Minimizing Electronic Line Terminals for Automatic Ring Protection in General WDM Optical Networks

G. Calinescu, Ophir Frieder, *Senior Member, IEEE*, and Peng-Jun Wan

***Abstract*—Automatic ring protection provides simple and rapid fault protection and restoration in telecommunication networks. To implement the automatic ring protection in general wavelength-division multiplexing (WDM) optical networks, the lightpaths are partitioned into groups each of which can be carried in a simple cycle of the underlying network. As the electronic line terminals are the dominant cost factor in the deployment of WDM optical networks, we study how to generate these partitions with minimum electronic line terminals. This optimization problem is NP-hard. We develop two polynomial-time approximation algorithms, with performance guarantees between 1.5 and 1.6 and between 1.5 and $1.5+\epsilon$, respectively. The second algorithm can be adapted, with the same performance guarantees, to the problem in which lightpaths are not prespecified and only the endpoints of each connection are given. Both algorithms can be easily adapted, with the same performance guarantees, to the problem in which only link protection is desired, and each group must be carried in a closed trail. The first algorithm matches and the second algorithm improves the approximation ratio obtained independently by Eilam *et al.* (2000).**

***Index Terms*—Approximation algorithm, automatic ring protection, lightpath, traffic grooming.**

I. Introduction

With the advent of dense wavelength-division multiplexing (DWDM) and its increasing deployment in fiber optical networks, the risk of losing vast volumes of data due to a span cut or node failure has escalated [1], [2]. Because of fierce competition among service providers and customers' intolerance of disruption of service, survivability of an optical network has assumed great importance. Survivability refers to the ability of a network to provide continuity of service with no disruption, no matter how much the network may be damaged due to events such as fiber cable cuts or node failures (due to equipment breakdown at a central office or other events such as fires, flooding, etc.). Consequently, optical network designers are beginning to incorporate provisioning of services over disjoint lightpaths, so that if the primary lightpath fails due to a link or node failure, the secondary lightpath can carry the traffic to its destination. We study the survivability provisioning in the general mesh topologies, which are typically deployed in the wide-area networking environment. In addition, we assume no wavelength conversion, and thus each lightpath is carried in a single wavelength.

Among the many fault protection mechanisms, automatic ring protection is very attractive due to its easy implementation and rapid fault restoration [9]. In automatic ring protection, the lightpaths are partitioned into groups of lightpaths belonging to a simple cycle of the optical network, and in case of a failure, the impaired lightpath(s) is reversed backward along the cycle(s) to circumvent the failure. A partition of a set of lightpaths is said to be *proper* if the lightpaths in each group of this partition belong to a simple cycle. To implement the automatic ring protection, all lightpaths in each group are assigned with the same wavelength to form a *logical* ring, and the logical rings, instead of individual lightpaths, become the basic entities for wavelength assignment under the constraint that two logical rings can share the same wavelength if and only if they do not share any common link. However, we do not intend to minimize the total number of wavelengths, as the number of wavelengths is no longer a bottleneck with hundreds wavelengths to be carried in a single fiber link enabled by a recent advance in DWDM technology. Instead, we will try to minimize the total number of electronic line terminals, which are a dominant cost factor [7], [8]. Note that the number of line terminals required by a logical ring is the number of different nodes which are the endpoints of lightpaths in this logical ring, and the number of line terminals required by a proper partition is the total number of line terminals required by all logical rings in this proper partition. In this paper, we refer to the problem of finding a proper partition with minimum number of line terminals as the *minimum ring generation* problem.

The minimum ring generation problem in ring topologies is studied in [7] and [10] in the context of wavelength assignment to lightpaths in WDM/SONET ring networks to minimize the SONET add–drop multiplexers (ADMs). The NP-hardness of the minimum ring generation problem in ring topologies proven in [10] implies the same hardness of the minimum ring generation problem in general topologies. Thus, the objective is to develop provably good polynomial-time approximation algorithms for the minimum ring generation problem in general topologies. We observe that most of the constant approximation algorithms proposed in [3] and [12] cannot be extended to general topologies and the rest of them need substantial modifications in both design and analysis to be applicable in general topologies. The minimum ring generation problem in general topologies was recently studied by Eilam *et al.* [5] independently of us. The main result of their paper [5] is an algorithm whose output is guaranteed to be at most $\mathrm{opt}+(3/5)|P|$, where opt is the cost of the optimum solution, and P is the set of lightpaths. In this paper, we propose a 1.6-approximation algorithm and a $(1.5+\epsilon)$-approximation algorithm for an arbitrarily small number $\epsilon>0$. The 1.6-approximation algorithm does in fact guarantee a solution of cost at most $\mathrm{opt}+(3/5)|P|$,

Manuscript received February 15, 2001; revised July 30, 2001.

The authors are with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: calinescu@cs.iit.edu; ophir@cs.iit.edu; wan@cs.iit.edu).

Publisher Item Identifier S 0733-8716(02)00155-5.

thus matching the algorithm of Eilam *et al.* Though not faster, our 1.6-approximation algorithm is conceptually simpler, and it does have a much shorter proof. The $(1.5+\epsilon)$-approximation algorithm guarantees a solution of cost at most $(1/2)\text{opt}+(1+\epsilon)|P| \le \text{opt}+((1/2)+\epsilon)|P|$, thus improving the bounds of Eilam *et al.* Both algorithms can also be easily adapted, with the same performance guarantees, to the problem in which only link protection is desired, and each group must be carried in a closed trail. In addition, we extend the second algorithm with the same performance guarantees for the case in which the lightpaths are not prespecified and only the endpoints of each connection are given.

The rest of this paper is laid out as follows. In Section II, we introduce some basic terminology and problem formulations. In Section III, we derive a lower bound on the minimum number of line terminals required by a set of lightpaths. In Section IV, we present a general and trivial upper bound on the performance guarantees of all nontrivial algorithms. In Sections V and VI, we propose two approximation algorithms with performance guarantees 1.6 and $1.5+\epsilon$, respectively. In Section VII, we discuss how to extend the algorithm design and analysis to the case in which no routing is prespecified. Finally, we conclude this paper in Section VIII.

II. Terminology and Formulation

Let P be any set of lightpaths in a graph $G=(V,E)$ satisfying that each path in P is contained in some simple cycle of G. We construct a multigraph graph $G(P)$ as follows. The vertex set of $G(P)$ is V. For any path in P with endpoints u and v, add an edge between u and v in $G(P)$. A walk in $G(P)$ is also referred to as a *walk of lightpaths* in P. A walk is said to be *closed* if the endpoints of the walk are identical and *open* otherwise. The *length* of a walk W, denoted by $|W|$, is the number of lightpaths contained in this walk. A walk of length ℓ is called an ℓ-walk. The *cost* of an open walk is one plus its length, and the *cost* of a closed walk is equal to its length. A walk is said to be a *chain of lightpaths* if *all* its lightpaths lie in some simple cycle of G. When it is clear from the context, we identify a chain consisting of one lightpath with the lightpath itself. Two lightpaths *overlap* if they share one vertex which is not an endpoint of both lightpaths. More generally, two chains of lightpaths *overlap* if there is a vertex of G which appears in lightpaths in both chains and is not an endpoint of both chains.

For the link-protection problem, we define a *link-valid-chain* as a walk in $G(P)$ with its lightpaths lying in a closed trail of G. A trail can repeat vertices but cannot use the same edge twice. Similarly, two link-valid-chains *overlap* if there is an edge which appears in lightpaths in both chains. Every bound, example, or theorem in this paper can be extended to the link-protection problem by simply replacing chains by link-valid-chains and using the appropriate notion of overlapping lightpaths and chains.

The following small examples illustrate how bad groupings will cause nonoptimal results.

Example 1: Let G be the ring with four vertices labeled by 0, 1, 2, 3. Assume four lightpaths are given: $P_1=(0,1,2), P_2=(2,3,0), P_3=(0,1)$, and $P_4=(1,2,3,0)$. An optimum grouping will put P_1 and P_2 in a chain, and P_3 and P_4 in a second one, for a cost of 4. A bad grouping will put P_2 and P_3 in a chain, while P_1 and P_4 will each make a chain by itself, at a cost of 7. Note that no two chains in the bad groupings can be merged into a larger chain.

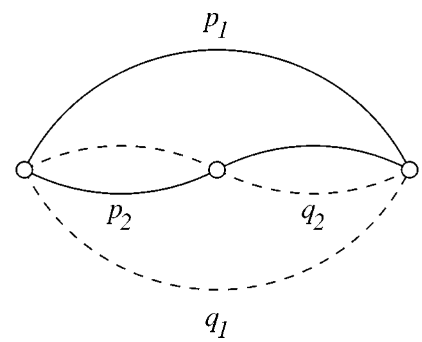

Fig. 1. Grouping a pair of lightpaths (p_1 and q_1) into a closed chain may lead to nonoptimal solution.

Note that we can restrict the solutions of the minimum ring generation to partitions into chains without sacrificing the optimum cost, although such restrictions may require a larger number of wavelengths. Therefore, we will focus on the chain generations from a set of lightpaths. As the cost of chains produced by a chain generation is the number of lightpaths plus the number of open chains, an optimal solution to minimum ring generation corresponds to a chain generation with a minimum number of open chains.

Clearly, the heuristic proposed in [7] and [10] based on a cut-and-merge approach in ring topologies cannot be extended to general topologies. There are two fundamental differences between ring and general topologies, which will affect applicability of the design and analysis of other algorithms proposed in [3] and [12]. The first difference is the computational complexity of the following decision problem: whether a given a set of lightpaths contains a subset of lightpaths that form a closed chain. While this decision problem in ring topologies is solvable in polynomial time, this decision problem in general topologies is NP-complete (the proof for this NP-completeness is omitted in this paper). The second difference is the impact of forming a closed chain from two lightpaths on the optimality of the solution. While one can always take an arbitrary pair of complementary lightpaths in a ring topology to form a closed chain without changing the optimality, it is no longer true in general topologies. This can be illustrated by the following instance.

Example 2: Let p_1 and q_1 be two lightpaths that can form a closed chain, and let p_2 and q_2 be two lightpaths sharing one interior node such that both p_1, p_2 and q_1, q_2 can form a closed chain, respectively (see Fig. 1). The optimal solution for these four paths is the two closed chains formed by p_1, p_2 and q_1, q_2, respectively. However, if we put p_1 and q_1 in a closed chain, then p_2 and q_2 must each form an open chain by itself.

Throughout the paper, we use OPT to denote an optimal chain generation from P and use opt to denote the cost of OPT.

III. Lower Bounds

A trivial lower bound on opt is $|P|$. We derive two other improved lower bounds. The second one will be used by the algorithm of Section V.

For any node v of the network, let $\deg(v)$ denote the degree of the node v in the graph $G(P)$, i.e., the number of lightpaths in P that contain node v as one endpoint. It is well known that the number of nodes with odd degree in any graph is even. The *deficiency* of P, denoted by $d(P)$, is defined as the half of the number of nodes with odd degree. Since each node v requires at least $\lceil(\deg(v)/2)\rceil$ line terminals, the total cost required by P is at least

$$\begin{aligned}\mathrm{opt} &\geq \frac{1}{2}\sum_{v=0}^{n-1}\deg(v) + d(P) = \frac{1}{2}\cdot 2|P| + d(P)\\ &= |P| + d(P).\end{aligned}$$

So $|P| + d(P)$ is a lower bound on opt. This lower bound holds whether lightpaths are prespecified or not. It also holds when groups can be carried by closed trails instead of simple cycles.

The second lower bound is based on the concept of circuit cover. Recall that a *circuit cover* of a graph is a collection of vertex-disjoint circuits which together cover all the vertices. If the graph is weighted, then the weight of a circuit cover of is the sum of the weights of all edges in this circuit cover. It is well known that a minimum-weighted circuit cover of a graph can be found in polynomial time by a reduction to minimum-weighted perfect bipartite matching.

We define a weighted graph $H(P)$ over P as follows.

- For each path p with endpoints u and v, we add three vertices: u_p, v_p, and p. Thus, $H(P)$ consists of $3|P|$ vertices.
- For each path p with endpoints u and v, we add three edges $(u_p, v_p), (u_p, p)$ and (v_p, p) and assign them with the weights 2, 0, and 0, respectively.
- For any pair of nonoverlapping lightpaths p_1 and p_2 between the same two endpoints, say u and v, we add two edges (u_{p_1}, u_{p_2}) and (v_{p_1}, v_{p_2}) of weight 1.
- For any pair of lightpaths p_1 and p_2 which share one endpoint, say v, and lie in some simple cycle of G, we add two edges as follows: let u and w be the other endpoints of p_1 and p_2, respectively, and add two edges (u_{p_1}, w_{p_2}) with weight 2 and (v_{p_1}, v_{p_2}) with weight 1.
- For any pair of node-disjoint lightpaths p_1 with endpoints u and v, and p_2 with endpoints u' and v', we add four edges $(u_{p_1}, u'_{p_2}), (u_{p_1}, v'_{p_2}), (v_{p_1}, u'_{p_2})$ and (v_{p_1}, v'_{p_2}), all of weight 2.

The total number of edges in $H(P)$ is

$$3|P| + 2|P|(|P| - 1) \leq 2|P|^2 + |P|.$$

Note that $H(P)$ has a trivial circuit cover

$$\{(u_p, p, v_p) : p \in P\}.$$

In addition, any chain generation induces naturally a circuit cover of $H(P)$ whose weight is equal to the cost of this chain generation. Thus, the weight of a minimum-weighted circuit cover of $H(P)$ is a lower bound of opt.

IV. Upper Bounds

A trivial upper bound on the cost of any chain generation is $2|P|$, which is at most $2\cdot\mathrm{opt}$, as the number of open chains is at most $|P|$. A slightly better upper bound of 1.75 can be obtained on the cost of any chain generation in which no pair of chains can be merged into a larger chain. Let $\mathcal{C}$ be any collection of chains in which every lightpath of P appears exactly in one chain of $\mathcal{C}$.

Lemma 3: If any pair of chains in $\mathcal{C}$ cannot be merged into a larger chain, the cost of $\mathcal{C}$ is at most $(7/4)\cdot\mathrm{opt}$.

Proof: We call a lightpath *unmatched* in $\mathcal{C}$ if it forms a 1-chain in $\mathcal{C}$. Then out of any two consecutive lightpaths in any chain of OPT at most one is unmatched in $\mathcal{C}$. Let C be any chain in OPT. If C is closed, then at most $\lfloor(|C|/2)\rfloor$ lightpaths in C are unmatched in $\mathcal{C}$. If C is open, then at most $\lceil(|C|/2)\rceil$ lightpaths in C are unmatched in $\mathcal{C}$. Let i denote the unmatched lightpaths in $\mathcal{C}$ and k denote the number of odd open chains in OPT. Then, $i \leq (|P|+k)/2$. Thus, the total number of chains in $\mathcal{C}$ is at most

$$i + \left\lfloor\frac{|P| - i}{2}\right\rfloor \leq \frac{|P| + i}{2} \leq \frac{|P| + \frac{|P|+k}{2}}{2} = \frac{3}{4}|P| + \frac{k}{4}.$$

So the cost of $\mathcal{C}$, being equal to the number of lightpaths plus the number of open chains, is at most

$$|P| + \left(\frac{3}{4}|P| + \frac{k}{4}\right) = \frac{7}{4}|P| + \frac{k}{4} \leq \frac{7}{4}(|P| + k) \leq \frac{7}{4}\cdot\mathrm{opt}.$$

■

By repeatedly merging, if possible, two chains into a larger chain, one can convert any chain generation into one in which any pair of chains cannot be merged into a larger chain. Lemma 3 implies that any algorithm followed by this postprocessing will have an approximation ratio of at most $7/4$. As shown in Example 1 given in Section II, postprocessing does not guarantee anything better than $7/4$. Next, we present several algorithms whose approximation ratios beat $7/4$.

V. Walk Splitting

We propose a three-phased algorithm, referred to as minimum circuit cover—walk splitting (MCC-WS). The first phase, referred to as walk generation phase, generates a set of walks with cost at most opt. We find a minimum-weighted circuit cover of $H(P)$ in polynomial time. Removing all edges of weight two from the minimum-weighted circuit cover, we obtain a collection of paths and circuits in $H(P)$. Note that in any circuit cover of $H(P)$, for any $p \in P$ the three nodes u_p, p, v_p are in the same circuit with p adjacent to u_p and v_p. By replacing the three nodes u_p, p, and v_p with p, each path (circuit respectively) induces an open (closed, respectively) walk in the graph $G(P)$. Then the total cost of the obtained walks is exactly equal to the weight of the minimum-weighted circuit cover of $H(P)$. In addition, any two consecutive lightpaths in each walk lie in a simple cycle of G.

The second phase, referred to as walk splitting phase, splits the walks obtained in the first phase into chains. Specifically, an open walk is split into chains by traversing along this open walk from the first lightpath and generating a chain whenever an overlap occurs; a closed walk is split into chains by traversing along this closed walk from an arbitrary lightpath and generating a chain whenever overlap occurs.

The third phase, referred to as the chain merging phase, repeatedly merges any pair of open chains into a larger chain until no more merging can occur.

As an example, consider five lightpaths over a ring G with five vertices, labeled 0, 1, 2, 3, 4. Assume five lightpaths are given: $P_1 = (0,1,2), P_2 = (2,3,4), P_3 = (4,0,1), P_4 = (1,2,3)$, and $P_5 = (3,4,0)$. The walk generation phase produces just one walk: P_1, P_2, P_3, P_4, P_5, of cost 5. The walk splitting phase first produces the chain P_1, P_2, to which P_3 is not added since an overlap occurs. Then the algorithm produces a second chain, P_3, P_4. P_5 is left as a chain by itself. No merging is possible during the chain merging phase. On this particular example, the algorithm produces an optimum solution.

The following lemma gives an upper bound on the cost of the obtained chains.

Lemma 4: MCC-WS produces a chain generation with cost at most $\mathrm{opt} + (3/5)|P|$.

Proof: Let ω denote the weight of a minimum-weighted circuit cover of $H(P)$. Then, the total cost of the walks generated by the first phase is exactly ω. The splitting of walks into chains in the second phase may increase the cost. Let W be any walk that is not a chain and i be the number of chains split from C. Then the splitting of W into i chains increases the cost by $i-1$ if W is open or i if C is closed. From the construction of the graph $H(P)$, the lengths of all chains obtained from C, except the last one, are at least two. Thus

$$i \le \left\lceil \frac{|W|}{2} \right\rceil \le \frac{|W|+1}{2}.$$

So splitting W creates an additional cost of at most $(|W|-1)/2$ if W is open, at most $(|W|/2)$ if W is closed and $|W|$ is even, and at most $(|W|+1)/2$ if W is closed and $|W|$ is odd. Let j be the number of closed walks of odd length generated by the first phase that are not chains. Then the cost of the chains generated by the second phase is at most $\omega + (|P|+j)/2$. As the length of any odd closed walk generated by the first phase that is not a chain is at least five, $j \le (|P|/5)$. So the cost of the chains produced by the second phase is at most

$$\begin{aligned}\omega + \frac{|P|+j}{2} &\le \omega + \frac{|P| + \frac{|P|}{5}}{2} = \omega + \frac{3}{5}|P| \\ &\le \mathrm{opt} + \frac{3}{5}|P|\end{aligned}$$

as ω is a lower bounds of opt. This completes the proof of the lemma. ■

The above lemma and the fact that $|P| \le \mathrm{opt}$ imply an upper bound of 1.6 on the approximation ratio of the algorithm MCC-WS. A lower bound of 1.5 can be obtained from Example 13 given in [3], which is included for the sake of completeness. We use R_n to denote a ring of n nodes labeled by $0, 1, \ldots, n-1$ clockwise. A path (i.e., arc)

$$i, (i+1) \bmod n, (i+2) \bmod n, \ldots, j$$

is represented by (i,j). Let G be R_n with $n = 2(2k+1)$ for some $k > 1$, and let P consist of the following $3n/2$ paths:

$$\begin{aligned}&\left\{p_i = \left(2i, 2i+\frac{n}{2}\right) \,\middle|\, 0 \le i < \frac{n}{2}\right\} \cup \\ &\left\{p_i' = \left(2i+\frac{n}{2}, 2i+\frac{n}{2}+1\right) \,\middle|\, 0 \le i < \frac{n}{2}\right\} \\ &\cup \left\{p_i'' = \left(2i+\frac{n}{2}+1, 2i\right) \,\middle|\, 0 \le i < \frac{n}{2}\right\}\end{aligned}$$

as illustrated in Fig. 2(a). Note that for any $0 \le i < n$, the three paths p_i, p_i', p_i'' form a closed chain. So $\mathrm{opt} = |P| = (3n/2)$. On the other hand, MCC-WS may produce in the first phase two circuits

$$p_k'', p_{2k}'', p_{k-1}'', p_{2k-1}'', \ldots, p_1'', p_{k+1}'', p_0'';$$

and

$$\begin{aligned}&p_0, p_0', p_{k+1}, p_{k+1}', p_1, p_1', p_{k+2}, p_{k+2}', \ldots, \\ &p_{k-1}, p_{k-1}', p_{2k}, p_{2k}', p_k, p_k'.\end{aligned}$$

Each circuit induces a closed walk which is not a closed chain. In the second phase, the first closed walk is split into k open 2-chains

$$\{\{p_i'', p_{i+k}''\} \,|\, 1 \le i \le k\}$$

and one open 1-chain $\{p_0''\}$. Since the $2k+1$ arcs

$$\{p_i \,|0 \le i \le 2k\}$$

are pairwise overlapping, at least $2k+1$ open chains must be used by the second closed walk. One solution is the following:

$$\{\{p_i, p_i'\} \,|\, 0 \le i \le 2k\}.$$

In the third phase, the two open chains $\{p_0\}$ and $\{p_0', p_0''\}$ are merged into a closed chain $\{p_0, p_0', p_0''\}$. So totally $3k+1$ chains are obtained, among which $3k$ are open [see Fig. 2(b)]. The total cost of all these chains is

$$|P| + 3k = 3(2k+1) + 3k = 9k + 3 = \frac{3}{2} \cdot \mathrm{opt} - \frac{3}{2}.$$

Thus the approximation ratio of MCC-WS is at least $3/2$.

In summary, we have the following theorem.

Theorem 5: The approximation ratio of **MCC-WS** is between 1.5 and 1.6.

VI. Iterative Matching

Let $\mathcal{C}$ be a collection of chains. The *fit graph* of $\mathcal{C}$, denoted by $F(\mathcal{C})$, is a weighted undirected graph in which the vertex set is $\mathcal{C}$, and there is an edge between two chains C_1 and C_2 if and only if C_1 and C_2 can be merged into a larger chain, and the weight of the edge between C_1 and C_2 is equal to the number of endpoints shared by C_1 and C_2.

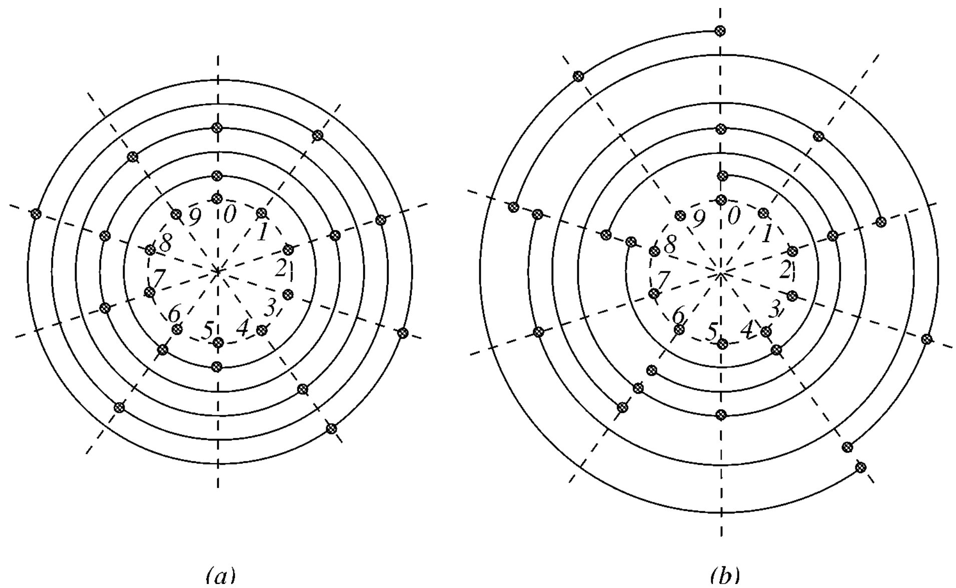

Fig. 2. An instance used for obtaining the 1.5 lower bound: (a) the optimal solution and (b) the solution produced by the algorithm MCC-WS.

We start by introducing a simple algorithm referred to as Iterative Matching (IM) proposed in [3]. This algorithm maintains a set of chains of lightpaths $\mathcal{C}$ throughout its execution. Initially $\mathcal{C}$ consists of 1-chains each of which is a lightpath in P. While the fit graph $F(\mathcal{C})$ of $\mathcal{C}$ has nonempty edge set, we find a maximum-weighted matching M in $F(\mathcal{C})$ and then merge each matched pair of chains in M into a larger chain. When $F(\mathcal{C})$ has empty edge set, $\mathcal{C}$ is output as the chain generation.

It is obvious that the algorithm IM has polynomial run-time. Next, we show that its approximation ratio is at most 5/3.

Lemma 6: The approximation ratio of IM is at most 5/3.

Proof: From any chain C in OPT, a matching of cardinality $\lfloor (|C|/2) \rfloor$ can be obtained. Let i be the number of odd open chains in OPT, and j be the number of odd closed chains in OPT. Then from the chains in OPT, we can obtain a matching of cardinality $(|P| - i - j)/2$. Thus the cardinality of any maximum-weighted matching obtained in the first iteration is at least $(|P| - i - j)/2$, and consequently after the first iteration, the total number of chains is at most

$$|P| - \frac{|P| - i - j}{2} = \frac{|P| + i + j}{2}.$$

Note that any odd closed chain must contain at least three lightpaths. Thus $j \leq (|P| - i)/3$. So after the first iteration, the total cost is at most

$$\begin{aligned} |P| + \frac{|P| + i + j}{2} &= \frac{3|P|}{2} + \frac{i}{2} + \frac{j}{2} \\ &\leq \frac{3|P|}{2} + \frac{i}{2} + \frac{|P| - i}{6} = \frac{5|P| + i}{3} \\ &\leq \frac{5}{3}(|P| + i) \\ &\leq \frac{5}{3} \cdot \mathrm{opt}. \end{aligned}$$

Therefore, the approximation ratio of IM is at most 5/3. ∎

We propose the preprocessed iterative matching with an *odd* parameter l (l-PIM) algorithm, which runs in two phases.

1) Preprocessing Phase: for $k = 1$ to l, repeatedly take closed k-chains out of the remaining lightpaths until no more closed k-chain can be obtained from the remaining lightpaths. This is done by trying all combinations of k lightpaths and checking if a combination results in a closed k-chain.
2) Matching Phase: apply the algorithm IM to the remaining lightpaths.

It is obvious that for any fixed constant l, the algorithm has polynomial run-time. A straightforward implementation of the first phase has time $O(|P|^{l+3}|V|^2)$. Next, we show that its approximation ratio of is at most $1.5 + 1/(2(l+2))$.

Lemma 7: l-PIM produces a chain generation with cost at most $(1/2)\mathrm{opt} + (1 + 1/(2(l+2)))|P|$.

Proof: We call the lightpaths appearing in the closed chains obtained in the preprocessing phase *blue* lightpaths, and the others are *red* lightpaths. We use B and R to denote the set of blue lightpaths and the set of red lightpaths, respectively. Then in any closed chain of length at most l in OPT, at least one lightpath is blue. From OPT, we remove all blue lightpaths and obtain a collection of red chains. Note that the number of red chains obtained from a closed chain C is at most the number of blue lightpaths in C; the number of red chains obtained from an open chain C is at most the number of blue lightpaths in C plus one. Thus the total number of red open chains is at most $|B|$ plus the total number of open chains, which is $\mathrm{opt} - |P|$ and consequently is at most

$$|B| + \mathrm{opt} - |P| = \mathrm{opt} - |R|.$$

In addition, all red closed chains have length at least $l+1$, and using the fact l is *odd*, all *odd* red closed chains have length at least $l+2$. Let k be the number of *odd* red chains, open or closed. Then

$$k \leq \mathrm{opt} - |R| + \left\lfloor \frac{|R|}{l+2} \right\rfloor \leq \mathrm{opt} - \frac{l+1}{l+2}|R|.$$

As each red chain C can contribute a matching of cardinality $\lfloor(|C|/2)\rfloor$, the red lightpaths R admit a matching of $(|R|-k)/2$ pairs. Thus, the first iteration in the matching phase generates a maximum matching of at least $(|R|-k)/2$ pairs. So after this first iteration, the total number of red open chains is at most

$$|R| - \frac{|R|-k}{2} = \frac{|R|+k}{2}.$$

At this moment, the total cost is at most

$$\begin{aligned} |P| + \frac{|R|+k}{2} &\leq |P| + \frac{|R| + \mathrm{opt} - \frac{l+1}{l+2}|R|}{2} \\ &= |P| + \frac{\mathrm{opt}}{2} + \frac{1}{2(l+2)}|R| \leq \frac{\mathrm{opt}}{2} + \left(1 + \frac{1}{2(l+2)}\right)|P| \end{aligned}$$

thus completing the proof of the lemma. ■

The above lemma and $\mathrm{opt} \leq |P|$ imply an upper bound of $3/2 + 1/(2(l+2))$ on the approximation ratio of the algorithm l-PIM. Thus when $l = 3$, the approximation ratio is at most 1.6. By increasing the parameter l, the approximation ratio can be arbitrarily close to 3/2, although at the cost of more running time. However, the approximation ratio can not be less than 3/2 no matter how large l is. This can be illustrated by Example 2. As the algorithm does not specify exactly which chain will be picked first during the preprocessing step, we assume it picks the closed chain p_1, q_1. Iterative matching cannot merge any two chains, and the output is a solution of cost 6, but optimum has cost 4.

In summary, we have the following theorem.

Theorem 8: The approximation ratio of l-PIM is between 3/2 and $3/2 + 1/(2(l+2))$.

VII. Chain Generation Without Prespecified Routing

Now we drop the assumption that the routing is prespecified. Instead, the input is a set of pairs of nodes in a two-vertex-connected graph G. We call these pairs *requests*. Routing a request means selecting a lightpath in between the two endpoints of a request. The solution should provide a lightpath for each request in the input and a partition of these lightpaths into chains. Finding an optimal solution in general topologies is NP-hard as this optimization problem is NP-hard even in ring topologies [3]. In the following, we discuss the extension of our results obtained earlier in this paper.

l-PIM for nonprespecified routing is adapted from l-PIM, and has the same two phases.

1) Preprocessing Phase: for $k = 1$ to l, repeatedly select, route, and remove a set of requests such that they can be routed to form a closed k-chain, as long as possible.
2) Matching Phase: Let $\mathcal{R}$ be the set of remaining requests and $\mathcal{C}$ be the empty set of chains of lightpaths. While the fit graph $F(\mathcal{R} \cup \mathcal{C})$ has nonempty edge set, find a maximum-weighted matching M in $F(\mathcal{R} \cup \mathcal{C})$ and then merge each matched pair (routing, if necessary, the requests of R) into a larger chain. The two matched vertices are removed from $(\mathcal{R} \cup \mathcal{C})$ and the larger chain resulting from their merge is added to $\mathcal{C}$. When $F(\mathcal{R} \cup \mathcal{C})$ has empty edge set, arbitrarily route the remaining requests in $\mathcal{R}$ and output the resulting chains.

Both phases require additional explanations. Checking if a set of requests can be routed to form a closed k-chain can be done in polynomial time for fixed k [11]. The same algorithm of Robertson and Seymour can check if two given requests can be routed on one ring—thus being joined by an edge in the fit graph $F(\mathcal{R} \cup \mathcal{C})$. Checking if a chain of lightpaths and a request can routed on one ring can done by an application of the Fan lemma (see, for example, [6, p. 146]), as follows. Construct the subgraph G' of G obtained by removing all the interior vertices of the chain. If the endpoints of the request coincide with the endpoints of the chain, then just checking if the endpoints are connected in G' is enough. If the endpoints of the request are disjoint from the endpoints of the chain, there is no point in adding this edge to $F(\mathcal{R} \cup \mathcal{C})$. If the request and the chain share exactly one endpoint, say v, then, if we denote by u the other endpoint of the chain and by s the other endpoint of the request, the problem becomes finding two paths in $G' : P_1$ from s to u, and P_2 from s to v, which intersect only in s. The Fan lemma assures the existence of these two paths (and its proof finds them) if and only if there is no vertex in $V(G') \setminus \{s\}$ whose removal disconnects s from either u or v. This condition can be easily checked in $O(|V(G')||E(G')|)$ time.

Following the argument in Lemma 7, we can prove that the approximation ratio of the algorithm above is at most $3/2 + 1/(2(l+2))$.

VIII. Conclusion

Motivated by support of automatic ring protection in optical networks with minimum line terminal cost, we studied the Minimum Ring Generation problem. We developed several gradually improved lower bounds on the minimum cost and upper bounds on the cost of nontrivial solutions. We also proposed two approximation algorithms MCC-WS based on minimum-weighted circuit cover, and l-PIM based on maximum-weighted matching. We proved that the performance guarantee of MCC-WS is between 1.5 and 1.6, and the performance guarantee of l-PIM is between 3/2 and $3/2 + 1/(2(l+2))$.

An important variation of the Minimum Ring Generation problem is allowing the splitting of lightpaths to achieve lower cost [4], [7]. Such a splitable version has been studied in ring topologies [4], [7]. It would be interesting to develop provably good algorithms for this splitable version in general topologies.

References

[1] M. Alanyali and E. Ayanoglu, "Provisioning algorithms for WDM optical networks," *Proc. IEEE INFOCOM'98*, pp. 910–918, 1998.

[2] J. Armitage, O. Crochat, and J. Y. Le Bouder, "Design of survivable WDM photonic network," *Proc. IEEE INFOCOM'97*, pp. 244–252, 1997.

[3] G. Călinescu and P.-J. Wan, Traffic partition in WDM/SONET rings to minimize SONET ADMs, in *J. Combinatorial Optimization*, to be published.

[4] ——, "Splitable traffic partition in WDM/SONET rings to minimize SONET ADMs," in *Theoretical Computer Sci. Preliminary Results I-SPAN*, 2000.

[5] T. Eilam, S. Moran, and S. Zaks, "Approximation algorithms for survivable optical networks," presented at the *14th Int. Symp. Distributed Computing*, 2000.

[6] *Handbook of Combinatorics*, R. L. Graham, M. Grötschel, and L. Lovász, Eds., MIT Press, 1995. Connectivity and Network Flows.

[7] O. Gerstel, P. Lin, and G. Sasaki, "Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings," *Proc. IEEE INFOCOM'98*, vol. 1, pp. 94–101.
[8] ——, "Combined WDM and SONET network design," *Proc. IEEE INFOCOM'99*, vol. 2, pp. 734–743.
[9] I. Haque, W. Kremer, and K. Raychauduri, "Self-healing rings in a synchronous environment," in *SONET/SDH: A Sourcebook of Synchronous Networking*, C. A. Siller and M. Shafi, Eds. New York: IEEE Press, 1996, pp. 131–139.
[10] L. W. Liu, X.-Y. Li, P.-J. Wan, and O. Frieder, "Wavelength assignment in WDM rings to minimize SONET ADMs," in *Proc. INFOCOM 2000*, vol. 2, Israel, pp. 1020–1025.
[11] N. Robertson and P. D. Seymour, "Graph minors XIII: The disjoint paths problem," *J. Combin. Theory B*, vol. 63, pp. 65–110, 1995.
[12] P.-J. Wan, G. Călinescu, L.-W. Liu, and O. Frieder, "Grooming of arbitrary traffic in SONET/WDM BLSRs," *J. Select. Areas Commun.*, vol. 18, pp. 1995–2003, Oct. 2000.

G. Calinescu received the Diploma from the University of Bucharest and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, in 1998.

He joined the Computer Science Department at Illinois Institute of Technology in February 2000. During the academic year 1998–1999 he was a postdoctoral Fellow at DIMACS.

Ophir Frieder (SM'93) is the IITRI Professor of Computer Science at the Illinois Institute of Technology. His research interests include information retrieval, databases, and parallel and distributed systems. He co-authored three books, published over 100 refereed publications, was granted four patents, and has received research support from a variety of governmental and industrial grants.

In 1993, Dr. Frieder was a recipient of the International Information Science Foundation Award from Japan and the NSF National Young Investigator Award. He is currently on the editorial boards of the *Journal of the American Society for Information Science*, *Information Retrieval and Supercomputing*, and *Optical Networks Magazine* and he is a member of Phi Beta Kappa and ACM.

Peng-Jun Wan received the B.Sc. degree in applied mathematics, in 1990, from Tsinghua University, P. R. China, the M.S. degree in operations research, in 1993, from the Chinese Academy of Sciences, P. R. China, and the Ph.D. degree in computer science, in 1997, from the University of Minnesota, Mineapolis.

He has been an Assistant Professor of Computer Science at the Illinois Institute of Technology since 1997. He is currently active in the fields of wavelength division multiplexing optical networks and wireless *ad hoc* networks. He has published over 20 journal papers and 30 conference papers.