Spectrum Assignment in Rings with Shortest-Path Routing: Complexity and Approximation Algorithms

Sahar Talebi[†], Furqan Alam[‡], Iyad Katib[‡], George N. Rouskas[†‡]
[†]Operations Research and Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206 USA
[‡]King Abdulaziz University, Jeddah, Saudi Arabia

***Abstract*—We study the spectrum assignment (SA) problem in ring networks with shortest path (or, more generally, fixed) routing. With fixed routing, each traffic demand follows a predetermined path to its destination. In earlier work, we have shown that the SA problem can be viewed as a multiprocessor problem. Based on this insight, we prove that, under the shortest path assumption, the SA problem can be solved in polynomial time in small rings, and we develop constant-ratio approximation algorithms for large rings. For rings of size up to 16 nodes (the maximum size of a SONET/SDH ring), the approximation ratios of our algorithms are strictly smaller than the best known ratio to date.**

I. Introduction

Elastic optical networking has been the subject of considerable research and development activities in recent years due to its potential to accommodate efficiently the ongoing growth in traffic demands [4], [6], [12]. Key enabling technologies of elastic networking include optical OFDM, distance-adaptive modulation, flexible spectrum selective switches, and bandwidth-variable transponders [18]. These technologies make it possible for network operators to support multirate connections and adapt to variable bandwidth requests dynamically, by "slicing off" just the right amount of spectrum for each traffic demand [6].

Routing and spectrum assignment (RSA) [2], [7], [16] has emerged as the essential problem for network-wide management of spectral resources in the context of design and control of elastic optical networks. The objective of the RSA problem is to (1) assign a physical path to each demand, and (2) allocate continuous and contiguous spectrum to the demand along the links of each path, so as to optimize a metric of interest typically related to spectrum utilization. Several offline and online variants of the problem have been studied; for a survey and classification of existing approaches, the reader is referred to [14]. Since general versions of the RSA problem are computationally intractable, common solution approaches include integer linear programming (ILP) formulations (for small network sizes) and heuristics.

While most studies of the RSA problem consider general network topologies, we note that not only are large parts of the current infrastructure based on SONET/SDH rings, but DWDM networks with topological rings are being deployed that are based on technologies other than SONET (e.g., Ethernet, IP/MPLS, etc.). Therefore, more recently there has been increasing interest in RSA solutions for ring networks [8]–[10], [13], [17]. Most of these studies employ heuristics, although some interesting theoretical results do exist. For instance, using results from graph coloring theory, it was shown in [13] that there exists a $(4+2\epsilon)$-approximation algorithm for the spectrum assignment problem in rings; whereas the work in [9] proves that the contiguity (i.e., adjacency) constraint in spectrum assignment can always be satisfied starting from an optimal solution to a corresponding coloring problem.

In this paper we present a formal study of the spectrum assignment (SA) problem in ring networks with shortest path (or, more generally, fixed) routing. With fixed routing, each traffic demand follows a predetermined path to its destination. In earlier work [15], we have shown that the SA problem can be viewed as a multiprocessor problem. We extend those results to prove that the SA problem can be solved in polynomial time in small rings, and to develop constant-ratio approximation algorithms for large rings. For rings of size up to 16 nodes (the maximum size of a SONET/SDH ring), the approximation ratios of our algorithms are strictly smaller than the best known $(4+2\epsilon)$ ratio presented in [13]. We also note that our results apply to the wavelength assignment problem, a special case of spectrum assignment in which all demands are of equal size.

The paper is organized as follows. In Section II, we show that the SA problem in rings with shortest path routing transforms to a classical multiprocessor scheduling problem. Based on this insight, in Section III, we investigate the complexity of the SA problem and prove that, under shortest path routing, SA can be solved in polynomial time on rings of small size. In Section IV, we develop new constant-ratio approximation algorithm for large rings, and we conclude the paper in Section V.

II. Spectrum Assignment in Bidirectional Rings

Consider the following general definition of the spectrum assignment (SA) problem in elastic optical networks:

Definition 2.1 (SA): Given

- a graph $G = (\mathcal{V}, \mathcal{A})$ where $\mathcal{V}$ is the set of nodes and $\mathcal{A}$ the set of arcs (directed links),

This work was supported by the National Science Foundation under Grant CNS-1113191, and in part by the Deanship of Scientific Research (DSR), King Abdulaziz University, under Grant No. 2-611-1434-HiCi.

- a spectrum demand matrix $T = [t_{sd}]$, where t_{sd} is the amount of spectrum required to carry the traffic from source node s to destination node d, and
- a physical path r_{sd} from node s to node d,

assign spectrum to each demand so as to minimize the total amount of spectrum used on any link in the network, under three constraints:

1) each demand is assigned contiguous spectrum (spectrum contiguity constraint);
2) each demand is assigned the same spectrum along all links of its path (spectrum continuity constraint); and
3) demands that share a link are assigned non-overlapping parts of the available spectrum (non-overlapping spectrum constraint).

In this work, we study the SA problem in bidirectional rings under the assumption that each traffic demand is carried over the shortest path from its source to the destination node. Let N be the number of nodes (and links) of the ring network. Note that, whenever N is even, there are two shortest paths between every pair of nodes that are diametrically opposite each other. In this case, we assume that one of these paths (in either the clockwise or counter-clockwise direction) is selected and is provided as input to the SA problem.

As we have shown in our recent work [15], the SA problem in mesh networks transforms to the multiprocessor scheduling problem, denoted as $P|fix_j|C_{max}$, in which some tasks are to be executed on multiple processors simultaneously. Problem $P|fix_j|C_{max}$, which we will be referring to throughout this paper, is defined as [1], [5]:

Definition 2.2 ($P|fix_j|C_{max}$): Given

- a set of m identical processors,
- a set of n tasks with processing time $p_j, j = 1, \dots, n$, and
- a prespecified set fix_j of processors for executing each task $j, j = 1, \dots, n$,

schedule the tasks so as to minimize the makespan $C_{max} = \max_j C_j$, where C_j denotes the completion time of task j, under the constraints:

1) preemptions are not allowed;
2) each task must be processed simultaneously by all processors in fix_j; and
3) each processor can work on at most one task at a time.

Also, we denote by $Pm|fix_j|C_{max}$ the special case of $P|fix_j|C_{max}$ in which the number of processors m is considered to be fixed. The proof of the transformation is available in [15]. Briefly, each link in the SA problem transforms to a processor, each traffic demand (s, d) to a task j, the demand size t_{sd} and path r_{sd} to the processing time p_j and set fix_j of the corresponding task j, respectively, the maximum spectrum assigned to any link to C_{max}, and each of the three constraints of the SA problem to one of the three constraints of problem $P|fix_j|C_{max}$.

Since any algorithm that solves the $P|fix_j|C_{max}$ problem also solves the corresponding SA problem, in the following we will derive results for the SA problem in rings by studying the corresponding multiprocessor scheduling problem. In our discussion, we will make use of two concepts related to $P|fix_j|C_{max}$.

Definition 2.3 (Compatible Tasks): A set $\mathcal{T}$ of tasks for the $P|fix_j|C_{max}$ problem are said to be *compatible* if and only if their prespecified sets of processors are pairwise disjoint, i.e., $fix_i \cap fix_j = \emptyset, \forall\ i, j \in \mathcal{T}$.

Compatible tasks may be paired with each other (i.e., they can be executed simultaneously), as they do not share any processors.

Definition 2.4 (Dominant Processor and Lower Bound): Consider an instance of $P|fix_j|C_{max}$, and let $\mathcal{T}_k$ denote the set of tasks that require processor k, i.e., $\mathcal{T}_k = \{j : k \in fix_j\}$. Clearly, all the tasks in $\mathcal{T}_k$ are pairwise incompatible, hence they have to be executed sequentially. Therefore, a lower bound LB for the problem instance can be obtained as:

$$LB = \max_{k=1,\dots,m} \left\{ \sum_{j \in \mathcal{T}_k} p_j \right\}. \tag{1}$$

We will refer to a processor that achieves the lower bound LB as the *dominant* processor.

III. Complexity Results

We first note that, under shortest path routing, the clockwise and counter-clockwise directions of the ring become decoupled and completely independent of each other. Consequently, the SA problem in bidirectional rings is decomposed into two disjoint subproblems, one for each direction, that can be solved separately; the subproblem in the clockwise (respectively, counter-clockwise) direction takes as input the subset of clockwise (respectively, counter-clockwise) links and the subset of demands with shortest paths along these links. It can be seen that this decomposition is optimal, in that finding the optimal solution (i.e., minimum total spectrum on any link) for each subproblem and taking the maximum of the two is an optimal solution to the original problem on the bidirectional link. Therefore, for the remainder of this paper, we will only consider the SA subproblem for the clockwise direction of the ring; because of symmetry, the same results apply to the subproblem defined on the counter-clockwise direction.

We have shown in [15] that the SA problem in unidirectional rings can be transformed to a $P|fix_j|C_{max}$ problem. Moreover, in the general case, i.e., whenever there are traffic demands between any pair of nodes, the SA problem in unidirectional rings with $N = 3$ nodes transforms [15] to the $P3|fix_j|C_{max}$ problem that is strongly NP-hard [5]. On the other hand, the SA subproblem defined on the clockwise direction of a bidirectional ring is a special case of the unidirectional ring problem inasmuch as its input consists of only the subset of demands that are routed in that direction. Therefore, the problem can be solved in polynomial time for small rings, and approximation algorithms with constant ratios exist, as we show next.

A. Rings with $N = 3, 4$ Nodes

The following two lemmas establish that, under shortest path routing, the SA problem can be solved in polynomial time in three- and four-node bidirectional rings, since the subproblems defined on the clockwise (and, hence, also the counter-clockwise) direction yield polynomial solutions. Note also that the wavelength assignment (WA) problem [11] is a special case of the SA problem in which all demands are of size $t_{sd} = 1$. Consequently, these two lemmas also establish that the WA problem is solvable in polynomial time in three- and four-node rings with shortest path routing.

Lemma 3.1: The SA subproblem defined in the clockwise direction of a bidirectional ring with $N = 3$ nodes and shortest path routing is solvable in polynomial time.

Proof. In a bidirectional ring with $N = 3$ nodes, the shortest path for each demand consists of a single link. Consider the SA subproblem defined on the clockwise direction. This subproblem has three demands, each carried on exactly one of the three clockwise links of the ring. The corresponding $P3|fix_j|C_{max}$ multiprocessor scheduling problem has three tasks, each requiring exactly one of the three processors (i.e., $|fix_j| = 1, j = 1, 2, 3$). Since the tasks are pairwise compatible, they can be scheduled simultaneously. Hence, the optimal value of the total amount of spectrum required in the network (respectively, C_{max}) is equal to the maximum demand size (respectively, the maximum task processing time). ■

Lemma 3.2: The SA subproblem defined in the clockwise direction of a bidirectional ring with $N = 4$ nodes and shortest path routing is solvable in polynomial time.

Proof. In a four-node ring, the clockwise and counter-clockwise paths between two non-adjacent nodes are of equal length (i.e., two), and either may be selected as the shortest path. Let us consider the case where all demands between non-adjacent nodes are routed in the clockwise direction. In other words, if nodes 1 and 3 are non-adjacent, then both traffic from 1 to 3 and traffic from 3 to 1 is routed clockwise; and similarly for the other pair (2,4) of non-adjacent nodes. Hence, the input to the SA subproblem consists of four one-link demands and four two-link demands. Consequently, the input to the corresponding $P4|fix_j|C_{max}$ problem consists of four single-processor tasks and four two-processor tasks. Let us denote these tasks as T_1, T_2, T_3, T_4, T_{12}, T_{23}, T_{34}, and T_{41}, where the subscript of each task denotes the processors in the corresponding set fix_j.

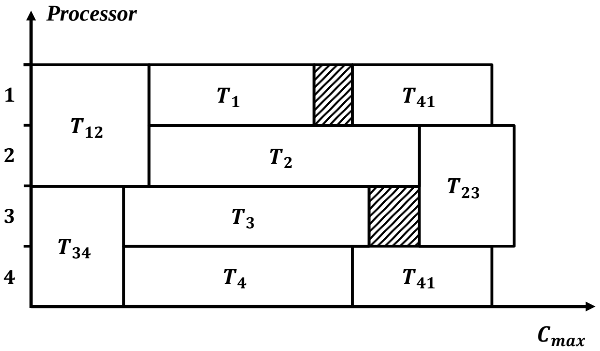

Fig. 1. Optimal schedule for the clockwise direction of a four-node bidirectional ring with shortest path routing

The proof is by construction of the optimal schedule, as shown in Figure 1. Specifically, first schedule the task T_{12} in parallel with the task T_{34} starting at time $t = 0$. Then, add all the single processor tasks T_1, T_2, T_3, T_4 to this initial schedule without any gaps. Finally, execute the two-processor tasks T_{23} and T_{41} as soon as both processors of each task are available. For the instance depicted in Figure 1, the schedule is optimal as it is equal to the lower bound determined by the sum of the processing times of tasks requiring processor 2 (the dominant processor). In fact, because of symmetry, the schedule is optimal regardless of which processor is the dominant one.

If some of the demands between non-adjacent nodes are routed in the counter-clockwise direction, then the instance of $P4|fix_j|C_{max}$ defined on the clockwise direction will not include the corresponding two-processor tasks. Again, it can be seen that the above algorithm yields an optimal schedule. For instance, if task T_{23} is excluded from Figure 1, then the schedule remains optimal. The same is true if T_{41} is excluded, or T_{23} and T_{41} are both excluded, or any combination of two-processor tasks is excluded. If all two-processor tasks are excluded (i.e., all demands between non-adjacent nodes are routed in the counter-clockwise direction), then the problem contains only single-processor tasks and the algorithm again produces an optimal schedule. ■

The above lemma shows that as long as traffic demands in a four-node bidirectional ring are routed along a shortest path (with ties broken arbitrarily), the SA problem is solvable in polynomial time using a simple algorithm that is linear in the number of tasks (spectrum demands). The following lemma shows that if one of the demands between adjacent nodes takes a non-shortest path, the SA problem becomes NP-complete. The proof is by reduction from the PARTITION problem [3] which is defined as:

Definition 3.1 (PARTITION): Given a set of k integers $A = \{a_1, a_2, \ldots, a_k\}$ such that $B = \sum_{j=1}^{k} a_j$, does there exist a partition of A into two sets, A_1 and A_2, such that $\sum_{a_j \in A_1} a_j = \sum_{a_j \in A_2} a_j = \frac{B}{2}$?

Lemma 3.3: The SA subproblem defined in the clockwise direction of a bidirectional ring with $N = 4$ nodes and such that:

- all demands between non-adjacent nodes are routed in the clockwise direction, and
- all demands between adjacent nodes are routed along their (one-link) shortest path in the clockwise or counter-clockwise direction, *except* for one such demand that is directed to a three-link path in the clockwise direction,

is NP-complete.

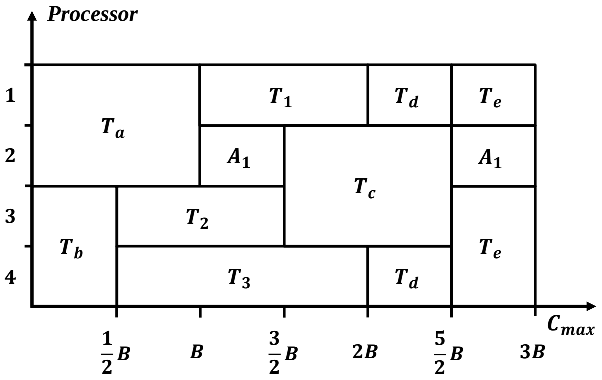

Fig. 2. A feasible schedule with $C_{max} = 3B$ for the clockwise direction of a four-node bidirectional ring with shortest-path routing except for one demand routed along a three-link path.

Proof. If a traffic demand with a one-link shortest path in the counter-clockwise direction is routed along the alternate clockwise three-link path, then the $P4|fix_j|C_{max}$ problem defined on the clockwise direction will include a three-processor task. Without loss of generality, assume that this three-processor task requires processors 3, 4, and 1 (similar arguments apply for any other three-processor task). Given an instance of PARTITION, we create an instance of this $P_4|fix_j|C_{max}$ as follows. For each $a_j \in A$ we create a task τ_j with processing time $p_j = a_j$ and $fix_j = \{2\}$ (note that these tasks must be executed by processor 2, the one that is *not* required by the three-processor task). We also create the following eight gadget tasks:

task	p_j	fix_j
T_a	B	$\{1, 2\}$
T_b	$B/2$	$\{3, 4\}$
T_c	B	$\{2, 3\}$
T_d	$B/2$	$\{4, 1\}$
T_e	$B/2$	$\{3, 4, 1\}$
T_1	B	$\{1\}$
T_2	B	$\{3\}$
T_3	$3B/2$	$\{4\}$

If A can be partitioned into two disjoint sets A_1 and A_2 such that $\sum_{a_j \in A_1} = \sum_{a_j \in A_2} = B/2$, then there is a feasible schedule with $C_{max} = 3B$, as shown in Figure 2.

Conversely, let us assume that there exists a feasible schedule $\mathcal{S}$ with $C_{max} \leq 3B$. Without loss of generality, suppose that T_a and T_b are executed before T_c and T_d in $\mathcal{S}$; otherwise, we can use similar arguments and reach the same conclusion. Then, all the single processor tasks T_1, T_2, and T_3 must be executed immediately after T_a or T_b complete, as scheduling any other task at that time would lead to a makespan greater than $3B$. T_c must also be scheduled exactly right after T_2 and before T_e, otherwise it would not be possible to obtain the schedule with length of at most $3B$. Using a similar argument, T_d must be scheduled right after T_1 and T_3 and before T_e, and in parallel with T_c. The schedule corresponding to this set of tasks is shown in Figure 2 where only the intervals $[B, 3B/2]$ and $[5B/2, 3B]$ are available for the execution of the PARTITION jobs on processor 2. Therefore, a partition must exist. ■

B. Rings with $N \geq 5$ Nodes

The next theorem states that the SA problem on five-node bidirectional rings (and, hence, on any larger ring) is intractable.

Theorem 3.1: The SA subproblem defined in the clockwise direction of a bidirectional rings with $N = 5$ nodes and shortest path routing is NP-complete.

Proof. As the number of nodes is odd, there is a unique shortest path for each traffic demand between any two non-adjacent nodes; therefore, the problem in the clockwise direction includes only the demands with a shortest path along the clockwise links. The proof is by reduction from the PARTITION problem, and follows an approach similar to the one we used in the proof of Lemma 3.3. Specifically, for each $a_j \in A$, we create a task τ_j with processing time $p_j = a_j$ and $fix_j = \{2\}$. We also create the following set of tasks:

task	p_j	fix_j
T_a	$3B/2$	$\{1, 2\}$
T_b	$5B/2$	$\{2, 3\}$
T_c	$B/2$	$\{3, 4\}$
T_d	B	$\{4, 5\}$
T_e	$2B$	$\{5, 1\}$
T_1	$3B/2$	$\{1\}$
T_2	$2B$	$\{3\}$
T_3	$7B/2$	$\{4\}$
T_4	$2B$	$\{5\}$

If there exists a partition of A into two disjoint sets A_1 and A_2 such that $\sum_{a_j \in A_1} = \sum_{a_j \in A_2} = B/2$, then we can execute the tasks as shown in Figure 3 and create a feasible schedule with $C_{max} = 5B$.

Conversely, assume that there exists a feasible schedule $\mathcal{S}$ with $C_{max} \leq 5B$. Similar to the proof of Lemma 3.3 and without loss of generality, suppose that T_a and T_d are executed before T_c and T_e in $\mathcal{S}$; otherwise, we can use similar arguments and reach the same conclusion. We need to schedule T_2 in parallel with T_a and T_d, otherwise the schedule length will exceed $5B$. As T_d completes earlier than T_a, we need to execute T_4 before T_e. Therefore, T_1 must be scheduled right after T_a and before T_e. On the other hand, T_3 must be executed immediately after T_d, and T_c must be scheduled at the very end of $\mathcal{S}$, since if we change the order of execution of T_3 and T_c in $\mathcal{S}$, the makespan will be greater than $5B$. Finally, executing T_c between $[9B/2, 5B]$ means that T_b must be scheduled immediately after T_2. A feasible schedule corresponding to this set of tasks is shown in Figure 3 where only the intervals $[3B/2, 2B]$ and $[9B/2, 5B]$ are available for the execution of the PARTITION jobs on

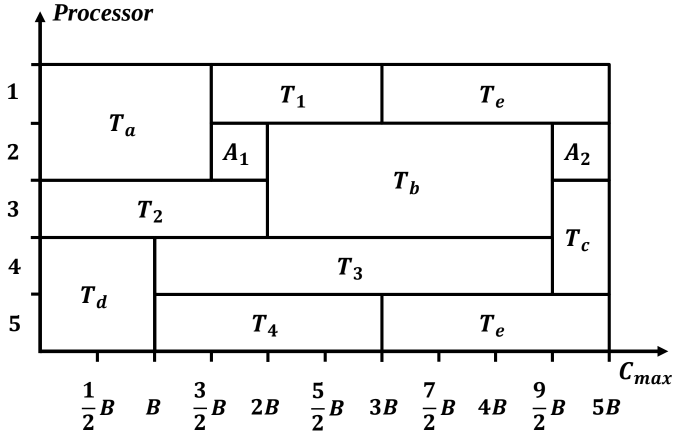

Fig. 3. A feasible schedule with $C_{max} = 5B$ for the clockwise direction of a five-node ring with shortest path routing.

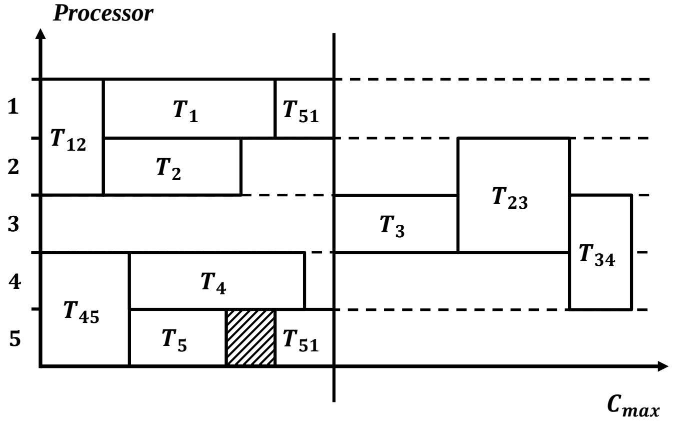

Fig. 4. Two-part schedule for a five-node bidirectional ring with shortest path routing

processor 2. Thus, we conclude that a partition of A must exist. ■

IV. Approximation Algorithms

In this section, we first provide approximation algorithms for the SA problem on bidirectional rings with $N = 5, 6$ and 7 nodes under shortest path routing. We then build upon the approximation algorithms for path networks we presented in [15] to develop approximation algorithms for bidirectional rings with $N \geq 8$ nodes. Since, as we mentioned earlier, the WA problem is a special case of SA, all approximation algorithms in this section also apply to WA.

A. Rings with $N = 5 - 7$ Nodes

Lemma 4.1: There exists an 1.5-approximation algorithm for the SA subproblem defined on the clockwise direction of a bidirectional ring with $N = 5$ nodes and shortest path routing.

Proof. As we mentioned earlier, in a five-node ring each traffic demand has a unique shortest path. Therefore, the clockwise direction serves $10 (= 5 * 4/2)$ demands, and the corresponding scheduling problem has 10 tasks as shown in Figure 4, where the subscript of each task indicates the processors required by the task. Without loss of generality, let processor 3 be the dominant processor, i.e., the one that achieves the lower bound LB in (1). Let OPT denote the optimal value of the makespan for this problem; clearly, $LB \leq OPT$.

Consider now the seven tasks that do *not* require processor 3, shown in the left part of the schedule in Figure 4. The scheduling problem consisting of these seven tasks can be viewed as the scheduling problem on a four-processor system (i.e., one without processor 3), similar to the one depicted in Figure 2 – but with three rather than four two-processor tasks. In essence, this scheduling problem corresponds to the SA problem on the clockwise direction of the five-node after removing the link corresponding to processor 3 and the three traffic demands using that link. Based on our earlier result regarding the four-node rings, these seven tasks can be scheduled optimally, as shown on the left part of Figure 4. Let OPT' be the makespan of this schedule; then, $OPT' \leq OPT$.

Now consider the three tasks that require processor 3. These can be scheduled back-to-back without any gaps, as shown in the right part of Figure 4. The makespan of this schedule is equal to LB. Hence, the makespan of the two-part, 10-task schedule depicted in Figure 4 is equal to: $OPT' + LB \leq 2 \times OPT$.

We can improve the approximation ratio of 2 by modifying the above two-part schedule as follows. Without loss of generality, assume that $T_{23} \geq T_{34}$ as indicated in Figure 4; if T_{34} is larger than T_{23}, then simply reverse the roles in the following discussion. In this case, we have that:

$$\begin{aligned} T_{34} &\leq T_3 + T_{23} \\ \Rightarrow 2T_{34} &\leq T_3 + T_{23} + T_{34} \; = \; LB \; \leq \; OPT \\ \Rightarrow T_{34} &\leq 0.5 \times OPT \end{aligned} \quad (2)$$

Now slide the right part of the schedule in Figure 4 (i.e., the three tasks T_3, T_{23} and T_{34}) as far left as possible so that tasks T_3 and/or T_{23} overlap with the tasks in the left part of the schedule. Consider the resulting nine-task schedule, i.e., the one consisting of all tasks of the problem except T_{34}. It can be seen that this schedule is optimal for these nine tasks. Let OPT'' be the makespan of this nine-task schedule, and $OPT'' \leq OPT$. Scheduling task T_{34} immediately after the end of this schedule results in a ten-task schedule of length $OPT'' + T_{34}$. Using (2), we conclude that the makespan of this schedule is no larger than $1.5 \times OPT$.

Lemma 4.2: There exist 2-approximation algorithms for the SA subproblem defined on the clockwise direction of bidirectional rings with $N = 6, 7$ nodes and shortest path routing.

Proof. The proof is by construction of a two-part schedule similar to the one we created for the proof of Lemma 4.1. The proof is omitted due to its length, and the details are

Ring size N	Approximation ratio
3-4	1
5	1.5
6-7	2
8-10	3
11-14	3.5
15-16	4

TABLE I
APPROXIMATION RATIOS FOR THE SA PROBLEM ON THE CLOCKWISE DIRECTION OF BIDIRECTIONAL RINGS OF VARIOUS SIZES UNDER SHORTEST PATH ROUTING.

available in the first author's dissertation. ■

B. Rings with $N \geq 8$ Nodes

We now present a general approximation algorithm for rings that builds upon corresponding algorithms for directed paths we developed in [15]. Consider the SA problem defined on the clockwise direction of a ring with $N \geq 8$ nodes and shortest path routing. The key idea is based on the observation that if we remove a link from the ring along with the traffic demands whose shortest paths use this link, the resulting SA subproblem is equivalent to the SA problem on a directed path with $N-1$ nodes. Therefore, the approximation algorithm for rings consists of the following steps:

1) Formulate the $PN|fix_j|C_{max}$ problem for the clockwise direction of the original ring.
2) Let processor N be the dominant processor (and relabel the processors appropriately if necessary).
3) Remove processor N and all tasks j that use this processor (i.e., tasks j such that $N \in fix_j$) and formulate the resulting $P(N-1)|line_j|C_{max}$ scheduling problem defined in [15][1].
4) Use the approximation algorithm in [15] to create schedule $\mathcal{S}_1$ for the $P(N-1)|line_j|C_{max}$ problem; let $\alpha(N-1)$ be the approximation ratio of this algorithm.
5) Schedule all tasks that use processor N sequentially without any gaps to create schedule $\mathcal{S}_2$.
6) Concatenate schedules $\mathcal{S}_1$ and $\mathcal{S}_2$ to create schedule $\mathcal{S}$ for the ring network.

Let OPT be the optimal makespan for the ring network. By construction, the makespan of $\mathcal{S}_2$ is equal to $LB \leq OPT$, while the makespan of $\mathcal{S}_1$ is no longer than $\alpha(N-1)OPT$. Hence, the approximation ratio of the above algorithm for an N-node ring is $1+\alpha(N-1)$. Using our earlier results and the values for $\alpha(N-1)$ from [15], we obtain the approximation ratios shown in Table I for various ring sizes. As we can see, for rings of sizes encountered in practice, the approximation ratio of our algorithm is strictly better than the $4+2\epsilon$ algorithm presented in [13].

V. Concluding Remarks

We have studied the complexity of the spectrum assignment problem in bidirectional rings with shortest path routing, and we have developed new constant-ratio approximation algorithms for rings of large size. In ongoing work, we study the RSA problem in bidirectional rings, also from a multiprocessor scheduling perspective.

References

[1] E. Bampis, M. Caramia, J. Fiala, A. Fishkin, and A. Iovanella. Scheduling of independent dedicated multiprocessor tasks. In *Proceedings of the 13th Annual International Symposium on Algorithms and Computation*, volume LNCS 2518, pages 391–402, 2002.

[2] K. Christodoulopoulos, I. Tomkos, and E.A. Varvarigos. Elastic bandwidth allocation in flexible OFDM–based optical networks. *Journal of Lightwave Technology*, 29(9):1354–1366, 2011.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Co., New York, 1979.

[4] O. Gerstel, M. Jinno, A. Lord, and S. J B Yoo. Elastic optical networking: a new dawn for the optical layer? *IEEE Communications Magazine*, 50(2):s12–s20, 2012.

[5] J. A. Hoogeveen, S. L. Van de Velde, and B. Veltman. Complexity of scheduling multiprocessor tasks with prespecified processor allocations. *Discree Applied Mathematics*, 55:259–272, 1994.

[6] M. Jinno, H. Takara, and B. Kozicki. Dynamic optical mesh networks: Drivers, challenges and solutions for the future. In *Proceedings of 35th European Conference on Optical Communication (ECOC)*, page 7.7.4, September 2009.

[7] M. Klinkowski and K. Walkowiak. Routing and spectrum assignment in spectrum sliced elastic optical path network. *IEEE Communications Letters*, 15(8):884–886, 2011.

[8] F. Musumeci, F. F. Puleio, and M. Tornatore. Dynamic grooming and spectrum allocation in optical metro ring networks with flexible grid. In *Proceedings of ICTON 2013*, page We.A1.2, June 2013.

[9] I. Popescu, I. Cerutti, N. Sambo, and P. Castoldi. On the optimal design of a spectrum-switched optical network with multiple modulation formats and rates. *IEEE/OSA Journal of Optical Communications and Networking*, 5(11):1275–1284, 2013.

[10] C. Rottondi, M. Tornatore, A. Pattavina, and G. Gavioli. Routing, modulation level, and spectrum assignment in optical metro ring networks using elastic transceivers. *IEEE/OSA Journal of Optical Communications and Networking*, 5(4):305–315, 2013.

[11] G. N. Rouskas. Routing and wavelength assignment in optical WDM networks. In *J. Proakis (Editor), Wiley Encyclopedia of Telecommunications*. John Wiley & Sons, 2001.

[12] G. Shen and M. Zukerman. Spectrum-efficient and agile CO-OFDM optical transport networks: architecture, design, and operation. *IEEE Communications Magazine*, 50(5):82–89, 2012.

[13] S. Shirazipourazad, Ch. Zhou, Z. Derakhshandeh, and A. Sen. On routing and spectrum allocation in spectrum-sliced optical networks. In *Proceedings of IEEE INFOCOM*, pages 385–389, April 2013.

[14] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Khalifah, and G. N. Rouskas. Spectrum management techniques for elastic optical networks: A survey. *Optical Switching and Networking*, 13:34–48, July 2014.

[15] S. Talebi, E. Bampis, G. Lucarelli, I. Katib, and G. N. Rouskas. The spectrum assignment (SA) problem in optical networks: A multiprocessor scheduling perspective. In *Proceedings of ONDM 2014*, May 2014.

[16] L. Velasco, M. Klinkowski, M. Ruiz, and J. Comellas. Modeling the routing and spectrum allocation problem for flexgrid optical networks. *Photonic Network Communications*, 24:177–186, 2012.

[17] Y. Wang, X. Cao, and Y. Pan. A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks. In *Proceedings of IEEE INFOCOM*, pages 1503–1511, 2011.

[18] G. Zhang, M. De Leenheer, A. Morea, and B. Mukherjee. A survey on OFDM-based elastic core optical networking. *IEEE Communications Surveys & Tutorials*, 15(1):65–87, First Quarter 2013.

[1] This scheduling problem corresponds to the SA problem on the $(N-1)$-link path, as discussed in [15].