last cycle. For given requests $\mathbf{r}$, $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$ is desired to achieve equal $a_w^l + \gamma_w$ among all wavelengths.

When $0 < \tau < +\infty$, both preemptive and nonpreemptive scheduling with minimizing the latest job completion time as the objective are NP-hard. $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ cannot be expressed as a closed-form function of $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$. We estimate that the smaller the τ, the less dependency of $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau)$ on $\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}\}$.

VII. Simulation Results and Analysis

In this section, we first investigate the cycle duration in the single DBA cycle case, and then continue to discuss the performances in the multiple-cycle case.

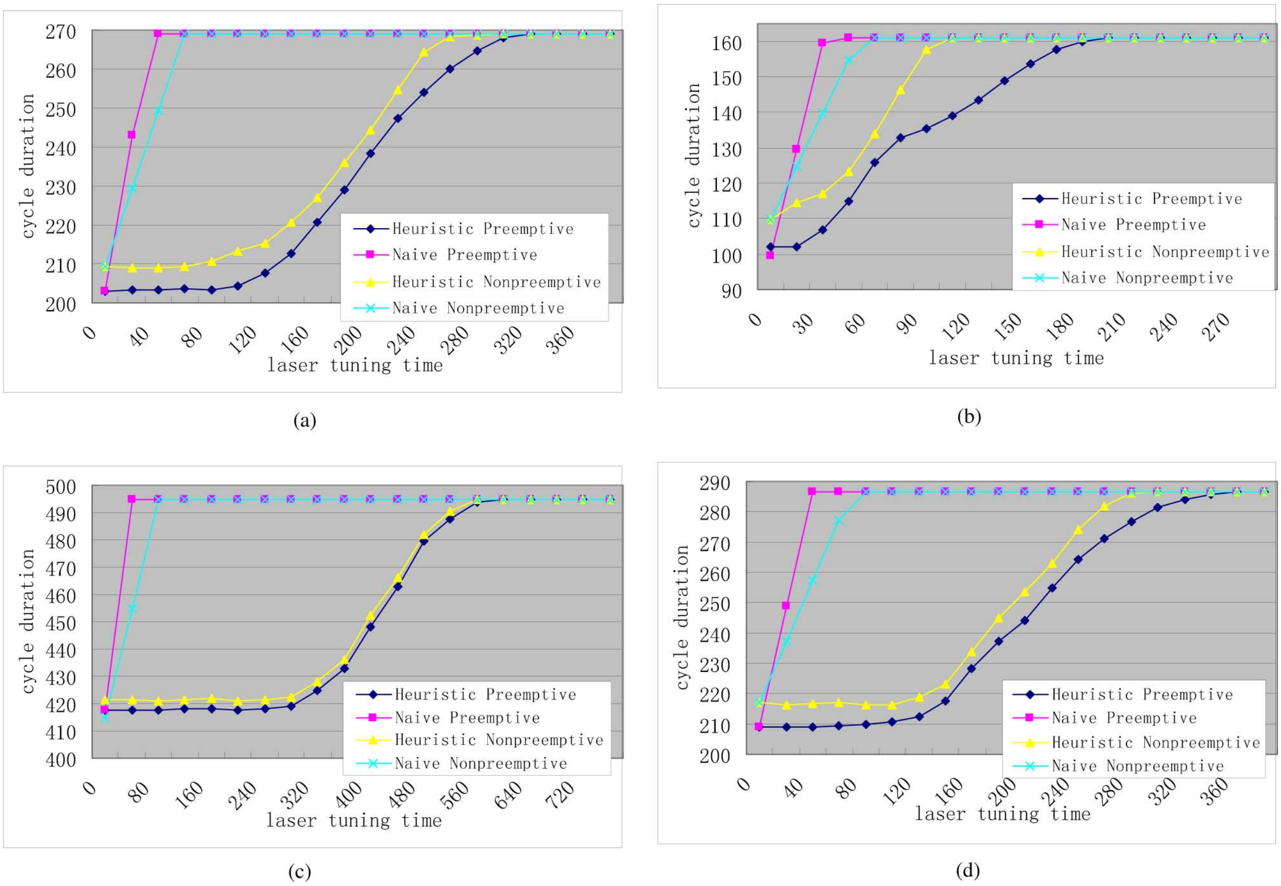


Fig. 4. Cycle duration versus laser tuning time. (a) $n = 16$, $m = 4$. (b) $n = 16$, $m = 8$. (c) $n = 32$, $m = 4$. (d) $n = 32$, $m = 8$.

A. Single-Cycle Case

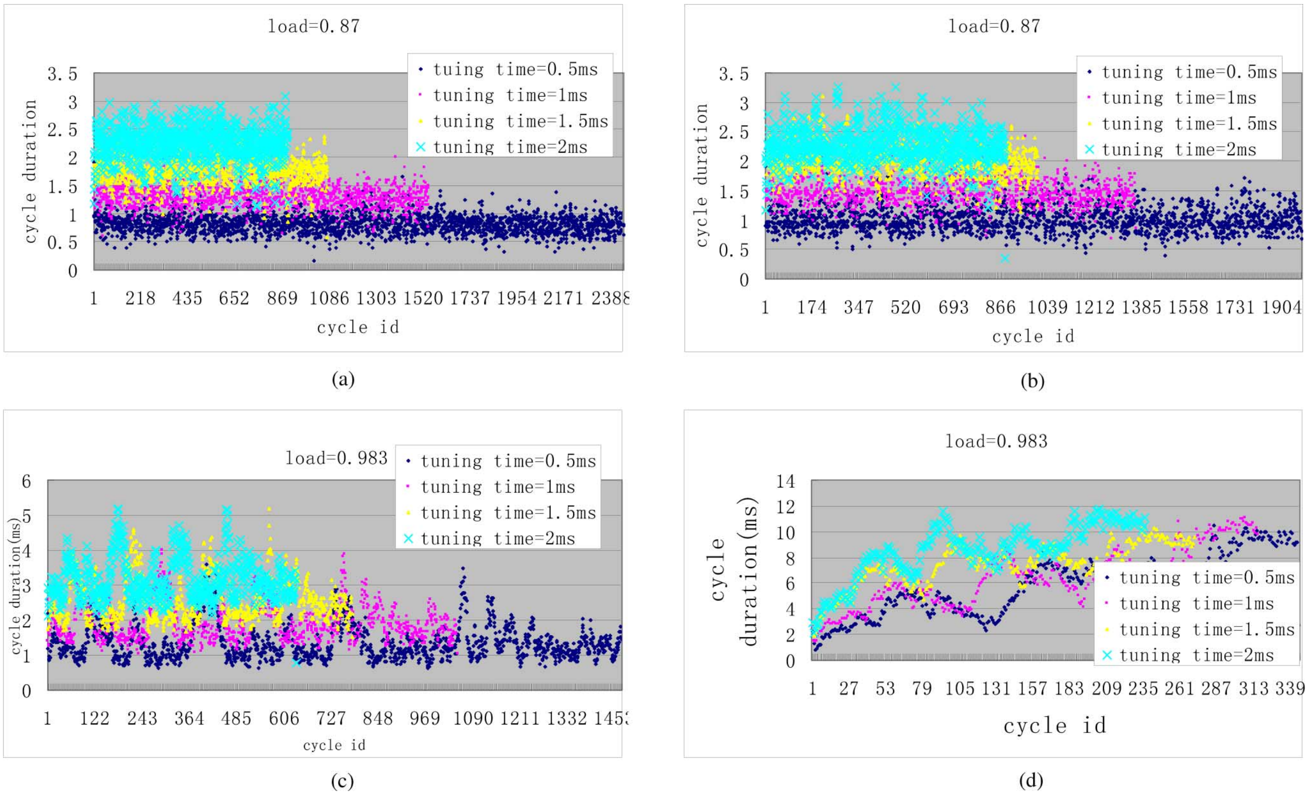
In the simulation, requests are expressed in terms of time durations, i.e., requested time durations. The number of ONUs n is set as 16 or 32, and the number of wavelengths m is set as 4 or 8. All the channels are available from time 0. The request sizes are uniformly distributed between 0 and 100, and the originally tuned wavelength λ_i^{-1} of ONU i at time 0 is set as $\lfloor i/m \rfloor$ such that each wavelength channel was tuned to by the same amount of ONUs at time 0. The cycle duration equals the latest request completion time in this case. We randomly generate 200 sets of requests and investigate their average performances.

Fig. 4 shows the relation between the cycle duration and the laser tuning time. From the figure, we can also observe the gap between $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$ and the gap between $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$. Since the cycle duration increases with the increase of the laser tuning time, $1 - C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)/C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$ can be interpreted as the maximum relative saving of the cycle duration benefited from the laser tunability. Simulation results in Fig. 4 show that $1 - C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)/C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$ increases with the increase of the number of ONUs and the decrease of the number of wavelengths. This is reasonable because large multiplexing gain can be exploited when the number of ONUs is large and the number of wavelengths is small.

Owing to the NP-hard property of the nonpreemptive scheduling problem, the exact value of $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ cannot be obtained. However, $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ is between $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ and the schedule length $C_{\max}^{S,\bar{p}}(\mathbf{r},0)$ produced by naive preemptive scheduling. Simulation results in Fig. 4 show that, in the four cases of combinations of ONU number and wavelength number, $C_{\max}^{S,\bar{p}}(\mathbf{r},0)$ exceeds $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$ by no more than 10%. Hence, we can infer that $1 - C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)/C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$ also increases with the increase of the number of ONUs and the decrease of the number of wavelengths.

For naive preemptive scheduling, Section IV-A shows that, for a given $\mathbf{r}$, $C_{\max}^{S,p}(\mathbf{r},\tau) = \min\{C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0) + 2\tau, C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)\}$. Therefore, when τ is not greater than $(C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty) - C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0))/2$, the cycle duration increases with the increase of τ by 2τ. When the laser tuning time increases beyond $(C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty) - C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0))/2$, the cycle duration equals $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$. For naive nonpreemptive scheduling, $C_{\max}^{S,\bar{p}}(\mathbf{r},\tau) = \min\{C_{\max}^{S,\bar{p}}(\mathbf{r},0)+\tau, C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)\}$. Therefore, the cycle duration remains constant as $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},+\infty)$ when the laser tuning time increases beyond $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\infty) - C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},0)$.

For heuristic preemptive scheduling and heuristic nonpreemptive scheduling, simulation results shown in Fig. 4 demonstrate that they yield significant performance improvement as compared to naive preemptive scheduling and naive nonpreemptive scheduling, respectively. The general trend of the relation between the cycle duration and the laser tuning time τ is as follows. When τ is below some value, referred to as "knee point 1," the cycle duration almost keeps as low as the case that $\tau = 0$. Beyond "knee point 1," the cycle duration increases with the increase of τ. Further increasing τ to



(a) (b) (c) (d)

Fig. 5. Variation of the cycle duration over time ($n = 16$, $m = 4$). (a) Preemptive scheduling, $\text{load} = 0.87$. (b) Nonpreemptive scheduling, $\text{load} = 0.87$. (c) Preemptive scheduling, $\text{load} = 0.983$. (d) Nonpreemptive scheduling, $\text{load} = 0.983$.

another value, referred to as "knee point 2," the cycle duration almost equals the value obtained in the case that $\tau = +\infty$. When $n = 16$ and $n = 4$, Fig. 4(a) shows that, in the curve describing the relation between cycle durations produced by heuristic preemptive scheduling and the laser tuning time, "knee point 1" happens when τ is around 120 and $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, 0)$ is around 202. Therefore, "knee point 1" is even larger than $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, 0)/2$. In addition, from Fig. 4, we can see that "knee point 1"$/C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, 0)$ increases with the increase of the ONU number and increases with the decrease of the wavelength number. This is again due to the fact that the multiplexing gain is large when the number of ONUs is large and the number of wavelengths is small. On the other hand, "knee point 2" is always around $C_{\max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r}, +\infty)$ in all cases of n and m.

Besides, heuristic preemptive scheduling performs better than heuristic nonpreemptive scheduling because of the scheduling flexibility benefited from the allowance of preemption. For a given n and m, "knee point 2" of heuristic nonpreemptive scheduling is smaller than that of heuristic preemptive scheduling, and "knee point 2" of two algorithms are similar. The outperformance is not obvious when $n = 32$ and $m = 4$, but is significant when $n = 16$ and $m = 8$. When $n = 16$ and $m = 4$, the outperformance is less than 5%.

B. Multiple-Cycle Case

For the multiple-cycle case, we take the configuration of $n = 16$ and $m = 8$ for example to investigate the performance. The simulation setup is as follows. The data rate on each wavelength channel is set as 1 Gb/s. A finite-time horizon with the time duration of 2 s is chosen. We assume the traffic of an ONU arrives in bursts. The burst size obeys Pareto distribution with the Pareto index $\alpha = 1.4$ and the mean equal to 31.25 kB, which takes about 0.25 ms to transmit. The burst interarrival time also obeys the Pareto distribution with $\alpha = 1.4$ and the mean equal to x. We vary x to obtain different traffic loads. The traffic load is defined as the ratio between the total size of the arrival bursts and the maximum value that can be accommodated, which is $4 * 1$ Gb/s $* 2$ s $= 8$ Gb.

We consider the offline scheduling framework, in which the OLT performs DBA after receiving requests from all ONUs. Since naive preemptive scheduling and naive nonpreemptive scheduling perform much worse than heuristic preemptive scheduling and heuristic nonpreemptive scheduling, respectively, we only investigate performances of heuristic preemptive scheduling and heuristic nonpreemptive scheduling in the multiple-cycle case.

1) Cycle Duration versus Tuning Time: The cycle duration is adaptive to the incoming traffic. Therefore, the cycle duration varies over time. Different scheduling schemes produce different cycle durations, thus resulting in different total number of cycles during the 2 s.

Fig. 5(a) shows the cycle duration of heuristic preemptive scheduling under the condition that the traffic load equals to 0.87. When the laser tuning time equals 0.5 ms, the cycle duration is around 0.75 ms, and the total number of cycles in the 2-s time period is around 2388. With the increase of the laser tuning time, the cycle duration increases, and the total number of cycles decreases. When the laser tuning time equals 2 ms,

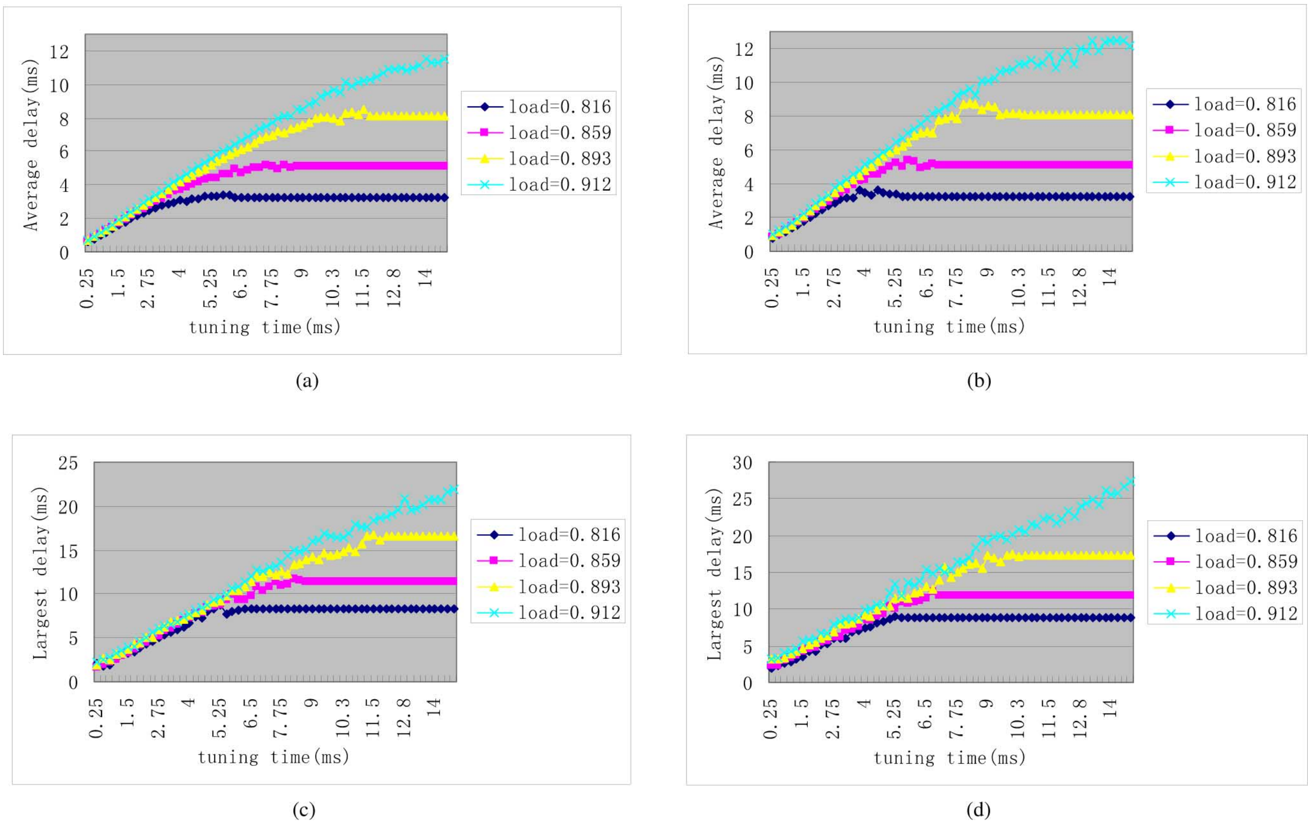


Fig. 6. Average delay and the largest delay versus laser tuning time ($n = 16,\ m = 4$). (a) Preemptive scheduling. (b) Nonpreemptive scheduling. (c) Preemptive scheduling. (d) Nonpreemptive scheduling.

most of the cycles last between 1.75 and 2.75 ms, and the total number of cycles in the 2-s time period is decreased to around 869. The cycle durations produced by heuristic nonpreemptive scheduling are shown in Fig. 5(b). As compared to Fig. 5(a), the cycle durations produced by nonpreemptive scheduling algorithm in each case of laser tuning time increase slightly.

Fig. 5(c) and (d) shows the cycle durations when the traffic load increases to 0.983. In comparing Fig. 5(c) and (a), the cycle duration greatly increases although there is only 0.113 increase of the traffic load. The cycle durations as shown in Fig. 5(c) fluctuate a lot when the load is 0.983. When the laser tuning time equals 0.5 ms, the cycle duration varies between 0.5 and 3.75 ms. Although the cycle duration fluctuates over time, its average value remains almost constant over time. This implies that heuristic preemptive scheduling can achieve throughput as high as 0.983. Fig. 5(d) shows the cycle durations produced by heuristic nonpreemptive scheduling. The cycle duration fluctuation is more severe than that produced by heuristic preemptive scheduling. Also, cycle durations are much greater than those produced by heuristic preemptive scheduling. Moreover, the average values of the cycle durations keep increasing over time, implying that heuristic nonpreemptive scheduling is not able to admit traffic load as high as 0.983. Therefore, when the network is heavily loaded, heuristic preemptive scheduling can achieve significant better performance than heuristic nonpreemptive scheduling. However, when the network is lightly loaded, heuristic nonpreemptive scheduling yields similar performance as heuristic preemptive scheduling.

Fig. 6 shows the average delay and the largest delay performance of the heuristic preemptive scheduling and the heuristic nonpreemptive scheduling. In offline scheduling of WDM PON, the arrival traffic in the current cycle will be transmitted in the next cycle. Therefore, the average delay should be around the average cycle duration, and the largest delay should be around two times that of the largest cycle duration. The largest delay as shown in Fig. 6(c) is around twice of the average delay as shown in Fig. 6(a); it agrees with the analysis. In addition, for a given traffic load, delay generally increases with the increase of the laser tuning time. This is due to the fact that large laser tuning time results in large cycle duration, which further introduces large delay. When the laser tuning time increases to a certain value that the laser tunability cannot help improve the performance, delay will become constant.

Since heuristic preemptive scheduling produces a shorter cycle duration than heuristic nonpreemptive scheduling, delay performance of heuristic preemptive scheduling is better than that of heuristic nonpreemptive scheduling. However, the outperformance of heuristic preemptive scheduling over heuristic nonpreemptive scheduling is not obvious in four cases of traffic loads. When the laser tuning time is too large that tunability cannot help improve the system performance, delay performance only depends on the incoming traffic profile, and thus both heuristic preemptive scheduling and heuristic nonpreemptive scheduling achieve the same performance. However, when the traffic load is too large (such as 0.983), preemptive scheduling may still be able to achieve relatively stable delay performance, while nonpreemptive scheduling may result in the phenomenon that delay keeps increasing over time.

VIII. Conclusion

In this paper, we have investigated the dynamic wavelength assignment and time allocation problem in a single resource allocation cycle in hybrid WDM/TDM PONs with tunable lasers as optical light generators. We map the scheduling problem into a multiprocessor scheduling problem, with wavelength channels as machines and ONU requests as jobs. Owing to the laser tuning time, jobs in this particular problem possess the property that sufficient guard time should be given when scheduling one job on two machines. We have shown that both preemptive and nonpreemptive scheduling problems with the objective of minimizing the schedule length are NP-hard when the laser tuning time is nonzero. Thus, we have proposed heuristic scheduling schemes for the case of arbitrary laser tuning time. Theoretical analyses show that the approximation ratios of the heuristic preemptive scheduling algorithm and the heuristic nonpreemptive scheduling algorithm are at most 2 and $2 - 1/m$, respectively, where m is the number of wavelengths. Simulation results show that our proposed heuristic scheduling algorithms achieve significantly better performances than naive algorithms that are directly derived from existing algorithms. It is also shown that the preemptive scheduling algorithm achieves slightly better performances than the nonpreemptive scheduling algorithm when the network is lightly loaded. However, the preemptive scheduling algorithm performs significantly better when the traffic load exceeds some value.

References

[1] F. Effenberger, D. Clearly, O. Haran, G. Kramer, R. D. Li, M. Oron, and T. Pfeiffer, "An introduction to PON technologies," *IEEE Commun. Mag.*, vol. 45, no. 3, pp. S17–S25, Mar. 2007.

[2] L. Hutcheson, "FTTx: Current status and the future," *IEEE Commun. Mag.*, vol. 46, no. 7, pp. 90–95, Jul. 2008.

[3] F.-T. An, D. Gutierrez, K. S. Kim, J. W. Lee, and L. Kazovsky, "SUCCESS-HPON: A next-generation optical access architecture for smooth migration from TDM-PON to WDM-PON," *IEEE Commun. Mag.*, vol. 43, no. 11, pp. S40–S47, Nov. 2005.

[4] K. Grobe and J.-P. Elbers, "PON in adolescence: From TDMA to WDM-PON," *IEEE Commun. Mag.*, vol. 46, no. 1, pp. 26–34, Jan. 2008.

[5] F. Ponzini, F. Cavaliere, G. Berrettini, M. Presi, E. Ciaramella, N. Calabretta, and A. Bogoni, "Evolution scenario toward WDM-PON," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 1, no. 4, pp. C25–C34, Sep. 2009.

[6] C.-H. Lee, S.-M. Lee, K.-M. Choi, J.-H. Moon, S.-G. Mun, K.-T. Jeong, J. H. Kim, and B. Kim, "WDM-PON experiences in korea," *J. Opt. Netw.*, vol. 6, no. 5, pp. 451–464, 2007.

[7] S. Oh, J. Shin, K. Kim, D. Lee, S. Park, H. Sung, Y. Baek, and K. Oh, "200 GHz-spacing 8-channel multi-wavelength lasers for WDM-PON optical line terminal sources," *Opt. Express*, vol. 17, no. 11, pp. 9401–9407, 2009.

[8] M. Yoshino, N. Miki, N. Yoshimoto, and K. Kumozaki, "Multiwavelength optical source for OCDM using sinusoidally modulated laser diode," *J. Lightw. Technol.*, vol. 27, pp. 4524–4529, 2009.

[9] Y. Luo and N. Ansari, "Bandwidth allocation for multiservice access on EPONs," *IEEE Commun. Mag.*, vol. 43, no. 2, pp. S16–S21, Feb. 2005.

[10] J. Angelopoulos, H.-C. Leligou, T. Argyriou, S. Zontos, E. Ringoot, and T. Van Caenegem, "Efficient transport of packets with QoS in an FSAN-aligned GPON," *IEEE Commun. Mag.*, vol. 42, no. 2, pp. 92–98, Feb. 2004.

[11] J. Zhang, N. Ansari, Y. Luo, F. Effenberger, and F. Ye, "Next-generation PONs: A performance investigation of candidate architectures for next-generation access stage 1," *IEEE Commun. Mag.*, vol. 47, no. 8, pp. 49–57, Aug. 2009.

[12] L. Coldren, G. Fish, Y. Akulova, J. Barton, L. Johansson, and C. Coldren, "Tunable semiconductor lasers: A tutorial," *J. Lightw. Technol.*, vol. 22, no. 1, pp. 193–202, Jan. 2004.

[13] J. Buus and E. J. Murphy, "Tunable lasers in optical networks," *J. Lightw. Technol.*, vol. 24, no. 1, pp. 5–11, Jan. 2006.

[14] J. Zhang and N. Ansari, "On analyzing the capacity of WDM PONs," in *Proc. IEEE GLOBECOM*, Honolulu, HI, 2009, pp. 1–5.

[15] J. Zhang and N. Ansari, "On the capacity of WDM passive optical networks," *IEEE Trans. Commun.*, vol. 59, no. 1, Jan. 2011, to be published.

[16] C. Bock, J. Prat, and S. Walker, "Hybrid WDM/TDM PON using the AWG FSR and featuring centralized light generation and dynamic bandwidth allocation," *J. Lightw. Technol.*, vol. 23, no. 12, pp. 3981–3988, Dec. 2005.

[17] G. Das, B. Lannoo, H.-D. Jung, T. Koonen, D. Colle, M. Pickavet, and P. Demeester, "A new architecture and MAC protocol for fully flexible hybrid WDM/TDM PON," in *Proc. 35th ECOC*, Sep. 2009, pp. 1–2.

[18] J. Zhang and N. Ansari, "Design of WDM PON with tunable lasers: The upstream scenario," *J. Lightw. Technol.*, vol. 28, no. 2, pp. 228–236, Jan. 15, 2010.

[19] K. H. Kwong, D. Harle, and I. Andonovic, "Dynamic bandwidth allocation algorithm for differentiated services over WDM EPONs," in *Proc. 9th Int. Conf. Commun. Syst.*, Sep. 2004, pp. 116–120.

[20] M. McGarry, M. Reisslein, and M. Maier, "WDM Ethernet passive optical networks," *IEEE Commun. Mag.*, vol. 44, no. 2, pp. 15–22, Feb. 2006.

[21] A. R. Dhaini, C. M. Assi, M. Maier, and A. Shami, "Dynamic wavelength and bandwidth allocation in hybrid TDM/WDM EPON networks," *J. Lightw. Technol.*, vol. 25, no. 1, pp. 277–286, Jan. 2007.

[22] M. P. McGarry, M. Reisslein, C. J. Colbourn, M. Maier, F. Aurzada, and M. Scheutzow, "Just-in-time scheduling for multichannel EPONs," *J. Lightw. Technol.*, vol. 26, no. 10, pp. 1204–1216, May 15, 2008.

[23] M. P. McGarry, M. Reisslein, M. Maier, and A. Keha, "Bandwidth management for WDM EPONs," *J. Opt. Netw.*, vol. 5, no. 9, pp. 637–654, 2006.

[24] L. Meng, J. El-Najjar, H. Alazemi, and C. Assi, "A joint transmission grant scheduling and wavelength assignment in multichannel SG-EPON," *J. Lightw. Technol.*, vol. 27, no. 21, pp. 4781–4792, Nov. 1, 2009.

[25] G. Pieris and G. Sasaki, "Scheduling transmissions in WDM broadcast-and-select networks," *IEEE/ACM Trans. Netw.*, vol. 2, no. 2, pp. 105–110, Apr. 1994.

[26] A. Bianco, M. Guido, E. Leonardi, and D. di Elettronica, "Incremental scheduling algorithms for WDM/TDM networks with arbitrary tuning latencies," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 464–475, Mar. 2003.

[27] M. Borella and B. Mukherjee, "Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 5, pp. 923–934, Jun. 1996.

[28] G. Rouskas and V. Sivaraman, "Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies," *IEEE/ACM Trans. Netw.*, vol. 5, no. 3, p. 370, Jun. 1997.

[29] Z. Ortiz, G. Rouskas, and H. Perros, "Scheduling of multicast traffic in tunable-receiver WDM networks with non-negligible tuning latencies," *Comput. Commun. Rev.*, vol. 27, no. 4, pp. 301–310, 1997.

[30] G. Rouskas and V. Sivaraman, "On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies," in *Proc. IEEE INFOCOM*, 1996, vol. 3, pp. 1217–1224.

[31] I. Baldine and G. Rouskas, "Dynamic load balancing in broadcast WDM networks with tuning latencies," in *Proc. IEEE INFOCOM*, 1998, pp. 78–85.

[32] E. Hou, N. Ansari, and H. Ren, "A genetic algorithm for multiprocessor scheduling," *IEEE Trans. Parallel Distrib. Syst.*, vol. 5, no. 2, pp. 113–120, Feb. 1994.

[33] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Upper Saddle River, NJ: Prentice-Hall, 2002.

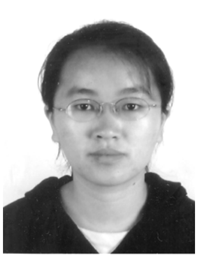
[34] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[35] D. Friesen and M. Langston, "Evaluation of a MULTIFIT-based scheduling algorithm," *J. Algor.*, vol. 7, no. 1, pp. 35–59, 1986.

[36] C. Lee, "Parallel machines scheduling with nonsimultaneous machine available time," *Discrete Appl.Math.*, vol. 30, no. 1, pp. 53–61, 1991.

[37] G. Lin, Y. He, Y. Yao, and H. Lu, "Exact bounds of the modified LPT algorithm applying to parallel machines scheduling with nonsimultaneous machine available times," *Appl. Math. J. Chinese Univ. B*, vol. 12, no. 1, pp. 109–116, 1997.

[38] S. Chang and H. Hwang, "The worst-case analysis of the MULTIFIT algorithm for scheduling nonsimultaneous parallel machines," *Discrete Appl. Math.*, vol. 92, no. 2–3, pp. 135–147, 1999.



Jingjing Zhang (S'09) received the B.E. degree in electrical engineering from Xi'an Institute of Posts and Telecommunications, Xi'an, China, in 2003, the M.E. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006, and is currently pursuing the Ph.D. degree in electrical engineering at the New Jersey Institute of Technology (NJIT), Newark.

Her research interests include planning, capacity analysis, and resource allocation of broadband access networks, QoE provisioning in next-generation networks, and energy-efficient networking.

Ms. Zhang received a 2010 New Jersey Inventors Hall of Fame Graduate Student Award.

Nirwan Ansari (S'78–M'83–SM'94–F'09) received the B.S.E.E. (*summa cum laude* with a perfect GPA) from the New Jersey Institute of Technology (NJIT), Newark, in 1982, the M.S.E.E. degree from University of Michigan, Ann Arbor, in 1983, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988.

He joined NJIT's Department of Electrical and Computer Engineering as Assistant Professor in 1988, became a tenured Associate Professor in 1993, and has been a Full Professor since 1997. He has also assumed various administrative positions at NJIT. He was Visiting (Chair) Professor at several universities. He authored *Computational Intelligence for Optimization* (Springer, 1997, translated into Chinese in 2000) with E. S. H. Hou and edited *Neural Networks in Telecommunications* (Springer, 1994) with B. Yuhas. He has also contributed over 350 technical papers, over one third of which were published in widely cited refereed journals/magazines. For example, one of his works was the sixth most cited article published in the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, according to the journal EIC report in February 2010. He has also guest-edited a number of special issues, covering various emerging topics in communications and networking. His current research focuses on various aspects of broadband networks and multimedia communications.

Prof. Ansari has served on the Editorial Board and Advisory Board of eight journals, including as a Senior Technical Editor of the *IEEE Communications Magazine* (2006–2009). He has served the IEEE in various capacities such as Chair of the IEEE North Jersey Communications Society (COMSOC) Chapter, Chair of the IEEE North Jersey Section, Member of the IEEE Region 1 Board of Governors, Chair of the IEEE COMSOC Networking Technical Committee Cluster, Chair of the IEEE COMSOC Technical Committee on Ad Hoc and Sensor Networks, and Chair/Technical Program Committee Chair of several conferences/symposia. Some of his recent recognitions include a 2007 IEEE Leadership Award from the Central Jersey/Princeton Section, NJIT Excellence in Teaching in Outstanding Professional Development in 2008, a 2008 IEEE MGA Leadership Award, the 2009 NCE Excellence in Teaching Award, a couple of best paper awards (IC-NIDC 2009 and IEEE GLOBECOM 2010), a 2010 Thomas Alva Edison Patent Award, and designation as an IEEE Communications Society Distinguished Lecturer (2006–2009, two terms).