Traffic Partition in WDM/SONET Rings to Minimize SONET ADMs

Gruia Călinescu *
Peng-Jun Wan
Dept. of Computer Science, Illinois Institute of Technology, Chicago, IL 60616
E-mail:{calinescu, wan}@cs.iit.edu

Abstract

SONET (Synchronous Optical NETworks) add-drop multiplexers (ADMs) are the dominant cost factor in the WDM(Wavelength Division Multiplexing)/SONET rings. The number of SONET ADMs required by a set of traffic streams is determined by the routing and wavelength assignment of the traffic streams. Previous works took as input the traffic streams with routings given a priori and developed various heuristics for wavelength assignment to minimize the SONET ADM costs. This paper contributes mainly in two aspects. First, in addition to the traffic streams with pre-specified routing, this paper also studies minimizing the ADM requirement by traffic streams without given routings, a problem which is shown to be NP-hard. Second, new heuristics are proposed and analyzed for both the prespecified routing and the non-prespecified routing versions. The Preprocessed Iterative Matching heuristic has approximation ratio in between $4/3$ *and* $3/2$*. A local-improvement algorithm has approximation ratio exactly* $3/2$*. An algorithm based on Eulerian tour decomposition has approximation ratio exactly* $3/2$ *for the non-prespecified routing version.*

1. Introduction

WDM/SONET rings form a very attractive network architecture that is being deployed by a growing number of telecom carriers. In this network architecture, each wavelength channel carries a high-speed (e.g., OC-48) SONET ring [4]. The key terminating equipments are optical add-drop multiplexers (OADMs) and SONET add-drop multiplexers (ADMs). Each node is equipped with one OADM. The OADM can selectively drop wavelengths at a node. Thus if a wavelength does not carry any traffic from or to a particular node, the OADM allows that wavelength to optically bypass that node rather than being electronically terminated. Consequently, in each SONET ring a SONET ADM is required at a node if and only if it carries some traffic terminating at this node. Therefore, the SONET ADMs required by a set of traffic streams is determined by their routing and the wavelength assignment. As the SONET ADMs are the dominant cost factor in the WDM/SONET rings, it is essential to find an efficient routing and wavelength assignment to a given set of traffic streams such that the total ADM cost is minimal. This optimization problem is referred to as *minimum ADM cost problem*.

A number of previous works [2, 3, 5] studied the minimum ADM cost problem in which each traffic stream has a predetermined routing given by a lightpath. As the lightpaths can be treated as circular arcs over the ring, this special version of minimum ADM cost problem is reduced to the following minimization problem:

- Instance: a set of circular-arcs A along a ring.
- Solution: a partition of A, $\Pi = \{A_1, A_2, \cdots, A_w\}$, such that for any $1 \leq i \leq w$ all arcs in each A_i are non-intersecting.
- Cost: the cost of each set of arcs A_i is the number of different nodes of the ring that are the endpoints of the arcs in A_i (note that a node which is the endpoint of two arcs of A_i is only counted once towards this number), and the cost of the partition Π is the sum of the costs of A_i for all $1 \leq i \leq w$.

  The minimum cost over all proper solutions is called the minimum ADM cost of A.

We refer to this special version as the *arc-version* minimum ADM cost problem. It was proven in [5] that this arc-version minimization problem is NP-hard. Several heuristics have been proposed in [2, 5]. Most of them have approximation ratio at least $3/2$, while the best previously proved ratio is $\frac{3+e}{1+e} = 1.537\ldots$ [5]. Note that an approximation ratio of 2 is trivial: optimum is at least the number of arcs, and any solution uses at most twice the number of arcs. Lemma 4 from [9] (see Lemma 3) shows that any non-trivial heuristic has an approximation ratio of at most 1.75.

*Work performed at DIMACS and supported by a DIMACS postdoctoral fellowship

In this paper we present two algorithms for the minimum ADM cost problem: Preprocessed Iterative Matching and a local-improvement algorithm. We prove that Preprocessed Iterative Matching has approximation ratio in between $4/3$ and $3/2$. Without a proof of the approximation ratio, the algorithm is described in the survey paper [9], where it is called Closed Segment First, and it is also mentioned in the conclusion section of [5]. We also prove that the new local-improvement algorithm has approximation ratio exactly $3/2$. We point out that the number of wavelengths used by any of these heuristics is at most twice the maximum link load of A.

This paper also addresses the minimum ADM cost problem in which the routing of each traffic stream is part of the solution instead of part of the input. Our algorithm and analysis applies to both bidirectional line-switched rings with two fibers (BLSR/2) and bidirectional line-switched rings with four fibers (BLSR/4) [4]. For simplicity, we assume that each traffic stream is symmetrically duplex and its two portions in opposite directions must be routed along the same path (in opposite directions). Under this assumption, we can treat the two working fiber rings as one (undirected) ring, and each traffic stream as a (undirected) chord. Thus this general minimum ADM cost problem can be stated as the following minimization problem:

- Instance: a set of chords C along a ring.
- Solution: a proper partition of C, $\Pi = \{C_1, C_2, \cdots, C_w\}$, such that for any $1 \leq i \leq w$ all chords in each C_i can be routed as non-intersecting arcs over the ring.
- Cost: the cost of each C_i is the number of different nodes of the ring that are the endpoints of the chords in C_i, and the cost of the partition Π is the sum of the costs of C_i for all $1 \leq i \leq w$. The minimum cost over all proper solutions is called the minimum ADM cost of C.

We refer to this general version as the *chord-version* minimum ADM cost problem. First, we claim the NP-hardness of this integrated problem, and point out that any nontrivial heuristic has an approximation ratio of at most 1.75 (again, approximation ratio of 2 is trivial). Several previously proposed heuristics, including minimum-load routing [7, 10], minimum-wavelength routing [6], Edge-Avoidance routing and Shortest-Path routing [1], fail to beat the 1.75 ratio. Even after the preprocessing by pairing up identical chords, their approximation ratios is still at least $\frac{5}{3} \approx 1.67$.

Preprocessed Iterative Matching and the local-improvement algorithm can be used, with the same performance ratios, for the chords-version minimum ADM cost problem. We also propose an algorithm based on Eulerian tour decomposition, and prove it has an approximation ratio of exactly $3/2$. Eulerian tour decomposition can also be applied to the arc version, but has approximation ratio exactly 1.75.

The rest of this paper is laid out as follows. In Section 2, we introduce some basic terminology and problem formulations. In Section 3 we present the proof of NP-hardness of the chord-version minimum ADM cost problem. In Section 4, Section 5, and Section 6, we describe and analyze our three new algorithms.

2. Terminology and Formulation

2.1. Ring, Arc, Chord

We assume that a ring network consists of n nodes numbered clockwise by $0, 1, \cdots, n-1$. It is oriented clockwise and is treated as a directed graph. All arithmetic involving nodes is performed implicitly using modulo n operations. The link from the node i to node $i+1$ in the ring is referred to as link i.

An (clockwise) arc a over a ring is represented by an *ordered* pair $(t(a), h(a))$, where $t(a)$ is the tail of a and $h(a)$ is the head of a. Two arcs are said to be *intersecting* (or *overlapping*) if they contain a common link of the ring, and *adjacent* if they are not intersecting but share at least one endpoint.

A chord in a ring is specified by an *unordered* pair (i, j) where i and j are the two endpoints of the chord. *Routing* a chord means selecting from its two endpoints a tail and a head, thus replacing the chord by an arc. Two chords are said to be *adjacent* if they have at least one common endpoint, and be *identical* if they have the same endpoints. Note that two adjacent but not identical pair of chords have a unique routing without intersection.

2.2. Chains of Arcs or Chords

To avoid confusion, we recall some graph theory concepts. A *walk* in a graph G is a sequence $W := (v_0, e_1, v_1, e_2, v_2, \cdots, e_\ell, v_\ell)$, where $v_0, v_1, \cdots, v_\ell$ are vertices of G, $e_1, e_2, , \cdots, e_\ell$ are edges G, and e_i is an edge joining v_{i-1} and v_i for $i = 1, \cdots, \ell$. The walk W is *closed* if $n > 0$ and $v_0 = v_n$, and *open* otherwise. If W is open, the vertex v_0 is the tail of W and the vertex v_ℓ is its head. A *trail* is a walk with all its edges distinct; a *path* is a open walk with all its vertices distinct; a *circuit* is a closed trail of positive length whose vertices are all distinct. An *Eulerian tour* in a graph is a closed trail which includes every edge. It is well known that a graph is Eulerian if and only if it is connected and every vertex has even degree.

Let A be any set of arcs over a ring. It induces naturally a directed graph $G(A)$ with the nodes of the ring as its vertex set and A as its edge set. A trail in $G(A)$ is also called a *trail of arcs* in A. A trail of arcs induces a (clockwise) walk over the ring starting from its tail to its head. A trail of arcs is said to be a *chain of arcs* if any pair of consecutive arcs are non-overlapping. A chain of arcs is said to be *valid* if all arcs in this chain are non-overlapping.

Let C be any set of chords. It induces naturally an undirected graph $G(C)$ with the nodes of the ring as its vertex set and C as its edge set. A trail in $G(C)$ is also called as a *trail of chords* in C. A trail of chords is said to be a *chain of chords* if either itself or its reverse induces a (clockwise) walk over the ring in which the two arcs corresponding to two consecutive chords are non-overlapping. A *routing* of a chain of chords is the (clockwise) walk induced by either itself or its reverse in which two arcs corresponding to consecutive chords are non-overlapping. Note that if a chain of chords is a path of at least two chords or a circuit of at least three chords, it has a unique routing. A chain of a single chord or two identical chords can have two different routings. A chain of chords is said to be *valid* if it has a routing in which all arcs corresponding to the chords are non-overlapping.

For simplicity, we also refer to the length of a chain (of arcs or chords) as its *size*, and the number of vertices in a chain as its *cost*. Thus the cost of a closed chain is exactly its size, and the cost of an open chain is one plus its size. Let $\mathcal{P}$ be a collection of chains. The *cost* of $\mathcal{P}$ is defined as the sum of the costs of all chains in $\mathcal{P}$, which equals to the total number of arcs in the chains in $\mathcal{P}$ plus the total number of open chains in $\mathcal{P}$. The *fit graph* of $\mathcal{P}$, denoted by $F(\mathcal{P})$, is a undirected graph in which the vertex set is $\mathcal{P}$, there is an edge between two chains P_1 and P_2 if and only if P_1 and P_2 can be merged into a larger valid chain.

For both arc-version and chord version of the minimum ADM cost problem, we can restrict the solutions to partitions of the input arcs or chords into valid chains, referred to as *valid chain generations*. This restriction does not change the optimum value, but may requires larger wavelength requirement. The wavelength requirement by any solution can be further reduced by treating each chain as an arc and applying Tucker's algorithm for circular-arc coloring [8]. For the arc-version, this processing can reduce the number of required wavelengths within twice the minimum. Indeed, suppose that the load of the original arcs is l (the *load of a link* is the number of arcs using that link, and the *load of a set of arcs* is the maximum load of a link) and the number of (valid) closed chains is k. Then the load of the open chains is $l-k$. The Tucker's algorithm for circular-arc coloring guarantees that the number of wavelengths used by the open chains is at most $2(l-k)-1$. Thus the total number of wavelengths is at most $k+2(l-k)-1=2l-k-1$, while the minimum number of wavelengths required is at least l.

2.3. Deficiency of Arcs or Chords

Let A be any set of arcs over a ring. For any node i of the ring, let $\sigma_A(i)$ denote the total number of arcs in A with node i as the head, and $\tau_A(i)$ denote the total number of arcs in A with node i as the tail. The deficiency of A, denoted by $d(A)$, is defined by

$$d(A)=\frac{1}{2}\sum_{i=0}^{n-1}\left|\sigma_A(i)-\tau_A(i)\right|.$$

Since each node i requires at least $\max\{\sigma_A(i),\tau_A(i)\}$ ADMs, the total number of ADMs required by A is at least

$$\begin{aligned}
&\sum_{i=0}^{n-1}\max\{\sigma_A(i),\tau_A(i)\}\\
&=\sum_{i=0}^{n-1}\frac{\sigma_A(i)+\tau_A(i)+|\sigma_A(i)-\tau_A(i)|}{2}\\
&=\sum_{i=0}^{n-1}\frac{\sigma_A(i)+\tau_A(i)+|\delta_A(i)|}{2}\\
&=\frac{\sum_{i=0}^{n-1}(\sigma_A(i)+\tau_A(i))}{2}+\frac{\sum_{i=0}^{n-1}|\delta_A(i)|}{2}\\
&=|A|+d(A).
\end{aligned}$$

So $|A|+d(A)$ is a lower bound on the minimum ADM cost required by A. This fact was first noticed by [2].

Let C be any set of chords. For any node i of the ring, let $\deg_C(i)$ denote the total number of chords in C that contain node i as one endpoint. It is well known the number of nodes with odd degree is even. The *deficiency* of C, denoted by $d(C)$, is defined as the half of the number of nodes with odd degree. Since each node i requires at least $\left\lceil\frac{\deg_C(i)}{2}\right\rceil$ ADMs, the total number of ADMs required by C is at least

$$\frac{1}{2}\sum_{i=0}^{n-1}\deg_C(i)+d(C)=|C|+d(C).$$

So $|C|+d(C)$ is a lower bound on the minimum ADM cost required by C.

3. NP-Hardness

The NP-hardness of the arc-version minimum ADM cost problem follows from the result proved in [5] that:

Theorem 1 *[5] It is NP-hard to decide whether the minimum ADM cost of a set of arcs A is equal to $|A|$ or not.*

In this section, we prove the NP-hardness of the chord-version minimum ADM cost problem .

Theorem 2 *It is NP-hard to decide whether the minimum ADM cost of an arbitrary set of chords C is equal to $|C|$ or not.*

Proof. The reduction is from the decision problem from Theorem 1. Let A be any set of arcs over a ring R. We orient the ring and each circular arcs in the clockwise direction. For each arc $a \in A$, we add a unique node v_a in the ring R within the arc a and create two chords, $(t(a), v_a)$ and $(v_a, h(a))$, as illustrated in Figure 1. We use R' and C to denote the obtained ring and set of chords respectively. From the construction,

$$|R'| = |R| + |A|,$$
$$|C| = 2|A|.$$

From Theorem 1, it is sufficient to show that the minimum ADM cost of C is equal to $|C| = 2|A|$ if and only if the minimum ADM cost of a set of A is equal to $|A|$. The if part is obvious. So we prove the only if part. Assume that $opt(C) = |C|$, then C can be partitioned into closed valid chains. Each such partition requires that the two chords created from each arc $a \in A$ be in the same valid closed chain and therefore must be routed along the arc a. By replacing these two chords with the arc a for each $a \in A$, we obtain a partition of A such the arcs in each group of the partition form a closed valid chain. ■

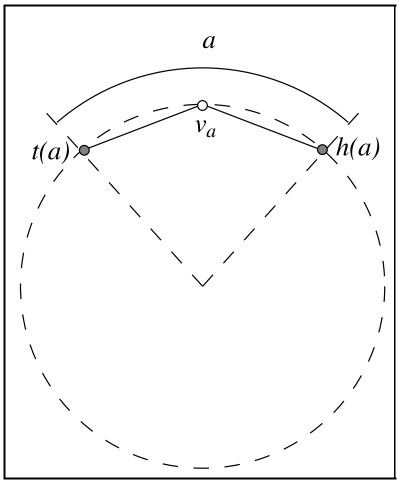

Figure 1. Replace each arc by two chords: the filled circles are the endpoints of the arc, and the empty cycle is the new node inserted.

Because of NP-hardness of both versions of the minimum ADM cost problem, we will develop approximation algorithms in the next, instead of seeking optimal solutions. We use OPT to denote an optimum solution and *opt* to denote the optimum cost.

The next lemma (Lemma 4 from [9]) shows that any nontrivial heuristic has an approximation ratio of at most 1.75 for the arc version. The same argument applies to the chord version.

Lemma 3 ([9]) *If any pair of chains in $\mathcal{P}$ cannot be merged into a larger valid chain, the cost of $\mathcal{P}$ is at most $\frac{7}{4} \cdot opt$.*

4. The Preprocessed Iterative Matching Algorithm

In this section, we propose the **Preprocessed Iterative Matching (PIM)** algorithm, which runs in two phases:

1. Preprocessing Phase: repeatedly take valid closed chains out of the remaining arcs until no more closed chain can be obtained from the remaining arcs.
2. Matching Phase: While the fit graph $F(\mathcal{P})$ of $\mathcal{P}$ has nonempty edge set, we find a maximum matching M in $F(\mathcal{P})$ and then merge each matched pair of chains of arcs in M into a larger chain. When $F(\mathcal{P})$ has empty edge set, $\mathcal{P}$ is output as the valid chain generation.

The following procedure is used by the Preprocessing Phase to obtain a closed valid chain containing a specified arc a, if there is any, from a set of arcs S. We build a directed acyclic graph (DAG) that consists of only those arcs in S that do not overlap with a. Obviously there is a path from $h(a)$ to $t(a)$ in the DAG if and only if there is a closed valid chain in S that contains a. By using *breadth-first search* in the DAG, we can obtain a path, if there is any, from the from $h(a)$ to $t(a)$. Once this path is obtained, we merge it with a to obtain a closed valid chain. For a link e, let A_e denote the set of arcs that contain the link e. After picking any link e, and setting $i \leftarrow 1$, the Preprocessing Phase runs as follows:

- While A_e is nonempty: pick any $a \in A_e$, and set $A_e \leftarrow A_e \setminus \{a\}$; if A has a valid closed chain P containing a, set $P_i \leftarrow P$, $A \leftarrow A \setminus P$, and $i \leftarrow i + 1$.

It is obvious that the algorithm has polynomial run-time. In the next, we show that its approximation ratio of is at most 1.5.

Lemma 4 *The approximation ratio of **PIM** is at most $\frac{3}{2}$.*

Proof. For the simplicity of description, we call the arcs appearing in the closed valid chains obtained in the Preprocessing Phase as *blue* arcs, and the others as *red* arcs. We use B and R to denote the set of blue arcs and the set of red arcs respectively. Then in any closed (valid) chain in OPT, at least one circular arc is blue. From OPT, we remove all blue arcs and obtain a collection of red (valid) chains. Note that the number of red chains obtained from a closed chain

P is at most the number of blue arcs in P; the number of red chains obtained from an open chain P is at most the number of blue arcs in P plus one. Thus the total number of red chains is at most $|B|$ plus the total number of open chains, and consequently is at most $|B| + opt - |A| = opt - |R|$. Let k be the number of *odd* red chains, and note that

$$k \leq opt - |R| . \tag{1}$$

From any red chain P, a matching of cardinality $\left\lfloor \frac{|P|}{2} \right\rfloor$ can be obtained. Overall, from the red chains we can generate a matching of cardinality $\frac{|R|-k}{2}$. Thus the cardinality of any maximum matching obtained in the first iteration in the Matching Phase is at least $\frac{|R|-k}{2}$, and consequently after this first iteration, the total number of remaining open chains is at most

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

So after the first maximum matching in the Matching Phase, using equation 1, the total ADM cost of the output of **PIM** is at most:

$$\begin{aligned} |A| + \frac{|R|+k}{2} &\leq |A| + \frac{|R| + opt - |R|}{2} \\ &= |A| + \frac{opt}{2} \leq opt + \frac{opt}{2} \\ &\leq \frac{3}{2} \cdot opt. \end{aligned}$$

Therefore the approximation ratio of **PIM** is at most $\frac{3}{2}$. ■

For completeness we include the following example from [9], showing that the approximation ratio of PIM is at least $4/3$:

Example 5 *([9]) Let $n = 6$, and $A = A_1 \cup A_2 \cup A_3$ where $A_1 = \{(0,2),(2,5),(5,0)\}$, $A_2 = \{(0,3),(3,4),(4,0)\}$, $A_3 = \{(1,2),(2,4),(4,1)\}$.*

Since the three arcs in A_i form a closed valid chain for any $1 \leq i \leq 3$, $opt = |A| = 9$. If the preprocessing takes out the following closed valid chain $\{(0,2),(2,4),(4,0)\}$, then the remaining 6 arcs do not contain any closed valid chain. Since the three arcs $(0,3)$ and $(2,5)$ and $(4,1)$ are pairwise intersecting, at least three open valid chains must be used by the remaining six arcs. One (optimal) solution is the following: $\{(2,5),(5,0)\},\{(4,1),(1,2)\},\{(0,3),(3,4)\}$. Thus the total ADM cost of all these valid chains is $12 = \frac{4}{3} \cdot opt$.

Thus we have the following theorem:

Theorem 6 *The approximation ratio of **PIM** is between $\frac{4}{3}$ and $\frac{3}{2}$.*

4.1. PIM for the Chord-Version

The **Preprocessed Iterative Matching (PIM)** for chords is adapted from the PIM algorithm for arcs. The algorithm is described, without proofs, in [9], where it is called chord-version Closed Chain First. The chord-version PIM also runs in two phases:

1. Preprocessing Phase: greedily take valid closed chains out of the remaining chords until no more closed chain can be obtained from the remaining chords.

2. Matching Phase: While the fit graph $F(\mathcal{P})$ of $\mathcal{P}$ has nonempty edge set, we find a maximum matching M in $F(\mathcal{P})$ and then merge each matched pair of chains of arcs in M into a larger chain. When $F(\mathcal{P})$ has empty edge set, $\mathcal{P}$ is output as the valid chain generation.

The following procedure is used by the Preprocessing Phase to obtain a closed valid chain containing a specified chord c, if there is any, from a set of chords S. Let c^- be the circular arc between the two endpoints of c that passes through the link between node $n-1$ and node 0, and let c^+ be the arc complementary to c^-. Let S_c^+ (S_c^- respectively) be the set of chords in $S - \{c\}$ whose two endpoints are both in c^+ (c^- respectively). Let G_c^+ (G_c^- respectively) be the directed graph with the nodes in c^+ (c^- respectively) as its vertices and directed edges obtained from orienting the chords in S_c^+ to not use the the link between node $n-1$ and node 0, (orienting the chords in S_c^- to use the link between node $n-1$ and node 0, respectively). There is a closed valid chain in S that contains c if and only if either there is a path between the two endpoints of c in G_c^+, or there is a path between the two endpoints of c in G_c^-. After constructing G_c^+ and G_c^-, such a path, if there is any, can be found by breadth-first search. Once this path is obtained, we add c to it to obtain a closed valid chain.

Following the argument in Lemma 4, we can prove that the approximation ratio of **PIM** for chords is at most 1.5. Indeed, first note that the proof of Lemma 4 only uses one iteration of matching. For this iteration, the fit graph F has a set of chords as vertices and has an edge in between two chords if they share an endpoint. Note that any two chords which share an endpoint can be routed into one valid chain. Further iterations, in which one can 'fit' chains with chords, are possible, but not necessary for the $3/2$ approximation ratio. Second, any closed valid chain of OPT must contain a blue chord, or otherwise this chain would have been found by the algorithm in the Preprocessing Phase.

Replacing each arc in Example 5 with a chord between the same endpoints leads to an instance for which the **PIM** for chords may produce a solution of $\frac{4}{3}$ times the optimum. So we have the following theorem.

Theorem 7 *The approximation ratio of **PIM** for chords is between $\frac{4}{3}$ and $\frac{3}{2}$.*

5. A Local-Improvement Algorithm

In this section we describe a simple local-improvement algorithm, and prove it has approximation ratio $3/2$.

Let $\mathcal{P}$ denote any valid partition of A into valid chains. The first arc and the last arc of an open chain P of length at least two are called the *end arcs* of P. We color the arcs in $\mathcal{P}$ as follows: each arc in a 1-chain with *red* color, each end-arc with *green* color, and all other arcs with *blue* color. This coloring is referred to as $\mathcal{P}$-coloring. Repeat the following procedure:

- **Repeatedly** merge any pair of open chains which can be merged into a larger valid chain.
- **If** there is a valid closed chain P which consists of green arcs and at least one red arc, remove all arcs in P from their original chains and add P to $\mathcal{P}$;
- **Else** if there is a valid open chain P which consists of green arcs and a pair of red arcs as its end arcs, remove all arcs in P from their original chains and add P to $\mathcal{P}$;
- **Else** exit.

Note that each operation in the first three bullet items reduce the number of open chains by at least one. Thus the algorithm terminates in linear number of iterations. The output partition $\mathcal{P}$ has the following three properties:

1. Any pair of open chains cannot be merged into a larger valid chain.
2. Each red arc cannot form a closed chain together with green arcs in $\mathcal{P}$.
3. Each pair of red arcs cannot be joined by green arcs in $\mathcal{P}$ into a valid chain.

A partition satisfying these three properties is said to *canonical*. In the remaining we show that the cost of any canonical partition $\mathcal{P}$ is at most 1.5 times opt.

We distribute the cost of $\mathcal{P}$ to arcs as follows: Each red arc costs 2, each green arc costs 1.5, and each blue arc costs 1. It is sufficient to show that the cost of all arcs in an closed (open, respectively) chain P in OPT is at most $1.5\,|P|$ ($1.5\,(|P|+1)$, respectively).

Let P be any chain of optimum. Let i and j be the number of red arcs and blue arcs in $\mathcal{P}$ respectively. Note that all these red arcs cannot be consecutive in P from Property 1. Then the total cost of the arcs of P is

$$2i + j + 1.5\,(|P| - i - j) = 1.5\,|P| + 0.5\,(i - j)\,.$$

We first assume that P is closed. It is sufficient to show that $j \geq i$. This is true when $i = 0$. If $i = 1$, then $j \geq 1$ for otherwise it violates the Property 2. If $i > 1$, then $j \geq i$ for otherwise at least one pairs of the red arcs can be joined by green arcs, which violates Property 3.

Now we assume that P is open. It is sufficient to show that $j \geq i-1$ (actually, we only have to show that $j \geq i-3$). This is true when $i = 0$ or 1. If $i > 1$, then $j \geq i-1$ for otherwise at least one pairs of the red arcs can be joined by green arcs, which violates Property 3.

We also have an example for which the local-improvement algorithm stops with a solution of cost exactly $3/2$ the optimum solution. Consider six arcs: $(0,1), (1,2), (2,0), (4,1), (1,3), (3,4)$. Optimum has cost six, while, if breaking ties the wrong way, the local-improvement algorithm selects the three chains: $\{(0,1),(1,3)\}$, $\{(3,4),(4,1)\}$ and $\{(1,2),(2,0)\}$, of total cost nine.

In conclusion, we have:

Theorem 8 *The approximation ratio of the local-improvement algorithm is $3/2$.*

5.1. The Local-Improvement Algorithm for the Chord-Version

In this subsection, we show that the local-improvement can be adapted to the chord-version problem, with the same bounds (and basically, same proofs) on the approximation ratio.

We describe how to implement each operation of the algorithm when applied to chords. The algorithms maintains a collection of already routed chains, plus the collection of single chords.

Any two non-identical chords which share an endpoint can be routed in a unique way as to form a valid chain. If a valid chain and a chord share an endpoint, they can be merged in a valid chain if and only if the other endpoint of the chord is not in the interior of the chain. Thus we can determine easily if merging two chains, two chords, or a chain with a chord, is possible in the current solution. This takes care of the first step of the algorithm.

A valid closed chain which consists of a red chord c and one or more green or red chords can be found, if it exists, by the methods used in the preprocessing step of the PIM algorithm.

A valid open chain which consists of green chords and a pair of red chords as its end arcs can also be found, as described below. For any pair of non-identical red chords (and by this stage there are no identical red chords, since they would have been merged in the first step), there is at most one way to route them such that the two obtained arcs do not overlap. Also, the green chords can be routed in at most one way to not overlap the two arcs obtained from the red chords. This unique routing reduces the chords to arcs, and therefore the methods used for arcs can be applied.

6. Eulerian Tour Decomposition for the Chord-Version Problem

In this section we propose the following **Eulerian Tour Decomposition (ETD)** heuristic, for the chord-version minimum ADM cost problem. A similar technique can be applied to the arc version, but has approximation ratio 1.75. Without loss of generality, we assume that $G(C)$ is connected, for otherwise we can apply the algorithm to each connected component separately. We consider two cases.

In the first case, some nodes in the ring have odd degree. The algorithm ETD runs in the following four steps:

1. Step 1: Divide the set of nodes with odd degree into disjoint pairs. Add one *fake* chord between the two nodes in each pair. Let C' be the set of fake chords. Then $|C'| = d(C)$ and the graph $G(C \cup C')$ is Eulerian. Find an Eulerian tour of $G(C \cup C')$.

2. Step 2: Remove all fake chords from the Eulerian tour to obtain $d(C)$ trails of $G(C)$.

3. Step 3: Split each trail into valid chains by walking along the trail from the first chord and generating a valid chain right before overlap occurs;

4. Step 4: Repeatedly merge any pair of open valid chains into a larger valid chain until no more merging can occur.

In the second case, all nodes in the ring have even degree. Thus the graph $G(C)$ is Eulerian and has an Eulerian tour. If $|C|$ is even, we apply Step 3 and Step 4 in the first case to any Eulerian tour of $G(C)$. If $|C|$ is odd, we obtain an Eulerian tour of $G(C)$ whose first three chords form a valid chain, if there is any, as follows. Fix a pair of adjacent but non-identical chords c_1 and c_2. Let (u, v) denote the (clockwise) arc induced by the unique routing of the 2-chain consisting of c_1 and c_2. Without loss of generality, we assume that u is the an endpoint of c_1 and v is the an endpoint of c_2. If there is a chord c with one endpoint being u and the other endpoint being v or inside the arc (v, u), then we can obtain a desired Eulerian tour whose first three chords are c, c_1, and c_2 sequentially. Otherwise if there is a chord c with one endpoint being v and the other endpoint being inside the arc (v, u), then we can obtain a desired Eulerian tour whose first three chords are c_1, c_2, and c. sequentially. Otherwise, we repeat the previous procedure to other unselected pair of adjacent but non-identical chords. Eventually, we either obtain a desired Eulerian tour, or conclude that any three chords cannot form a valid chain. In the former situation, we apply Step 3 and Step 4 in the first case to the obtained Eulerian tour. In the latter situation, we apply Step 3 and Step 4 in the first case to any Eulerian tour.

Lemma 9 *The approximation ratio of* **ETD** *is at most* 1.5.

Proof. We first consider the first case, i.e. $d(C) \geq 1$. Step 2 generates $d(C)$ trails. Let P be any such trail. Since each pair of consecutive chords in P can form a valid chain, all valid chains except the last one split from P at Step 3 contains at least two chords. Thus at most $\left\lceil \frac{|P|}{2} \right\rceil$ chains are split from P. So, in total, at most $\frac{|C|+d(C)}{2}$ chains are generated at Step 3, and costs at most

$$\begin{aligned} |C| + \frac{|C| + d(C)}{2} &= \frac{3|C| + d(C)}{2} \\ &\leq \frac{3}{2}(|C| + d(C)) \\ &\leq \frac{3}{2} \cdot opt. \end{aligned}$$

This solves the first case, when $d(C) \geq 1$.

In the next we consider the second case, i.e. $d(C) = 0$. If $|C|$ is even, then following the same argument above, at most $\frac{|C|}{2}$ chains are produced. So the total cost is at most

$$|C| + \frac{|C|}{2} = \frac{3}{2}|C| \leq \frac{3}{2} \cdot opt.$$

So we now assume that $|C|$ is odd. If some three chords form a valid chain, let $k \geq 3$ be the number of chords in the first valid chain split from the obtained Eulerian tour. Then again from the previous argument, at most $1 + \left\lceil \frac{|C|-k}{2} \right\rceil$ chains are produced. So the total cost is at most

$$\begin{aligned} |C| + 1 + \left\lceil \frac{|C| - k}{2} \right\rceil &\leq |C| + 1 + \frac{|C| - k + 1}{2} \\ &= \frac{3|C| - k + 3}{2} \leq \frac{3}{2}|C| \\ &\leq \frac{3}{2} \cdot opt. \end{aligned}$$

If any three chords in C cannot form a valid chain, we prove by contradiction that $opt > |C|$. Assume to the contrary. Then in any optimal solution, each valid chain is closed, and therefore must be a closed 2-chain. This implies that $|C|$ is even, which is a contradiction. Thus $opt \geq |C| + 1$. Since exactly $\frac{|C|+1}{2}$ chains are generated, the total cost is at most

$$|C| + \frac{|C| + 1}{2} = \frac{3|C| + 1}{2} < \frac{3}{2}(|C| + 1) \leq \frac{3}{2} \cdot opt.$$

Therefore, in all cases, the lemma is true. ■

The next example shows that the approximation ratio of **ETD** is at least 1.5.

Example 10 *Let $n = 2(2k+1)$ for some $k > 1$, and C consist of the following $\frac{3n}{2}$ chords*

$$\left\{c_i = \left(2i, 2i + \frac{n}{2}\right) \mid 0 \leq i < \frac{n}{2}\right\} \cup$$
$$\left\{c_i' = \left(2i + \frac{n}{2}, 2i + \frac{n}{2} + 1\right) \mid 0 \leq i < \frac{n}{2}\right\} \cup$$
$$\left\{c_i'' = \left(2i + \frac{n}{2} + 1, 2i\right) \mid 0 \leq i < \frac{n}{2}\right\}.$$

*Note that for any $0 \leq i < n$, the three chords c_i, c_i', c_i'' form a closed valid chain. So opt $= |C| = \frac{3n}{2} = 3(2k+1)$. As $d(C) = 0$ and $|C|$ is odd, the algorithm **ETD** tries to find an Eulerian tour in which the first three chords form a valid chain. The algorithm, if breaking ties the wrong way, produces the following Eulerian tour*

$$c_k', c_k'', c_{2k}'', c_{k-1}'', c_{2k-1}'', \cdots, c_1'', c_{k+1}'', c_0'',$$
$$c_0, c_0', c_{k+1}, c_{k+1}', c_1, c_1', c_{k+2}, c_{k+2}', \cdots,$$
$$c_{k-1}, c_{k-1}', c_{2k}, c_{2k}', c_k.$$

At Step 3, the following valid chains are generated:

- *one open valid 3-chain* $\{c_k', c_k'', c_{2k}''\}$,
- $k-1$ *open valid 2-chains* $\{\{c_i'', c_{i+k}''\} \mid 1 \leq i < k\}$,
- *one closed 3-chain* $\{c_0'', c_0, c_0'\}$,
- $2k - 1$ *open valid 2-chains* $\{\{c_i, c_i'\} \mid 1 \leq i \leq 2k, i \neq k\}$.
- *one open valid 1-chain* $\{c_k\}$.

So, in total, $3k$ open chains and one closed chain are obtained. These chains costs

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

*Thus the approximation ratio of **ETD** is at least $\frac{3}{2}$.*

From Lemma 9 and Example 10, we have the following theorem.

Theorem 11 *The approximation ratio of **ETD** is exactly* 1.5.

References

[1] Tamra Carpenter, Steven Cosares, and Iraj Saniee, "Demand Routing and Slotting on Ring Networks", *DIMACS Technical Report* 97-02, 1997.

[2] O. Gerstel, P. Lin, and G. Sasaki, "Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings", *Proc. IEEE INFOCOM'98* (San Francisco, CA), Vol. 1, pp. 94 –101.

[3] O. Gerstel, P. Lin, and G. Sasaki, "Combined WDM and SONET Network Design", *Proc. IEEE INFOCOM'99* (New York, NY), Vol. 2, pp. 734 –743.

[4] I. Haque, W. Kremer, and K. Raychauduri, "Self-Healing Rings in a synchronous environment", in *SONET/SDH: a sourcebook of synchronous networking*, Eds. C.A. Siller and M. Shafi, IEEE Press, New York, pp. 131–139, 1996.

[5] L.W. Liu , X.-Y. Li, P.-J. Wan, and O. Frieder, "Wavelength Assignment in WDM Rings to Minimize SONET ADMs", *Proc. INFOCOM 2000*, (Tel-Aviv, Israel), Vol. 2, pp. 1020 –1025.

[6] P. Raghavan and E. Upfal, "Efficient routing in all-optical networks", in *Proc. 26th ACM Symp. Theory of Computing*, pp. 134-143, 1994.

[7] A. Schrijver, P.D. Seymour, and P. Winkler, " The Ring Loading Problem", *SIAM J. Disc. Math.*, Vol. 11, No. 1, pp. 1-14, February 1998.

[8] A. Tucker, "Coloring a family of circular arcs", *SIAM Journal of Applied Math*, vol. 29, no. 3, pp. 493-502, 1975.

[9] P.-J. Wan, G. Călinescu, L.-W. Liu, O. Frieder, "Grooming of Arbitrary Traffic in SONET/WDM Rings", *IEEE Journal of Selected Area on Communications*, vol. 18, no. 10, pp. 1995-2003, 2000.

[10] P. Wilfong, P. Winkler, Ring Routing and Wavelength Translation, *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 333-341, 1998.