

Lightpath Arrangement in Survivable Rings to Minimize the Switching Cost

Tamar Eilam, Shlomo Moran, and Shmuel Zaks

Abstract—This paper studies the design of low-cost survivable wavelength-division-multiplexing (WDM) networks. To achieve survivability, lightpaths are arranged as a set of rings. Arrangement in rings is also necessary to support SONET/SDH protection schemes such as 4FBLSR above the optical layer. This is expected to be the most common architecture in regional (metro) networks [9]. We assume that we are given a set of lightpaths in an arbitrary network topology and aim at finding a partition of the lightpaths to rings adding a minimum number of lightpaths to the original set. The cost measure that we consider (number of lightpaths) reflects the switching cost of the entire network. In the case of a SONET/SDH higher layer, the number of lightpaths is equal to the number of add-drop multiplexers (ADMs) (since two subsequent lightpaths in a ring can share an ADM at the common node).

We prove some negative results on the tractability and approximability of the problem and provide an approximation algorithm with a worst case approximation ratio of 8/5. We study some special cases in which the performance of the algorithm is improved.

A similar problem was introduced, motivated, and studied in [9] and recently in [13] (where it was termed *minimum ADM problem*). However, these two works focused on a ring topology while we generalize the problem to an arbitrary network topology.

Index Terms—Optical network design, SONET add/drop multiplexers (ADMs), SONET rings, wavelength-division multiplexing (WDM).

I. INTRODUCTION

A. Background

OPTICAL networks play a key role in providing high bandwidth and connectivity in today's communication world and are currently the preferred medium for the transmission of data. While first-generation optical networks simply served as a transmission medium, second-generation optical networks perform some switching and routing functions in the optical domain. In these networks, routing is performed by using *lightpaths*. A lightpath is an end-to-end circuit-switched communication connection that traverses one or more links and uses

one wavelength-division-multiplexing (WDM) channel (wavelength) per link. We assume a static setting in which lightpaths are designed and constructed in the setup time of the network.

Since the capacity enabled by this technology substantially exceeds the one provided by conventional networks, it is important to incorporate the ability to recover from failures into the optical layer. *Survivability* is the ability of the network to recover from failures of hardware components. In this paper, we study the design of a survivable optical layer. Our goal is the construction of a low-cost survivable set of lightpaths in a given topology. We assume that an initial set of lightpaths (designed according to the expected communication pattern) is given, and we are targeted at augmenting this initial set with additional lightpaths such that the resulting set will guarantee survivability. For this purpose, we define a *survivability condition* that the solution must satisfy and a *cost function* according to which we evaluate the cost of the solution found.

We focus on the *ring partition survivability condition*. Informally, this condition states that lightpaths are partitioned to rings, and that all lightpaths in a ring traverse disjoint routes in the underlying topology. The motivation for the ring partition survivability condition is two folded. First, it supports a simple and fast protection mechanism. In the case of a failure, the data are rerouted around the impaired lightpath, on the alternate path of lightpaths in its ring. The demand that all lightpaths in one ring traverse disjoint routes guarantees that this protection mechanism is always applicable in the case of one failure. Second, a partition of the lightpaths to rings is necessary in order to support a higher layer in the form of synchronous optical network (SONET)/synchronous digital hierarchy (SDH) self-healing rings (e.g., 4FBLSR protection mechanism). While it is anticipated that shared-mesh restoration schemes will dominate in the core network, regional (metro) networks will continue to deploy ring restoration (e.g., [9]).

Another issue is determining the cost of the design. We assume that a *uniform cost* is charged for every lightpath, namely, the cost of the design is the number of lightpaths in it. This cost measure is justified for two reasons. First, in regional area networks, it is reasonable to assume that the same cost will be charged for all the lightpaths ([16]). Second, every lightpath is terminated by a pair of line terminals (LTs). In regional (metro) networks, the *switching cost* of the entire network is dominated by the number of LTs (port cost), which is proportional to the number of lightpaths ([9]). In the case of a SONET/SDH higher layer, the line terminals are add-drop multiplexers (ADMs). Two subsequent lightpaths in a ring can share one ADM at the common node. Thus, the number of lightpaths in a ring partition arrangement is equal to the

Manuscript received February 14, 2001; revised July 29, 2001. An extended abstract of this work appeared in DISC 2000. The work of S. Zaks was supported in part by the fund for the promotion of research at the Technion, by Technion V.P.R. fund, and by the Bar-Nir Bergreen Software Technology Center of Excellence, Department of Computer Science, Technion. The work of S. Moran was supported in part by the fund for the promotion of research at the Technion and by the Bernard Elkin Chair in Computer Science.

T. Eilam was with the Department of Computer Science, The Technion, 32000 Haifa, Israel. She is now with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10532 USA (e-mail: eilamt@us.ibm.com).

S. Moran and S. Zaks are with the Department of Computer Science, The Technion, 32000 Haifa, Israel (e-mail: moran@cs.technion.ac.il; zaks@cs.technion.ac.il).

Publisher Item Identifier S 0733-8716(02)00157-9.

number of ADMs, and minimizing the number of lightpaths is equivalent to minimizing the number of ADMs, which corresponds to the port cost of the network.

We assume that the network topology is given in the form of a simple graph. A lightpath is modeled as a pair (ID, P) , where ID is a unique identifier and P is a simple path in the graph. A *design* D for a set of lightpaths C is a set of lightpaths that subsumes C (i.e., $C \subseteq D$). A design is termed a *ring partition* if it satisfies the ring partition condition. The *cost* of a design is the number of lightpaths in it (namely, $\text{cost}(D) = |D|$). We end up with the following optimization problem, which we term the *minimum cost ring partition design* (MCRPD) problem. The input is a graph G and an initial set C of lightpaths in G . The goal is to find a ring partition design D for C with minimum cost.

B. Results

We prove that the MCRPD problem is NP-hard for every family of topologies that contains cycles with unbounded length, e.g., rings (see formal definition in Section III). A similar proof only for the family of rings recently appeared in [13]. We prove that there is no polynomial time approximation algorithm A that constructs a design D that satisfies $\text{Cost}(D) \leq \text{OPT} + n^\alpha$ for any constant $\alpha < 1$, where n is the number of lightpaths in the initial set and OPT is the cost of an optimal solution for this instance (unless $P = \text{NP}$). For $\alpha = 1$, a trivial approximation algorithm constructs a solution within this bound.

We present a *ring partition algorithm* (RPA) that finds in polynomial time a ring partition design for every given instance of MCRPD (if it exists). We analyze the performance of RPA and show that for the general case (arbitrary topology), RPA guarantees $\text{Cost}(D) \leq \min(\text{OPT} + (3/5) \cdot n, 2n) \leq (8/5)\text{OPT}$, where n and OPT are as defined above. We analyze the performance of RPA also for some interesting special cases in which better results are achieved.

This paper is structured as follows. We first present the model, followed by a description of the MCRPD problem (Section II). We then present and discuss the results; the focus of Section III is the negative results, while Section IV presents and analyzes the approximation algorithm and the special cases. Last, we summarize and discuss future research directions (Section VI).

C. Related Works

This paper [9] studies ring partition designs for the special case where the physical topology is a ring. In fact, the MCRPD problem is a generalization of this problem for arbitrary topologies. Reference [9] also motivates the focus on the number of lightpaths rather than the total number of wavelengths in the design. Some heuristics to construct ring partition designs in rings are given and some lower and upper bounds on the cost (as a function of the load) are proved. The paper also considers *lightpath splitting*—a lightpath might be partitioned to two or more lightpaths. It is shown that better results can be achieved by splitting lightpaths.

Recently, [13] presented more results on the same problem (termed *minimum ADM problem*) only for ring topology. The paper presents a similar NP-hardness result for the family of

rings, a randomized approximation scheme for the problem, and some heuristics and simulation results.

Other works in this field refer to different models than what we considered. Reference [10] presents methods for recovering from channel, link, and node failures in first-generation WDM ring networks with limited wavelength conversion. Reference [15] assumes that lightpaths are dynamic and focuses on management protocols for setting them up and taking them down.

When the set of lightpaths is static, the survivability is achieved by providing disjoint routes to be used in the case of a failure. References [12] and [1] study this problem, but the objective is the minimization of the total number of wavelengths and not the number of lightpaths.

Reference [2] offers some heuristics and empirical results for the following problem. Given the physical topology and a set of connections requests (i.e., requests for lightpaths in the form of pairs of nodes), find routes for the requests so as to minimize the number of pairs (l, e) consisting of a routed request (i.e., a lightpath) l and a physical link e for which there is no alternative path of lightpaths between the endpoints of l in the case that e fails. Note that this survivability condition is less restrictive than the ring partition condition that we consider in this paper.

A related problem in this area with a similar goal of minimizing the electronic multiplexing cost in WDM networks is traffic grooming (see, e.g., [5], [3], and [11]).

II. MODEL AND DEFINITIONS

For our purposes, lightpaths are modeled as *connections*, where every connection c has a unique identifier $\text{ID}(c)$ and is associated with a simple path $\mathcal{R}(c)$ in the network. \mathcal{R} is termed the *routing function*. Note that two different connections might have the same route. We assume that routes of connections are always simple (i.e., they do not contain loops). We say that two connections are *disjoint* if their routes are disjoint, namely, they do not share any edge and any node that is not an end node of both connections. We use the terms “connections” and “lightpaths” interchangeably.

A *virtual path* P is a sequence $\langle v_1, c_1, v_2, c_2, \dots, c_k, v_{k+1} \rangle$, where c_i is a connection with endpoints v_i and v_{i+1} (for $i = 1, \dots, k$). P is termed a *virtual cycle* if $v_1 = v_{k+1}$. We denote by $S(P)$ the set $\{c_1, c_2, \dots, c_k\}$ of connections in P . The routing function \mathcal{R} is naturally generalized to apply to virtual paths (and cycles) by concatenating the corresponding paths of connections. A virtual path (or cycle) P is termed *plain* if $\mathcal{R}(P)$ is a simple path (or cycle) in the network.

A *design* D for a set of connections C in a network G is a set of connections that subsumes C (i.e., $C \subseteq D$). A *ring partition design* D for a set of connections C satisfies $D = \cup_{t \in T} S(P_t)$, where every $P_t, t \in T$, is a plain virtual cycle and $S(P_{t_1}) \cap S(P_{t_2}) = \emptyset$ for every $t_1, t_2 \in T$. The partition $\{P_t\}_{t \in T}$ is termed *the ring partition of the design* D . For a design D , $\text{cost}(D) = |D|$, i.e., the number of lightpaths in the design.

The *minimum cost ring partition design* problem is formally defined as follows. The input is a graph G and a set of connections C in G . The goal is to find a ring partition design D for C that minimizes $\text{cost}(D)$. The corresponding decision problem

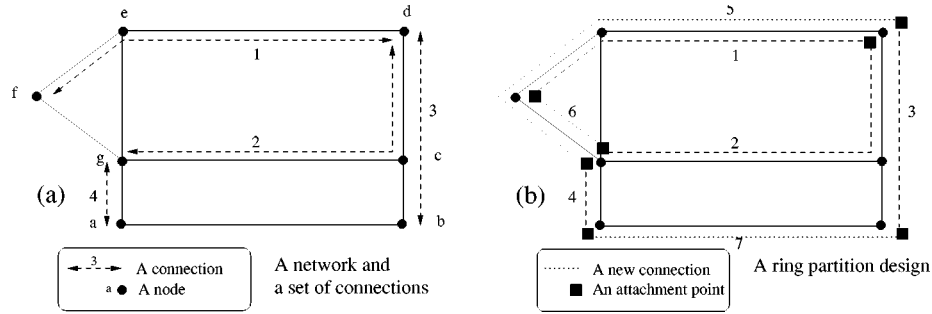


Fig. 1. The MCRPD problem.

is to decide for a set of connections C in G and a positive integer s whether there is a ring partition design D for C such that $\text{cost}(D) \leq s$.

$\text{MCRPD}_{\mathcal{G}}$ denotes the version of the problem in which the input is restricted to a family \mathcal{G} of networks (e.g., the family \mathcal{R} of rings).

Fig. 1 is an example of the MCRPD problem, where (a) shows an instance with an initial set of size four and (b) shows a solution that consists of two rings and three new connections. The cost of the solution is thus seven.

III. THE MCRPD PROBLEM

In this section, we begin our study of the MCRPD problem by providing some negative results regarding the tractability and approximability of the problem.

We say that a family of topologies $\mathcal{G} = G_1, G_2, \dots$ has the *unbounded cycle* (UBC) property if there exists a constant k such that for every n , there exists a graph $G_{i_n} \in \mathcal{G}$ with size $O(n^k)$ that contains a cycle of length n . Examples for families of topologies having the UBC property are the family \mathcal{R} of ring topologies and the family of complete graphs.

Theorem 1: The $\text{MCRPD}_{\mathcal{G}}$ problem is NP-hard for every family of topologies \mathcal{G} having the UBC property.

Proof: We prove Theorem 1 in two steps. In Step 1, we prove that $\text{MCRPD}_{\mathcal{R}}$ is NP-hard, and in Step 2, we extend the result to every family of topologies with the UBC property. ■

Step 1: We prove that the $\text{MCRPD}_{\mathcal{R}}$ problem is NP-hard by a polynomial transformation from the *circular arc coloring problem*, which is known to be NP-hard [8].

A graph G is termed a *circular arc graph* if its nodes can be placed in a one-to-one correspondence with a set \mathcal{F} of routes (paths) in a ring in such a way that two nodes of G are joined by an edge iff the corresponding two routes intersect.

The circular arc coloring problem is formalized as follows (definitions are adapted from definitions in this paper). A *set* \mathcal{F} of circular arcs in a ring R_m is a set $\{A_0, A_1, \dots, A_{n-1}\}$, where each A_i is an ordered pair (a_i, b_i) of positive integers, with $a_i \neq b_i$. For our needs, there is no difference between circular arcs and connections. Note that we can assume also in this case that $m \leq 2n$. The circular arc coloring problem is formally defined as follows. Given a pair $I = (R_m, \mathcal{F})$, where \mathcal{F} is a set of circular arcs in the ring R_m , and a positive integer K , can \mathcal{F} be partitioned into K classes so that no two arcs in the same class intersect?

Recall that load l_e of an edge e is the number of connections (or circular arcs) in the set C that use e , and let l_I be the maximum load of an edge (for a given instance $I = (G, C)$). We say that an instance I is *full* if it satisfies $l_e = l_I$ for every edge. Given an instance $I = (R_m, C)$, for every $t \geq l_I$, $\text{full}_t(I) = (R_m, C \cup C')$ is a full instance with load t , which is constructed from I by adding connections of length one. Formally, $C' = \cup_{x=0}^{m-1} C_x$, where $C_x = \cup_{i=1}^{t-l_{\{x, x+1\}}} \{(x, x+1)\}$, for every $x = 0, \dots, m-1 \pmod{m}$.

For an instance $I = (R_m, \mathcal{F})$ of the circular arc coloring problem and a positive integer K , we construct the instance $\text{full}_K(I) = (R_m, C)$ of the $\text{MCRPD}_{\mathcal{R}}$ problem, where $C = \mathcal{F} \cup C'$ (and the set C' contains the appropriate connections of length 1). Note that we can assume that K satisfies $l_C \leq K \leq |\mathcal{F}|$ since otherwise the answer for this instance of the circular arc coloring problem can be determined immediately. Note that $|C| \leq K \cdot m$; thus the construction is polynomial (since $K \leq |\mathcal{F}|$ and $m \leq 2 \cdot |\mathcal{F}|$). See Fig. 2 for an example of the transformation.

Proposition 1: The set \mathcal{F} can be partitioned into K classes of pair-wise nonintersecting arcs iff there is a ring partition design D for C with $\text{cost}(D) = |C|$.

Proof: In one direction, assume that there is a ring partition design $D = \cup_t S(C_t)$ for C such that $\text{cost}(D) = |C|$. Clearly, this situation where the cost of a design is equal to the size of the set of connections C (i.e., no new connections are added) is possible only if the instance is full (as is the case with $\text{full}_K(I)$) and only if the number of rings in D is equal to the load of the instance, i.e., $|T| = K$ in our case. Such a solution D induces a partition of \mathcal{F} to K classes of pairwise nonintersecting arcs in the obvious way, since routes of connections in the same virtual cycle C_t do not intersect.

In the other direction, consider a partition of \mathcal{F} to K classes $\{C_i\}_{i=1}^K$ such that in every class all the arcs do not intersect. Since $\text{full}_K(I)$ is a full instance with load K , and since C' contains only paths of length one, it can be easily seen that from such a partition of \mathcal{F} into K classes one can construct a ring partition of C simply by completing every class to a cycle by using paths (of length 1) from C' . We get a ring partition design D for C with $\text{cost}(D) = |C|$. ■

Step 2: Let \mathcal{G} be a family of topologies with the UBC property. We prove that the $\text{MCRPD}_{\mathcal{G}}$ problem is NP-hard by a polynomial transformation from the $\text{MCRPD}_{\mathcal{R}}$ (which is NP-hard by Step 1).

Let $I = (R_m, C)$ be an instance of the $\text{MCRPD}_{\mathcal{R}}$ problem. Let $G_{i_m} \in \mathcal{G}$ be the graph in \mathcal{G} that contains a cycle of length m (by the definition of the UBC property, such graph G_{i_m} exists in

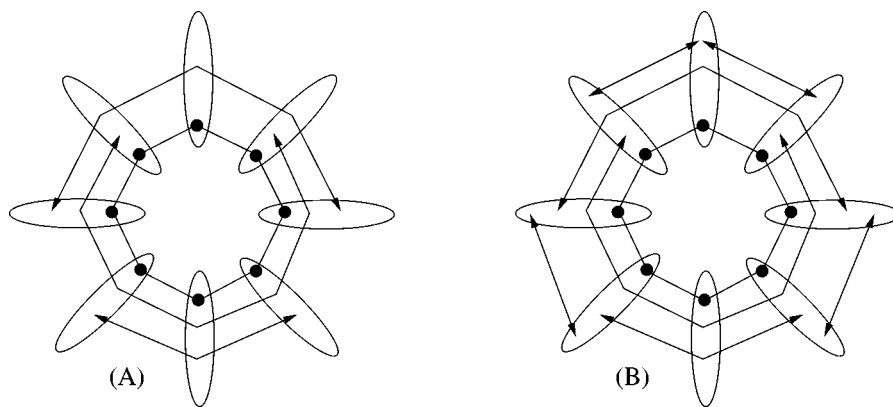


Fig. 2. (a) An instance I of the circular arc coloring problem. (b) The instance $\text{full}_2(I)$ of the $\text{MCRPD}_{\mathcal{R}}$ problem.

\mathcal{G}). We construct an instance $I' = (G_{i_m}, C')$ of the $\text{MCRPD}_{\mathcal{G}}$ problem in the obvious way such that there is a one-to-one corresponds (which preserves the intersection pattern) between paths from C in R_m and paths in the ring R'_{i_m} (of size m), which is a subgraph of G_{i_m} .

Clearly, for every ring partition design D for C there is a ring partition design D' for C' with the same cost. In the other direction, let $D' = \cup_{t \in T} S(P'_t)$ be a ring partition design for C' . Note that by the construction, all connections in the set C' have routes that are paths in the ring R'_{i_m} , but new connections might have any route in G_{i_m} . It is clear that we can construct from D' a ring partition design D such that all connections in D have routes on the ring and $\text{Cost}(D) \leq \text{Cost}(D')$ as follows. We can take the induced subgraph partition of D' (see Section IV-C) and complete every virtual path in it (whose route must be a path on the ring) to a virtual cycle by adding one new connection with the complement route on the ring. By the observations in Section IV-C, $\text{Cost}(D) \leq \text{Cost}(D')$.

We note that the definition of the UBC property can be relaxed so as to include families of topologies in which for every $n > 0$, there is a graph of size $\Theta(n^{k_1})$ that includes a cycle of length $\Theta(n^{k_2})$ for some constants K_1, K_2 . The same proof holds for the relaxed definition with slight technical extensions. ■

We continue by studying approximation algorithms for the MCRPD problem. A trivial approximation algorithm is achieved by adding for every connection c a new disjoint connection between c 's endpoints. Note that if there is no such route, then there is no ring partition design for this instance. The resulting ring partition design will include virtual cycles, each with two connections, one of which belongs to the initial set C . For an algorithm A , we denote by $A(I)$ the value of a solution found by A for an instance I and by $\text{OPT}(I)$ the value of an optimal solution. Clearly, $\text{TRIV}(I) = 2n \leq \text{OPT}(I) + n$, for every instance $I = (G, C)$ of MCRPD, where $|C| = n$. A question that arises naturally is whether there exists an approximation algorithm A for the MCRPD problem that guarantees $A(I) \leq \text{OPT}(I) + n^\alpha$ for some constant $\alpha < 1$. We give a negative answer for this questions (for every constant $\alpha < 1$).

Theorem 2: Let \mathcal{G} be any family of topologies having the UBC property. Then for any constant $\alpha < 1$, $\text{MCRPD}_{\mathcal{G}}$ has no polynomial-time approximation algorithm A that guarantees $A(I) \leq \text{OPT}(I) + n^\alpha$ (unless $P = \text{NP}$).

Proof: We prove Theorem 2 for the case of a ring. The proof is generalized to every family of topologies having the UBC property by using the same transformation as in Step 2 in the proof of Theorem 1. Assume to the contrary that there is a constant $\alpha < 1$ and a polynomial approximation algorithm A such that for every instance I of the $\text{MCRPD}_{\mathcal{R}}$ problem, $A(I) \leq \text{OPT}(I) + n^\alpha$. We show a polynomial algorithm A' that optimally solves the $\text{MCRPD}_{\mathcal{R}}$ problem, in contradiction with Theorem 1.

Given an instance $I = (R_m, C)$ (where $|C| = n$) of the $\text{MCRPD}_{\mathcal{R}}$ problem, we construct a new instance $I' = (R_{m'}, C')$, where $|C'| = n \cdot n^\beta, m' = m \cdot n^\beta$, and β is a constant whose value will be determined later as follows. The ring $R_{m'}$ is divided into n^β sections $J_0, \dots, J_{n^\beta-1}$, each of size m . The section J_t contains the interval of nodes $m \cdot t, \dots, (m+1) \cdot t - 1$, for every $t = 0, \dots, n^\beta$. The set of connections C' is the union of n^β sets of connections, where all the sets are isomorphic to the original set C but shifted in steps of m to different sections on the ring. Formally, $C' = \cup_{j=0}^{n^\beta-1} C^j$, where $C^j = \{(a_i^j, b_i^j) \mid i = 0, \dots, n-1\}$, $a_i^j = a_i + j \cdot m$, and $b_i^j = b_i + j \cdot m$, for every $j \in 0, \dots, n^\beta - 1$. See Fig. 3 for an example of the construction.

Recall that a canonical ring partition design satisfies that the number of new connections in every virtual ring is at most one. By the discussion in Section IV-C, for every ring partition design there is an equivalent conical ring partition design, therefore without loss of generality we can consider only canonical ring partition designs. Consider a canonical ring partition design $D = \cup_{t \in T} S(P_t)$ for C' . Note that in canonical ring partition designs, every attachment point of two connections in a virtual ring is an endpoint of a connection in the original set. Since every connection in I' has its two endpoints in one of the (disjoint) sections of size m on the ring, all the connections in every virtual cycle $P_t, t \in T$, have their endpoints in the same section. Formally, ■

Observation 1: Consider a canonical ring partition design $D = \cup_{t \in T} S(C_t)$ for the instance $I' = (R_{m'}, C')$. For every virtual cycle $C_t, t \in T$, there exists $j, j \in 0, \dots, n^\beta - 1$, such that every connection $c = (x, y) \in S(C_t)$ satisfies $j \cdot m \leq x, y \leq j \cdot m + (m - 1)$.

Proof: By the above discussion. ■

It follows that D can be partitioned into n^β sets of virtual cycles $C_j, j = 0, \dots, n^\beta$, according to the endpoints of their con-

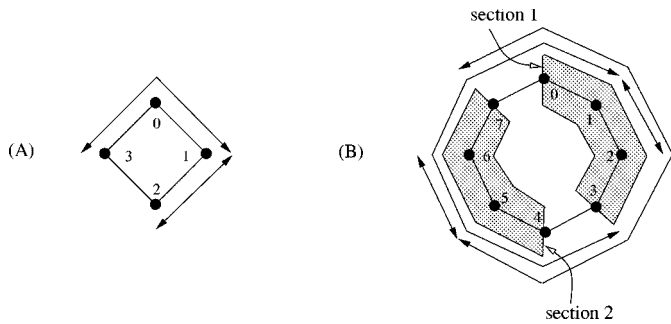


Fig. 3. (a) The input instance I , where $m = 4, n = 2$, and $C = \{(1,2), (3,1)\}$. (b) The new instance I' , where $\beta = 1$, i.e., a multiplication by two.

nections; the family \mathcal{C}_j includes all virtual cycles with endpoints in the j th section on the ring. Every family \mathcal{C}_j naturally induces a canonical ring partition design D_j to the original set C as follows: $D_j = \{(a,b) \mid (m \cdot j + a, m \cdot j + b) \in S(P_t), P_t \in \mathcal{C}_j\}$.

Claim 1: $\text{OPT}(I') = \text{OPT}(I) \cdot n^\beta$.

Proof: On one direction, one can easily see that from a canonical ring partition design D to the original set C , we can construct a (canonical) ring partition design D' for C' such that $\text{Cost}(D') = \text{Cost}(D) \cdot n^\beta$ simply by multiplying it by n^β in the same manner as we constructed the instance I' from I . Thus, $\text{OPT}(I') \leq \text{OPT}(I) \cdot n^\beta$.

On the other direction, by Observation 5, there is an optimal canonical ring partition design $D' = \cup_{T \in S} S(P_t)$ to C' . Such a design can be partitioned, as explained above, to n^β subsets of connections. We have to show that $D_j = \{(a,b) \mid (m \cdot j + a, m \cdot j + b) \in S(P_t), P_t \in \mathcal{C}_j\}$ is indeed a ring partition design for I . But this is clear from the construction, since for every connection $c = (a,b) \in C$, there is a connection $c' = (m \cdot j + a, m \cdot j + b) \in C'$. Thus, in D' , there is a virtual cycle P_x , such that $c' \in S(P_x)$. From the discussion above, since D' is canonical, $P_x \in \mathcal{C}_j$. It follows that D_j is a ring partition design for the original set C . Now, since $\sum_{i=0}^{n^\beta-1} \text{Cost}(D_j) = \text{Cost}(D')$, we get $\text{OPT}(I) \cdot n^\beta \leq \text{OPT}(I')$. ■

The idea of the algorithm A' that we construct from A should be clear by now. Given an instance $I = (R_m, C)$ of the MCRPD \mathcal{R} problem, the algorithm A' first constructs the instance $I' = (R_{m'}, C')$. It then executes A on the new instance. Let D' be the ring partition design found by A . Let us assume that D' is canonical (otherwise A' can construct in polynomial time a canonical ring partition design from D'). Then D' induces n^β ring partition designs $D_0, \dots, D_{n^\beta-1}$ for C such that $\sum_{j=0}^{n^\beta-1} \text{Cost}(D_j) = \text{Cost}(D')$. The output of A' is a ring partition design D_t for $C, t \in 0, \dots, n^\beta - 1$, such that $\text{Cost}(D_t) = \text{Min}_{j=0}^{n^\beta-1} \text{Cost}(D_j)$. Clearly, for a constant β , A' runs in time polynomial in the size of I .

Claim 2: $A'(I) \cdot n^\beta \leq A(I')$.

Proof: Immediate from the discussion, since $A'(I) = \text{Cost}(D_t) = \text{Min}_{j=0}^{n^\beta-1} \text{Cost}(D_j)$ and $A(I') = \text{Cost}(D') = \sum_{j=0}^{n^\beta-1} \text{Cost}(D_j)$. ■

Now

$$\begin{aligned} A'(I) \cdot n^\beta &\leq A(I') \\ &\leq \text{OPT}(I') + (n \cdot n^\beta)^\alpha \\ &= \text{OPT}(I) \cdot n^\beta + (n \cdot n^\beta)^\alpha. \end{aligned}$$

We get

$$A'(I) \leq \text{OPT}(I) + n^{(\beta+1)\alpha-\beta}.$$

Fixing β such that $\beta > (\alpha)/(1-\alpha)$, we get $n^{(\beta+1)\alpha-\beta} < 1$; thus A' actually finds an optimal solution for every instance I , in contradiction with Theorem 1. We conclude that there is no constant $\alpha < 1$ for which there is an approximation algorithm that guarantees $A(I) \leq \text{OPT}(I) + n^\alpha$ (unless $P = \text{NP}$). ■

The next question is whether there is an approximation algorithm A for MCRPD that guarantees $A(I) \leq \text{OPT}(I) + k \cdot n$, where $k < 1$ is a constant (clearly, the trivial algorithm TRIV satisfies this bound for $k = 1$). In the sequel we answer this question positively for $k = (3/5)$.

IV. A RING-PARTITION APPROXIMATION ALGORITHM

In this section, we provide a polynomial approximation algorithm, the *ring partition algorithm*, for the MCRPD problem. We analyze RPA and show that it guarantees $\text{RPA}(I) \leq \min(\text{OPT}(I) + (3/5) \cdot n, 2n)$ for every instance I (where n is the number of connections in the initial set). We also study some special cases in which better results are achieved.

Informally, the first step of the algorithm is to connect as many endpoints of pairs of lightpaths with a common node as possible. This is done by constructing a certain graph for every node and finding a maximum matching in it. We get a set of chains (and cycles) of lightpaths. The problem is that some of the chains (and cycles) of lightpaths may traverse nonsimple paths in the physical topology [see example in Fig. 4(a)]. Even if the path traversed by a chain of lightpaths is simple, it might be impossible to complete it to a cycle in the physical topology [see example in Fig. 4(b)]. The second step is then to partition any chain or cycle of lightpaths into a set of chains such that all remaining chains and cycles traverse simple paths in the physical topology that can be completed to a cycle. The last step is to actually complete every chain to a cycle by adding one lightpath per chain. This is done by finding a disjoint path in the physical topology using standard breadth first search (BFS) techniques.

Unless stated otherwise, we assume an arbitrary network topology $G = (V, E)$, where $V = \{v_1, \dots, v_m\}$, and an initial set of connections C in G , where $|C| = n$. We assume that the route $\mathcal{R}(c)$ of every connection c in C is a subpath in some simple cycle in G (observe that this assumption can be verified in polynomial time, and without it there is no ring partition design D for C).

A. Preliminary Constructions

We define some preliminary constructions that are used later for the definition of RPA. Recall that a virtual path P is a sequence $\langle v_1, c_1, v_2, c_2, \dots, c_k, v_{k+1} \rangle$, where c_i is a connection with endpoints v_i and v_{i+1} (for $i = 1, \dots, k$). P is termed a *virtual cycle* if $v_1 = v_{k+1}$. The pair of connections c_i and c_{i+1} are termed *attached at node* v_{i+1} in P (or simply, *attached in* P). If P is a virtual cycle, then the pair c_1 and c_k are also considered *attached (at node* $v_{k+1})$ in P .

Let C be a set of connections in G , and let v be a node in G . We denote by $C(v) \subseteq C$ the set of connections for which

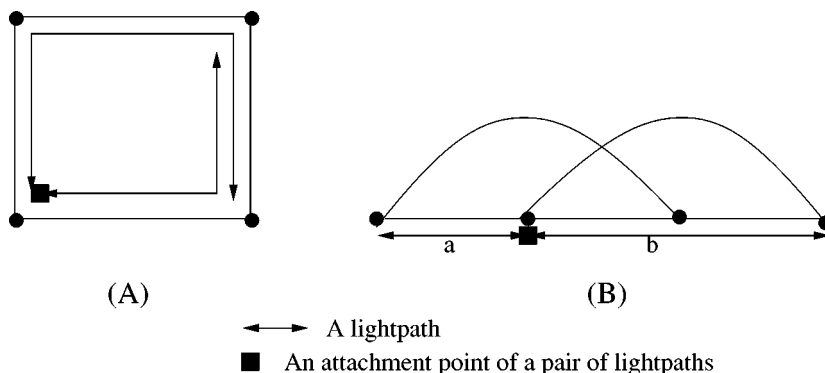


Fig. 4. (a) A chain of lightpaths $\langle a, b \rangle$ that traverses a nonsimple path in the physical topology. (b) A chain of lightpaths $\langle a, b \rangle$ that cannot be completed to a simple cycle in the physical topology.

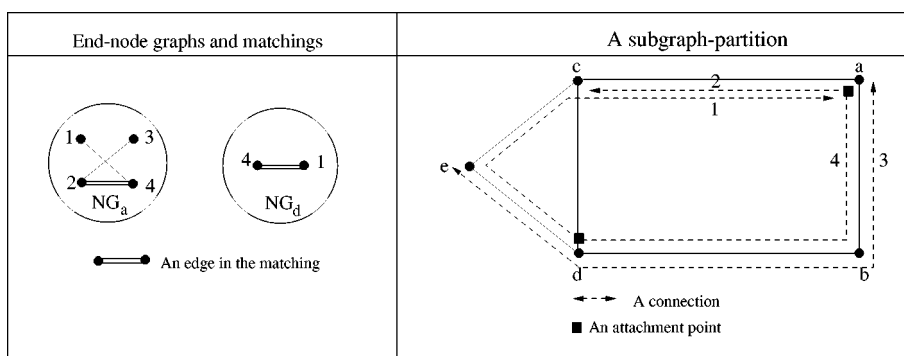


Fig. 5. A graph, a set of connections, a matching set (where only matchings in nontrivial end-node graphs are shown), and the equivalent subgraph partition.

v is an endpoint. Let Q be the symmetric binary relation over the set C of connections that is defined as follows: $(c_1, c_2) \in Q$ iff c_1 and c_2 are disjoint and there exists a simple cycle in G that contains both routes $\mathcal{R}(c_1)$ and $\mathcal{R}(c_2)$. Then Q defines an *end-node graph* $NG_v = (NV_v, NE_v)$ for every node v , where the set of nodes NV_v is $C(v)$ and NE_v is the set of edges, as follows. For every pair of connections $c_i, c_j \in C(v)$, $\{c_i, c_j\} \in NE_v$ iff $(c_i, c_j) \in Q$. A *matching* for a graph $G = (V, E)$ is a set $E' \subseteq E$ such that no two edges in E' share a common endpoint. A *maximum matching* is a matching of maximum size. We denote by $\text{match}(G)$ the size of a maximum matching for G . A matching in an end-node graph NG_v for a node v describes a set of attachments of pairs of connections (which satisfy Q) in v .

Consider a graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_m\}$, and a set of connections C in G . A *matching set* for G and C is a set of matchings $\mathcal{E} = \{NE'_{v_1}, NE'_{v_2}, \dots, NE'_{v_m}\}$, where $NE'_{v_i} \subseteq NE_{v_i}$ is a matching in the end-node graph NG_{v_i} (see Fig. 5 as an example).

A *subgraph partition* $\mathcal{G} = \mathcal{G}_p \cup \mathcal{G}_c$, for a set of connections C , is a partition of the connections in C into virtual paths and cycles (which are also termed *subgraphs*) as follows. Recall that $S(g)$ is the set of connections that are included in a virtual path (or cycle) g . \mathcal{G}_p is a set of virtual paths, \mathcal{G}_c is a set of virtual cycles, $C = \cup_{g \in \mathcal{G}} S(g)$, and $S(g_1) \cap S(g_2) = \emptyset$ for every $g_1, g_2 \in \mathcal{G}$. Note that the ring partition $\{P_t\}_{t \in T}$ of a ring partition design

$D = \cup_{t \in T} S(P_t)$ is actually a subgraph partition for D (where $\mathcal{G} = \mathcal{G}_c, \mathcal{G}_p = \emptyset$). In general, the virtual paths and cycles in a subgraph partition might not be plain.

Note that there is a one-to-one correspondence between matching sets and subgraph partitions as follows. Consider a matching set $\mathcal{E} = \{NE'_{v_1}, NE'_{v_2}, \dots, NE'_{v_m}\}$ and a subgraph partition $\mathcal{G} = \mathcal{G}_p \cup \mathcal{G}_c$ for a set of connections C in G . \mathcal{E} and \mathcal{G} are termed *equivalent* if the following condition is satisfied. For every pair of connections $c_1, c_2 \in C$, there exists a subgraph $g, g \in \mathcal{G}$, such that c_1 and c_2 are attached at node v_i in g , iff $\{c_1, c_2\} \in NE'_{v_i}$.

For a matching set \mathcal{E} we denote by $\mathcal{G}_{\mathcal{E}}$ the (unique) equivalent subgraph partition. Similarly, $\mathcal{E}_{\mathcal{G}}$ is the (unique) equivalent matching set for a given subgraph partition \mathcal{G} . Clearly, for a matching set \mathcal{E} , $\mathcal{E}_{\mathcal{G}_{\mathcal{E}}} = \mathcal{E}$. As an example, see Fig. 5.

B. Ring Partition Algorithm (RPA)

We present a ring partition algorithm that finds a ring partition design for a set of connections C in G in four main stages. First, the end-node graph NG_{v_i} is constructed and a maximum matching in it is found for every node $v_i, i = 1, \dots, m$. This defines a maximum matching set \mathcal{E} . Then, the equivalent subgraph partition $\mathcal{G} = \mathcal{G}_{\mathcal{E}}$ is constructed.

Next, we partition every nonplain virtual path or virtual cycle in \mathcal{G} to plain virtual paths. In addition, we make sure that for

every virtual path $P \in \mathcal{G}$, there is a simple cycle in G in which $\mathcal{R}(P)$ is a subpath. Lastly, the subgraph partition is completed to a ring partition by adding for every virtual path $P \in \mathcal{G}$ a connection that completes it to a plain virtual cycle. Following is the description of RPA followed by an informal description of the operations taken by its main functions.

```

1: RPA( $G, C$ )
2: ( $\mathcal{G}_p, \mathcal{G}_c$ ) := ConstructPartition( $G, C$ )
3: ( $\mathcal{G}_p, \mathcal{G}_c$ ) := AdjustPartition( $\mathcal{G}_p, \mathcal{G}_c, G$ )
4:  $D$  :=  $C \cup$  CompletePartition( $\mathcal{G}_p, \mathcal{G}_c, G$ )
5: return  $D$ 
6: ConstructPartition( $G, C$ )
7: for every  $i \in 1, \dots, m$ 
8:   construct  $NG_{v_i} = (NV_{v_i}, NE_{v_i})$ 
9:   find maximum matching  $NE'_{v_i} \subseteq NE_{v_i}$ 
10:   $\mathcal{E} := \{NE'_{v_1}, NE'_{v_2}, \dots, NE'_{v_m}\}$ 
11:  construct the equivalent subgraph-partition  $\mathcal{G}_{\mathcal{E}} = (\mathcal{G}_p, \mathcal{G}_c)$ 
12:  return ( $\mathcal{G}_p, \mathcal{G}_c$ )
13: AdjustPartition( $\mathcal{G}_p, \mathcal{G}_c, G$ )
14: for every  $P \in \mathcal{G}_p \cup \mathcal{G}_c$ 
15:   $\mathcal{G}_c := \mathcal{G}_c \setminus \{P\}$  /* in case  $P$  is a cycle */
16:   $C_P :=$  Partition( $P$ )
17:   $\mathcal{G}_p := (\mathcal{G}_p \setminus \{P\}) \cup C_P$ 
18: for every  $P \in \mathcal{G}_p$ 
19:  if ( $cycle(P)$ ) then
20:     $\mathcal{G}_p := \mathcal{G}_p \setminus \{P\}$ 
21:     $\mathcal{G}_c := \mathcal{G}_c \cup \{P\}$ 
22: return ( $\mathcal{G}_p, \mathcal{G}_c$ )
23: CompletePartition( $\mathcal{G}_p, \mathcal{G}_c, G$ )
24:  $D' := \emptyset$ 
25: for every  $P \in \mathcal{G}_p$ 
26:   $P^c :=$  findDisjoint( $P$ )
27:   $D' := D' \cup \{P^c\}$ 
28: return  $D'$ 
29: Partition( $P$ )
30: Assume that  $P := \langle v_1, c_1, v_2, c_2, \dots, v_l, c_l, v_{l+1} \rangle$ 
31:  $C_P := \emptyset$ ;  $first := 1$ 
32: for  $i := 1$  to  $l$ 
33:   $P' := \langle v_{first}, c_{first}, \dots, c_i, v_{i+1} \rangle$ 
34:  if ( $\neg(plain(P') \wedge cycleExists(P'))$ ) then
35:     $C_P := C_P \cup \langle v_{first}, c_{first}, \dots, c_{i-1}, v_i \rangle$ 
36:     $first := i$ 
37: return  $C_P \cup \langle v_{first}, c_{first}, \dots, c_l, v_{l+1} \rangle$ 

```

The function *ConstructPartition* first constructs the end-node graphs. The algorithm to construct the end-node graphs is straightforward and is not elaborated. It consists of determining for every pair of connections with a common endpoint whether they are disjoint and whether the path that is formed by concatenating them can be completed to a simple cycle in G . This could be done using standard BFS techniques (see, e.g., [7]). *ConstructPartition* then finds maximum matchings in the end-node graphs. Efficient algorithms for finding maximum matchings in graphs can be found in, e.g., [14] (for a survey, see [17, pp. 580–588]). Finding maximum matching in the end-node graphs corresponds to connecting as many as possible

endpoints of pairs of disjoint lightpaths. Last, the construction of the equivalent subgraph partition is straightforward.

The function *AdjustPartition* partitions every virtual path and virtual cycle in the subgraph partition using the function *Partition*. After the partition, every virtual path is plain and can be completed to a simple cycle in G . Every virtual path is then checked, and if it is actually a cycle (i.e., its endpoints are equal), then it is inserted into \mathcal{G}_c .

The task of *Partition* is to partition a virtual path (or cycle) to a set $\{P_1, \dots, P_l\}$ of plain virtual paths, such that for every P_i , $\mathcal{R}(P_i)$ is a subpath in some simple cycle in G . The function *cycleExists*(P) returns *true* if there is a disjoint path in G between P 's endpoints. The function *cycle*(P) returns *true* if the endpoints of a given virtual path are equal.

Lastly, the function *CompletePartition* completes every virtual path in \mathcal{G}_p to a virtual cycle by adding a new disjoint connection P^c between P 's endpoints.

C. Correctness and Analysis

We first present four observations that are used for the proof of the main theorem (Theorem 3). Observation 2 shows a connection between the sizes of matching sets and the equivalent subgraph partitions.

Observation 2: Let $\mathcal{E} = \{NE'_{v_1}, NE'_{v_2}, \dots, NE'_{v_m}\}$ be a matching set for a set of connections C in $G = (V, E)$, where $|C| = n$ and $V = \{v_1, \dots, v_m\}$. Let $\mathcal{G}_{\mathcal{E}} = \mathcal{G}_p \cup \mathcal{G}_c$ be the equivalent subgraph partition. Then $|\mathcal{G}_p| = n - \sum_{i=1}^m |NE'_{v_i}|$.

Proof: Let an *attachment point* in $\mathcal{G}_{\mathcal{E}}$ be an ordered pair $(\{c_1, c_2\}, v)$, where the connections c_1 and c_2 are attached at node v in some subgraph $g \in \mathcal{G}_{\mathcal{E}}$. Clearly, the number of unique attachment points in a virtual path $P_x \in \mathcal{G}_p$ is one less than the number of connections in P_x , i.e., $|S(P_x)| - 1$. The number of unique attachment points is equal to $|S(P_x)|$ if $P_x \in \mathcal{G}_c$ is a virtual cycle. It follows that the number of unique attachment points is equal to $(\sum_{g \in \mathcal{G}_{\mathcal{E}}} |S(g)|) - |\mathcal{G}_p| = n - |\mathcal{G}_p|$. Now by the definitions, there is a one-to-one correspondence between attachment points and edges in the matchings. It follows that the number of attachment points is equal to the number of edges in the matching set, i.e., $n - |\mathcal{G}_p| = \sum_{i=1}^m |NE'_{v_i}|$. ■

Let $\mathcal{G}(D)$ be a subgraph partition for a set of connections D . The *projection* $\mathcal{G}(D)|_C$ of $\mathcal{G}(D)$ on a set of connections $C \subseteq D$ is a subgraph partition for C , which is obtained from $\mathcal{G}(D)$ by deleting all the connections that are not in C (i.e., all the connections in $D \setminus C$). Note that a virtual path (or cycle) in $\mathcal{G}(D)$ might be cut by this process into few virtual paths. Similarly, let $\mathcal{E}(D)$ be a matching set for D . Then the *projection* $\mathcal{E}(D)|_C$ of $\mathcal{E}(D)$ on a set of connections $C \subseteq D$ is a matching set for C , which is obtained from $\mathcal{E}(D)$ by deleting the end-node graphs (and the matchings) nodes that correspond to connections in $D \setminus C$ and the edges that meet them. Clearly, if $\mathcal{G}(D)$ and $\mathcal{E}(D)$ are equivalent, then so are $\mathcal{G}(D)|_C$ and $\mathcal{E}(D)|_C$.

Consider a ring partition design $D = \cup_{t \in T} S(P_t)$ for a set of connections C . We denote by $\mathcal{G}(D)$ the ring partition $\{P_t\}_{t \in T}$ of D , and by $\mathcal{E}(D)$ the equivalent matching set for D (i.e., $\mathcal{E}(D) = \mathcal{E}_{\mathcal{G}(D)}$). The subgraph partition $\mathcal{G}(D)|_C$ and the matching set $\mathcal{E}(D)|_C$ for the initial set of connections C are termed the *induced subgraph partition* and the *induced matching set*, respectively (note that they are equivalent).

Observation 3 associates the cost of ring partition designs with the sizes of the induced matching sets and subgraph partitions.

Observation 3: Let $D = \cup_{t \in T} S(P_t)$ be a ring partition design for a set of connections C in a physical topology $G = (V, E)$, where $|C| = n$ and $|V| = m$. Let $\mathcal{E}(D)|_C = \{NE'_{v_1}, NE'_{v_2}, \dots, NE'_{v_m}\}$ and $\mathcal{G}(D)|_C = \mathcal{G}_p \cup \mathcal{G}_c$ be the induced matching set and subgraph partition for C . Then $\text{cost}(D) \geq n + |\mathcal{G}_p| = 2n - \sum_{i=1}^m |NE'_{v_i}|$.

Proof: By the definitions, $\text{Cost}(D) = \sum_{t \in T} S(P_t)$. Let $\text{new}(P_t)$ be the number of new connections in the virtual cycle P_t , i.e., $\text{new}(P_t) = S(P_t) \cap (D \setminus C)$. Clearly, $\text{Cost}(D) = n + \sum_{t \in T} \text{new}(P_t)$. Consider now the induced subgraph partition $\mathcal{G}(D)|_C = \mathcal{G}_p \cup \mathcal{G}_c$. Recall that it is obtained from D by deleting all the new connections. In this process, a virtual cycle in the ring partition might be cut into few virtual paths. Clearly, the number of such virtual paths for each virtual cycle is at most the number of new connections in it. It follows that $|\mathcal{G}_p| \leq \sum_{t \in T} \text{new}(P_t)$; thus $\text{Cost}(D) \geq n + |\mathcal{G}_p|$. By Observation 2, $n + |\mathcal{G}_p| = 2n - \sum_{i=1}^m |NE'_{v_i}|$. Note that strict inequality occurs when two new connections are attached in one of the virtual cycles. ■

A *maximum matching set* is a matching set $\mathcal{E} = \{NE'_{v_1}, \dots, NE'_{v_m}\}$ for a set of connections C , such that the matching NE'_{v_i} is a maximum matching for the end-node graph NG_{v_i} , for every $i = 1, \dots, m$. Recall that $\text{match}(G)$ is the size of a maximum matching for G . Observation 4 is a lower bound on the value of an optimal solution.

Observation 4: Every ring partition design D for C satisfies $\text{cost}(D) \geq 2n - \sum_{i=1}^m \text{match}(NG_{v_i})$ (where n and m are defined as above).

Proof: Let $D = \cup_{t \in T} S(P_t)$ be a ring-partition design for C . Note that every two connections that are attached in a virtual cycle $P_t, t \in T$, in the design satisfy the relation Q , i.e., they are disjoint and there is a simple cycle that contains both routes. Clearly, the same holds also for the induced subgraph partition $\mathcal{G}(D)|_C$ and matching set (since we only delete connections). Consider the equivalent matching set $\mathcal{E}(D)|_C = \{NE'_{v_1}, NE'_{v_2}, \dots, NE'_{v_m}\}$. It follows that NE'_{v_i} is actually a matching in the end-node graph NG_{v_i} , for $i = 1, \dots, m$, and thus $|NE'_{v_i}| \geq \text{match}(NG_{v_i})$. It follows, from Observation 3, that $\text{Cost}(D) \geq 2n - \sum_{i=1}^m |NE'_{v_i}| \geq 2n - \sum_{i=1}^m \text{match}(NG_{v_i})$. ■

Consider a ring-partition design $D = \cup_{t \in T} S(P_t)$ for a set of connections C in G . Let $\text{new}(P_t)$ be the number of new connections in $S(P_t)$ (i.e., connections in $S(P_t) \cap (D \setminus C)$). A *canonical ring-partition design* satisfies that $\text{new}(P_t) \leq 1$ for every $t \in T$. Note that it is always possible to construct from a given ring-partition design D a canonical ring-partition design D' such that $\text{Cost}(D') \leq \text{Cost}(D)$ as follows. Let $\mathcal{G}(D)|_C = \mathcal{G}_p \cup \mathcal{G}_c$ be the induced subgraph partition of D . To construct a canonical ring partition design D' with at most the same cost, we complete every virtual path in \mathcal{G}_p to a plain virtual cycle by adding one new connection. (This is always doable since every virtual path in \mathcal{G}_p is plain and is included in some simple cycle in G). From the discussion above, $\text{Cost}(D') = n + |\mathcal{G}_p| \leq \text{Cost}(D)$. Observation 5 follows.

Observation 5: If there is a ring partition design for a set of connections C in G , then there is a canonical ring-partition design with minimum cost.

It can be proved that Observation 3 holds for canonical ring-partition designs D' with equality, i.e., $\text{cost}(D') = n + |\mathcal{G}_p|$. It is therefore sometimes convenient to consider for simplicity only canonical ring-partition designs.

We are now ready to prove the main theorem.

Theorem 3: $\text{RPA}(I) \leq \min(\text{OPT}(I) + (3/5) \cdot n, 2n)$, for every $I = (G, C)$, where $|C| = n$.

Proof: For the analysis, we denote by \mathcal{G}_p^1 and \mathcal{G}_c^1 the sets \mathcal{G}_p and \mathcal{G}_c right after the execution of *ConstructPartition*, and by \mathcal{G}_p^2 and \mathcal{G}_c^2 the corresponding sets right after the execution of *AdjustPartition*. ■

We now examine the partition procedure *Partition*. Recall that the end-node graphs are constructed with respect to the relation Q , which is *true* for a pair of connections c_1 and c_2 iff their routes $\mathcal{R}(c_1)$ and $\mathcal{R}(c_2)$ are disjoint and there is a simple cycle that contains both routes (as subpaths). Consider a virtual path $P = \langle v_1, c_1, v_2, c_2, \dots, v_{l-1}, c_l, v_l \rangle \in \mathcal{G}_p^1$. Since P is a virtual path in the equivalent subgraph partition \mathcal{G}_E , it holds that $(c_i, c_{i+1}) \in Q$, for every $i = 1, \dots, l-1$. Let C_P be the set of virtual paths which is the output of *Partition(P)*. By the above discussion, and by the definition of *Partition*, at most one virtual path in C_P contains less than two connections. Such a virtual path can be only the last one, which contains the connection c_l . Let $n_P = |S(P)|$ (i.e., the number of connections in the virtual path P). Let $m_P = |C_P|$ (i.e., the number of plain virtual paths that are the result of applying the partition procedure on P). It follows that $m_P \leq \lfloor n_P + 1/2 \rfloor$.

Now consider a nonplain virtual cycle $P \in \mathcal{G}_c^1$. Then, by the same considerations, $m_P \leq \lfloor n_P + 1/2 \rfloor$, where n_P and m_P are defined similarly.

Let $\mathcal{G}_c' \subseteq \mathcal{G}_c^1$ and $\mathcal{G}_c'' \subseteq \mathcal{G}_c^1$ be the sets of nonplain virtual cycles with, respectively, odd and even number of connections after *ConstructPartition*. Note that *CompletePartition* adds one new connection for every virtual path $P \in \mathcal{G}_p^2$. We get

$$\begin{aligned} \text{RPA}(I) &= |\mathcal{G}_p^2| + n \\ &\leq \sum_{P \in \mathcal{G}_c'} \left(\frac{n_P}{2} + \frac{1}{2} \right) + \sum_{P \in \mathcal{G}_c''} \left(\frac{n_P}{2} \right) \\ &\quad + \sum_{P \in \mathcal{G}_p^1} \left(\frac{n_P}{2} + \frac{1}{2} \right) + n \\ &\leq \frac{3n}{2} + \frac{1}{2} |\mathcal{G}_c'| + \frac{1}{2} |\mathcal{G}_p^1|. \end{aligned}$$

Observe that a nonplain virtual cycle in \mathcal{G}_p^1 contains at least four connections, since otherwise clearly there are two consecutive connections that are not disjoint in the cycle, which is not possible by the definition of the algorithm. It follows that $|\mathcal{G}_c'| \leq (n/5)$. We get $\text{RPA}(I) \leq n + (3/5) \cdot n + (1/2) |\mathcal{G}_p^1|$. Now, by Observation 4, we can show that $\text{OPT}(I) \geq n + |\mathcal{G}_p^1|$ (since in the first step RPA finds maximum matchings in the end-node graphs). Thus, $\text{RPA}(I) \leq \text{OPT}(I) + (3/5) \cdot n$.

Observe that RPA constructs a canonical solution, i.e., there is at most one new connection in every ring. Clearly, there is at least one connection from the initial set in every ring. It follows that $\text{RPA}(I) \leq 2n$. ■

Note that since $\text{OPT}(I) \geq n$, this is actually better than an 8/5-approximation.

The time complexity depends on the exact format of the input for the algorithm and the data structures that are used in order to represent the physical topology, the set of connections, and the auxiliary combinatorial constructions (i.e., the end-node graphs and the subgraph partition). It is clear, however, that this time is polynomial in the size of C and G . It is well known that it takes $O(\sqrt{|V|} \cdot |E|)$ time to find a maximum matching in a graph $G = (V, E)$ ([14]) and that it takes $O(|E|)$ time to find whether two paths are disjoint, or whether there exists a disjoint path between a given path's endpoints. For special topologies, these tasks can be significantly simpler. For instance, clearly in the ring physical topology case, every plain virtual path can be completed to a plain virtual cycle; thus the relation Q can be simplified to $Q(c_1, c_2) = \text{disjoint}(c_1, c_2)$. The end-node graphs are bipartite, and finding maximum matchings in bipartite graphs is considerably easier ([17]). Also, to find a disjoint path between the endpoints of a given simple path is trivial. In any case, for the applications of RPA for the design of optical networks, time efficiency is not crucial since the algorithm is applied only in the design stage of the network, and it is reasonable to invest some preprocessing time once in order to achieve better network designs.

Some practical considerations for using the RPA in real networks are due. Note that in some "bad" cases $\text{OPT}(I) + (3/5) \cdot n$ might be larger than $2n$, in which case the RPA will not outperform the trivial algorithm. (Recall that the trivial algorithm simply adds one new lightpath for every lightpath in the original set.) From the description of the RPA, it is clear that this poor result will occur in the following two cases: 1) if there are no pairs of lightpaths that share a common endpoint or 2) if every pair of lightpaths with a common endpoint violates the *disjointness* or the *completion* conditions (namely, they do not traverse disjoint routes in the underlying topology or their route cannot be completed to a simple cycle). However, especially in regional area networks, it is unlikely that this will be the case; there will be many nodes that are end-nodes of more than one lightpath, and it is expected that many pairs of lightpaths can be completed to cycles (a common physical topology in regional area networks is a tree of rings in which any two disjoint lightpaths on the same ring can be completed to a cycle). A simple example to illustrate the behavior of RPA relative to the trivial algorithm is a ring $\langle 0, \dots, n-1 \rangle$ with $n/2$ disjoint lightpaths of length 2 in the original set (e.g., between nodes 0 and 2, 2 and 4, etc.). The trivial algorithm will add one new lightpath for every lightpath in the original set, while RPA will connect every two consecutive lightpath; thus, will add one new lightpath in the case that n is odd and no new lightpaths otherwise. Therefore, the cost of the design constructed by RPA is 50% less than the design constructed by the trivial algorithm.

D. Special Cases

In this section, we study some special cases of the MCRPD problem. Due to space limitations, proofs are omitted in this section.

1) *Optimal Cases*: Since the MCRPD problem is NP-hard (Theorem 1), it is natural to try and find restricted families of topologies for which it can be solved in polynomial time. Un-

fortunately, we actually proved in Theorem 1 that the MCRPD problem is NP-hard for every family of topologies that contains cycles with unbounded length (e.g., rings). Since trees do not support ring partition designs, this implies that the problem is NP-hard for every family of topologies that is of interest in this setting. This observation motivates the question of finding polynomially solvable classes of instances of the problem when taking into account not only the topology of the network but also the initial set of connections.

The *induced graph* $\text{IG}_C = (IV_C, IE_C)$ for a set of connections C in G is the subgraph of G that includes all the edges and nodes of G that are used by at least one connection in C .

A natural question is whether applying restrictions on the induced graph suffices to guarantee efficient optimal solution to the problem. We answer this question negatively by showing that the problem remains NP-hard even for the most simple case where the induced graph is a chain.

Theorem 4: The MCRPD problem is NP-hard even if the induced graph for the set of connections C in G is a chain (or a set of chains).

Proof: See [6]. ■

Next we show that if, in addition to an induced graph with no cycles, the network topology satisfies a certain condition (with respect to the initial set of connections), then RPA finds a minimum cost ring partition design.

Theorem 5: $\text{RPA}(I) = \text{OPT}(I)$ for every instance $I = (G, C)$, which satisfies the following two properties.

- 1) *No Cycles*. The induced graph $\text{IG}_C = (IV_C, IE_C)$ is a forest.
- 2) *Completion*. For every plain virtual path P over C , there is a simple cycle in G that contains the route of P , $\mathcal{R}(P)$, as a subpath.

Proof: See [6]. ■

We discuss below some cases in which the conditions in Theorem 5 are satisfied. A perfectly connected graph (PC) satisfies that every simple path in it is included in a simple cycle. Clearly, if a graph is perfectly connected, then the completion property is satisfied for every initial set of connections. This property also guarantees that there is a ring partition design D for every initial set of connections C . A natural question is to characterize perfectly connected graphs. We give a full characterization of perfectly connected graphs by proving that a graph is PC iff it is randomly Hamiltonian. Randomly Hamiltonian graphs are defined and characterized in [4].

Theorem 6: A graph G is perfectly connected iff it is one of the following: a ring, a complete graph, or a complete bipartite graph with equal number of nodes in both sets.

Proof: See [6]. ■

We note that RPA does not have to be modified in order to give an optimal result for instances that satisfy the conditions in Theorem 5. However, we can benefit from recognizing in advance such instances since in these cases, the procedure *Adjust-Partition* can be skipped. The recognition can be done easily for specific topologies (e.g., rings) and in polynomial time in the general case.

2) *Bounded Length Connections in Rings*: We analyze the performance of RPA in the case of a ring physical topology

when there is a bound on the length of connections in the initial set.

Theorem 7: $\text{RPA}(I) \leq \min(\text{OPT}(I) + (3k/2m) \cdot n, 2n)$, for every instance $I = (R_m, C)$ of $\text{MCRPD}_{\mathcal{R}}$, if for every connection $c \in C$, $\text{length}(\mathcal{R}(c)) \leq k$, for any constant k , $1 \leq k \leq m - 1$.

Proof: See [6]. ■

Note that RPA does not guarantee that the same bound on the length holds also for connections in the ring partition design that is constructed. Indeed, the case where the length of connections in the solution must be bounded is inherently different, and the main results in this paper do not hold for it.

3) *Approximations Based on the Load:* Let the load l_e of an edge $e \in E$ be the number of connections in C that use e , and $l_I = \max_{e \in E} l_e$. Recall the definition of an induced graph $\text{IG}_C = (IV_C, IE_C)$ for a set of connections C in G (Section IV–D1). We add to this definition a weight function $w: IE_C \rightarrow N$ that assigns a weight for every edge that is equal to its load. Although in the worst case the load of an instance is equal to the number of connections $|C|$, usually it is substantially smaller. Therefore, it is interesting to bound the cost of a design as a function of the load.

For this purpose, we assume that the route of every virtual path is a subpath of some simple cycle in G (i.e., the completion property). Let $W = \sum_{e \in E} l_e$. Now consider the weighted induced graph $\text{IG}_C = (IV_C, IE_C, W_C)$ for C . Let T_{\max} be a maximum-weight spanning tree in IG_C , $W_{T_{\max}} = \sum_{e \in T_{\max}} l_e$ and $W_{G-T_{\max}} = W - W_{T_{\max}}$. Following is a description of a modified version of RPA, termed RPA_I . We temporarily remove all connections that use edges that are not in T_{\max} . Next, we find a ring partition design for the remaining set of connections (using RPA). Lastly, we reinsert the removed connections and complete each one of them to a virtual cycle by adding a new connection. We prove that the cost of the resulting ring partition design is larger by at most $2W_{G-T_{\max}}$ than the optimal one. (Note that an improved heuristics might be to repeat the same process with the remaining set of connections.)

Theorem 8: $\text{RPA}_I(I) \leq \text{OPT}(I) + 2W_{G-T_{\max}}$, for every instance $I = (G, C)$ that satisfies the completion property.

Proof: See [6]. ■

For the case of a ring physical topology, it holds $\text{RPA}_I(I) \leq \text{OPT}(I) + \min_{e \in E} l_e$. A slightly better bound is given for this case in [9].

Note that there might be a set of connections C'_{\min} with size smaller than $W_{G-T_{\max}}$ such that the induced graph for the remaining set $C \setminus C'_{\min}$ is a forest. However, we prove in Proposition 9 that finding a minimum set of connections whose removal leaves us with an induced graph with no cycles is NP-hard.

Theorem 9: Find a minimum set of connections $C' \subseteq C$ in a graph G such that the induced graph for the remaining set $C \setminus C'$ does not contain cycles is NP-hard.

Proof: See [6]. ■

V. SUMMARY AND FUTURE RESEARCH

In this paper, we studied the MCRPD problem for which the input is an initial set of lightpaths in a network and the goal is to augment this set by adding lightpaths such that the

result is a ring-partition design with minimum cost. We have shown an approximation algorithm for this problem that guarantees $\text{Cost}(D) \leq \min(\text{OPT} + k \cdot n, 2n)$, where $k = (3/5)$, n is the number of lightpaths in the initial set, and OPT is the cost of an optimal solution. Moreover, we have shown that unless $P = \text{NP}$, there is no approximation algorithm A for this problem that guarantees $\text{Cost}(D) \leq \text{OPT} + n^\alpha$, for every constant $\alpha < 1$. The main open question here is whether the constant k can be improved.

Ring-partition designs are necessary for the near-term future of optical networks since they support a SONET/SDH higher layer network that is configured in the form of rings. However, it is claimed that the core network architecture will have to change and that SONET/SDH will give way to a smart optical layer. Incorporating new technologies, it might be possible to reroute lightpaths dynamically. In these cases, other less restrictive survivability conditions might be considered. While less restrictive survivability conditions might be less expensive to implement, the price to pay is of a more complex protection mechanism that is executed for every failure. The challenge here is two-fold: first, to study the gain in the cost of the network when less restrictive survivability conditions are considered, and second, to study the algorithmic and technological issues of implementing protection mechanisms in the optical domain based on these conditions.

ACKNOWLEDGMENT

The authors would like to thank O. Gerstel for introducing them to this problem and for very helpful discussions and E. Korach for some useful references on randomly Hamiltonian graphs.

REFERENCES

- [1] M. Alanyali and E. Ayanoglu, "Provisioning algorithms for WDM optical networks," in *Proc. IEEE INFOCOM '98*, 1998, pp. 910–918.
- [2] J. Armitage, O. Crochat, and J. Y. Le Boudec, "Design of survivable WDM photonic network," in *Proc. IEEE INFOCOM '97*, 1997, pp. 244–252.
- [3] R. Berry and E. Modiano, "Reducing electronic multiplexing costs in SONETWDM rings with dynamically changing traffic," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1961–1972, Oct. 2000.
- [4] G. Chartrand and H. V. Kronk, "Randomly traceable graphs," *SIAM J. Appl. Math.*, vol. 16, pp. 696–700, 1968.
- [5] A. Chiu and E. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *IEEE J. Lightwave Technology*, vol. 18, pp. 2–12, Jan. 2000.
- [6] T. Eilam, S. Moran, and S. Zaks, "Approximation algorithms for survivable optical networks," Computer Science Dept., Technion, Israel, Tech. Rep. CS-2000-05, 2000.
- [7] S. Even, *Graph Algorithms*. Woodland Hills, CA: Computer Science Press, 1979.
- [8] M. Garey, D. Johnson, G. Miller, and C. Papadimitriou, "The complexity of coloring circular arc graphs and chords," *SIAM J. Discrete Math.*, vol. 1, no. 2, pp. 216–227, 1980.
- [9] O. Gerstel, P. Lin, and G. Sasaki, "Wavelength assignment in WDM ring to minimize cost of embedded SONET rings," in *Proc. IEEE INFOCOM '98*, 1998, pp. 94–101.
- [10] O. Gerstel, R. Ramaswami, and G. H. Sasaki, "Fault tolerant multiwavelength optical rings with limited wavelength conversion," in *Proc. IEEE INFOCOM '97*, 1997, pp. 507–515.
- [11] O. Gerstel, R. Ramaswami, and G. Sasaki, "Cost effective traffic grooming in WDM rings," *IEEE Trans. Networking*, pp. 618–630, 2000.

- [12] Y. Hamazumi, N. Nagatsu, and K. Sato, "Number of wavelengths required for optical networks with failure restoration," *Opt. Fiber Commun.*, pp. 67–68, Feb. 1994.
- [13] L. Liu, X. Li, P. J. Wan, and O. Frieder, "Wavelength assignment in WDM rings to minimize SONET ADMs," in *Proc. INFOCOM*, 2000, pp. 1020–1025.
- [14] S. Micali and V. V. Vazirani, "An $O(\sqrt{V} \cdot E)$ algorithm for finding maximum matching in general graphs," in *Proc. 21st Ann. Symp. Foundations of Computer Science*, 1980, pp. 17–27.
- [15] R. Ramaswami and A. Segall, "Distributed network control for optical networks," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 936–943, 1997.
- [16] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*: Academic Press/Morgan Kaufmann, 1998.
- [17] *Handbook of Theoretical Computer Science*, vol. A, J. van Leeuwen, Ed., MIT Press, Cambridge, MA, 1990.



Tamar Eilam received the B.Sc. and Ph.D. degrees in computer science from The Technion, Haifa, Israel, in 1995 and 2000, respectively.

Since September 2000, she has been a Research Staff Member of the IBM T. J. Watson Research Center, Yorktown Heights, NY. Her research interests include dynamic virtual partition of clusters, automatic and dynamic resource management systems, high availability of distributed systems, optical networks, and compact routing.



Shlomo Moran received the B.Sc. and D.Sc. degrees in mathematics from The Technion, Haifa, Israel, in 1975 and 1979, respectively.

In 1981, he joined the Department of Computer Science at The Technion, where he is currently a full Professor and a Chairman. He has coauthored 70 journal papers covering various aspects of computer science, such as complexity theory, distributed algorithms, communication networks, computational geometry, parallel algorithms, and search methods on the Internet. In addition, he has coauthored many conference papers. He also served as the Advisor of more than 20 graduate students.



Shmuel Zaks received the B.Sc. and M.Sc. degrees in mathematics from The Technion, Haifa, Israel, in 1971 and 1972, respectively, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 1979.

He has been with the Department of Computer Science at The Technion since 1979. His research interests include theory of distributed computing, ATM and optical networks, combinatorial and graph algorithms, and discrete mathematics.