Scheduling Transmissions in WDM Optical Networks

Bhaskar DasGupta
Department of Computer Science
Rutgers University
Camden, NJ 08102, USA

Michael A. Palis
Department of Computer Science
Rutgers University
Camden, NJ 08102, USA

Abstract *This paper addresses the problem of scheduling packet transmissions in wavelength-division multiplexed (WDM) networks with tunable transmitters and fixed-tuned receivers. Unlike previous work which assume that all packets are known in advance, this paper considers the* on-line *case in which packets may arrive at any time. An on-line algorithm is presented that achieves a performance ratio of* 3 *with respect to an optimal off-line algorithm. In addition, off-line algorithms are presented for the case when there are two wavelength channels. Even this special case of the problem is known to be NP-complete and the currently best known algorithm for this case achieves a performance ratio of* 2*. Using a more rigorous analysis, it is shown that this algorithm has, in fact, a performance ratio of* $\frac{3}{2}$*, and an example is presented where this algorithm achieves this performance ratio even when the tuning delay is zero. Furthermore, for this case, a new polynomial-time approximation algorithm is presented with a performance ratio better than* $\frac{3}{2}$*, provided the tuning delay* δ *is less than* $\left(\frac{3}{2}-\alpha\right)\frac{S}{6}$*, where* S *is the total number of packets to be transmitted and* $\alpha \approx 1.4142136$*.*

Keywords: Wavelength-division multiplexed networks, Scheduling, Online algorithms, Approximation Algorithms, Performance Ratios.

1 Introduction

Wavelength division multiplexing is a promising approach to utilize the enormous bandwidth of optical fiber and offers the capability of building very large wide-area networks consisting of thousands of nodes with per-node throughputs in the gigabits-per-second range.

In a wavelength division multiplexed (WDM) optical network, n transmitters and r receivers communicate through m non-interfering wavelength channels. In practice, m is typically much less than either n or r and hence the channels are shared by the transmitters and the receivers. Transmitters and receivers that can tune from one wavelength to another are called *tunable*, while those that cannot are called *fixed-tuned*. The network is packet switched and time slotted. That is, transmitters transmit data in fixed-length packets and a packet's transmission time equals one time slot. Packets are transmitted within slot boundaries.

An important parameter in the design of WDM optical networks is the *tuning delay*, which is the amount of time required for a transmitter to tune from one wavelength to another. Current WDM networks have large tuning delays, sometimes in the order of milliseconds for transmitters and receivers with wide tuning ranges [4]. Consequently, algorithms for scheduling packet transmissions in WDM networks must explicitly take into account the effect of tuning delay on performance.

The problem of scheduling transmissions in WDM networks has been studied by various researchers [11, 1, 2, 3, 4, 8, 9, 10]. In this paper, we are interested in the scheduling problem for WDM networks with *tunable transmitters* and *fixed-tuned receivers*. This model has previously been studied in [9] and [4]. In [9], Pieris and Sasaki considered the *all-to-all broadcast* problem (i.e., a single packet is to be transferred between every transmitter/receiver pair) and presented upper and lower bounds on the minimum-length schedule for this problem. Subsequently, Choi, Choi and Azizoğlu [4] improved upon [9]'s lower bound and showed that the latter's all-to-all broadcast algorithm is, in fact, optimal. In the same paper [4], the authors considered the general problem in which arbitrary (but known) number of packets are to be transferred between transmitter/receiver pairs. They presented an algorithm based on the well-known *list scheduling* algorithm [5, 7] which produces schedule lengths that are at most twice the optimal length.

In this paper, we consider the *on-line* version of the general transmission scheduling problem, which applies to more practical situations that does the off-line version. In on-line scheduling, packets arrive at the transmitters at arbitrary times; consequently, scheduling decisions must be made on the basis of the packets that have arrived so far, without knowledge of future packets. We show that this problem, while more difficult than the off-line case, admits efficient solutions as well. In particular, we give an on-line algorithm that produces schedule lengths that are at most three times the optimal length. Interestingly, our on-line algorithm reduces to the off-line list scheduling algorithm when all packets are known in advance (i.e., arrive at time 0).

For the off-line case, the interesting question is whether the performance ratio of 2 achieved by the list scheduling algorithm of [4] is the best possible. To gain further insight into this problem, we consider the special case when there are only two wavelength channels. Even this special case of the transmission scheduling problem is known to be NP-complete [11].

For the two-channel case, a more rigorous analysis shows that the list scheduling algorithm actually has a performance ratio of $\frac{3}{2}$. We also show that this ratio is tight even when the tuning delay is zero. This leads to the question of whether $\frac{3}{2}$ is the best ratio achievable by any off-line algorithm. We answer this question in the negative by presenting a polynomial-time approximation algorithm that achieves a performance ratio better than $\frac{3}{2}$, provided the tuning delay δ is less than $\left(\frac{3}{2} - \alpha\right)\frac{S}{6}$, where S is the total number of packets to be transmitted and $\alpha \approx 1.4142136$. This result opens up the possibility of even better performing off-line algorithms not only for the two-channel case, but for the general case as well.

2 The On-Line Algorithm

An instance of the on-line transmission scheduling problem consists of n tunable transmitters T_i $(1 \le i \le n)$, r fixed-tuned receivers R_i $(1 \le i \le r)$ and m wavelength channels C_i $(1 \le i \le m)$. Each receiver R_i is tuned permanently to a specific channel C_j; hence, all packets destined for R_i must be transmitted over channel C_j. On the other hand, each transmitter T_i may tune to, and transmit packets over, any channel. However, at any given time, a transmitter may transmit over at most one channel and a channel may carry at most one packet. All packets have the same length and a packet's transmission time equals one time unit. When a transmitter tunes to a channel, it incurs a tuning delay equal to δ time units. Initially, the transmitters are not tuned to any specific channel.

Packets arriving at a transmitter T_i are placed in a queue Q_i. For notational convenience, we denote by $Q_i[j]$ the set of packets in Q_i that are to be transmitted over channel C_j. T_i also maintains a ready queue $READY_i$ of packets already scheduled for transmission.

We now present the on-line algorithm. The algorithm maintains an array F of m elements, one for each channel $C_j, 1 \le j \le m$. $F[j] = t$ means that channel C_j will become free (i.e., no packet transmission is scheduled) after t time units (relative to current time). $F[j]$ is decremented by one after each time unit. Initially, $F[j] = 0$ for all j.

Each transmitter goes through a sequence of *transmit cycles*; during each cycle the transmitter tunes to a channel, waits (if necessary) until the channel becomes free, then sends one or more packets over the channel. Specifically, each transmitter T_i cycles through the steps given in Algorithm A.

Before analyzing the performance of the above on-line algorithm, we first derive some useful properties of an optimal schedule and the schedule produced by Algorithm A. Let:

- $p(T_i)$ = total number of packets to be transmitted by transmitter T_i,
- $p(C_i)$ = total number of packets to be transmitted over channel C_i, and
- $c(T_i)$ = number of distinct channels over which the packets of T_i have to be transmitted.

Let L_{OPT} be the length of an optimal schedule. The following facts are obvious:

Fact 2.1 $L_{OPT} \geq \max_{1 \leq i \leq n}\{\, p(T_i) + \delta c(T_i)\,\}$.

Fact 2.2 $L_{OPT} \geq \max_{1 \leq i \leq m}\{\, p(C_i) + \delta\,\}$.

Let L be the length of the schedule produced by Algorithm A. Let T be the transmitter which completed transmission at time L. Suppose that T goes through a sequence of l transmit cycles $\langle \Gamma_1, \Gamma_2, \ldots, \Gamma_l \rangle$. Suppose further that during the last transmit cycle Γ_l, T transmitted packets over channel C. Let ρ be the packet with the earliest arrival time among all packets transmitted during Γ_l. Let i be the largest integer such that the arrival time of $\rho \geq$ start time of Γ_i.

Fact 2.3 *For any two consecutive transmit cycles Γ_j and Γ_{j+1} in $\langle \Gamma_i, \ldots, \Gamma_l \rangle$, there is no idle*[1] *period between the end of Γ_j and the start of Γ_{j+1}.*

Proof: From Algorithm A, it is clear that once a transmitter has sent all packets over a channel, it immediately tunes to a new channel (and hence begins the next transmit cycle) whenever there are packets still waiting to be sent. Since packet ρ arrived during Γ_i and was not transmitted till Γ_l, T always had at least one packet to send at the completion of every transmit cycle $\Gamma_j, i \leq j \leq l$. The fact follows. ∎

[1] A transmitter is *busy* if it is either tuning to a channel or transmitting a packet; otherwise, it is *idle*.

Fact 2.3 implies that the idle periods of T occur only within transmit cycles; specifically, only when T has finished tuning to a channel but is forced to wait until the channel becomes free before transmitting any packets.

Fact 2.4 *At any time during $\langle \Gamma_i, \ldots, \Gamma_l \rangle$, channel C is busy whenever transmitter T is idle.*

Proof: Note that transmitter T has at least one packet to send (i.e., packet ρ) over channel C during $\langle \Gamma_i, \ldots, \Gamma_l \rangle$. Suppose to the contrary that during some transmit cycle $\Gamma_j, i \leq j \leq l$, T remained idle when channel C became free. If T were tuned to C, then it should have started transmitting as soon as C became free and not remained idle. If T were tuned to another channel D, then it should have instead tuned to C because C would be available earlier than D. In either case, we have a contradiction. ∎

The following theorem shows the performance ratio of the above algorithm.

Theorem 2.1 *Algorithm A produces a schedule of length $L \leq 3L_{OPT}$, where L_{OPT} is the length of an optimal schedule.*

Proof: During $\langle \Gamma_i, \ldots, \Gamma_l \rangle$, either:

(1) all cycles transmit over distinct channels; or

(2) two or more cycles transmit over the same channel.

Case 1. Consider first the case when all transmit cycles in $\langle \Gamma_i, \ldots, \Gamma_l \rangle$ use distinct channels. Let:

- t_1 = arrival time of packet ρ at transmitter T;
- t_2 = sum of all idle periods of T during $\langle \Gamma_i, \ldots, \Gamma_l \rangle$; and

- t_3 = sum of all busy periods of T during $\langle \Gamma_i, \ldots, \Gamma_l \rangle$.

Clearly, the finish time L of transmitter T satisfies:

$$L \leq t_1 + t_2 + t_3$$

Since packet ρ arrived at time t_1, any schedule must finish no earlier than t_1. Hence,

$$t_1 \leq L_{OPT}$$

Recall from Fact 2.4 that whenever T is idle, channel C is busy. Using this fact and Fact 2.2, we have,

$$t_2 \leq p(C) \leq L_{OPT} - \delta$$

Finally, since T transmits over distinct channels during $\langle \Gamma_i, \ldots, \Gamma_l \rangle$, then

$$t_3 \leq p(T) + \delta c(T) \leq L_{OPT},$$

where we used Fact 2.1 for the second inequality.

It follow that:

$$L \leq t_1 + t_2 + t_3 \leq 3L_{OPT} - \delta \leq 3L_{OPT}$$

Case 2. Suppose that in $\langle \Gamma_i, \ldots, \Gamma_l \rangle$, two or more transmit cycles used the same channel. Find the largest integer $j, i \leq j \leq l$, such that:

- no transmit cycles in $\langle \Gamma_{j+1}, \ldots, \Gamma_l \rangle$ used the same channel; and
- Γ_j used the same channel C' as some transmit cycle Γ_k in $\langle \Gamma_{j+1}, \ldots, \Gamma_l \rangle$.

Let ρ' be the packet with the earliest arrival time among all packets transmitted by T during Γ_k. Furthermore, let:

- t'_1 = arrival time of packet ρ' at T;
- t'_2 = sum of all idle periods of T during $\langle \Gamma_j, \ldots, \Gamma_l \rangle$; and
- t'_3 = sum of all busy periods of T during $\langle \Gamma_j, \ldots, \Gamma_l \rangle$.

Clearly, packet ρ' should have arrived no earlier than the start of transmit cycle Γ_j, since otherwise ρ' would have been transmitted during Γ_j and not during Γ_k. Thus, the finish time L of T satisfies:

$$L \leq t'_1 + t'_2 + t'_3$$

Moreover,

$$t'_1 \leq L_{OPT}$$

and

$$t'_2 \leq p(C) \leq L_{OPT} - \delta$$

Note that during $\langle \Gamma_j, \ldots, \Gamma_l \rangle$ no channel was used more than once except for channel C'. Therefore,

$$t'_3 \leq p(T) + (\delta + 1) \cdot c(T) \leq L_{OPT} + \delta,$$

by Fact 2.1. It follows that:

$$L \leq t'_1 + t'_2 + t'_3 \leq 3L_{OPT} \quad \blacksquare$$

3 Off-Line Scheduling: Better Polynomial-Time Approximation Algorithms for the Two-Channel Case

When all packets to be transmitted are known in advance (i.e., all packets arrive at time 0), the on-line algorithm described in the previous section reduces to the off-line *list scheduling* algorithm described in [4]. In [4] it was shown that this algorithm produces schedules which are within a factor 2 of the optimal schedule. We should point out that the alternative algorithms given in [4] (viz., Theorem 3 and its corollaries) are *not* polynomial-time approximation algorithms[2] and hence could not be used to get a polynomial-time approximation with a ratio better than 2.

[2] This is because the time taken by these algorithms is proportional to the size s of the largest packet. But, only $\lceil \log_2(s+1) \rceil$ bits are needed to encode s. In other words, all these algorithms run in *pseudo-polynomial* time (see, for example, [6, pages 387-391] for a discussion on pseudo-polynomial time algorithms).

We attempt to provide further insight into the off-line scheduling problem by considering the special case when there are only two channels. Even this special case of the problem is known to be NP-complete [11]. For this special case, a more rigorous analysis shows that the list scheduling algorithm actually has a better performance ratio of $\frac{3}{2}$. We also show that this ratio is tight by demonstrating a problem instance (with even zero tuning delay) for which the algorithm achieves exactly this ratio. This leads to the interesting question of whether $\frac{3}{2}$ is the best ratio achievable by any polynomial-time off-line algorithm. We partially answer this question by exhibiting an algorithm that achieves a performance ratio better than $\frac{3}{2}$, provided the tuning delay δ is less than $\left(\frac{3}{2}-\alpha\right)\frac{S}{6}$, where S is the total number of packets to be transmitted and $\alpha \approx 1.4142136$.

3.1 A $\frac{3}{2}$ Performance Bound for Two-Channel List Scheduling

For the case of two channels, we can obtain an improved performance ratio for the list scheduling algorithm.

Theorem 3.1 *The off-line list scheduling algorithm achieves a performance ratio of $\frac{3}{2}$ when there are 2 channels. Moreover, this ratio is tight.*

3.2 Breaking the $\frac{3}{2}$ Barrier

Intuitively, in order to improve upon the $\frac{3}{2}$ ratio, we need to ensure that the transmission schedules over the two channels are *balanced* in a better way. As before, assume that transmitter T_i, $1 \leq i \leq n$, has a_i and b_i packets to transmit over channels C_1 and C_2, respectively. Let $S_1 = \sum_{i=1}^{n} a_i$, $S_2 = \sum_{i=1}^{n} b_i$, and assume, without loss of generality, that $S_2 \geq S_1 > 0$. Obviously, $L_{OPT} \geq \max\{\ \delta + S_2,\ \max_{1\leq i\leq n}(2\delta + a_i + b_i)\ \}$. Also, since every transmitter has at least one packet to send, $S_2 \geq \frac{n}{2}$. The scheduling algorithm is given as Algorithm B.

Theorem 3.2 *Algorithm B runs in polynomial time and achieves a performance ratio of $r < \frac{3}{2}$, provided the tuning delay δ satisfies $\delta < \left(\frac{3}{2}-\alpha\right)\frac{S}{6}$, where $\alpha = \frac{1}{\frac{2}{3}+\epsilon_1}$ (Notice that $\alpha \approx 1.4142136$)*

Proof: Assume that $\delta < \left(\frac{3}{2}-\alpha\right)\frac{S}{6}$. Since $S_2 \geq \frac{S}{2}$, $\delta < \left(\frac{3}{2}-\alpha\right)\frac{S_2}{3}$. For notational simplicity, let $c = \frac{1}{\frac{3}{2}-\alpha} > 1$. Hence, $S_2 > 3c\delta$. First, notice that it is always the case that $\Sigma_1 \cap \Sigma_1' = \Sigma_2 \cap \Sigma_2' = \phi$; hence during the first (respectively, second) round, transmitters from Σ_1 (respectively, Σ_2) do not compete with the transmitters from Σ_1' (respectively, Σ_2') for the same channel. Also, notice that $\Sigma_1 \cup \Sigma_2 = \Sigma_1' \cup \Sigma_2' = \{T_1, T_2, \ldots, T_n\}$; hence at the end of the algorithm, all transmitters finish their transmissions. Finally, due to the choice of the particular value of the constant ϵ_1, it is true that $\frac{4}{3} + 2\epsilon_1 = \frac{1}{\frac{2}{3}+\epsilon_1}$ (ϵ_1 is the positive root of the quadratic equation $18\epsilon_1^2 + 24\epsilon_1 - 1 = 0$).

If the algorithm found some i such that $(a_i + b_i) \geq (\frac{2}{3}+\epsilon_1)(S_1+S_2)$, then $L_{OPT} \geq 2\delta + (\frac{2}{3}+\epsilon_1)(S_1+S_2)$, whereas the schedule length L of Algorithm B is $L = 2\delta + S_1 + S_2$. Hence,

$$\begin{aligned}
r &= \frac{L}{L_{OPT}} \\
&\leq \frac{S_1+S_2}{2\delta+(\frac{2}{3}+\epsilon_1)(S_1+S_2)} + \frac{2\delta}{2\delta+(\frac{2}{3}+\epsilon_1)(S_1+S_2)} \\
&< \frac{1}{\frac{2}{3}+\epsilon_1} + \frac{1}{1+(\frac{2}{3}+\epsilon_1)\frac{S_2}{2\delta}} \\
&< \frac{1}{\frac{2}{3}+\epsilon_1} + \frac{1}{1+c} \\
&< \frac{1}{\frac{2}{3}+\epsilon_1} + \frac{1}{c} \\
&= \alpha + \frac{3}{2} - \alpha \\
&= \frac{3}{2}
\end{aligned}$$

as desired.

Otherwise, $(a_i + b_i) < (\frac{2}{3}+\epsilon_1)(S_1+S_2)$ for every i. Algorithm B now ensures that $\Sigma_1, \Sigma_2, \Sigma_1', \Sigma_2' \neq \phi$. Define

$$\sigma_1 = \delta + \sum_{T_j \in \Sigma_1} a_j \quad \sigma_1' = \delta + \sum_{T_j \in \Sigma_1'} b_j$$

$$\sigma_2 = \delta + \sum_{T_j \in \Sigma_2} a_j \quad \sigma_2' = \delta + \sum_{T_j \in \Sigma_2'} b_j$$

Notice that $\sigma_1 + \sigma_2 = 2\delta + S_1$ and $\sigma_1' + \sigma_2' = 2\delta + S_2$. Let $t = |\sigma_1 - \sigma_1'|$. Depending on

Step 1. Select a j such that $Q_i[j] \neq \emptyset$ and channel C_j has the earliest available time (i.e., $F[j]$ is minimum).

Step 2. Move the packets in $Q_i[j]$ to the ready queue $READY_i$.

Step 3. If already tuned to channel C_j, then update $F[j] = |READY_i|$ and transmit all packets in $READY_i$ over channel C_j. Go to step 1.

Step 4. If not tuned to channel C_j, then do the following:
(a) Let $f = F[j]$ and $\tau = \max\{ F[j], \delta \}$. Update $F[j] = \tau + |READY_i|$.
(b) Tune to channel C_j (for δ time units).
(c) Wait $\max\{ f - \tau, 0 \}$ time units, then transmit all packets in $READY_i$ over channel C_j. Go to step 1.

Algorithm A

$\epsilon_1 = \frac{-24+\sqrt{648}}{36} \approx 0.040440115$, $\epsilon_2 = \frac{1}{6} - 2\epsilon_1 \approx 0.085786438$

if $\exists i$ such that $(a_i + b_i) \geq (\frac{2}{3} + \epsilon_1)(S_1 + S_2)$ **then**
  $\Sigma_1 = \Sigma_2' = \{T_1, T_2, \ldots, T_n\}$, $\Sigma_1' = \Sigma_2 = \phi$
else
  if $\exists i$ such that $|a_i + b_i - S_2| \leq (\frac{1}{2} - \epsilon_2)S_2$ **then**
    $\Sigma_1 = \{a_i\}$, $\Sigma_1' = \{b_j \mid j \neq i\}$, $\Sigma_2 = \{a_j \mid j \neq i\}$, $\Sigma_2' = \{b_i\}$
  else
    find k such that

$$S_2 - \sum_{1 \leq i \leq k-1} (a_i + b_i) > (\frac{1}{2} - \epsilon_2)S_2 \textbf{ and}$$

$$S_2 - \sum_{1 \leq i \leq k} (a_i + b_i) \leq (\frac{1}{2} - \epsilon_2)S_2$$

    (the proof will show that such a k exists)
    $\Sigma_1 = \Sigma_2' = \{T_1, T_2, T_3, \ldots, T_k\}$
    $\Sigma_2 = \Sigma_1' = \{T_{k+1}, T_{k+2}, T_{k+3}, \ldots, T_n\}$
  endif
endif
Transmit the packets in two rounds of transmission as follows:
During first round,
  Transmitters $T_j \in \Sigma_1$ transmit over channel C_1 one after another in any order.
  Transmitters $T_j \in \Sigma_1'$ transmit over channel C_2 one after another in any order.
All transmitters wait (if necessary) until both C_1 and C_2 are not busy.
During second round,
  Transmitters $T_j \in \Sigma_2$ transmit over channel C_1 one after another in any order.
  Transmitters $T_j \in \Sigma_2'$ transmit over channel C_2 one after another in any order.
endif

Algorithm B

the relative magnitudes of $\sigma_1, \sigma_1', \sigma_2, \sigma_2'$, there could be four possibilities:

(a) $\sigma_1 \le \sigma_1'$ **and** $\sigma_2 \le \sigma_2'$. Then, $L = S_2 + 2\delta$, $L_{OPT} \ge S_2 + \delta$ and hence

$$r = \frac{L}{L_{OPT}} \le 1 + \frac{\delta}{S_2+\delta} \le 1 + \frac{1}{1+3c} < \frac{5}{4}$$

(b) $\sigma_1 \le \sigma_1'$ **and** $\sigma_2 > \sigma_2'$. Since $S_1 \le S_2$,

$$\begin{aligned} \sigma_2 - \sigma_2' = & \ (S_1 + 2\delta - \sigma_1) - (S_2 + 2\delta - \sigma_1') \\ \le & \ \sigma_1' - \sigma_1 \end{aligned}$$

Hence,

$$L = S_2 + 2\delta + (\sigma_2 - \sigma_2') \le S_2 + 2\delta + t$$

Since $L_{OPT} \ge S_2 + \delta$, we have

$$r = \frac{L}{L_{OPT}} \le \frac{S_2 + 2\delta + t}{S_2 + \delta} < 1 + \frac{1}{c} + \frac{t}{S_2+\delta}$$

(c) $\sigma_1 > \sigma_1'$ **and** $\sigma_2 \le \sigma_2'$. Then, $L = S_2 + 2\delta + t$, $L_{OPT} \ge S_2 + \delta$, and hence again (similar to (b) above) $r = \frac{L}{L_{OPT}} < 1 + \frac{1}{c} + \frac{t}{S_2+\delta}$.

(d) $\sigma_1 > \sigma_1'$ **and** $\sigma_2 > \sigma_2'$. But, $\sigma_1 + \sigma_2 = S_1 + 2\delta \le S_2 + 2\delta = \sigma_1' + \sigma_2'$. Hence, this case is **not** possible.

So, combining all the items above, $r < \max\{\frac{5}{4}, 1 + \frac{t}{S_2} + \frac{1}{c}\}$. Our goal is to show that t is not too large. We have two major cases:

Case 1. The algorithm found some i such that $|a_i + b_i - S_2| \le (\frac{1}{2} - \epsilon_2)S_2$. Then, $t = |\sigma_1 - \sigma_1'| = |a_i - (S_2 - b_i)| \le (\frac{1}{2} - \epsilon_2)S_2$. Hence, $r < 1 + \frac{1}{2} - \epsilon_2 + \frac{1}{c} = \frac{3}{2} - \epsilon_2 + \frac{1}{c} = \frac{4}{3} + 2\epsilon_1 + \frac{1}{c} = \frac{1}{\frac{2}{3}+\epsilon_1} + \frac{1}{c} = \alpha + \frac{3}{2} - \alpha = \frac{3}{2}$.

Case 2. The algorithm found no such i as in Case 1. That is, for all i, $|a_i + b_i - S_2| > (\frac{1}{2} - \epsilon_2)S_2 > 0$.

First we show that, for all i, $a_i < S_2 - b_i$. Assume, for the sake of contradiction, that $a_i > S_2 - b_i$ for some i. This implies $a_i + b_i - S_2 > (\frac{1}{2} - \epsilon_2)S_2$. Hence, $a_i + b_i > (\frac{3}{2} - \epsilon_2)S_2 = (\frac{4}{3} + 2\epsilon_1)S_2$. But, we already have, $(a_i + b_i) < (\frac{2}{3} + \epsilon_1)(S_1 + S_2) \le (\frac{4}{3} + 2\epsilon_1)S_2$, since $S_1 \le S_2$. This is a contradiction.

Hence, for all i, $a_i < S_2 - b_i$. That means $S_2 - b_i - a_i > (\frac{1}{2} - \epsilon_2)S_2$. That is, $a_i + b_i < (\frac{1}{2} + \epsilon_2)S_2$. Algorithm B now tries to find an appropriate index k. The index k must exist, since

$$S_2 - \sum_{1 \le i \le 0} (a_i + b_i) = S_2 > (\frac{1}{2} - \epsilon_2)S_2,$$

$$S_2 - \sum_{1 \le i \le n} (a_i + b_i) = -S_1 < (\frac{1}{2} - \epsilon_2)S_2.$$

Hence, the index k can be found. Let

$$P = S_2 - \sum_{1 \le i \le k} (a_i + b_i) \le \left(\frac{1}{2} - \epsilon_2\right) S_2$$

and

$$P' = S_2 - \sum_{1 \le i \le k-1} (a_i + b_i) > \left(\frac{1}{2} - \epsilon_2\right) S_2$$

Then, $t = |P|$. How large $|P|$ can be? Notice that, since $a_k + b_k < (\frac{1}{2} + \epsilon_2)S_2$,

$$\begin{aligned} P &= P' - (a_k + b_k) \\ &> (\tfrac{1}{2} - \epsilon_2)S_2 - (\tfrac{1}{2} + \epsilon_2)S_2 = -2\epsilon_2 S_2 \end{aligned}$$

Hence, $-2\epsilon_2 S_2 < P \le (\frac{1}{2} - \epsilon_2)S_2$, and

$$\begin{aligned} t = |P| &\le \max\{\, 2\epsilon_2 S_2,\ (\tfrac{1}{2} - \epsilon_2)S_2 \,\} \\ &= (\tfrac{1}{2} - \epsilon_2)S_2 \end{aligned}$$

and, hence

$$\begin{aligned} r &< 1 + (\tfrac{1}{2} - \epsilon_2) + \tfrac{1}{c} \\ &= \tfrac{1}{\frac{2}{3}+\epsilon_1} + \tfrac{1}{c} \\ &= \alpha + \tfrac{3}{2} - \alpha \\ &= \tfrac{3}{2} \end{aligned}$$

Combining all cases, it is always true that $r < \frac{3}{2}$. ■

4 Conclusion

The results presented in this paper point to several interesting questions that still remain to be addressed:

- Is there an on-line transmission scheduling algorithm that achieves a performance ratio better than 3? (It is possible that a better analysis would show that the on-line algorithm presented here has a performance ratio less than 3.)

- Can the off-line algorithm for two channels be generalized to m channels and be shown to achieve a performance ratio better than $\frac{3}{2}$?

- Can efficient on-line and off-line algorithms be developed for the general case of tunable transmitters and tunable receivers?

Furthermore, to be of practical use, extensive simulations need to be carried out to test the algorithms under a variety of system configurations and traffic distribution patterns. We encourage other researchers to investigate these problems so as to gain better insight into the capabilities (and limitations) of WDM optical networks.

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan, "Efficient Routing and Scheduling Algorithms for Optical Networks", IBM Research Report, Tech. Rep. RC 18967, June 1993.

[2] M. Azizoğlu, R. Barry, and A. Mokhtar, "Impact of Tuning Delay on the Performance of Bandwidth-Limited Optical Broadcast Networks with Uniform Traffic", *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, June 1996, pp. 935–944.

[3] M. S. Borella and B. Mukherjee, "Efficient Scheduling of Nonuniform Packet Traffic in a WDM/TDM Local Lightwave Network with Arbitrary Transceiver Tuning Latencies", *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, June 1996, pp. 923–934.

[4] H. Choi, H.-A. Choi and M. Azizoğlu, "Efficient Scheduling of Transmissions in Optical Broadcast Networks", *IEEE/ACM Trans. Networking*, vol. 4, no. 6., Dec. 1996, pp. 913–920.

[5] E. G. Coffman and P. J. Denning, *Operating Systems Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1973.

[6] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall Inc., 1982.

[7] R. L. Graham, "Bounds for Certain Multiprocessor Anomalies", *Bell Sys. Tech. Journal*, 45, 1966, pp. 1563–1581.

[8] F. Jia, B. Mukerjee, and J. Iness, "Scheduling Variable-Length Messages in a Single-Hop Multichannel Local Lightwave Network", *IEEE/ACM Trans. Networking*, vol. 3, no. 4, Aug. 1995, pp. 477–487.

[9] G. R. Pieris and G. H. Sasaki, "Scheduling Transmissions in WDM Broadcast-and-Select Networks", *IEEE/ACM Trans. Networking*, vol. 2, no. 2, Apr. 1994, pp. 105–110.

[10] G. N. Rouskas and V. Sivaraman, "On the Design of Optimal TDM Schedules for Broadcast WDM Networks with Arbitrary Transceiver Tuning Latencies", *Proc. IEEE INFOCOM'96*, 1996, pp. 1217–1224.

[11] G. N. Rouskas and V. Sivaraman, "Packet Scheduling in Broadcast WDM Networks with Arbitrary Transceiver Tuning Latencies", *IEEE/ACM Trans. Networking*, vol. 5, no. 3, June 1997, pp. 359–370.