On-line Maximum Matching in Complete Multipartite Graphs with Implications to the Minimum ADM Problem on a Star Topology

Mordechai Shalom[1] *, Prudence W.H. Wong **[2], and Shmuel Zaks[3]

[1] TelHai Academic College, Upper Galilee, 12210, Israel cmshalom@telhai.ac.il
[2] Department of Computer Science, The University of Liverpool, Liverpool, UK pwong@liverpool.ac.uk
[3] Department of Computer Science, Technion, Haifa, Israel zaks@cs.technion.ac.il

Abstract. One of the basic problems in optical networks is assigning wavelengths to (namely, coloring of) a given set of lightpaths so as to minimize the number of ADM switches. In this paper we present a connection between maximum matching in complete multipartite graphs and ADM minimization in star networks. A tight 2/3 competitive ratio for finding a maximum matching implies a tight 10/9 competitive ratio for finding a coloring that minimizes the number of ADMs.

Keywords: *Online Matching, Multi-Partite Graphs, Wavelength Assignment, Wavelength Division Multiplexing (WDM), Optical Networks, Add-Drop Multiplexer (ADM).*

1 Introduction

Optical wavelength-division multiplexing (WDM) is today the most promising technology that enables us to deal with the enormous growth of traffic in communication networks, like the Internet. A communication between a pair of nodes is done via a *lightpath*, which is assigned a certain wavelength. In graph-theoretic terms, a lightpath is a simple path in the network, with a color assigned to it.

Given a WDM network $G = (V, E)$ comprising optical nodes and a set of full-duplex lightpaths $P = \{p_1, p_2, ..., p_N\}$ of G, the wavelength assignment (WLA) task is to assign a wavelength to each lightpath p_i. Recent studies in optical networks dealt with the issue of assigning colors to lightpaths, so that every two lightpaths that share an edge get different colors.

When the various parameters comprising the switching mechanism in these networks became clearer the focus of studies shifted, and today many studies concentrate on the total hardware cost. The key point here is that each lightpath uses two Add-Drop Multiplexers (ADMs), one at each endpoint. If two adjacent lightpaths, i.e. lightpaths sharing a common endpoint, are assigned the same

* This research was supported in part by the Israel Science Foundation research grant.
** This research was partly supported by EPSRC Grant EP/E028276/1.

wavelength, then they can use the same ADM. Because ADMs are designed to be used mainly in ring and path networks in which the degree of a node is at most two, an ADM may be shared by at most two lightpaths. The total cost considered is the total number of ADMs. Lightpaths sharing ADMs in a common endpoint can be considered as concatenated, so that they form longer paths or cycles. These paths/cycles do not use any edge $e \in E$ twice, for otherwise they cannot use the same wavelength which is a necessary condition to share ADMs.

Minimizing the number of ADMs in optical networks is a main research topic in recent studies. The problem was introduced in [GLS98] for the ring topology. An approximation algorithm for the ring topology with approximation ratio of $\frac{3}{2}$ was presented in [CW02], and was improved in [SZ04],[EL04],[EL09] to $\frac{10}{7}+\epsilon$, $\frac{10}{7}$, and $\frac{98}{69}$ respectively. The off-line version of the Minimum ADM problem can be solved optimally for trees [ZCXG03].

The motivation for the on-line problem stems from the need to utilize the cost of use of the optical network. We assume that the switching equipment is installed in the network. Once a lightpath arrives, we need to assign it two ADMs, and our target is to determine which wavelength to assign to it so that we minimize the cost, measured by the total number of ADMs used.

An on-line algorithm with competitive ratio of $\frac{7}{4}$ for any network topology was presented in [SWZ07]. It was shown that this algorithm has an optimal $\frac{7}{4}$ competitive ratio for a ring topology, and an optimal $\frac{3}{2}$ competitive ratio for a path topology.

In this paper we present a connection between matchings in complete multipartite graphs and ADM minimization in star networks. Online bipartite maximum matching problem was introduced in [KVV90] and a $(1-1/e)$-competitive randomized algorithm was proposed , which is optimal [GM08,BM08]. The greedy algorithm is 1/2-competitive and is optimal for deterministic online algorithms. The problem has found applications in other related problems (e.g., [ANR02], [AR05], [AC06]). The problem has also been studied for general weighted graphs in [KP93] where a 1/3-competitive deterministic algorithm is given. In [Sit96], a formula for the cardinality of the maximum matching in complete multipartite graph is given. To the best of our knowledge, there is no work on online maximum matching in multipartite graphs.

We show a tight bound of 2/3 for the competitive ratio of deterministic algorithms for this maximum matching problem, implying a tight 10/9 competitive ratio for the ADM minimization problem.

In Section 2 we describe both problems. The lower bound and upper bound for the competitive ratio are presented in Sections 3 and 4, respectively. We conclude and discuss further research directions in Section 5.

2 Preliminaries

2.1 The ADM Minimization Problem

An instance α of the problem is a pair $\alpha = (G, P)$ where $G = (V, E)$ is an undirected graph and P is a multi-set of simple paths in G. In an on-line instance,

the graph G is known in advance and the set P of paths is given on-line. In this case we denote $P = \{p_1, p_2, ..., p_N\}$ where p_i is the i-th path of the input and $P_i = \{p_j \in P | j \leq i\}$ consists of the first i paths of the input.

A valid chain (resp. cycle) is a path (resp.cycle) formed by the concatenation of distinct paths $p_{i_0}, p_{i_1}, ... \in P$ that do not have an edge in common. A solution S of an instance $\alpha = (G, P)$ is a partition of P into valid chains and cycles.

The cost of a valid chain (resp. cycle) containing k paths is $k+1$ (resp. k) ADMs. The cost $cost(S)$ of a solution S is the sum of the costs of its valid chains and cycles. The objective is to find a solution S such that $cost(S)$ is minimum.

Let OPT be an optimal off-line algorithm. An online algorithm A to a minimization problem is said to be c-competitive (for $c \geq 1$) if there exists some $b \geq 0$ such that for any input I, $A(I) \leq c \times OPT(I) + b$, where $A(I)$ and $OPT(I)$ denote the cost of the output of A and OPT, respectively, on input I. Similarly, for a maximization problem A is said to be c-competitive (for $c \leq 1$) if there exists some $b \geq 0$ such that for any input I, $A(I) \geq c \times OPT(I) - b$.

2.2 The Star Topology and the d-PARTMM problem

Let G be a star with d edges, namely $G = (V, E), V = \{0, 1, ..., d\}, E = \{e_1, ..., e_d\}$ and $\forall 1 \leq i \leq d, e_i = (0, i)$. In this case paths in P are of length either 1 or 2. Let $p \in P$ be a path of length 2 with endpoints i and j. For a path p' to be concatenated to p, one of its endpoints should be either i or j. In this case p and p' would share one of the edges e_i, e_j. Therefore paths of length 2 constitute valid chains of size 1 in every solution, and each such path costs 2 ADMs. We therefore assume w.l.o.g. that all the paths are of length 1.

Two paths p, p' of length 1 have always a common endpoint 0. Let i (resp. j) be the other endpoint of p (resp. p'). They can form a valid chain if and only if $i \neq j$. In this case the cost of the valid chain is 3, or in other words 3/2 per path, whereas a path constituting a valid chain costs 2. Therefore our goal is to maximize the number of valid chains of size 2, that is equivalent to find a maximum matching in a complete d-partite graph. This problem will be called the d-PARTMM problem throughout the paper.

The following lemma is proven in [Sit96], we give a sketch of proof for completeness.

Lemma 2.1 *Let $G = (V, E)$ be a complete d-partite graph with N nodes (V is partitioned into parts $V_1, V_2, \cdots, V_d$). G contains a matching of size $\lfloor \frac{N}{2} \rfloor$ if and only if $|V_i| \leq \lceil \frac{N}{2} \rceil$ for every i.*

Sketch of Proof: The 'only if' part is obvious. The 'if' part is proved by induction on N. Assume w.l.o.g. that V_1 and V_2 are the two largest sets among $V_1, V_2, \cdots, V_d$. The induction step stems from the observation that, by matching any node $v \in V_1$ with any node $v' \in V_2$, and deleting these two nodes from G, results in a complete bipartite graph G' of $N-2$ nodes (and sets $V_1 - \{v\}, V_2 - \{v'\}, V_3, \cdots, V_d$). The proof follows by the inductive hypothesis.

□

A d-partite graph having a matching of size $\lfloor \frac{N}{2} \rfloor$ will be called *balanced*.

In an on-line instance, the input consists of d empty parts that are initially empty. The nodes of the graph are revealed one at a time where each node is an element of some part. For each node p_i of the input, an on-line algorithm has to decide to which node $p_j (j < i)$ to match it, or to leave it unmatched. As the graph is a complete d-partite graph p_j is eligible if and only if it is unmatched and it is not in the same part as p_i. Once two nodes are matched, the decision cannot be revoked. An on-line instance will be called *completely balanced* if every prefix of it is balanced.

When $d = 2$, the on-line problem can be solved optimally by the greedy algorithm, which matches to each p_i an unmatched node p_j in the other part of the graph as long as such a p_j exists. In the rest of our work we assume $d \geq 3$.

We conclude this section with the following claim that relates the performances of any solution with respect to these two problems.

Lemma 2.2 *A solution to the d-PARTMM problem is a c-approximation* ($0 \leq c \leq 1$)*, if and only if the corresponding solution to the Minimum ADM problem is a* $\frac{4-c}{3}$*-approximation, with the same additive term.*

Proof. Let MM be the size of a maximum d-partite matching, and let M be the size of a d-partite matching that constitutes c-approximation. There exists a constant $b \geq 0$, such that $M \geq cMM - b$. Let S^* be an optimal solution to the Minimum ADM problem and S be a solution corresponding to the c-approximation.

$$\begin{aligned} cost(S^*) &= 2|P| - MM \\ cost(S) &= 2|P| - M \leq 2|P| - cMM + b = \frac{2|P| - cMM}{2|P| - MM} cost(S^*) + b \\ &= \frac{2 - c(MM/|P|)}{2 - MM/|P|} cost(S^*) + b. \end{aligned}$$

As $0 \leq c \leq 1$, the coefficient of $cost(S^*)$ is a non decreasing function of $MM/|P|$. Considering that $MM/|P| \leq 1/2$ we conclude

$$cost(S) \leq \frac{2 - c/2}{2 - 1/2} cost(S^*) + b = \frac{4-c}{3} cost(S^*) + b.$$

On the other hand let S be a $c' = \frac{4-c}{3}$ approximation to the Minimum ADM problem and M the size of d-partite matching that it induces. There is a constant b' such that

$$\begin{aligned} cost(S) &\leq c' \cdot cost(S^*) + b' \\ 2|P| - M &\leq c'(2|P| - MM) + b' \\ M &\geq c'MM + (1 - c')2|P| - b' \geq c'MM + (1 - c')4MM - b' \\ &= (4 - 3c')MM - b' = cMM - b'. \end{aligned}$$

□

3 Lower Bound

Lemma 3.1 *For any $c > \frac{2}{3}$ and $d \geq 3$ there is no c-competitive deterministic on-line algorithm for the d-PARTMM problem.*

Proof. Assume, by contradiction that there is a $\left(\frac{2}{3}+\epsilon\right)$-competitive deterministic on-line algorithm ALG for some $\epsilon > 0$. Then there is a constant $b \geq 0$, such that for any online instance I

$$ALG(I) \geq \left(\frac{2}{3}+\epsilon\right) OPT(I) - b$$

where $ALG(I)$ is the size of the matching returned by the algorithm and $OPT(I)$ is the size of the maximum matching.

For any non-negative integer k, consider the instance I containing $2k$ nodes, such that k of them are in V_1 and k of them are in V_2. Obviously $OPT(I) = k$, then

$$ALG(I) \geq \left(\frac{2}{3}+\epsilon\right) k - b = \frac{2}{3}k + \epsilon k - b$$

and ALG leaves $k - ALG(I)$ unmatched nodes at each one of V_1 and V_2. Let I' be the online instance which is obtained by appending to I, $2k$ nodes from part 3. In this phase ALG can not do better than matching the $2(k - ALG(I))$ unmatched nodes to the nodes of V_3. Therefore

$$ALG(I') \leq ALG(I) + 2(k - ALG(I)) = 2k - ALG(I) \leq \frac{4}{3}k - \epsilon k + b. \quad (1)$$

On the other hand $OPT(I') = 2k$. Then

$$ALG(I') \geq \left(\frac{2}{3}+\epsilon\right) OPT(I') - b = \left(\frac{2}{3}+\epsilon\right) 2k - b \quad (2)$$

Combining (1) and (2) we get

$$\begin{aligned} \left(\frac{2}{3}+\epsilon\right) 2k - b &\leq \frac{4}{3}k - \epsilon k + b \\ 2\epsilon k - b &\leq -\epsilon k + b \\ k &\leq \tfrac{2b}{3\epsilon} \end{aligned}$$

For any input I with k bigger than the right hand side we reach a contradiction. □

By applying Lemma 2.2 for $c = \frac{2}{3}$, and using Lemma 3.1, we have thus proved:

Theorem 1 *For any $c < \frac{10}{9}$, there is no c-competitive deterministic on-line algorithm for the Minimum ADM problem in star networks.*

4 Upper Bound

4.1 Eliminating unbalanced instances

The following lemma shows that the difficult instances of the d-PARTMM problem are the completely balanced instances.

Lemma 4.1 *There is a c-competitive deterministic algorithm for the d-PARTMM problem, if and only if there is a c-competitive deterministic algorithm for it when the instances are restricted to be completely balanced.*

Sketch of Proof: The 'only if' part is immediate. We now show the 'if' part. Let ALG be a c-competitive deterministic on-line algorithm for completely balanced instances, where $0 \leq c \leq 1$. We claim that the following algorithm ALG' is c-competitive for all instances.

ALG'
Initialization:
  $U \leftarrow \emptyset$
On input p_i do: $//I = \{p_1, ..., p_i\}$
  $B \leftarrow I \setminus U$
  If B is completely balanced then
    follow the decision of ALG on input B
  else{// There is exactly one part h with more than $\lfloor \frac{B}{2} \rfloor$ nodes of B
    If p_i is in h then
      leave p_i unmatched
    else{
      choose an arbitrary unmatched node $p_j \in U$ (*)
      match p_i to p_j
    }
    $U \leftarrow U \uplus \{p_i\}$
  }

First note that any instance I which is not completely balanced has prefixes which are not balanced. Each such prefix has one part h which is the "heaviest" part containing more than half of the nodes. As the instance is initially balanced it can be uniquely divided into intervals $B_1, U_1, B_2, U_2, ...$ which are alternately balanced and unbalanced. Each unbalanced interval U_i has a corresponding "heaviest" part h_i.

We describe in detail how the intervals U_i are determined. Consider a step during which the input became unbalanced. This happens necessarily after some odd step $2s_i - 1$ with s_i nodes in h_i. After this step $s_i + 1$ nodes out of $2s_i$ are in h_i. Now consider the first step that the input becomes balanced again. It happens necessarily after some even step $2e_i$ with $2e_i - 2$ nodes in h_i. After this step e_i nodes out of $2e_i - 1$ are in h_i. In this case U_i is the interval from $2s_i$ to $2e_i - 1$ during which the input contained $2e_i - 2s_i$ nodes out of which $e_i - s_i$ are in h_i. As $2e_i - 1$ is the first step that this happens, at any time between these two steps any node not in h_i can be matched to a node in h_i in line (*) of

ALG'. Moreover the nodes of U_i admit a perfect matching where each edge of the matching has an adjacent node in h_i. If we remove the sub-instance U_i from the input we remain with the instance until step $2s_i - 1$ which is balanced. Note that an intervals B_i may possibly be empty.

If the instance terminates with an unbalanced interval, i.e. the instance is unbalanced, then for the last unbalanced interval U_l, more than half of the nodes revealed during U_l, say $x + \delta$ of them, are in h_l where x is the total number of nodes in the other parts. If the instance is balanced let $\delta = 0$. Let $B = B_1, B_2, ...$ and $U = U_1, U_2, ..., U_{l-1}$.

Then ALG' returns $\frac{|U_i|}{2}$ matchings at each interval $U_i, i < l$. And for U_l it returns x matchings. On the other hand ALG "sees" only the sub-instance B that is completely balanced. Therefore for some constant b, it returns at least $c \cdot OPT(B) + b$ matchings. We conclude

$$ALG'(I) = ALG(B) + \sum_{i<l} \frac{|U_i|}{2} + x \geq c \cdot OPT(B) + b + \frac{|U|}{2} + x$$
$$\geq c\left(OPT(B) + \frac{|U|}{2} + x\right) + b.$$

On the other hand

$$OPT(I) \leq \left\lfloor \frac{|I| - \delta}{2} \right\rfloor = \left\lfloor \frac{|B \cup U|}{2} \right\rfloor + x = \left\lfloor \frac{|B|}{2} \right\rfloor + \frac{|U|}{2} + x = OPT(B) + \frac{|U|}{2} + x.$$

We conclude that ALG' is c-competitive.

□

4.2 Algorithm MATCHBYRATIO

In this section we present the algorithm $MATCHBYRATIO(\alpha, d)$ for completely balanced instances, where $0 < \alpha \leq 2/3$ and d is the number of parts of the graph. We prove that for any α in this interval the algorithm is α-competitive.

Algorithm MATCHBYRATIO is designed with the lower bound proof in mind. It depends on some constant $0 < \alpha \leq 2/3$ and the number of parts d (we justify the dependency on d in Section 5).

. In the preprocessing step, it calculates a value β depending on d. The algorithm attempts to maintain the number of the matchings to be close to α times the optimum (with an additive offset of β). Each time it falls behind this threshold it adds one matching to the output. One node of the matching is the current input node by definition of the problem. We call this node the *matching* node. The other node is chosen arbitrarily from the part having the biggest ratio of unmatched nodes (ratio of number of unmatched nodes to total number of nodes in the part). This node is called the *matched* node.

The following pseudo-code of the algorithm will be helpful in the analysis. The algorithm partitions the nodes into three sets: U is the set of unmatched nodes, MG is the set of matching nodes, and MD is the set of matched nodes.

$MATCHBYRATIO(\alpha, d)$
Initialization:
Calculate β as a function of d // See analysis (Theorem 2)
$U \leftarrow \emptyset$
$MG \leftarrow \emptyset$
$MD \leftarrow \emptyset$

On input p_i do: $//I = \{p_1, ..., p_i\}$
$opt \leftarrow OPT(I)$ $// = \lfloor \frac{i}{2} \rfloor)$
if $|MG| < \lfloor \alpha \cdot opt - \beta \rfloor$ then { // (**)
Let h be the part containing p_i
choose an arbitrary unmatched node p_j from part $h' \neq h$ having a maximal ratio of unmatched nodes
if there is no such node then FAIL (*)
output the edge (p_i, p_j)
$U \leftarrow U \setminus \{p_j\}$
$MD \leftarrow MD \cup \{p_j\}$
$MG \leftarrow MG \cup \{p_i\}$
} else {
$U \leftarrow U \cup \{p_i\}$
}

By the description of the algorithm, it is clearly α-competitive unless it fails in the line marked by (*). It remains to prove that this does not happen if $\alpha \leq 2/3$.

We begin by introducing some notation. U_i (resp. MD_i, MG_i) is the value of the set U (resp. MD, MG) after step i of the algorithm, in other words after it has processed p_i. Let also $M_i \stackrel{def}{=} MG_i \uplus MD_i$ and $T_i \stackrel{def}{=} M_i \uplus U_i$. P is the set of all the input nodes. We denote by $P_1, ..., P_d$ the parts of the multipartite graph, clearly $P = \uplus_{h=1}^{d} P_h$. For any subset Q of P, $X_i(Q) \stackrel{def}{=} X_i \cap Q$ where X stands for any one of U, MD, MG, M or T. Whenever X is the name of a set, its lowercase counterpart x denotes its size. For instance $mg_i(P_h)$ is the number of matching nodes of P_h after input p_i is processed by the algorithm. For a nonempty subset Q of P we define its unmatched ratio as $\rho_i(Q) \stackrel{def}{=} \frac{u_i(Q)}{t_i(Q)}$.

A basic property of the algorithm is that the sizes $u_i, m_i, ...$ do not depend on the input and are functions of i only. However their subdivision, i.e. the sizes $u_i(P_h), m_i(P_h), ...$ depend on the input.

Lemma 4.2 *For all steps $1 \leq i \leq j$, in which the condition in line (**) is true, we have*

$$\alpha i - 2\beta + 2 - 2\alpha \leq m_i < \alpha i - 2\beta + 2 \tag{3}$$
$$(1-\alpha)i + 2\beta - 2 < u_i \leq (1-\alpha)i + 2\beta - 2 + 2\alpha \tag{4}$$

and

$$|m_j - m_i - \alpha(j-i)| < 2\alpha$$
$$|u_j - u_i - (1-\alpha)(j-i)| < 2\alpha.$$

Proof. Omitted. □

We assume by contradiction that the algorithm fails, and then use backward analysis to reach a contradiction. Under the failure assumption we define recursively the following two finite sequences:

i_0 is the step during which the algorithm failed and H_0 is the part containing the input at step i_0. Formally,
$H_0 = P_h$, where $p_{i_0} \in P_h$.

For any $k > 0$:

i_k is the last step before i_{k-1} that a matching is added to the output and none of its nodes are in H_{k-1}. If such a step does not exist then i_k is undefined and i_{k-1} terminates the sequence, otherwise:

$H_k = H_{k-1} \cup P_h$ where $p' \in P_h$ and p' is the matched node at step i_k.

Note that a matching or a failure may occur only at even steps, because $\alpha\lfloor\frac{i}{2}\rfloor$ does not increase at odd steps. Therefore i_k is even for all k. Note also that the length of the sequence is at most $d-1$, because each time a part of the graph is added to H, and at least one part (i.e. the part of the matching node) is left out.

Lemma 4.3 *For any $d \geq 3$ and any $\alpha < 1/2$, $MATCHBYRATIO(\alpha, d)$ does not FAIL if $\beta \geq 1$.*

Proof. Assume by contradiction that the algorithm fails at some step, and let i_0 be the step before the failure. By definition H_0 is the part of the graph containing the input node at this step. All the unmatched nodes should be in H_0, because otherwise the algorithm would pick an unmatched node from $P \setminus H_0$ and construct a matching, thus would not fail. Therefore we have

$$u_{i_0}(H_0) = u_{i_0} > (1-\alpha)i_0 + 2\beta - 2 \geq (1-\alpha)i_0 > i_0/2.$$

On the other hand as the instance is balanced we have $u_{i_0}(H_0) \leq t_{i_0}(H_0) \leq \lceil i_0/2 \rceil = i_0/2$, a contradiction. □

Lemma 4.4 *If $\alpha \leq 2/3$ and $\beta > d$ then*

$$u_{i_k}(H_k) \geq 2(1-\alpha)t_{i_k}(H_k) + 2\beta - 2 - k \geq 2(1-\alpha)t_{i_k}(H_k).$$

Proof. The second inequality follows from $\beta > d$, $k < d$ and $d \geq 3$. We will prove the first inequality by induction on k.

$k = 0$**:** The proof is similar to the proof of Lemma 4.3:

$$u_{i_0}(H_0) = u_{i_0} > (1-\alpha)i_0 + 2\beta - 2 = (1-\alpha)t_{i_0} + 2\beta - 2 \geq 2(1-\alpha)t_{i_0}(H_0) + 2\beta - 2$$

where the last inequality holds because the instance is balanced.

$k > 0$**:** Recall that $i_k < i_{k-1}$ and both even. For readability we denote $m = i_k$ and $n = i_{k-1}$. We will analyze the change in the sizes of the sets U, MG, MD, etc... from step $m + 1$ to step n.

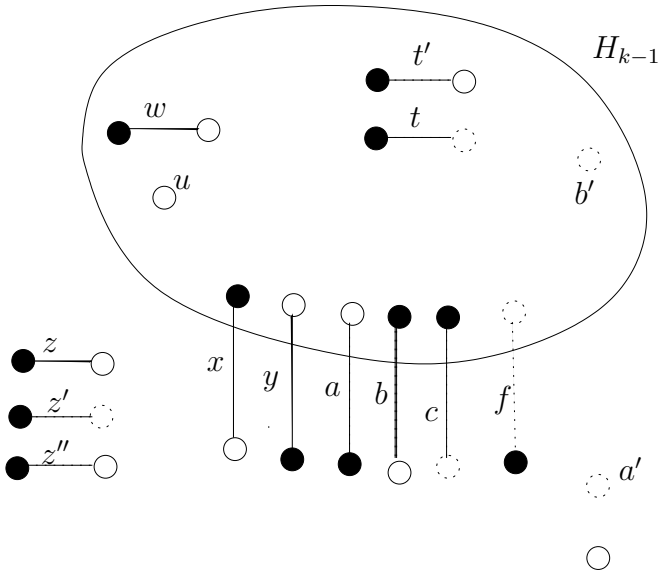

Fig. 1. The scenario discussed in the proof of Lemma 4.4

Consult Figure 1 for the following discussion. Solid edges are matchings that were output until step m and dotted edges are matchings that were output from step $m + 1$ to step n. A black node is a matching node, and a white node is a matched node. An unmatched node is drawn with a solid border if it was input until step m, and with a dotted border otherwise. The figure shows all the possibilities of matchings and all the possibilities of unmatched nodes. The letters on the nodes (resp. edges) are the number of such nodes (resp. edges).

First note that m is the last step during which two elements of $\overline{H_{k-1}} = P \setminus H_{k-1}$ are matched to each other. Therefore $z' = z'' = 0$ and the set sizes are as follows:

$$
\begin{aligned}
md_m(H_{k-1}) &= y + w \\
mg_m(H_{k-1}) &= x + w \\
m_m(H_{k-1}) &= x + y + 2w \\
u_m(H_{k-1}) &= a + u + t' \\
t_m(H_{k-1}) &= x + y + 2w + a + u + t' \\
md_n(H_{k-1}) &= y + w + a + f + t + t' \\
mg_n(H_{k-1}) &= x + w + b + c + t + t' \\
m_n(H_{k-1}) &= x + y + 2w + a + b + c + f + 2t + 2t' \\
u_n(H_{k-1}) &= u + b' \\
t_n(H_{k-1}) &= x + y + 2w + a + b + c + f + u + b' + 2t + 2t' \\
mg_n - mg_m &= a + b + c + f + t + t' \\
t_n - t_m &= a' + b' + a + b + 2c + 2f + 2t + t'
\end{aligned}
$$

Claim. $\frac{b'-a-t'}{b+c+f+b'+2t+t'} \leq 2(1-\alpha) + \frac{\alpha}{n-m}$.

Proof.

$$
\begin{aligned}
\frac{b'-a-t'}{b+c+f+b'+2t+t'} &\leq \frac{b'-a+t}{b+c+f+b'+2t+t'} \\
&\leq \frac{a'+b'+c+f+t}{a'+b'+a+b+2c+2f+2t+t'}. \qquad (5)
\end{aligned}
$$

The second inequality above holds by the following observation. If $b'-a+t \leq 0$ then the left hand side is non-positive and the right hand side is positive, therefore the inequality holds. Otherwise $b'-a+t > 0$ and the left hand side is a fraction with value of at most 1. Increasing the value of both nominator and denominator by the same value increases the fraction. Note that the right hand side is obtained from the left hand side by adding $a+a'+c+f$ to both nominator and denominator.

On the other hand we have

$$
\begin{aligned}
&\frac{a'+b'+c+f+t}{a'+b'+a+b+2c+2f+2t+t'} = \frac{(t_n - t_m) - (mg_n - mg_m)}{t_n - t_m} \\
&= \frac{(n-m) - (mg_n - mg_m)}{n-m} < \frac{(n-m) - (\frac{\alpha}{2}(n-m) - \alpha)}{n-m} \\
&= \left(1 - \frac{\alpha}{2}\right) + \frac{\alpha}{n-m} \leq 2(1-\alpha) + \frac{\alpha}{n-m}. \qquad (6)
\end{aligned}
$$

Note that the last inequality holds because $\alpha \leq 2/3$. By combining (5) and (6) we get the claim. □

By the inductive assumption we have

$$u_n(H_{k-1}) \geq 2(1-\alpha)t_n(H_{k-1}) + 2\beta - 2 - (k-1),$$

and by the above claim

$$\begin{aligned} b' - a - t' &\leq 2(1-\alpha)(b+c+f+b'+2t+t') + \alpha\frac{(b+c+f+b'+2t+t')}{n-m} \\ &< 2(1-\alpha)(b+c+f+b'+2t+t') + 1. \end{aligned}$$

We combine to get

$$\begin{aligned} u_m(H_{k-1}) &= u_n(H_{k-1}) - (b' - a - t') \\ &> 2(1-\alpha)t_n(H_{k-1}) + 2\beta - 2 - (k-1) \\ &\quad -2(1-\alpha)(b+c+f+b'+2t+t') - 1 \\ &= 2(1-\alpha)t_m(H_{k-1}) + 2\beta - 2 - k > 2(1-\alpha)t_m(H_{k-1}). \end{aligned}$$

Therefore $\rho_m(H_{k-1}) > 2(1-\alpha)$. Now recall that in step m the algorithm matched two nodes, both not from H_{k-1}. Let P_h be the part of the graph containing the matched node. By the behavior of the algorithm this means that the unmatched ratio of P_h is at least as much as each one of the parts of H_{k-1}, thus at least as much as entire H_{k-1}, therefore $\rho_m(P_h) \geq \rho_m(H_{k-1}) > 2(1-\alpha)$, i.e. $u_m(P_h) > 2(1-\alpha)t_m(P_h)$. Combining with the above, we get:

$$\begin{aligned} u_m(H_k) &= u_m(H_{k-1}) + u_m(P_h) \\ &> 2(1-\alpha)t_m(H_{k-1}) + 2\beta - 2 - k + 2(1-\alpha)t_m(P_h) \\ &= 2(1-\alpha)t_m(H_k) + 2\beta - 2 - k. \end{aligned}$$

□

Lemma 4.5 *For any $d \geq 3$ and $1/2 \leq \alpha \leq 2/3$, $MATCHBYRATIO(\alpha, d)$ does not FAIL if $\beta > \frac{3}{2}d + 3$.*

Proof. If the algorithm fails, the sequences $i_0, i_1, ..., i_l$ and the $H_0, H_1, ..., H_l$ are defined, where $l \leq d-2$. Let $\overline{H_l} = P \setminus H_l$. By the definition of the sequence H and the fact that H_l is the last item of the sequence no matching can have both nodes in $\overline{H_l}$. Such a matching would cause part of $\overline{H_l}$ to be added to H_l, to form H_{l+1}. In other words all the matchings contain at least one node in H_l. Therefore

$$m_{i_l}(\overline{H_l}) \leq m_{i_l}(H_l).$$

α and β satisfy the conditions of Lemma 4.4, by which we have

$$u_{i_l}(H_l) \geq 2(1-\alpha)t_{i_l}(H_l) + \delta = 2(1-\alpha)m_{i_l}(H_l) + 2(1-\alpha)u_{i_l}(H_l) + \delta$$

$$\frac{1}{3}u_{i_l}(H_l) \geq (2\alpha - 1)u_{i_l}(H_l) \geq 2(1-\alpha)m_{i_l}(H_l) + \delta \geq \frac{2}{3}m_{i_l}(H_l) + \delta$$

$$u_{i_l}(H_l) \geq 2m_{i_l}(H_l) + 3\delta.$$

for $\delta = 2\beta - 2 - k$. We conclude

$$u_{i_l} \geq u_{i_l}(H_l) \geq 2m_{i_l}(H_l) + 3\delta \geq m_{i_l}(H_l) + m_{i_l}(\overline{H_l}) + 3\delta = m_{i_l} + 3\delta$$

$$u_{i_l} - m_{i_l} \geq 3\delta.$$

Recalling that $\alpha \geq 1/2$ we get from (3) and (4)

$$u_{i_l} - m_{i_l} \leq (1-2\alpha)i_l + 4\beta + 4\alpha - 4 \leq 4\beta + 4\alpha - 4 \leq 4\beta.$$

Therefore

$$4\beta \geq 3\delta = 6\beta - 6 - 3k$$

$$2\beta \leq 3k + 6 \leq 3d + 6$$

contradicting our assumption. □

Combining Lemma 4.3 and Lemma 4.5 we get

Theorem 2 *For any $d \geq 3$ and any $\alpha \leq 2/3$, $MATCHBYRATIO(\alpha, d)$ does not FAIL if $\beta > \frac{3}{2}d + 3$.*

Corollary 4.1 *$MATCHBYRATIO(2/3, d)$ is a 2/3-competitive algorithm for the $d - PARTMM$ problem, with an additive term of $\frac{3}{2}d + 3$.*

5 Conclusion and Possible Improvements

In this paper, we have shown an interesting connection between maximum matchings in complete multipartite graphs and ADM minimization in star networks. We show a tight 2/3 competitive ratio for finding a maximum matching, implying a tight 10/9 competitive ratio for finding a coloring that minimizes the number of ADMs.

The algorithm used in the upper bound is 2/3-competitive with an additive term β that depends on the number of parts d of the graph, which is supposed to be known in advance. Actually this is the situation for the ADM minimization problem in which the star network (and therefore d) is given in advance. On the other hand our algorithm is usable also when d is not known a priori by a slight modification. We start with the assumption $d = 3$ and increment the value of d each time the first node of some part is revealed, and adjust β accordingly. In this case $\beta = O(d)$ is unbounded and depends on the on-line input. However this does not constitute a problem if d is $o(N)$.

An open question is to improve the competitive ratio by randomized algorithms. It is also interesting to consider other topologies like trees. We believe the result in star networks may be a starting point for the investigation of the more general tree networks.

Another important extension is to consider the ADM minimization problem when grooming is allowed; in graph-theoretic terms, this amounts to coloring the paths so that at most g of them are crossing any edge, and where each ADM can serve up to g paths that come from at most two of its adjacent edges (see [GRS98,ZM03]). Another direction of extension is to the case where more involved switching functions are under consideration.

References

[AC06] Y. Azar and Y. Chaiutin. Optimal node routing. In *The proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science, Marseille, France*, pages 596–607, February 2006.

[ANR02] Y. Azar, J. (Seffi) Naor, and R. Rom. The competitiveness of on-line assignments. In *The proceedings of the 13th SIAM Symposium on Discrete Algorithms. San Francisco, CA, USA*, pages 203–210, January 2002.

[AR05] Y. Azar and Y. Richter. Management of multi-queue switches in QoS networks. *Algorithmica*, 43(1-2):81–96, 2005.

[BM08] B. Birnbaum and C. Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, 2008.

[CW02] G. Călinescu and P-J. Wan. Traffic partition in WDM/SONET rings to minimize SONET ADMs. *Journal of Combinatorial Optimization*, 6(4):425–453, 2002.

[EL04] L. Epstein and A. Levin. Better bounds for minimizing SONET ADMs. In *2nd Workshop on Approximation and Online Algorithms, Bergen, Norway*, September 2004.

[EL09] L. Epstein and A. Levin. Better bounds for minimizing sonet adms. *J. Comput. Syst. Sci.*, 75(2):122–136, 2009.

[GLS98] 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*, pages 69–77, 1998.

[GM08] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, USA*, pages 982–991, January 2008.

[GRS98] O. Gerstel, R. Ramaswami, and G. Sasaki. Cost effective traffic grooming in WDM rings. In *INFOCOM'98, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 1998.

[KP93] B. Kalyanasundaram and K. Pruhs. On-line weighted matching. *J. Algorithms*, 14(3):478–488, 1993.

[KVV90] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching, 1990.

[Sit96] D. Sitton. Maximum matchings in complete multipartite graphs. *Furman University Electronic Journal of Undergraduate Mathematics*, 2:6–16, 1996.

[SWZ07] M. Shalom, P. W. H. Wong, and S. Zaks. Optimal on-line colorings for minimizing the number of ADMs in optical networks. In *DISC*, pages 435–449, 2007.

[SZ04] M. Shalom and S. Zaks. A $10/7 + \epsilon$ approximation scheme for minimizing the number of ADMs in SONET rings. In *First Annual International Conference on Broadband Networks, San-José, California, USA*, pages 254–262, October 2004.

[ZCXG03] F. Zhou, G. Chen, Y Xu, and J. Gu. Minimizing ADMs on WDM directed fiber trees. *Journal of Computer Science & Technology*, 18(8):725–731, Nov 2003.

[ZM03] K. Zhu and B. Mukherjee. A review of traffic grooming in WDM optical networks: Architecture and challenges. *Optical Networks Magazine*, 4(2):55–64, March-April 2003.