Journal of Combinatorial Optimization, 6, 425–453, 2002

Traffic Partition in WDM/SONET Rings to Minimize SONET ADMs

GRUIA CĂLINESCU* calinescu@cs.iit.edu
PENG-JUN WAN wan@cs.iit.edu
Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA

Received December 11, 2000; Revised September 12, 2001; Accepted September 13, 2001

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. However, little was known about the performance guarantees of these heuristics. 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. Several heuristics for integrated routing and wavelength assignment are proposed to minimize the SONET ADM costs. Second, the approximation ratios of those heuristics for wavelength assignment only and those heuristics for integrated routing and wavelength assignment are analyzed. The new Preprocessed Iterative Matching heuristic has the best approximation ratio: at most $3/2$.

Keywords: SONET, WDM, add-drop multiplexer, grooming, approximation algorithms

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 (Haque et al., 1996). 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 optimal 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*.

*The work of the first author was supported by a DIMACS postdoctoral fellowship.

A number of previous works (Gerstel et al., 1998, 1999; Liu et al., 2000) 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, $\Pi = \{A_1, A_2, \ldots, A_w\}$, such that for any $1 \leq i \leq w$ all arcs in each A_i are non-intersecting.
- *Cost*: the cost of each A_i is the number of different nodes of the ring that are the endpoints of the arcs in A_i, 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 Liu et al. (2000) that this arc-version minimization problem is NP-hard. Several heuristics have been proposed in Gerstel et al. (1998) and Liu et al. (2000). However, little is known about their analytical performances in terms of approximation ratios. 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 Wan et al. (2000) shows that any nontrivial heuristic has an approximation ratio of at most 1.75. Thus, it is interesting to know whether those heuristics proposed in Gerstel et al. (1998) and Liu et al. (2000) can beat the 1.75 ratio. In this paper, we focus on three heuristic approaches, namely cut-and-merging (Gerstel et al., 1998; Liu et al., 2000), trail splitting (Liu et al., 2000), and iterative matching (Liu et al., 2000). We prove that the former two both have an approximation ratio of exactly 1.75, and the last one has an approximation ratio of between 1.5 and $\frac{5}{3}$. We then propose improvements on these three heuristics. The improved cut-and-merging has an approximation ratio between 1.5 and $\frac{5}{3}$, the improved trail splitting has an approximation ratio between 1.5 and 1.6, and the improved iterative matching has the an approximation ratio between $\frac{4}{3}$ and 1.5. In addition, we point out that the number of wavelengths used by any of these heuristics is less than 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) (Haque et al., 1996). 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, $\Pi = \{C_1, C_2, \ldots, C_w\}$, 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 as 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. We first prove the NP-hardness of this integrated problem, and then points out that any nontrivial heuristic has an approximation ratio of at most 1.75 (again, approximation ratio of 2 is trivial). Thus, it is interesting to find heuristics which can beat the 1.75 ratio. We then show that both minimum-load routing (Schrijver et al., 1998; Wilfong and Winkler, 1998) and minimum-wavelength routing (Raghavan and Upfal, 1994) fail to beat the 1.75 ratio. We also show that the two well-known routing heuristics, Edge-Avoidance routing and Shortest-Path routing (Carpenter et al., 1997), fail to beat the 1.75 ratio as well. Even after the preprocessing by pairing up identical chords, their approximation ratios is still at least $\frac{5}{3} \approx 1.67$. We then propose two heuristic approaches based on Eulerian tour decomposition and matching respectively. While its counterpart for arcs performs poorly and requires improvement to get an approximation ratio between 1.5 and 1.6, the Eulerian Tour Decomposition for chords has an approximation ratio of exactly 1.5. The Iterative Matching and its variants have the similar performance as their counterpart for arcs. We prove that the approximation ratio of the basic Iterative Matching is between 1.5 and $\frac{5}{3}$, and the approximation ratio of its improvement by a preprocessing with repeatedly taking out chords which can form a ring is between $\frac{4}{3}$ and 1.5. A remark is that some partial or weaker results presented in this paper were claimed without proofs in our earlier survey paper (Wan et al., 2000).

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 prove the NP-hardness of the chord-version minimum ADM cost problem. In Section 4 and Section 5, we describe several heuristics for the arc-version and the chord version minimum ADM cost problem respectively. Their approximation ratios are analyzed. Finally we conclude this paper in Section 6.

2. Terminology and formulation

2.1. *Walk, trail, path, circuit*

To avoid confusion, we first present the standard definitions of some basic concepts in graph theory. A *walk* in a graph G is a sequence

$$W := (v_0, e_1, v_1, e_2, v_2, \ldots, e_\ell, v_\ell),$$

where $v_0, v_1, \ldots, v_\ell$ are vertices of G, $e_1, e_2, \ldots, e_\ell$ are edges G, and e_i is the edge joining v_{i-1} and v_i for $i = 1, \ldots, \ell$. In a simple graph, the edges in a walk is completely specified by its sequence of vertices $(v_0, v_1, \ldots, v_\ell)$. The walk W is *closed* if $n > 0$ and $v_0 = v_n$, and *open* otherwise. The *length* of W is the number of edges, namely ℓ, and is denoted by $|W|$; the *parity* of W is the parity of its length. A walk of length ℓ is called as an ℓ-walk. If W is open, the vertex v_0 is the tail of W, the vertices $v_1, \ldots, v_{\ell-1}$ its *internal* vertices and the vertex v_ℓ 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. It is obvious that a trail can be decomposed into paths and circuits. The definitions of these concepts in a directed graph are the same as in a graph with the exception that the edge must be traversed in the correct direction.

An *Eulerian trail* in a graph is a trail which includes every edge. A closed Eulerian trail is also called an *Eulerian tour*. A (directed) graph is said to be *Eulerian* if it has an Eulerian tour. It is well known that a graph is Eulerian if and only if it is connected and every vertex has even degree, and a directed graph is Eulerian if and only if it is connected and the in-degree and out-degree of any vertex are equal. An Eulerian tour can be generated in polynomial time.

A *circuit cover* of a (directed) graph G is a collection of vertex-disjoint circuits which together cover all vertices of G. If G is a weighted (directed) graph, then the weight of a circuit cover of G is the sum of the weights of all edges in this circuit cover. It is well-known that a minimum-weighted circuit cover of G can be found in polynomial time by a reduction to minimum-weight perfect matching in a bipartite graph.

2.2. *Ring, arc, chord*

We assume that a ring network consists of n nodes numbered clockwise by $0, 1, \dots, n-1$. 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, *disjoint* if they do not contain any common node of the ring, *adjacent* if they are not intersecting but share at least one endpoint, and *complementary* if they are not intersecting and share two endpoints.

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 *crossing* if they have different endpoints and a walk along the ring visits their endpoints alternatively. Two chords are said to be *parallel* if they have different endpoints and are not crossing. 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.

2.3. *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

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

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

$$\sum_{i=0}^{n-1} \max\{\sigma_A(i), \tau_A(i)\}$$

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

So $|A| + d(A)$ is a lower bound on the minimum ADM cost required by A. This fact was first noticed by (Gerstel et al., 1998).

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 $\lceil \frac{\deg_C(i)}{2} \rceil$ ADMs, the total number of ADMs required by C is at least

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

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

2.4. *Chains of arcs or chords*

Let A be any set of arcs over a ring. It induces naturally a directed (undirected respectively) 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. It is obvious that a trail of arcs can be decomposed into chains and a chain of arcs can be decomposed into valid chains of arcs.

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 chord 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. It is obvious that a trail of chords can be decomposed into chains of chords, and a chain of chords can be decomposed into valid chains.

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 weighted 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, and the weight of the edge between P_1 and P_2 is equal to the number of endpoints shared by P_1 and P_2.

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 for circular-arc coloring (Tucker, 1975). 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 and the number of (valid) closed chains is k. Then the load of those open chains is $l - k$. The Tucker's algorithm for circular-arc coloring guarantees that the number of wavelengths used by those open chains is at most $2(l - k) - 1$. Thus, the total number of wavelengths is at most $k + 2(l - k) - 1 = 2l - k - 1$, which is within twice the minimum wavelengths.

Notice that the cost of chains produced a valid chain generation is the number of input arcs or chords plus the number of open chains. Therefore, an optimal solution corresponds to the a valid chain generation with minimal number of open valid chains.

3. NP-hardness

The NP-hardness of the arc-version minimum ADM cost problem follows from the result proved in Liu et al. (2000) that:

Theorem 1 (Liu et al. 2000). *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: 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,

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

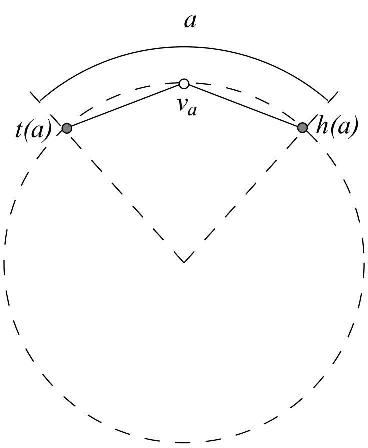

Figure 1. Replace each arc by two chords: the filled circles are the endpoints of the arc, and the empty cycle is the new node inserted.

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 ring. □

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.

Fix a set of arcs A (chords C respectively). Let *OPT* denote the optimal valid chain generation from A (C respectively) and let *opt* denote the cost of *OPT*. Consider any valid chain generation $\mathcal{P}$ of A (C respectively). As discussed in Section 2, the cost of $\mathcal{P}$ is $|A|$ ($|C|$ respectively) plus the open chains in $\mathcal{P}$. Since the number of open chains in $\mathcal{P}$ can never exceed $|A|$ ($|C|$ respectively), the cost of $\mathcal{P}$ is at most $2|A|$ ($2|C|$ respectively). Note that $|A|$ ($|C|$ respectively) is a lower bound on *opt*. Thus the cost of $\mathcal{P}$ is at most $2 \cdot opt$. This implies that two is a trivial upper bound on the approximation ratio of *any* valid chain generation.

The next lemma (Lemma 4 from Wan et al. (2000)) 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 (Wan et al., 2000)**.** *If any pair of chains in $\mathcal{P}$ can not be merged into a larger valid chain, the cost of $\mathcal{P}$ is at most $\frac{7}{4} \cdot opt$.*

By repeatedly merging two non-overlapping chains which share at least one endpoint into a larger (valid) chain, one can convert any valid chain generation into one in which any pair of (valid) chains can not be merged into a larger valid chain. Lemma 3 implies that any algorithm appended with this post-processing will have an approximation ratio of at most $\frac{7}{4}$. In the next, we will present several algorithms whose approximation ratios beat $\frac{7}{4}$.

4. Valid chain generation from arcs

In this section, we will describe three heuristic approaches for valid chain generation from arcs in Subsections 4.1, 4.2 and 4.3 respectively. Their approximation ratios are analyzed as well.

4.1. Cut and merge

If all circular arcs in A do not cross over some link of the ring, then an optimal solution can be obtained in polynomial time. In fact by removing this link from the ring, we can treat all arcs as intervals along a line. A simple *Greedy Sweeping* algorithm for an optimal partition of these intervals into valid chains works as follows (Gerstel et al., 1998). At the left-most node, each interval starting from this node forms a 1-chain. Subsequently at each node, if an interval starting from this node can be merged with an existing chain, then merge them to form a larger chain; otherwise create a new 1-chain from this interval. This procedure is repeated until no interval is left. The optimality of this algorithm follows from the fact that each node i uses max $\{\sigma_A(i), \tau_A(i)\}$ ADMs.

The *Cut And Merge (CM)* heuristic (Gerstel et al., 1998; Liu et al., 2000) is a slight revision from the well-known Tucker's algorithm (Tucker, 1975) for circular-arc coloring. It consists of the following two steps:

- *Step 1*: Choose a link i with minimum load, and let B_i denote the set of circular arcs in A that pass through link i. Partition $A \backslash B_i$ into an optimal set of chains using the Greedy Sweeping algorithm. Let $\mathcal{P}'$ denote the obtained chains.
- *Step 2*: Construct a weighted bipartite graph between B_i and $\mathcal{P}'$ as follows: if an arc a in B_i can be merged with a chain P in $\mathcal{P}'$, add an edge between a and P with weight equal to their common endpoints. Find a maximum-weighted bipartite matching in the resulting bipartite graph. Merge each matched pair of arc and chain into a larger chain. This process is repeated until no further matching (or merging) can be obtained.

We first use the following example to illustrate the algorithm and to derive a lower bound on its approximation ratio.

Example 4. Let $n = 4$, and $A = A_1 \cup A_2 \cup A_3 \cup A_4$ where

$$A_1 = \{(0, 2), (2, 0)\}, A_2 = \{(1, 2), (2, 1)\},$$
$$A_3 = \{(1, 3), (3, 1)\}, A_4 = \{(3, 0), (0, 3)\},$$

as illustrated in figure 2(a). Since the two arcs in each A_i form a closed valid chain for $1 \leq i \leq 4$, $opt = |A| = 8$. Since each link has the same load equal to four, during Step 1

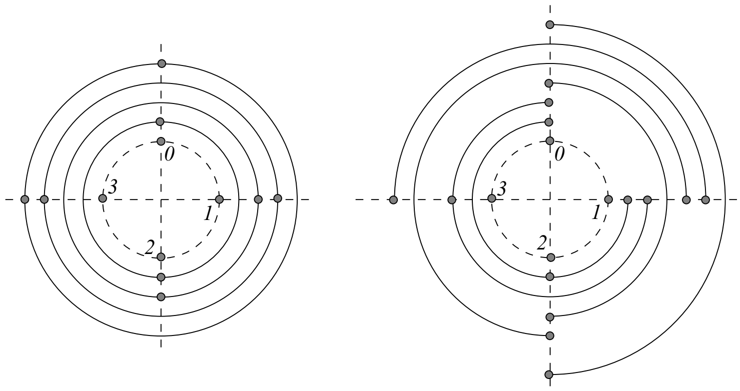

Figure 2. The instance in Example 4: (a) the optimal solution and (b) the solution produced by the algorithm Cut and Merge.

link 0 may be chosen as the link to be removed, and

$$B_0 = \{(0, 2), (2, 1), (3, 1), (0, 3)\}.$$

The algorithm, if unlucky, would partition $A \backslash B_0$ into two valid chains

$$\{(1, 2), (2, 0)\}, \{(1, 3), (3, 0)\}.$$

During Step 2, the four arcs in B_0 can not be merged with either of the above two chains. So they will form four 1-chains alone. Thus, totally six open chains are generated (see figure 2(b)). Their total ADM cost is

$$8 + 6 = 14 = \frac{7}{4} \cdot opt.$$

This implies that the approximation ratio of CM is at least $\frac{7}{4}$.

From Lemma 3 and Example 4, we have the following theorem.

Theorem 5. *The approximation ratio of CM is exactly* 1.75.

Example 4 suggests that it might improve the performance by taking out all complementary pairs of arcs before applying the algorithm CM. Indeed, it is easy to verify that there is an optimal solution in which these complementary arcs each form a closed chain. We refer the algorithm which takes out all complementary pairs of arcs and then applies CM to the remaining arcs by *Preprocessed Cut-And-Merge (PCM)*.

The next lemma provides an upper bound on the approximation ratio of PCM.

Lemma 6. *The approximation ratio of PCM is at most* $\frac{5}{3}$.

Proof: Let A' denote the set of arcs in A that survive after the preprocessing. Then all arcs in $A\backslash A'$ form closed valid 2-chains and thus has total cost of $|A\backslash A'|$. Let *OPT* be any optimal solution in which all arcs in $A\backslash A'$ form closed valid 2-chains.

Without loss of generality, assume link 0 has the least load. Let B_0 be the set of arcs in A' that pass through link 0. Let B_0' denote the set of arcs in B_0 that form 1-chains in *OPT*, and $B_0'' = B_0\backslash B_0'$. Thus each arc in B_0'' is within some chain of length at least two, and no two arcs in B_0'' can be in the same chain in *OPT*. Let P be any chain in *OPT* that contains an arc in B_0''. The cost of P is at least three, since if it is closed its length must be at least three, and if it is open, its length must be at least two. Since the total cost of these chains which contain arcs in B_0'' is at most $opt - |A\backslash A'| - 2|B_0'|$, we have:

$$|B_0''| \leq \frac{1}{3}(opt - |A\backslash A'| - 2|B_0'|).$$

Note that OPT induces a partition of $A'\backslash B_0$ into chains of cost at most $opt - |A\backslash A'| - 2|B_0'|$. On the other hand, Step 2 obtains an optimal partition of $A'\backslash B_0$ into valid chains. Therefore, the total cost of our solution is at most

$$\begin{aligned}
&|A\backslash A'| + (opt - |A\backslash A'| - 2|B_0'|) + 2|B_0| \\
&\quad = opt + 2|B_0''| \\
&\quad \leq opt + \frac{2}{3}(opt - 2|B_0'|) \\
&\quad \leq \frac{5}{3}\, opt - \frac{4}{3}|B_0'| \leq \frac{5}{3}\, opt.
\end{aligned}$$

□

The next examples indicates that even after the preprocessing, the approximation ratio of PCM is still at least $\frac{3}{2}$.

Example 7. Let $n = 8$, and $A = A_1 \cup A_2 \cup A_3 \cup A_4$ where

$$\begin{aligned}
A_1 &= \{(0, 3), (3, 4), (4, 0)\}, \\
A_2 &= \{(0, 6), (6, 7), (7, 0)\}, \\
A_3 &= \{(1, 2), (2, 3), (3, 1)\}, \\
A_4 &= \{(1, 5), (5, 6), (6, 1)\},
\end{aligned}$$

as illustrated in figure 3(a). Since the three arcs in each A_i form a closed valid chain for $1 \leq i \leq 4$, $opt = |A| = 12$. Note that all links have load equal to four. Thus during Step 1, link 0 may be chosen as the cutting node, and

$$B_0 = \{(0, 3), (0, 6), (3, 1), (6, 1)\}.$$

The algorithm, if unlucky, would partition $A\backslash B_0$ into two valid chains

$$\begin{aligned}
&\{(1, 2), (2, 3), (3, 4), (4, 0)\}, \\
&\{(1, 5), (5, 6), (6, 7), (7, 0)\}.
\end{aligned}$$

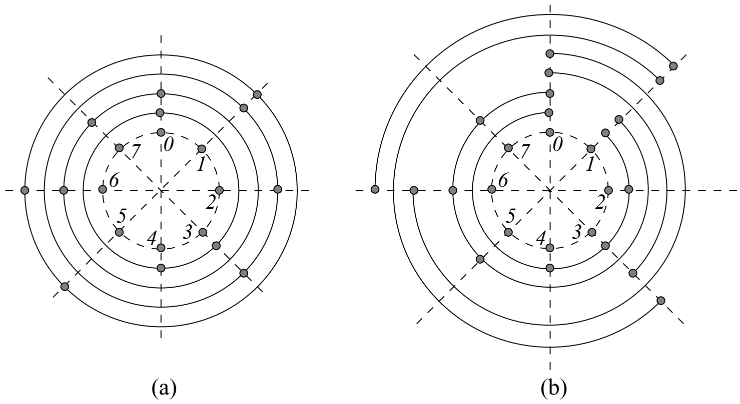

Figure 3. An instance used in Example 7: (a) the optimal solution and (b) the solution produced by the algorithm PCM.

During Step 2, the four arcs in B_0 can not be merged with either of the above two chains. So they will form four 1-chains alone. Thus, totally six open chains are generated (see figure 3(b)). Their total ADM cost is

$$12 + 6 = 18 = \frac{3}{2} \cdot opt.$$

This implies that the approximation ratio of PCM is at least $\frac{3}{2}$.

From Lemma 19 and Example 7, we have the following theorem.

Theorem 8. *The approximation ratio of PCM is between* $\frac{3}{2}$ *and* $\frac{5}{3}$.

4.2. Trail splitting

We start by describing the a trail splitting algorithm based on Eulerian tour, referred to as *ET-TS*.

1. *Eulerian tour phase*: Add a set of fake arcs A' such that $|A'| = d(A)$ and $d(A \cup A') = 0$. This can be easily done by adding one by one fake arc from a node i with $\sigma_A(i) > \tau_A(i)$ to a node j with $\sigma_A(j) < \tau_A(j)$, thus each fake arc decreasing the deficiency by one. Now the directed graph $G(A \cup A')$ is Eulerian. Find an Eulerian tour of $G(A \cup A')$. Remove all fake arcs from the Eulerian tour to obtain $d(A)$ trails of $G(A)$.
2. *Trail decomposing phase*: Decompose each trail into simple paths and circuits of $G(A)$. Each simple path induces an open chain, and each cycle induces a closed chain.
3. *Chain split phase*: Split each invalid open chain into valid chains by walking along the chain from the first arc and generating a valid chain whenever overlap occurs; split each invalid closed chain into valid chains by walking along the chain from *each* arc and

generating a valid chain whenever overlap occurs, and then choose the one with the smallest number of open chains.
4. *Chain merging phase*: Repeatedly merge any pair of open valid chains into a larger valid chain until no more merging can occur.

From Lemma 3, the approximation ratio of ET-TS is at most 1.75. The next example shows it is also at least 1.75.

Example 9. Let $n = 2k + 1$ for some $k > 1$, and A consist of the following $2n$ arcs

$$\{a_i = (i, i + k + 1) \mid 0 \le i < n\} \cup \{a'_i = (i + k + 1, i) \mid 0 \le i < n\},$$

as illustrated in figure 4(a). Note that for any $0 \le i < n$, the arcs a_i and a'_i are complementary and thus form a closed valid chain. So $opt = |A| = 2n$. On the other hand, *ET-TS* may produce at Eulerian Tour Phase an Eulerian tour

$$a_0, a_{k+1}, a_1, a_{k+2}, a_2, \ldots, a_k, a'_k, a'_{2k}, a'_{k-1}, a'_{2k-1}, a'_{k-2}, \ldots, a'_0.$$

It is then broken into two circuits at Trail Decomposing Phase:

$$a_0, a_{k+1}, a_1, a_{k+2}, a_2, \ldots, a_k;$$

and

$$a'_k, a'_{2k}, a'_{k-1}, a'_{2k-1}, a'_{k-2}, \ldots, a'_0.$$

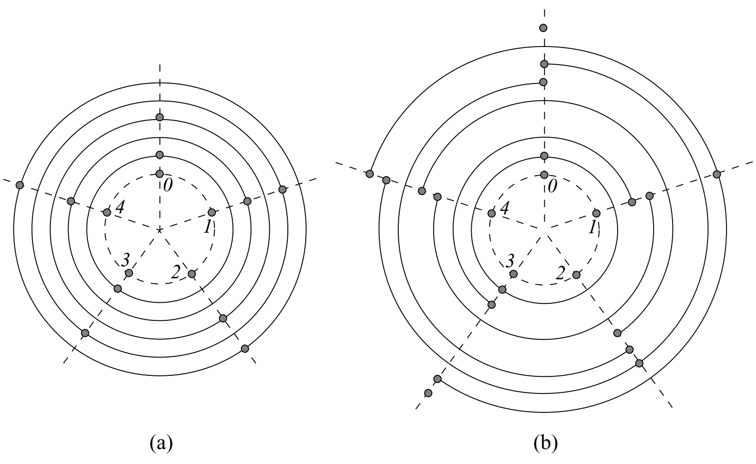

Figure 4. An instance used in Example 9: (a) the optimal solution and (b) the solution produced by the algorithm ET-TS.

Both circuits leads to an invalid closed chain. At Chain Split Phase, the first invalid closed chain is split into $n = 2k + 1$ open valid 1-chains $\{\{a_i\} \mid 0 \leq i < n\}$; the second invalid closed chain is split into k open valid 2-chains $\{\{a'_i, a'_{i+k}\} \mid 1 \leq i \leq k\}$ and one open valid 1-chain $\{a'_0\}$. At Chain Merging Phase, the two open chains $\{a_0\}$ and $\{a'_0\}$ are merged into a closed valid chain $\{a_0, a'_0\}$. So totally $n + k = 3k + 1$ chains are obtained, among which $3k$ are open (see figure 4(b)). The total ADM cost of all these chains is

$$|A| + 3k = 2(2k + 1) + 3k = 7k + 2 = \frac{7}{4} \cdot opt - \frac{3}{2}.$$

Thus, the approximation ratio of *ET-TS* is at least $\frac{7}{4}$.

From Lemma 3 and Example 9, we have the following theorem.

Theorem 10. *The approximation ratio of ET-TS is exactly* 1.75.

Example 9 suggests that it might improve the performance by taking out all complementary pairs of arcs before applying the algorithm ET-TS, as done in the algorithm PCM. However, the next example shows that after this preprocessing the approximation ratio is still at least 1.5.

Example 11. Let $n = 2(2k + 1)$ for some $k > 1$, and A consist of the following $3(2k + 1)$ arcs

$$\left\{a_i = (2i, 2(i + k + 1)) \mid 0 \leq i < \frac{n}{2}\right\} \cup \left\{a'_i = (2(i + k + 1), 2i - 1) \mid 0 \leq i < \frac{n}{2}\right\}$$
$$\cup \left\{a''_i = (2i - 1, 2i) \mid 0 \leq i < \frac{n}{2}\right\},$$

as illustrated in figure 5(a). Note that for any $0 \leq i < n$, the three arcs a_i, a'_i, a''_i form a closed valid chain. So $opt = |A| = 3(2k + 1)$. On the other hand, *ET-TS* may produce at Eulerian Tour Phase an Eulerian tour

$$a_0, a_{k+1}, a_1, a_{k+2}, a_2, \ldots, a_k, a_{2k}, a'_k, a''_k, a'_{2k}, a''_{2k}, a'_{k-1}, a''_{k-1},$$
$$a'_{2k-1}, a''_{2k-1}, a'_{k-2}, a''_{k-2}, \ldots, a'_0, a''_0.$$

This trail is then broken into two circuits at Trail Decomposing Phase:

$$a_0, a_{k+1}, a_1, a_{k+2}, a_2, \ldots, a_k, a_{2k};$$

and

$$a'_k, a''_k, a'_{2k}, a''_{2k}, a'_{k-1}, a''_{k-1}, a'_{2k-1}, a''_{2k-1}, a'_{k-2}, a''_{k-2}, \ldots, a'_0, a''_0.$$

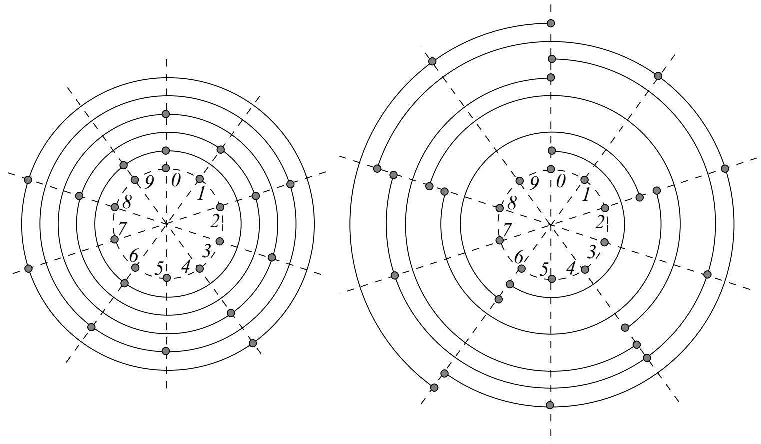

Figure 5. An instance used in Example 11: (a) the optimal solution and (b) the solution produced by the algorithm ET-TS with preprocessing.

Both circuits leads to an invalid closed chain. At Chain Split Phase, the first invalid closed chain is split into $\frac{n}{2} = 2k + 1$ open valid 1-chains $\{\{a_i\} \mid 0 \leq i < \frac{n}{2}\}$; the second invalid closed chain is split into k open valid 4-chains $\{\{a'_i, a''_i, a'_{i+k}, a''_{i+k}\} \mid 1 \leq i \leq k\}$ and one open valid 2-chain $\{a'_0, a''_0\}$. For an example, see figure 5(b). At Chain Merging Phase, the two open chains $\{a_0\}$ and $\{a'_0, a''_0\}$ are merged into a closed valid chain $\{a_0, a'_0, a''_0\}$. So totally $3k + 1$ chains are obtained, among which $3k$ are open. The total *ADM* cost of all these chains is

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

Thus, the approximation ratio of ET-TS is at least $\frac{3}{2}$.

However, even after this preprocessing, we have not been able to prove that the approximation ratio is less than 1.75. The difficulty lies in the fact that two consecutive arcs in a chain generated by Trail Decomposing Phase in ET-TS may intersect with each other. This implies that the size of an invalid open chain can be as low as two and an invalid closed chain can be as low as three. To avoid this situation, we propose below a new phase which would replace the Eulerian Tour Phase in ET-TS and can guarantee that any two consecutive arcs in a chain generated by Trail Decomposing Phase in ET-TS do not intersect with each other.

Since we can take all complementary pairs of arcs out of A, we assume that no pairs of arcs in A are complementary. We define a weighted directed graph $H(A)$ over A as follows. The vertex set of $H(A)$ is A. For any pair of non-overlapping arcs a_1 and a_2 in A, add one link from a_1 to a_2 and one link from a_2 to a_1. If a_1 and a_2 do not share any endpoints, the weights of both links are set to two. If $h(a_1) = t(a_2)$, the weight of the link from a_1 to a_2 is set to one and the weight of the link from a_2 to a_1 is set to two. If $h(a_2) = t(a_1)$, the weight of the link from a_2 to a_1 is set to one and the weight of the link from a_1 to a_2 is set to two. In addition, there is one loop link with weight two at each arc. Note that $H(A)$ has a circuit cover. Furthermore, any valid chain generation induces naturally a circuit cover of $H(A)$ whose weight is equal to the cost of the valid chain generation. Thus the weight of a minimum-weighted circuit cover of $H(A)$ is a lower bound of the minimum ADM cost.

Using these definitions, we propose the following algorithm, referred to as MCC-TS, for valid chain generation. This algorithm replaces Eulerian Tour Phase in ET-TS by the following MCC Phase:

- Construct the graph $H(A)$, and find a minimum-weighted circuit cover of $H(A)$.
- Remove all links of weight two from the minimum-weighted circuit cover and obtain a collection of paths and circuits. Each path (circuit respectively) induces an trail (closed trail respectively) along the ring of $G(A)$.

One key property of is that in any chain generated by the Trail Decomposition Phase, any two consecutive arcs do not intersect. This property implies that the size of any invalid open chain is at least three, and the size of any invalid closed chain is at least five. This observation leads to the following lemma.

Lemma 12. *The approximation ratio of MCC-TS is at most* 1.6.

Proof: Let ω denote the weight of a minimum-weighted circuit cover of $H(A)$. Then the total cost of the chains generated by Trail Decomposition Phase is exactly ω. The splitting of invalid chains into valid chains at Chain Split Phase will increase the cost. Let P be any invalid chain, and i be the number of valid chains split from P. Then the splitting of P into i valid chains increases the cost by $i - 1$ if P is open, or i if P is closed. From the construction of the graph $H(A)$, the sizes of all valid chains obtained from P, except the last one, are at least two. Thus,

$$i \leq \left\lceil \frac{|P|}{2} \right\rceil \leq \frac{|P|+1}{2}.$$

So P contributes to an additional cost of at most $\frac{|P|-1}{2}$ if P is open, or at most $\frac{|P|+1}{2}$ if P is closed. Let j be the number of invalid closed chains generated by Trail Decomposition Phase. Then the cost of the valid chains generated by Chain Split Phase is at most $\omega + \frac{|A|+j}{2}$. As the size of any invalid closed chain is at least five, $j \leq \frac{|A|}{5}$. So the cost of the valid chains produced by our algorithm is at most

$$\omega + \frac{|A|+j}{2} \leq \omega + \frac{|A| + \frac{|A|}{5}}{2} = \omega + \frac{3}{5}|A| \leq opt + \frac{3}{5} \cdot opt = \frac{8}{5} \cdot opt,$$

as both ω and $|A|$ are lower bounds of *opt*. Thus the lemma is true. □

In the next, we will derive a lower bound on the approximation ratio of MCC-TS. We notice that the execution of MCC-TS on the instance given in Example 11 will always lead to an optimal solution. Thus Example 11 can not provide us a good lower bound. So instead we developed the following example.

Example 13. Let $n = 2(2k+1)$ for some $k > 1$, and A consist of the following $\frac{3n}{2}$ arcs

$$\left\{a_i = \left(2i, 2i + \frac{n}{2}\right) \middle| 0 \leq i < \frac{n}{2}\right\} \cup \left\{a_i' = \left(2i + \frac{n}{2}, 2i + \frac{n}{2} + 1\right) \middle| 0 \leq i < \frac{n}{2}\right\}$$
$$\cup \left\{a_i'' = \left(2i + \frac{n}{2} + 1, 2i\right) \middle| 0 \leq i < \frac{n}{2}\right\},$$

as illustrated in figure 6(a). Note that for any $0 \leq i < n$, the three arcs a_i, a_i', a_i'' form a closed valid chain. So $opt = |A| = \frac{3n}{2}$. On the other hand, MCC-TS may produce at MCC Phase two circuits

$$a_k'', a_{2k}'', a_{k-1}'', a_{2k-1}'', \ldots, a_1'', a_{k+1}'', a_0'';$$

and

$$a_0, a_0', a_{k+1}, a_{k+1}', a_1, a_1', a_{k+2}, a_{k+2}', \ldots, a_{k-1}, a_{k-1}', a_{2k}, a_{2k}', a_k, a_k'.$$

Each circuit induces an invalid closed chain. At Chain Split Phase, the first invalid closed chain is split into k open valid 2-chains $\{\{a_i'', a_{i+k}''\} \mid 1 \leq i \leq k\}$ and one open valid 1-chain $\{a_0''\}$. Since the $2k+1$ arcs $\{a_i \mid 0 \leq i \leq 2k\}$ are pairwise intersecting, at least $2k+1$ open

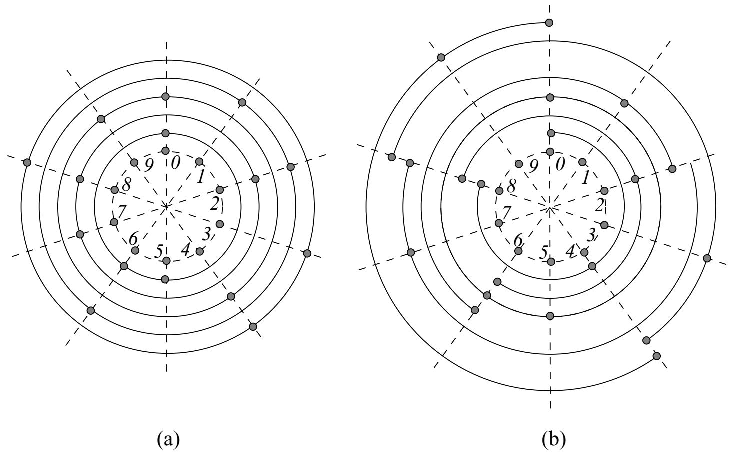

Figure 6. An instance used in Example 13: (a) the optimal solution and (b) the solution produced by the algorithm ET-TS with preprocessing.

valid chains must be used by the second closed invalid chain. One solution is the following: $\{\{a_i, a'_i\} \mid 0 \leq i \leq 2k\}$.

At Chain Merging Phase, the two open chains $\{a_0\}$ and $\{a'_0, a''_0\}$ are merged into a closed valid chain $\{a_0, a'_0, a''_0\}$. So totally $3k + 1$ chains are obtained, among which $3k$ are open (see figure 6(b)). The total ADM cost of all these chains is

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

Thus, the approximation ratio of MCC-TS is at least $\frac{3}{2}$.

From Lemma 12 and Example 13, we have the following theorem.

Theorem 14. *The approximation ratio of MCC-TS is between* 1.5 *and* 1.6.

Both Example 11 and Example 13 suggest the preprocessing by repeatedly taking out valid closed chains of sizes at most three. However, it is no longer guaranteed that there is an optimal solution that contains a specified valid closed chain. This can be illustrated by the Example 9 from Wan et al. (2000), which we include for the sake of completeness:

Example 15 (Wan et al., 2000). Let $n = 6$, and $A = A_1 \cup A_2 \cup A_3$ where

$$A_1 = \{(0, 2), (2, 5), (5, 0)\},$$
$$A_2 = \{(0, 3), (3, 4), (4, 0)\},$$
$$A_3 = \{(1, 2), (2, 4), (4, 1)\},$$

as illustrated in figure 7(a). 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

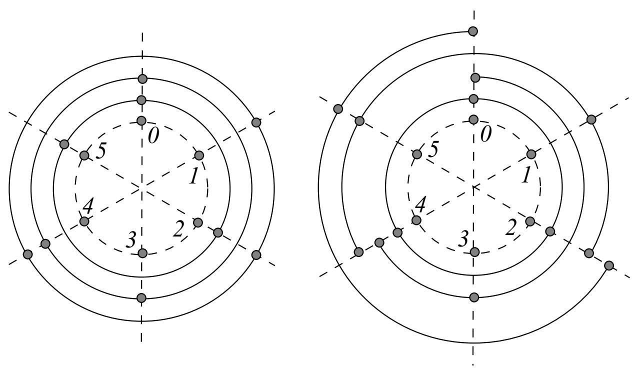

Figure 7. The instance in Example 15: (a) the optimal solution and (b) a solution produced after the preprocessing.

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 (see figure 7(b)):

$$\{(2, 5), (5, 0)\}, \{(4, 1), (1, 2)\}, \{(0, 3), (3, 4)\}.$$

Thus the total ADM cost of all these valid chains is $12 = \frac{4}{3} \cdot opt$.

Example 15 indicates that for any algorithm which first repeatedly taking out closed valid chains until no more is left, its approximation ratio can not be less than $\frac{4}{3}$. Note that Examples 11 and 13 can be generalized to show that even if an instance does not contain any closed chain of sizes at most l, the approximation ratios of ET-TS and MCC-TS is at least $1 + \frac{3}{2(l+1)}$. In particular, if $l = 3$, the lower bound on the approximation ratio is 1.375. It would be interesting to explore the effect of preprocessing by repeatedly taking out closed valid chains on the approximation ratios of both ET-TS and MCC-TS.

4.3. Iterative matching

We start by analyzing a simple algorithm referred to as Iterative Matching (IM) proposed in Liu et al. (2000). This algorithm maintains a set of valid chains of arcs $\mathcal{P}$ throughout its execution. Initially $\mathcal{P}$ consists of 1-chains each of which is an arc in A. 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.

It is obvious that the algorithm IM has polynomial run-time. In the next we show that its approximation ratio is at most $\frac{5}{3}$.

Lemma 16. *The approximation ratio of IM is at most* $\frac{5}{3}$.

Proof: From any valid chain P in OPT, a matching of cardinality $\lfloor \frac{|P|}{2} \rfloor$ 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 $\frac{|A|-i-j}{2}$. Thus the cardinality of any maximum matching obtained in the first iteration is at least $\frac{|A|-i-j}{2}$, and consequently after the first iteration, the total number of valid chains is at most

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

Note that any odd closed chain must contain at least three arcs. Thus $j \leq \frac{|A|-i}{3}$. So after the first iteration, the total ADM cost is at most

$$|A| + \frac{|A|+i+j}{2} = \frac{3|A|}{2} + \frac{i}{2} + \frac{j}{2} \leq \frac{3|A|}{2} + \frac{i}{2} + \frac{|A|-i}{6} = \frac{5|A|+i}{3}$$
$$\leq \frac{5}{3}(|A|+i) \leq \frac{5}{3} \cdot opt.$$

Therefore, the approximation ratio of IM is at most $\frac{5}{3}$. □

For the sake of completeness, we include Example 5 from Wan et al. (2000), which shows that the approximation ratio of IM is at least $\frac{3}{2}$:

Example 17 (Wan et al., 2000). Let $n = 5$ and $A = A_1 \cup A_2$ where

$$A_1 = \{(0, 1), (1, 3), (3, 0)\},$$
$$A_2 = \{(0, 2), (2, 4), (4, 0)\},$$

as illustrated in figure 8(a). Note that the three arcs in A_1 form a closed valid chain, so do the three arcs in A_2. Thus $opt = |A| = 6$. On the other hand, the algorithm IM may output the following open valid chains (see figure 7(b)):

$$\{(4, 0), (0, 1)\}, \{(0, 2), (2, 4)\}, \{(1, 3), (3, 0)\}.$$

The cost of these three chains is $9 = \frac{3}{2} \cdot opt$. This example can be scaled to the rings with size multiple of five. Thus, the approximation ratio of IM is at least $\frac{3}{2}$.

From Lemma 16 and Example 17, we have the following theorem.

Theorem 18. *The approximation ratio of the algorithm IM is between $\frac{3}{2}$ and $\frac{5}{3}$.*

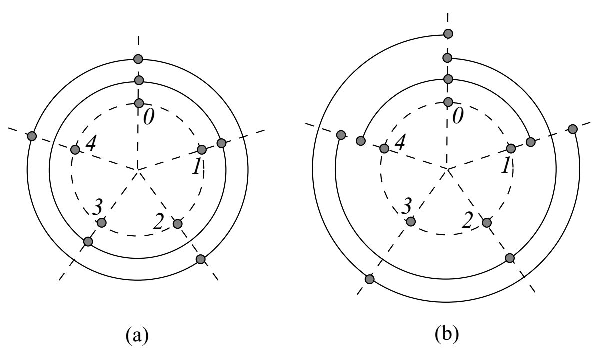

Figure 8. The instance in Example 17: (a) the optimal solution and (b) a solution produced produced by the algorithm IM.

From the proof of Lemma 16, the closed chains play a critical role in bounding the total ADM cost. In particular, if *OPT* does not contain any closed chains, the total ADM cost after the first iteration is at most $\frac{3}{2} \cdot opt$. Based on this observation, 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*: apply the algorithm IM to the remaining arcs.

Without a proof of approximation ratio, the algorithm is described in Wan et al. (2000), where it was called Closed Segment First.

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. The Preprocessing Phase runs as follows:

- Find the link e which has the minimum link load, and let A_e denote the number of arcs that contain the link e. Set $i \leftarrow 1$.
- While A_e is nonempty: pick any $a \in A_e$, and set $A_e \leftarrow A_e \backslash \{a\}$; if A has a valid closed chain P containing a, set $P_i \leftarrow P$, $A \leftarrow A \backslash P$, 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 19. *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. Following the same argument in the proof of Lemma 16, 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 valid red chains is at most

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

So after the first maximum matching in the Matching Phase, the total ADM cost is at most

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

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

The instance in Example 15 can be used to derive a lower bound of $\frac{4}{3}$ on the approximation ratio of PIM. Thus we have the following theorem.

Theorem 20. *The approximation ratio of PIM is between $\frac{4}{3}$ and $\frac{3}{2}$.*

5. Valid chain generation from chords

A straightforward approach to valid chain generation from chords consists of two separate stages. In the first stage, find a routing of the input chords with the low load or wavelength requirement. In the second stage, apply heuristics developed in the previous section to the obtained arcs. Given a set of chords, its minimum-load routing can be found in polynomial time (Schrijver et al., 1998; Wilfong and Winkler, 1998). However, it is NP-hard to find a minimum-wavelength routing (Raghavan and Upfal, 1994), and two is the best approximation ratio of heuristics for minimum-wavelength routing known in the literature. Among them, the Edge-Avoidance Routing and the Shortest-Path Routing (Carpenter et al., 1997) are the most well-known ones. The Edge-Avoidance Routing routes all chords to avoid an arbitrary link of the ring. The shortest-path routing always routes a chord that traverses the fewest links. Despite of their simplicity, both of them guarantee that the link load is at most twice the minimum, and at the same time they can lead to a wavelength assignment with at most twice the minimum the number of wavelengths.

We first show that both minimum-load routing and minimum-wavelength routing may have the worst performance in terms of the ADM cost.

Example 21. Let $n = 2k$ for some $k \geq 2$, and C consists of two copies of $(2i, 2i + 1)$ for all $0 \leq i < k$. In order to achieve minimum load (or wavelength), each chord must be routed in the short-way, as shown in figure 9(a). However, with this routing each chord must form a 1-chain, which would lead the total ADM cost of $2|C|$. On the other hand, one can form a valid closed 2-chain from each pair of identical chords, as shown in figure 9(b). So the minimum ADM cost is $|C|$.

Now we examine the performance of the Edge-Avoidance Routing and the Shortest-Path Routing. However, Example 21 shows that both of them can cost as high as twice the

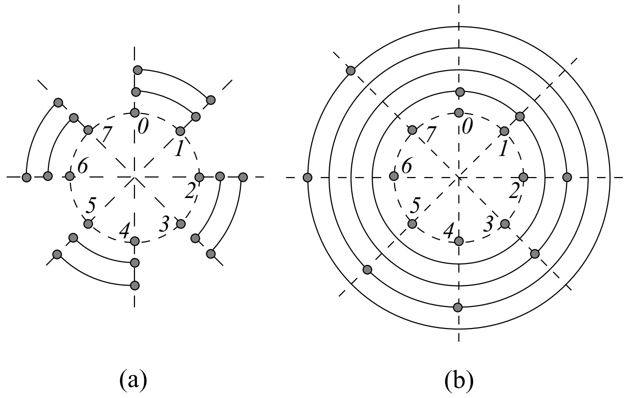

Figure 9. An instance used in Example 21: (a) the minimum load (or wavelength routing), and (b) the routing with minimum ADM cost.

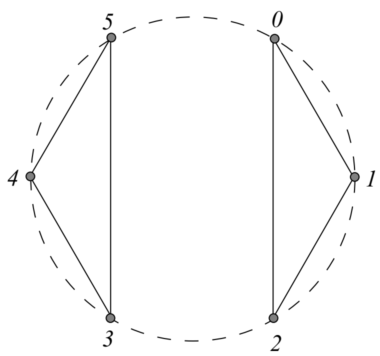

Figure 10. An instance used in Example 22 with $n = 6$.

minimum. Even when there are no identical chords, they can still cost as high as $\frac{5}{3}$ times the minimum as illustrated by the following example.

Example 22. Let $n = 3k$ for some $k \geq 2$, and $C = \bigcup_{i=0}^{k-1} C_i$ where

$$C_i = \{(3i, 3i + 1), (3i + 1, 3i + 2), (3i, 3i + 2)\},$$

as illustrated in figure 10. Note that the three chords in each C_i form a closed valid chain. Thus $opt = |C| = 3k$. On the other hand, it is easy to verify that both edge-avoidance routing and shortest-path routing would lead to at least $2k$ open chains. The cost of these three chains is $3k + 2k = 5k = \frac{5}{3} \cdot opt$.

The previous two examples imply that heuristics based on separated routing and valid chain generation from arcs can not provide good performance guarantees. In the next, we

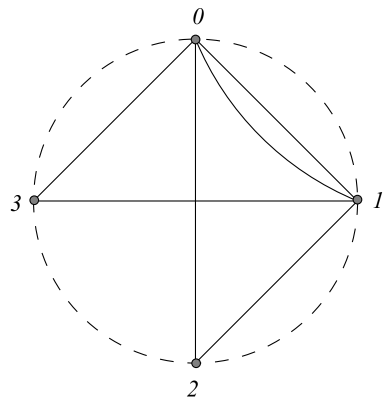

Figure 11. Pairing up two identical chords into a closed chain may not give rise to optimal solution.

propose heuristics with integrated routing and chain generation. Before we describe the heuristics, we would like to point out a subtle difference between valid chain generation from arcs and valid chain generation from chords. While there is an optimal valid chain generation from arcs which pair up a given pair of complementary arcs, there may be no optimal valid chain generation from chords which pair up a given pair of identical chords. In fact, the following example indicates that one can not expect the approximation ratio of any algorithm which first take out pairs of identical chords to be less than $\frac{4}{3}$.

Example 23. Let $n = 4$ and $C = C_1 \cup C_2$ where

$$C_1 = \{(0, 1), (1, 2), (2, 0)\},$$
$$C_2 = \{(0, 1), (1, 3), (3, 0)\},$$

as illustrated in figure 11. Note that the three chords in C_1 form a closed valid chain, so do the three chords in C_2. Thus $opt = |C| = 6$. On the other hand, if we take the two identical chords between 0 and 1 to form a closed 2-chain, then the remaining four chords require at least two open chains. The cost of these three chains is $6 + 2 = 8 = \frac{4}{3} \cdot opt$.

5.1. Eulerian tour decomposition

In this section we propose the following *Eulerian Tour Decomposition (ETD)* heuristic, which is a counter-part of the ET-TS heuristic described in Subsection 4.2. 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 whenever 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 of 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 an 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 an 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 can not 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.

While the approximation ratio of ET-TS is 1.75 according to Theorem 10, the next lemma shows that the approximation ratio of ETD is at most 1.5.

Lemma 24. *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 $\lceil \frac{|P|}{2} \rceil$ chains are split from P. So totally at most $\frac{|C|+d(C)}{2}$ chains are generated at Step 3, and costs at most

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

This implies that the approximation ratio of ETD is at most 1.5.

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 $1 + \lceil \frac{|C|-k}{2} \rceil$ chains are produced. So the total cost is at most

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

If any three chords in C can not 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, $opt > |C|$. 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 either case the lemma is true. □

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

Example 25. Let $n = 2(2k + 1)$ for some $k > 1$, and C consist of the following $\frac{3n}{2}$ chords

$$\left\{ c_i = \left(2i, 2i + \frac{n}{2}\right) \middle| 0 \leq i < \frac{n}{2} \right\} \cup \left\{ c'_i = \left(2i + \frac{n}{2}, 2i + \frac{n}{2} + 1\right) \middle| 0 \leq i < \frac{n}{2} \right\}$$
$$\cup \left\{ c''_i = \left(2i + \frac{n}{2} + 1, 2i\right) \middle| 0 \leq i < \frac{n}{2} \right\},$$

as illustrated in figure 12. This instance can be obtained from the instance in Example 13 by replacing each arc with a chord between its two endpoints. Note that for any $0 \leq i < n$,

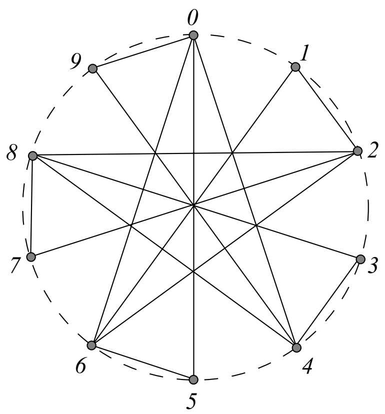

Figure 12. An instance used in Example 25 with $n = 10$.

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 may unluckily produce the following Eulerian tour

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

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 totally $3k$ 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 24 and Example 25, we have the following theorem.

Theorem 26. *The approximation ratio of ETD is exactly* 1.5.

5.2. *Iterative matching*

In this subsection, we modify the heuristics *Iterative Matching (IM)* and *Preprocessed Iterative Matching (PIM)* described in Subsection 4.3 such that they are applicable to chords. The IM for chords also maintains a set of valid chains of chords $\mathcal{P}$ throughout its execution. Initially $\mathcal{P}$ consists of 1-chains of single chord in C. 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 chords) in M into a larger chain (of chords). When $F(\mathcal{P})$ has empty edge set, $\mathcal{P}$ is output as the solution.

By using the same argument used in Lemma 16, one can prove the approximation ratio of IM for chords is at most $\frac{5}{3}$. On the other hand, replacing each arc in Example 17 by a chord between its two endpoints leads to an instance (see figure 13) for which the IM for chords may produce a solution of $\frac{3}{2}$ times the optimum. So we have the following theorem.

Theorem 27. *The approximation ratio of the algorithm IM for chords is between $\frac{3}{2}$ and $\frac{5}{3}$.*

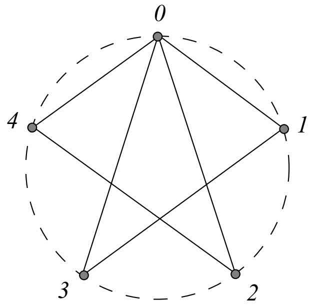

Figure 13. The instance obtained by replacing each arc in Example 17 with a chord between the same endpoints.

The Preprocessed Iterative Matching (PIM) for chords is adapted from the PIM algorithm for arcs. The algorithm is described, without proofs, in Wan et al. (2000), where it is called chord-version Closed Chain First. The chord-version PIM also runs in two phases:

1. *Preprocessing phase*: repeatedly take valid closed chains out of the remaining chords until no more closed chain can be obtained from the remaining chords.
2. *Matching phase*: apply the algorithm IM to the remaining chords.

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 19, we can prove that the approximation ratio of PIM for chords is at most 1.5. Replacing each arc in Example 15 with a chord between the same endpoints leads to an instance (see figure 14) for which the PIM for chords may produce a solution of $\frac{4}{3}$ times the optimum. So we have the following theorem.

Theorem 28. *The approximation ratio of PIM for chords is between* $\frac{4}{3}$ *and* $\frac{3}{2}$.

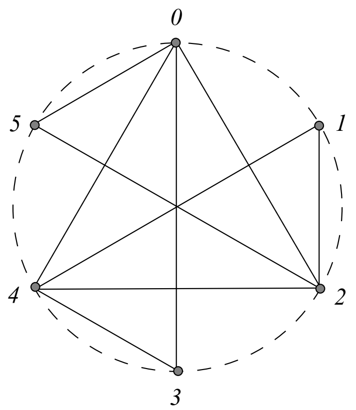

Figure 14. The instance obtained by replacing each arc in Example 15 with a chord between the same endpoints.

6. Conclusion

This paper addresses both arc-version and chord version of the minimum ADM problem. Both versions can be reduced to a valid chain generation problem. A number of algorithms have been studied and their performances are summarized in Tables 1 and 2.

From Tables 1 and 2, there is a gap between the lower bound and the upper bound for most heuristics. It will be interesting to reduce the gap, and ideally to get the exact values of the approximation ratios. For the chord-version ADM problem, we have an approximation

Table 1. Bounds on the approximation ratios of heuristics studied in this paper for arc-version minimum ADM cost problem.

Heuristic	Lower bound	Upper bound
CM	1.75	1.75
PCM	1.5	$\frac{5}{3}$
ET-TS	1.75	1.75
MCC-TS	1.5	1.6
IM	1.5	$\frac{5}{3}$
PIM	$\frac{4}{3}$	1.5

Table 2. Bounds on the approximation ratios of heuristics studied in this paper for chord-version minimum ADM cost problem.

Heuristic	Lower bound	Upper bound
ETD	1.5	1.5
IM	1.5	$\frac{5}{3}$
PIM	$\frac{4}{3}$	1.5

algorithm (a more specific PIM) with performance ratio slightly better than 1.5. The proof is very complicated and its ideas do not seem general enough to be published. For arcs, it is still an open problem whether or not there exists polynomial-time algorithms with approximation ratio less than 1.5.

References

1. T. Carpenter, S. Cosares, and I. 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," in *Proc. IEEE INFOCOM'98*, San Francisco, CA, 1998, vol. 1, pp. 94–101.
3. O. Gerstel, P. Lin, and G. Sasaki, "Combined WDM and SONET Network Design," in *Proc. IEEE INFOCOM'99*, New York, NY, 1999, 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*, C.A. Siller and M. Shafi (Eds.), IEEE Press, New York, 1996, pp. 131–139.
5. 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*, Tel-Aviv, Israel, 2000, 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*, 1994, pp. 134–143.
7. A. Schrijver, P.D. Seymour, and P. Winkler, "The ring loading problem," *SIAM J. Disc. Math.*, vol. 11, no. 1, pp. 1–14, 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, and 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 and P. Winkler, "Ring routing and wavelength translation," in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998, pp. 333–341.