A 10/7 + ϵ Approximation for Minimizing the Number of ADMs in SONET Rings

Mordechai Shalom and Shmuel Zaks
Faculty of Computer Science, Technion
32000 Haifa, Israel
(cmshalom,zaks)@cs.technion.ac.il

Abstract

SONET ADMs are dominant cost factors in WDM/SONET rings. Whereas most previous papers on the topic concentrated on the number of wavelengths assigned to a given set of lightpaths, more recent papers argue that the number of ADMs is a more realistic cost measure. Some of these works discuss various heuristic algorithms for this problem, and the best known result is a 3/2 approximation in [1]. Through the study of the relation between this problem and the problem of finding maximum disjoint rings in a given set of lightpaths we manage to shed more light onto this problem and to develop a $10/7 + \epsilon$ *approximation for it.*

1. Introduction

1.1. Background

A single fiber-optic cable offers a bandwidth that can potentially carry information at the rate of several terabits per second, much faster than any electronic device can handle. In order to utilize the potential of optical fiber, wavelength-division multiplexing (WDM) is used. The bandwidth is partitioned into a number of channels at different wavelengths. A single channel supplies bandwidth in the range of gigabits per second, and several signals can be transmitted through a fiber link simultaneously on different channels. The number of channels (wavelengths) available in WDM systems is limited by the chosen technology. One of the important parameters affected by the technology is the network cost. Tunable lasers or arrays of fixed-wavelength lasers are used to generate the laser beams that are to be transmitted on the optical channels. Add/drop multiplexers (ADMs) are employed at the network nodes to insert lightwaves into the fiber and extract them. Fixed-wavelength or tunable filters and receivers are used at the receiving side of a transmission. The electronic equipment is not required to operate faster than a single optical channel, thus, WDM allows existing electronic equipment to fully use the enormous potential of optical fiber.

WDM ring networks are deployed by a growing number of telecom carriers. The problem of minimizing the number of wavelengths has been extensively studied. Variants of this problem such as to maximize the number of lightpaths given a limited number of wavelengths (the $MAXPC$ problem) or minimize the blocking probability of a lightpath were also studied.

Recent studies (e.g.,[5],[9]) argue that a more realistic cost measure is the number of ADMs used by the network. Moreover, these studies concentrate on a ring topology for various reasons. One of the commonly stated reasons is that higher level networks which make use of the WDM network cannot necessarily support arbitrary topologies. The most widely deployed network above the WDM layer is the SONET/SDH self-healing rings. These networks have to be configured in rings for protection purposes.

We concentrate on the problem of minimizing the additional overhead resulting from the need of these lightpaths to be configured as rings. This can be split into two problems:

- Assign a route to a lightpath; namely, choose one of two possible directions on the ring such that the maximum number of lightpaths intersecting on an edge is minimal. This is called the ring loading problem. In [11] an optimal solution for the problem in directed rings is given. As for undirected rings, a polynomial time approximation scheme is given in [8].

- Given the routing above, assign wavelengths to the paths such that the number of ADMs used by the system is minimized. We focus on this problem. This problem is studied in [3] for general topology, although their motivation is slightly different.

1.2. Problem Definition

Given a WDM ring network $G = (V, E)$ such that $V = \{0, 1, ..., n-1\}$ comprising optical nodes and a set of full-duplex lightpaths $L = \{l_1, l_2, ...\}$ such that for all j, $l_j = (s_j, e_j)$ and $s_j, e_j \in V$, the wavelength assignment problem assigns a wavelength to each lightpath l_i. The forward part of the duplex lightpath (s_i, e_i) traverses from s_i to e_i and the reverse part traverses from e_i to s_i. Call s_i the starting node and e_i the ending node. $s(l_i) \stackrel{def}{=} s_i$ and $e(l_i) \stackrel{def}{=} e_i$.

Without loss of generality we assume that each lightpath l_i is routed clockwise on the ring from s_i to e_i. Under this assumption the following definitions are valid.

Definition 1.1 *$l, l' \in L$ are conflicting or overlapping if l and l' have an edge in common. This is denoted as $l \asymp l'$.*

Definition 1.2 *$len(l)$ is the length of the lightpath l, namely $(e(l) - s(l)) mod n$.*

Definition 1.3 *For any edge $e \in E$, its load $l(e)$ is the number of lightpaths containing it. $L_{min} \stackrel{def}{=} min_{e \in E} l(e)$.*

Definition 1.4 *A proper coloring (or wavelength assignment) of L is a function $W : L \mapsto \mathbb{N}$, such that $W(l) \neq W(l')$ whenever $l \asymp l'$.*

Another assumption is that L is given upfront, in other words we study the static (off-line) WLA problem. This assumption is reasonable for example in the case of very high-speed pipes in the telecom environment.

Electrical TDM line-terminals terminate the lightpaths. We assume this nodes are SONET/SDH add/drop multiplexers (ADMs). Each lightpath l uses two ADMs, one at $s(l)$ and another at $e(l)$. Although in $s(l)$ (resp. $e(l)$) only the downstream (resp. upstream) ADM function is needed, full ADMs will be installed on both nodes in order to complete the protection path around the ring. The full configuration would result in a number of SONET rings all circumventing a single optical ring. It follows that if two adjacent lightpaths are assigned the same wavelength, then they are used by the same SONET ring and the ADM in the common node can be shared by them. This would save the cost of one ADM. With this in mind, we define our goal, as follows:

For each node v of the ring, $cost(v)$ is defined as the number of different colors assigned to all lightpaths starting or ending at v, namely:

$$\forall v \in V, cost(v) \stackrel{def}{=} |\{W(l) | s(l) = v \vee e(l) = v\}| .$$

The goal is to minimize the total cost function:

$$cost(V) = \sum_{v \in V} cost(v).$$

A more convenient statement of the problem is discussed in Section 2.2.

1.3. Previous Work

A number of previous works [5, 6, 9, 1] studied the minimum ADM problem in which each traffic stream has a predetermined routing. This is called also the *arc version* of the problem. The problem is proved to be in $\mathcal{NP} - hard$ [9]. Several heuristics are proposed in [5, 9, 10], most of which have approximation ratio at least $3/2$. Some of the heuristics are proved to have approximation ratio at most $\frac{3+e}{1+e} = 1.537...$. Note that an approximation ratio of 2 is trivial: optimum is at least the number of lightpaths and any solution will use at most twice the number of lightpaths (one at each endpoint).

The Preprocessed Iterative Matching heuristic proposed in [1] solves the $arc\ version$ of the minimum ADM problem and is shown to have an approximation ratio of $3/2$ which is the best known result. In the same paper the *chord version* of the problem is also addressed. In this version, the routing of the lightpaths is not part of the input and they are to be determined by the solution as well as their wavelength assignments. An algorithm with approximation ratio of $3/2$ is proposed in [1] for this problem.

A $10/7$ approximation algorithm is presented recently in [4].

1.4. Our Contribution

Our main result is a $10/7 + \epsilon$ approximation algorithm for the minimum ADM problem. We start by presenting an algorithm which is a modified version of the *Assign First* algorithm presented in [5] and prove its approximation ratio to be between $3/2$ and $11/7$.

We then investigate the relationship between the approximability of the maximum disjoint paths problem and the approximability of the $arc\ version$ of the minimum ADM problem. Using good approximation algorithms for this problem we manage to improve our first algorithm and obtain a second algorithm with approximation ratio less than 1.48. Finally by using the same technique on the algorithm presented in [3] we obtain an algorithm with approximation ratio at most $10/7 + \epsilon$.

The rest of the paper is organized as follows: Section 2.1 gives some preliminary definitions and a formal statement of the problem. In section 3 we present our first algorithm, and in section 4 we analyze its performance. Section 5 presents and analyzes the improved algorithms. In section 6 we present simulation results and discuss further research directions.

2. Preliminaries

2.1. Notation and Definitions

Given a solution of the problem, we make the following definitions.

Definition 2.1 *A chain c is a maximal sequence of distinct consecutive lightpaths $(l_1, l_2, ..., l_k)$ assigned the same wavelength by the solution, s.t. $\forall i > 1, s(l_i) = e(l_{i-1})$ and $e(l_k) \neq s(l_1)$. $s(l_1)$ (resp. $e(l_1)$) will be called the start of the chain and will be denoted as $s(c)$ (resp. $e(c)$). $len(c) \stackrel{def}{=} \sum_{i=1}^{k} len(l_i)$, or in other words $(e(c) - s(c)) mod n$.*

Definition 2.2 *A cycle c is a sequence of distinct consecutive lightpaths $(l_1, l_2, ..., l_k)$ assigned the same wavelength in the solution, s.t. $\forall i > 1, s(l_i) = e(l_{i-1})$ and $e(l_k) = s(l_1)$.*

As the elements of the chains and cycles are distinct, we will refer to these sequences sets too.

Definition 2.3 *The wavelength of a chain/cycle c is the unique wavelength assigned to its lightpaths: $W(c) \stackrel{def}{=} W(l_1)$.*

Definition 2.4 *The (unique) chain/cycle that contains a lightpath l is denoted by $c(l)$.*

For our algorithms we will use the following two definitions, following [5] and [2], respectively.

Definition 2.5 *$\forall i \in V$:*

$$\tau_i \stackrel{def}{=} \{l \in L | s(l) = i\}$$

$$\sigma_i \stackrel{def}{=} \{l \in L | e(l) = i\}$$

$$X_i \stackrel{def}{=} \{l \in L | i \text{ is a node of } l\} \setminus (\tau_i \cup \sigma_i)$$

$$Y_i \stackrel{def}{=} X_i \cup \tau_i$$

Definition 2.6 *The node graph of a node $i \in V$ is the bipartite graph $G_i = (\tau_i, \sigma_i, E_i)$ where $(l, l') \in E_i \subseteq \tau_i \times \sigma_i$ whenever $l \not\preceq l'$.*

2.2. Restatement of The Problem

Given a wavelength w let $cost(w)$ be the number of nodes v that w contributes 1 to $cost(v)$. Clearly $cost(V) = \sum_{w=1}^{\infty} cost(w)$. Now consider all the chains and cycles c such that $W(c) = w$. If there is no lightpath l such that $W(l) = w$ and $v \in \{s(l), e(l)\}$ then w contributes 0 to $cost(v)$, thus v contributes 0 to $cost(w)$. All the other nodes contribute 1 to $cost(w)$. Therefore $cost(w)$ is the number of nodes of the chains (and cycles) colored w. The number of these nodes is exactly the number of lightpaths in all those chains/cycles, plus the number of the chains. This is because a cycle c has $|c|$ nodes and a chain c' has $|c'| + 1$ nodes. Summing over all w we conclude that $cost(V) = |L| + The\ number\ of\ chains$.

Since our goal is to minimize the number of ADMs and not the number of wavelengths, we slightly change the statement of the problem. In accordance with the above discussion we are not concerned with the wavelength assignment itself but only on the chains and cycles induced by the solution. Thus an optimal solution of the minimum ADM problem is a partitioning of L into chains and cycles such that the number of chains is minimum.

3. Algorithm PAF

In this section we present algorithm PAF (Preprocessed Assign First). This algorithm is a modification of the *Assign First* algorithm in [5]. We use the notations and definitions of the previous section.

We first briefly describe the Assign First algorithm:

- The nodes of the ring are renumbered from 0 to $n-1$ such that 0 is a node minimizing some objective function (which is not relevant in our case).
- All the lightpaths in Y_0 are colored with distinct colors.
- The nodes are scanned from 1 to $n-1$. At each node i the lightpaths in τ_i are colored. This coloring is done in the following manner: The colors of the lightpaths in σ_i are preferred colors. The preferred colors are used first, if they are exhausted, other colors are used from lowest numbered first. If a color is not valid for a lightpath, the next color is tried.

Now we restate Assign First in our terminology:

- Renumber the nodes of the ring from 0 to $n-1$.
- Designate each lightpath in Y_0 as a chain by itself.
- Scan the nodes from 1 to $n-1$. At each node i first try to expand the chains c ending at i, then form new chains of the lightpaths in τ_i which have not yet joined a chain.

PAF has two major differences from Assign First:

- It has a preprocessing phase.
- The attempts to expand the chains are done by trying the maximum matching of the *node graph*.

PAF (n, L) {
 Preprocessing:
 A) Remove a maximal set of cycles of two paths from L.
 B) Remove a maximal set of cycles from L.

 Processing:

```
For each path l ∈ L do c(l) = (l)
For each node i from 1 clockwise to 0 do{
  Find a maximum matching MM_i of the node graph G_i.
  τ'_i = The unmatched nodes of τ_i.
  σ'_i = The unmatched nodes of σ_i.
  G'_i =The complete bipartite graph (τ'_i, σ'_i, τ'_i × σ'_i).
  Find a maximum matching MM'_i OF G'_i.
  τ''_i = The unmatched nodes of τ'_i.
  σ''_i = The unmatched nodes of σ'_i.
  For each edge (a,b) ∈ MM_i {
    if len(c(a)) + len(c(b)) ≤ n
      UNION (c(a), c(b))
    else
      failure
  }
  For each edge (a,b) ∈ MM'_i unmatched
  For each b in τ''_i start
  For each b in σ''_i end
 }
}
```

The words failure, unmatched, start, end written in bold in the code are events which are generated for the sake of the analysis, otherwise they do nothing. Figure 1 describes the four cases that may cause such an extension to fail. A solid line depicts a chain consisting of one lightpath and dashed line depicts a chain with at list two lightpaths.

4. Analysis

4.1. Correctness and Complexity

Once a lightpath l is added to a chain it is not added to another one. This could happen only at node $s(l)$. This can not happen twice, because of the property of the matching. Therefore the output is a partitioning of L.

During the execution of the algorithm a chain or cycle's length can not exceed n, because this is checked before every potential extension of a chain. Moreover a lightpath l is added to a chain c only at node $e(c) = s(l)$ in a manner consistent with the definition of a chain. Thus every set in the partitioning is a valid chain or cycle.

The algorithm runs in polynomial time as implied by the following discussion: The removal of 2-cycles is done in linear time in the input. To check the existence of a cycle can be done with L_{min} calls to BFS or any shortest path algorithm, therefore in polynomial time. At each node maximum bipartite matching can be found in polynomial time using any maximum flow algorithm, all other operations can be done in constant time.

4.2. Approximation Ratio

Let ALG be any deterministic algorithm solving an approximation problem. It is customary to denote by $ALG(I)$ or simply ALG the cost of the solution of algorithm ALG on instance I. Similarly $OPT(I)$ or simply OPT is the cost of an optimal solution.

Lemma 4.1 *If there exist two lightpaths l_1, l_2 forming a cycle, there is an optimal solution in which they form a cycle.*

Proof: Consider an optimal solution OPT in which l_1 and l_2 do not form a cycle. In this solution l_1 and l_2 should be in different chains or cycles c_1 and c_2. We build a new solution OPT' by taking all the chains and cycles of OPT except c_1 and c_2 together with the cycle $c'_1 = (l_1, l_2)$ and $c'_2 = c_1 \cup c_2 \setminus c'_1$. Consider three cases:

- Both c_1 and c_2 are cycles In this case c'_1 and c'_2 are cycles, thus $OPT = OPT'$.
- c_1 is a cycle, c_2 is a chain In this case c'_1 is a cycle and c'_2 is a chain, again $OPT = OPT'$.
- Both c_1 and c_2 are chains In this case c'_1 is a cycle and c'_2 forms at most two chains $OPT' \leq OPT$.

In all cases $OPT' \leq OPT$, therefore optimal.

□

It follows from the above lemma that the first step of the preprocessing phase removes cycles which are guaranteed to be in an optimal solution. In other words if we can find an optimal solution for the rest of the problem, our solution will be optimal. Let R_2 be set number of lightpaths removed in the first step of the preprocessing step. Then:

$$\begin{aligned} PAF(L) = |R_2| &+ PAF(L \setminus R_2) \\ OPT(L) = |R_2| &+ OPT(L \setminus R_2) \\ \frac{PAF(L \setminus R_2)}{OPT(L \setminus R_2)} &\geq 1 \end{aligned}$$

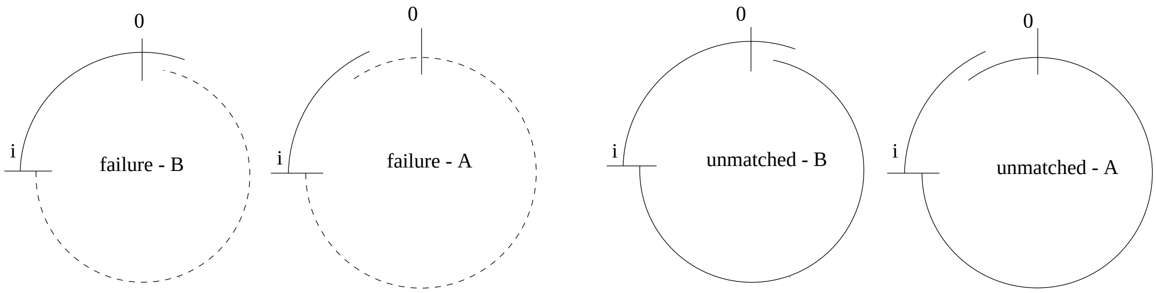
Therefore:

$$\frac{PAF(L)}{OPT(L)} = \frac{|R_2| + PAF(L \setminus R_2)}{|R_2| + OPT(L \setminus R_2)} \leq \frac{PAF(L \setminus R_2)}{OPT(L \setminus R_2)}$$

We conclude that the approximation ratio of the algorithm on an instance without 2-cycles can be only worse than a corresponding instance with 2-cycles. Without loss of generality, in the sequel we assume that no two lightpaths in the input form a cycle, or in other words, there are no 2-cycles in the instance.

During the execution of the algorithm each occurrence of an *unmatched* or *failure* event determines the end of a chain c_i and the start of a chain c_j. In this case we write:

- $c_i \prec_F c_j$
- $c_i \prec_U c_j$



depending on the event occurred.

For every start (resp. end) event we introduce the dummy chain s_i (resp. e_i) and write:

- $s_i \prec_S c_i$
- $c_i \prec_E e_i$

Observation 4.1 *A chain c occurs once at the right side of a $\prec$ relation and once at the left side of a $\prec$ relation.*

Proof: A chain c participates in a left (resp. right) side of a $\prec$ relation, as a result of an event generated at node $e(c)$ (resp. $s(c)$). It can be seen by code inspection that a chain ie either extended or is involved in exactly one event.

□

Observation 4.2 *A dummy chain s_i (resp. e_i) occurs once at the right (resp. left) side of a $\prec$ relation and never at the right (resp. left) side of a $\prec$ relation.*

Because of the preceding observations, the graph of the $\prec$ relation can be partitioned into cycles and maximal chains. Moreover the maximal chains start with s_i nodes and end with e_i nodes.

Definition 4.1 *In order to avoid confusion we will call the chains/cycles of the $\prec$ relation super chains (cycles) and will denote them by capital letters $(C_1, C_2, \ldots)$.*

Let C_i be a super cycle/chain. L_i is the set of lightpaths in the super cycle/chain, namely $L_i \stackrel{def}{=} \bigcup C_i$.

U_i (resp. F_i, S_i, E_i) is the number of the $\prec_U$ (resp. $\prec_F$, $\prec_S$, $\prec_E$) relationships in C_i. Note that:

$$S_i = E_i = \begin{cases} 1 & \text{if } C_i \text{ is a super chain} \\ 0 & \text{otherwise} \end{cases}$$

$U \stackrel{def}{=} \sum U_i$ and $F \stackrel{def}{=} \sum F_i$ are the total number of $\prec_U$ and $\prec_F$ relationships, or in other words the number of times $event - U$ and $event - F$ happen respectively.

$S \stackrel{def}{=} \sum S_i$ and $E \stackrel{def}{=} \sum E_i$ are the total number of $\prec_S$ and $\prec_E$ relationships, or in other words the number of times $start$ and end events happen respectively. Note that $S = E$ which is in turn equal to the number of super chains. Moreover $E = \sum_{i=0}^{n-1} max(0, |\sigma_i| - |\tau_i|)$. Note that this number is a function of the input and does not depend of the output.

Let $R = |L|$, C the number of cycles removed in the preprocessing phase of the algorithm and R_C the number of lightpaths in these cycles.

Let C^* the number of cycles in the output of an optimal algorithm and R_C^* the number of lightpaths in these cycles.

Lemma 4.2

$$2E + 2U + 3F + 2R_C \leq 2R \qquad (1)$$

Proof: Consider a super chain or cycle C_i in the output of the algorithm. For each $\prec_U$ relationship in C_i there are at least two lightpaths in L_i which are involved. For each $\prec_F$ relationship there are at least three lightpaths involved. For each $\prec_S$ or $\prec_E$ relationship there is at least one lightpath involved. Each lightpath in L_i is exactly in one chain thus involved in two relationships. Therefore : $2U_i + 3F_i + S_i + E_i \leq 2|L_i|$. Summing up over all the super chains/cycles we obtain: $S + E + 2U + 3F = 2U + 3F + S + E = 2U + 3F + 2E \leq 2\sum |L_i|$. The lightpaths which are involved in events are those who survived the preprocessing phase, therefore in $\sum |L_i| = R - R_C$.

□

Lemma 4.3

$$U + F + 2C \leq 2L_{min} \qquad (2)$$

Proof: Consider the set Y_i of ligthpats crossing an edge $(i, i+1)$ such that $|Y_i| = L_{min}$. Every lightpath is involved in two relationships. This is in particular true for the ligthpaths in Y_i. On the other hand each U or F event involves at least one lightpath from Y_i which survived the preprocessing phase. The number of these lightpaths is $L_{min} - C$. Therefore $U + F \leq 2(L_{min} - C)$.

□

Lemma 4.4

$$OPT \geq R + E + U - R_C + C. \qquad (3)$$

Proof: At each node i, the paths of σ_i can be classified as follows:

- $R_C(i)$ paths removed by the preprocessing phase.
- $E(i) + U(i)$**:** paths which did not take part in the maximum matching.

- $|MM_i|$ paths participating in the maximum matching.

Therefore, $|MM_i| = |\sigma_i| - R_C(i) - E(i) - U(i)$. Summing up over all nodes i and defining $MM \stackrel{def}{=} \sum_{i=0}^{n-1} |MM_i|$ we have:

$$MM = R - R_C - E - U. \tag{4}$$

Consider a maximum matching MM_i of the node graph after the removal of the C cycles of PAF and a maximum matching MM_i^0 of the node graph of node i before any preprocessing. Each pair of paths $p_1 \in \sigma_i, p_2 \in \tau_i$ removed by the preprocessing phase reduces the value of the maximum matching at most by two. Therefore

$$|MM_i| \geq \left|MM_i^0\right| - 2R_C(i)$$

Summing over all nodes we have $MM \geq MM^0 - 2R_C$. In fact we will later prove:

$$MM \geq MM^0 - 2R_C + C \tag{5}$$

On the other hand as it is pointed out in [9] and [3]:

$$OPT \geq 2R - MM^0$$

Combining with (5) and substituting the value of MM in (4) we get

$$\begin{aligned} OPT &\geq 2R - MM - 2R_C + C \\ &= 2R - (R - R_C - E - U) - 2R_C + C \\ &= R + E + U - R_C + C \end{aligned}$$

as required.

It remains to prove inequality (5). It is sufficient to show that in each one of the C cycles removed in the preprocessing phase, there is at least one path that does not reduce MM by two. Assume, by contradiction that there is a cycle $p_1, p_2, ..., p_k$ removed in the preprocessing phase such that each successive pair of paths p_{i-1}, p_i reduces MM by two. This means that both p_{i-1} and p_i are matched to two paths by OPT. Let these paths be b_{i-1} and a_i respectively (see Figure 4.2). Considering the fact that a_i, p_i, b_i is part of a chain/cycle of OPT:

$$len(a_i) + len(p_i) + len(b_i) \leq n$$

Summing over all nodes $v_1, v_2, ..., v_k$:

$$\sum len(a_i) + \sum len(p_i) + \sum len(b_i) \leq nk$$

$\sum len(p_i) = n$ because they form a cycle, therefore

$$\sum len(a_i) + \sum len(b_i) \leq n(k-1)$$

On the other hand:

$$len(a_i) + len(b_{i-1}) > n$$

for, otherwise they can be added to any matching which do not include any of them, and MM_i is not reduced by two. Summing over all nodes, we get:

$$\sum len(a_i) + \sum len(b_i) > nk$$

a contradiction.

□

The total length of the chains of OPT is at least $nL_{min} - nC^*$. Obviously $len(c) < n$ for each chain. Therefore there must be at least $n(L_{min} - C^*)/n = L_{min} - C^*$ chains in OPT.

$$OPT \geq R + L_{min} - C^*. \tag{6}$$

Any algorithm should use at least one ADM for the beginning of a lightpath and at least $|\sigma_i| - |\tau_i|$ ADMs at the end of lightpaths ending at node i. Therefore:

$$OPT \geq R + \sum_{i=0}^{n-1} \max(0, |\sigma_i| - |\tau_i|) = R + E$$

By our assumption all cycles consist of at least 3 lightpaths, thus:

$$3C \leq R_C \tag{7}$$
$$3C^* \leq {R_C}^* \tag{8}$$

Our preprocessing phase removes a maximal number of cycles. Therefore, each cycle of OPT should contain at least one lightpath from the cycles of PAF, for, otherwise there would be an entire cycle which is not removed by PAF in the preprocessing phase. This would be a contradiction to the maximality of the cycles removed in the preprocessing phase. We conclude:

$$C^* \leq R_C \cdot \tag{9}$$

Obviously, the number of lightpaths in the chains of any solution is at least as the load induced by them on any edge, in particular on an edge of minimum load, Thus:

$$R - {R_C}^* \quad \geq \quad L_{min} - C^* \tag{10}$$

Theorem 4.1

$$\frac{PAF}{OPT} \leq \frac{11}{7}$$

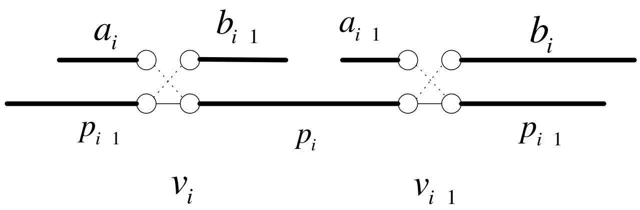
.

Proof: Assume the contrary, i.e. that for some $\rho > 11/7$, $PAF/OPT \geq \rho$. It is easy to see from the algorithm that $PAF = R + U + F + E$. Then:

$$R + U + F + E > \rho \cdot OPT \tag{11}$$

We substitute $\rho = 11/7$ and we seek for values of the variables which may satisfy all the constraints found so far. It can be shown that, the resulting Linear Program has no feasible solution. Therefore the corresponding ILP does not have a solution, a contradiction.

□



4.3. A Lower Bound

The maximum disjoint cycles problem in a ring $(MDCR)$, is the problem of partitioning L into chains and cycles, such that the number of cycles is maximum. Any algorithm solving the minimum ADM problem, is also solving the $MDCR$ problem. Its performance with respect the two problems may of course, be different.

In [1] the PIM (Preprocessed Iterative Matching) algorithm is presented and proven to have an approximation ratio between $\frac{4}{3}$ and $\frac{3}{2}$. A closer look to their proof reveals the following, general lower bound:

Any algorithm ALG with no performance guarantee on the $MDCR$ problem beyond maximality, has approximation ratio no better than $4/3$ for the minimum ADM problem.

The following improved lower bound is recently established in [4]:

Lemma 4.5 *Any algorithm ALG with no performance guarantee on the $MDCR$ problem beyond maximality, has approximation ratio no better than $3/2$ for the minimum ADM problem.*

Corollary 4.1 $3/2 \leq PAF/OPT \leq 11/7$.

5. Algorithms with Improved Preprocessing - IPAF and IEMZ

5.1. The Motivation

In this section we develop algorithms with approximation ratio better than $3/2$. In view of the lower bound, this necessarily requires a better performance guarantee on the $MDCR$ problem.

There is an infinite family of instances, such that for every $\epsilon > 0$ there is an instance on which the approximation ratio of the Assign First algorithm is more than $5/3 - \epsilon$. This family of instances is omitted from this extended abstract. Other algorithms which do not have the preprocessing phase are proven in [1] to have approximation ratio of exactly $\frac{3}{2}$. On the other hand PAF has approximation ratio at most $11/7$ which is better than $5/3$. This clearly indicates that the preprocessing phase improves the algorithm. Thus it is natural to investigate the approximability of the problem with respect to this preprocessing phase, more precisely the part of the solution which consists of cycles.

In this section we analyze the performance of PAF for special cases and present an improved version of it which is essentially PAF with improved preprocessing phase and then combine the improved preprocessing phase with an algorithm with approximation ratio $3/2$ and manage to reach an approximation ratio of $10/7 + \epsilon$.

Lemma 5.1 *A 2 approximation to the $MDCR$ problem, implies a $7/5$ approximation to the minimum ADM problem.*

Proof: We add the constraint $C^* \leq 2C$ to the LP in the proof of Theorem 4.1 and we show that the resulting LP has no solution for $\rho > 7/5$.

□

Corollary 5.1 *PAF is a $7/5$-approximation to the minimum ADM problem for instances with no cycles.*

The above result indicates that a better approximation to the $MDCR$ problem would lead a better approximation to the minimum ADM problem. We proceed with an algorithm with a better preprocessing phase.

5.2. Algorithm $IPAF_k$

$IPAF_k\{$
  Run preprocessing phase A of PAF
    Calculate all the possible cycles c such that $|c| \leq k$
    Find a maximum set packing (MSP) of these cycles
  Remove the maximum packing from L
  Run preprocessing phase B of PAF
  Run processing phase of PAF
$\}$

5.3. Analysis of $IPAF_k$

In the sequel *short cycles* are cycles containing at most k lightpaths and *long cycles* are cycles with at least $k+1$ lightpaths. The calculation of all the $short\ cycles$ may be done by choosing an edge e such that $l(e) = L_{min}$ and trying all the possible clockwise extensions of the lightpaths passing through this edge. We repeat this process $k-1$ times. The number of cycles with k lightpaths or less is at most $L_{min}(L_{max})^{k-1}$, in other words there are a polynomial number of cycles, and they can be computed in polynomial time as described. Moreover each cycle is as a set with at most k elements. A $(k/2+\epsilon)-approximation$ for the MSP problem is given in ([7]), for all $k \geq 3$. For any fixed ϵ and k, the running time of the algorithm is polynomial.

Note that for instances with cycles of at most 4 paths, our preprocessing is a 2-approximation for the $MDCR$ problem, then $IPAF_k$ is a $7/5$-approximation to the minimum ADM problem. Generally:

Theorem 5.1

$$\frac{IPAF_5}{OPT} \leq 1.48$$

.

Proof: We define the following variables:

C^*_- and C^*_+ are the number of $short$ and $long$ cycles in an optimal solution. Similarly we define C_- and C_+ are defined similarly with respect to the solution obtained by $IPAF_k$. In the same way we define R^*_{C-}, R^*_{C+}, R_{C-} and R_{C+} as the number of lightpaths in these cycles. The following equalities are immediate:

$$C^* = C^*_- + C^*_+ \quad (12)$$
$$C = C_- + C_+ \quad (13)$$
$$R^*_C = R^*_{C-} + R^*_{C+} \quad (14)$$
$$R_C = R_{C-} + R_{C+} \quad (15)$$

as are the following inequalities:

$$3C^*_- \leq R^*_{C-} \leq kC^*_- \quad (16)$$
$$3C_- \leq R_{C-} \leq kC_- \quad (17)$$
$$(k+1)C^*_+ \leq R^*_{C+} \quad (18)$$
$$(k+1)C_+ \leq R_{C+} \quad (19)$$

Let $\overline{C_-}$ be the maximum number of disjoint short cycles. The MSP algorithm guarantees $C_- \geq \frac{\overline{C_-}}{k/2+\epsilon'}$ for every $\epsilon' > 0$. On the other hand the optimal solution can not include more than $\overline{C_-}$ short cycles. Thus $C_- \geq \frac{C^*_-}{k/2+\epsilon'}$. For all $\epsilon'' > 0$, we have:

$$(k+\epsilon'')C_- \geq 2C^*_- \quad (20)$$

We extend the linear program in the proof of Theorem 4.1 by adding the above constraints. It can be shown that for $k = 5$ and $\rho > 1.48$, the resulting linear program has no feasible solution.

□

5.4. Algorithm $IEMZ_k$

The following algorithm has the same preprocessing phase as $IPAF_k$, it solves the remaining instance using algorithm EMZ introduced in [3].

$IEMZ_k$ (n, L) {
  Run preprocessing of $IPAF_k$
  For each path $l \in L$ do $c(l) = (l)$
  For each node i from 1 clockwise to 0 do{
    Find a maximum matching MM_i of the node graph G_i.
    $\tau'_i =$ The unmatched nodes of τ_i.
    $\sigma'_i =$ The unmatched nodes of σ_i.
    G'_i =The complete bipartite graph $(\tau'_i, \sigma'_i, \tau'_i \times \sigma'_i)$.
    Find a maximum matching MM'_i OF G'_i.
    $\tau''_i =$ The unmatched nodes of τ'_i.
    $\sigma''_i =$ The unmatched nodes of σ'_i.
    For each edge $(a, b) \in MM_i$ UNION $(c(a), c(b))$
    For each edge $(a, b) \in MM'_i$ **unmatched**
    For each b in τ''_i **start**
    For each b in σ''_i **end**
  }
  For each chain/cycle c do {
    Let $c = p_1, p_2, ..., p_k$
    i=1;
    For j=1 to k{
      If $p_{j+1} \asymp p_i$ {
        split c into two chains such that
        p_j and p_{j+1} are in different chains
        $i = j + 1$
        failure
      }
    }
  }
}

5.5. Analysis of $IEMZ_k$

Lemma 5.2

$$2F + E + U + R_C \leq R$$

Proof: In the second phase of EMZ algorithm, there is a one-to-one mapping from the F events to the successful matchings. This can be seen by the following simple argument taken from [3]: The first matching of a chain can not be broken, because otherwise the total length of paths involved are summing up to at least $n + 1$, which means that there is no edge joining them in the node graph, therefore they can not be part of a matching. Therefore to any broken matching (F event) there is a corresponding unbroken matching. In our notation this is denoted as:

$$F \leq MM - F.$$

Substituting the value of MM we get:

$$2F \leq R - R_C - E - U.$$

□

Theorem 5.2

$$\frac{IEMZ_5}{OPT} \leq 10/7 + \epsilon$$

.

Sketch of Proof: It is easy to show that all the inequations that hold form $IPAF_k$ hold for $IEMZ_k$ too, except Lemma 4.2. We replace the corresponding constraint in the linear program in the proof of Theorem 5.1 with the result of Lemma 5.2 and get a new linear program. This linear program has a solution for $\rho = 10/7$ but no solution for any $\rho > 10/7$. The details of the proof are omitted in this version of the paper. We assume a solution for $\rho = 10/7+\delta$ for any $\delta > 0$, we compare it to a solution of $\rho = 10/7$. Considering the tightly satisfied constraints, we reach a contradiction.

□

6. Simulation Results, Conclusions and Open Problems

The calculation of OPT is in NP-Hard. Therefore, we compared the performance of $IPAF_5$ and PIM on 200 random instances with $10 \leq n \leq 16$ and $20 \leq R \leq 150$. $IPAF_5$ led to better results for almost all the instances, where the difference in the performance grows with the size of the input, i.e. the number of the lightpaths.

In this work we investigated the relationship between the arc version of the minimum ADM problem and the maximum disjoint cycles problem. We saw that on instances without cycles we can obtain a $7/5 - approximation$ and generally we can not get better than $3/2 - approximation$ if we can not perform better than the trivial greedy algorithm for the $MDCR$ problem. We presented the algorithm $IPAF_5$ which has a provable upper bound of 1.48. Finally we presented algorithm $IEMZ_k$ which has the same preprocessing phase and proved it to have an approximation ratio at most $10/7 + \epsilon$.

The algorithm presented in [4] has a preprocessing phase removing short cycles and paths, whereas our preprocessing phase removes short cycles only, thus answering affirmatively an open question mentioned in [4].

A possible improvement to the preprocessing phase is to modify it to choose the value of k as a function of R/L_{min} or alternatively to try different values for k and get the best solution among them. This direction might lead to a provable increase in the performance.

Another possible direction is to improve the preprocessing phase by replacing the algorithm [7] which solves the general MSP problem for $k - sets$ with an algorithm that achieves better performance by taking advantage of the properties of the $k - cycles$.

Acknowledgements

We thank Jean-Claude Bermond and Stéphane Perennes from INRIA at Sophia-Antipolis for fruitful discussions with the first author while visiting them during the initial stage of this research.

References

[1] G. Călinescu and P.-J. Wan. Traffic partition in wdm/sonet rings to minimize sonet adms. In *15th International Parallel and Distributed Processing Symposium*, 2001.

[2] T. Eilam. *Cost Vs. Quality, Tradeoffs in Communication Networks*. PhD thesis, Faculty of Computer Science, Technion-Israel Institute of Technology, Feb 2000.

[3] T. Eilam, S. Moran, and S. Zaks. Lightpath arrangement in survivable rings to minimize the switching cost. *IEEE Journal of Selected Area on Communications*, 20(1):172–182, Jan 2002.

[4] L. Epstein and A. Levin. Better bounds for minimizing sonet adms. *To appear in WAOA, Workshop on Approximation and Online Algorithms*, Sep 2004.

[5] O. Gerstel, P. Lin, and G. Sasaki. Wavelength assignment in a wdm ring to minimize cost of embedded sonet rings. In *Infocom'98, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 69–77, 1998.

[6] O. Gerstel, P. Lin, and G. Sasaki. Combined wdm and sonet network design. In *INFOCOM'99, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 734–43, 1999.

[7] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal Of Discrete Mathematics*, 2(1):68–722, Feb 1989.

[8] S. Khanna. A polynomial time approximation scheme for the sonet ring loading problem. *Bell Labs Technical Journal*, pages 36–41, Spring 1997.

[9] L. Liu, X. Li, P.-J. Wan, and O. Frieder. Wavelength assignment in a wdm rings to minimize sonet adms. In *INFOCOM'2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Tel-Aviv, Israel*, pages 1020–1025, 2000.

[10] P. Wan, G. Călinescu, L.-W. Liu, and O. Frieder. Grooming of arbitrary traffic in sonet/wdm rings. *IEEE Journal of Selected Area on Communications*, 18(10):1995–2003, 2000.

[11] G. Wilfong and P. Winkler. Ring routing and wavelength translation. In *Proceedings of the ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'98, San Francisco, California*, pages 333–341, Jan 1998.