On Randomized Algorithms for Matching in the Online Preemptive Model

Ashish Chiplunkar[1,⋆], Sumedh Tirodkar[2], and Sundar Vishwanathan[2]

[1] Amazon Development Centre, Banglore
ashish.chiplunkar@gmail.com
[2] Department of Computer Science and Engineering, IIT Bombay
{sumedht,sundar}@cse.iitb.ac.in

Abstract. We investigate the power of randomized algorithms for the maximum cardinality matching (MCM) and the maximum weight matching (MWM) problems in the online preemptive model. In this model, the edges of a graph are revealed one by one and the algorithm is required to always maintain a valid matching. On seeing an edge, the algorithm has to either accept or reject the edge. If accepted, then the adjacent edges are discarded. The complexity of the problem is settled for deterministic algorithms [7,9].

Almost nothing is known for randomized algorithms. A lower bound of 1.693 is known for MCM with a trivial upper bound of two. An upper bound of 5.356 is known for MWM. We initiate a systematic study of the same in this paper with an aim to isolate and understand the difficulty. We begin with a primal-dual analysis of the deterministic algorithm due to [7]. All deterministic lower bounds are on instances which are trees at every step. For this class of (unweighted) graphs we present a randomized algorithm which is $\frac{28}{15}$-competitive. The analysis is a considerable extension of the (simple) primal-dual analysis for the deterministic case. The key new technique is that the distribution of primal charge to dual variables depends on the "neighborhood" and needs to be done after having seen the entire input. The assignment is asymmetric: in that edges may assign different charges to the two end-points. Also the proof depends on a non-trivial structural statement on the performance of the algorithm on the input tree.

The other main result of this paper is an extension of the deterministic lower bound of Varadaraja [9] to a natural class of randomized algorithms which decide whether to accept a new edge or not using *independent* random choices. This indicates that randomized algorithms will have to use *dependent* coin tosses to succeed. Indeed, the few known randomized algorithms, even in very restricted models follow this.

We also present the best possible $\frac{4}{3}$-competitive randomized algorithm for MCM on paths.

⋆ This work was done when the author was a student at IIT Bombay.

© Springer-Verlag Berlin Heidelberg 2015
N. Bansal and I. Finocchi (Eds.): ESA 2015, LNCS 9294, pp. 325–336, 2015.
DOI: 10.1007/978-3-662-48350-3_28

1 Introduction

Matching has been a central problem in combinatorial optimization. Indeed, algorithm design in various models of computations, sequential, parallel, streaming, etc., have been influenced by techniques used for matching. We study the maximum cardinality matching (MCM) and the maximum weight matching (MWM) problems in the online preemptive model. In this model, edges $e_1, \ldots,$ e_m of a graph, possibly weighted, are presented one by one. An algorithm is required to output a matching M_i after the arrival of each edge e_i. This model constrains an algorithm to accept/reject an edge as soon as it is revealed. If accepted, the adjacent edges, if any, have to be discarded from M_i.

An algorithm is said to have a *competitive ratio* α if the cost of the matching maintained by the algorithm is at least $\frac{1}{\alpha}$ times the cost of the offline optimum over all inputs. The deterministic complexity of this problem is settled. For maximum cardinality matching (MCM), it is an easy exercise to prove a tight bound of two.

The weighted version (MWM) is more difficult. Improving an earlier result of Feigenbaum et al. [5], McGregor [7] gave a deterministic algorithm together with an ingenious analysis to get a competitive ratio of $3 + 2\sqrt{2} \approx 5.828$. Later, this was proved to be optimal by Varadaraja [9].

Very little is known on the power of randomness for this problem. Recently, Epstein et al. [4] proved a lower bound of $1+\ln 2 \approx 1.693$ on the competitive ratio of randomized algorithms for MCM. This is the best lower bound known even for MWM. Epstein et al. [4] also give a 5.356-competitive randomized algorithm for MWM.

In this paper, we initiate a systematic study of the power of randomness for this problem. Our main contribution is perhaps to throw some light on where lies the difficulty. We first give an analysis of McGregor's algorithm using the traditional Primal-Dual framework (see Appendix A in [3]). All lower bounds for deterministic algorithms (both for MCM and MWM) employ *growing trees*. That is, the input graph is a tree at every stage. It is then natural to start our investigation for this class of inputs. For this class, we give a randomized algorithm (that uses two bits of randomness) that is $\frac{28}{15}$ competitive. While this result is modest, already the analysis is considerably more involved than the traditional primal dual analysis. In the traditional primal dual analysis of the matching problem, the primal charge (every selected edge contributes one to the charge) is distributed (perhaps equally) to the two end-points. In the online case, this is usually done as the algorithm proceeds. Our assignment depends on the structure of the final tree, so this assignment happens at the end. Our charge distribution is *not* symmetric. It depends on the position of the edge in the tree (we make this clear in the analysis) as also the behavior of neighboring edges. The main technical lemma shows that the charge distribution will depend on a neighborhood of distance at most four. We also note that these algorithms are (restricted versions of) randomized greedy algorithms even in the offline setting. Obtaining an approximation ratio less than two for general graphs, even in the offline setting is a notorious problem. See [8,2] for a glimpse of the difficulty.

The optimal maximal matching algorithm for MCM, and McGregor's [7] optimal deterministic algorithm for MWM are both local algorithms. The choice of whether a new edge should accepted or rejected is based only on the weight of the new edge and the weight of the conflicting edges, if any, in the current matching.

It is natural to add randomness to such local algorithms, and to ask whether they do better than the known deterministic lower bounds. An obvious way to add randomness is to accept/reject the new edge with certain probability, which is only dependent on the new edge and the conflicting edges in the current matching. The choice of adding a new edge is independent of the previous coin tosses used by the algorithm. We call such algorithms *randomized local algorithms*. We show that randomized local algorithms cannot do better than optimal deterministic algorithms. This indicates that randomized algorithms may have to use dependent coin tosses to get better approximation ratios. Indeed, the algorithm by Epstein et al. does this. So do our randomized algorithms.

The randomized algorithm of Epstein et al. [4] works as follows. For a parameter θ, they round the weights of the edges to powers of θ randomly, and then they update the matching using a deterministic algorithm. The weights get distorted by a factor $\frac{\theta \ln \theta}{\theta - 1}$ in the rounding step, and the deterministic algorithm has a competitive ratio of $2 + \frac{2}{\theta - 2}$ on θ-*structured graphs*, i.e., graphs with edge weights being powers of θ. The overall competitive ratio of the randomized algorithm is $\frac{\theta \ln \theta}{\theta - 1} \cdot \left(2 + \frac{2}{\theta - 2}\right)$ which is minimized at $\theta \approx 5.356$. A natural approach to reducing this competitive ratio is to improve the approximation ratio for θ structured graphs. However, we prove that the competitive ratio $2 + \frac{2}{\theta - 2}$ is tight for θ-structured graphs, as long as $\theta \geq 4$, for deterministic algorithms.

One (minor) contribution of this paper is a randomized algorithms for MCM on paths, that achieves a competitive ratio of $\frac{4}{3}$, with a matching lower bound.

2 Barely Random Algorithms for MCM

In this section, we present barely random algorithms, that is, algorithms that use a constant number of random bits, for MCM on growing trees.

The ideal way to read the paper, for a reader of leisure, is to first read our analysis of McGregor's algorithm (presented in Appendix A in [3], then the analysis of the algorithm for trees with maximum vertex degree three (presented in Appendix B.2 in [3]) and then this section. The dual variable management which is the key contribution gets progressively more complicated. It is local in the first two cases. Here are the well known Primal and Dual formulations of the matching problem.

Primal LP	Dual LP
$\max \sum_e x_e$	$\min \sum_v y_v$
$\forall v : \sum_{v \in e} x_e \leq 1$	$\forall e : y_u + y_v \geq 1$
$x_e \geq 0$	$y_v \geq 0$

2.1 Randomized Algorithm for MCM on Growing Trees

In this section, by using only two bits of randomness, we beat the deterministic lower bound of 2 for MCM on growing trees.

Algorithm 1. Randomized Algorithm for Growing Trees

1. The algorithm maintains four matchings: M_1, M_2, M_3, and M_4.
2. On receipt of an edge e, the processing happens in two phases.
   (a) **The augment phase.** The new edge e is added to each M_i in which there are no edges adjacent to e.
   (b) **The switching phase.** For $i = 2, 3, 4$, in order, e is added to M_i (if it was not added in the previous phase) and the conflicting edge is discarded, provided it decreases the quantity $\sum_{i,j\in[4],i\neq j} |M_i \cap M_j|$.
3. Output matching M_i with probability $\frac{1}{4}$.

We begin by assuming (we justify this below) that all edges that do not belong to any matching are leaf edges. This helps in simplifying the analysis. Suppose that there is an edge e which does not belong to any matching, but is not a leaf edge. By removing e, the tree is partitioned into two subtrees. The edge e is added to the tree in which it has 4 neighboring edges. (There must be such a subtree, see next para.) Each tree is analysed separately.

We will say that a vertex(/an edge) is *covered* by a matching M_i if there is an edge in M_i which is incident on(/adjacent to) the vertex(/edge). We also say that an edge is *covered* by a matching M_i if it belongs to M_i. We begin with the following observations.

- After an edge is revealed, its end points are covered by all 4 matchings.
- An edge e that does not belong to any matching has 4 edges incident on one of its end points such that each of these edges belong to a distinct matching. This holds when the edge is revealed, and does not change subsequently.

An edge is called *internal* if there are edges incident on both its end points. An edge is called *bad* if its end points are covered by only 3 matchings.

We begin by proving some properties about the algorithm. The key structural lemma that keeps "influences" of bad edges local is given below. The two assertions in the Lemma have to be proved together by induction.

Lemma 1. *1. An internal edge is covered by at least four matchings (counted with multiplicities). It is not necessary that these four edges be in distinct matchings.*
2. If p, q and r are three consecutive vertices on a path, then bad edges cannot be incident on all 3 of these vertices, (as in figure 1).

The proof of this lemma is in the Appendix B.4 in [3].

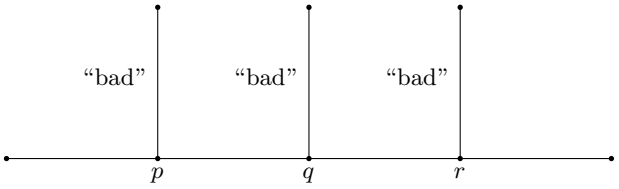

Fig. 1. Forbidden Configuration

Theorem 1. *The randomized algorithm for finding MCM on growing trees is* $\frac{28}{15}$*-competitive.*

A local analysis like the one in Appendix B.2 in [3] will not work here. For a reason, see Appendix B.3 in [3]. The analysis of this algorithm proceeds in two steps. Once all edges have been seen, we impose a partial order on the vertices of the tree and then with the help of this partial order, we distribute the primal charge to the dual variables, and use the primal-dual framework to infer the competitive ratio. If every edge had four adjacent edges in some matching (counted with multiplicities) then the distribution of dual charge is easy. However we do have edges which have only three adjacent edges in matchings. We would like the edges in matchings to contribute more to the end-points of these edges. Then, the charge on the other end-point would be less and we need to balance this through other edges. Details follow.

Ranks: Consider a vertex v. Let $v_1, \ldots, v_k$ be the neighbors of v. For each i, let d_i denote the maximum distance from v to any leaf if there was no edge between v and v_i.The rank of v is defined as the minimum of all the d_i. Observe that the rank of v is one plus the second highest rank among the neighbors of v. Thus there can be at most one neighbor of vertex v which has rank at least the rank of v. All leaves have rank 0. Rank 1 vertices have at most one non-leaf neighbor.

Lemma 2. *There exists an assignment of the primal charge amongst the dual variables such that the dual constraint for each edge* $e \equiv (u, v)$ *is satisfied at least* $\frac{15}{28}$ *in expectation, i.e.* $\mathbb{E}[y_u + y_v] \geq \frac{15}{28}$.

Proof. Consider an edge $e \equiv (u, v)$ where rank of u is i and rank of v is j. We will show that $y_u + y_v \geq 2 + \epsilon$ for such an edge, when summed over all four matchings. The value of ϵ is chosen later. The proof is by induction on the lexicographic order of $< j, i >$, $j \geq i$.

Dual Variable Management: Consider an edge e from a vertex of rank i to a vertex of rank j, such that $i \leq j$. This edge will distribute its primal weight between its end-points. The exact values are discussed in the proof of the claim below. In general, we look to transfer all of the primal charge to the higher ranked vertex. But this does not work and we need a finer strategy. This is detailed below.

- If e does not belong to any matching, then it does not contribute to the value of dual variables.

- If e belongs to a single matching then, depending on the situation, one of 0, ϵ or 2ϵ of its primal charge will be assigned to the rank i vertex and rest will be assigned to the rank j vertex. The small constant ϵ is determined later.
- If e belongs to two matchings, then at most 3ϵ of its primal charge will be assigned to the rank i vertex as required. The rest is assigned to the rank j vertex.
- If e belongs to three or four matchings, then its entire primal charge is assigned to the rank j vertex.

The analysis breaks up into six cases.

Case 1. Suppose e does not belong to any matching. Then it must be a leaf edge. Hence, $i = 0$. There must be 4 edges incident on v besides e, each belonging to a distinct matching. Of these 4, at least 3 say e_1, e_2, and e_3, must be from lower ranked vertices to the rank j vertex v. The edges e_1, e_2, and e_3, each assign a charge of $1 - 2\epsilon$ to y_v. Therefore, $y_u + y_v \geq 3 - 6\epsilon \geq 2 + \epsilon$.

Case 2. Suppose e is a bad edge that belongs to a single matching. Since no internal edge can be a bad edge, $i = 0$. This implies (Lemma 1) that, there is an edge e_1 from a rank $j - 1$ vertex to v, which belongs to a single matching. Also, there is an edge e_2, from v to a higher ranked vertex, which also belongs to a single matching. The edge e assigns a charge of 1 to y_v. If e_1 assigns a charge of 1 (or $1 - \epsilon$) to y_v, then e_2 assigns ϵ (or 2ϵ respectively) to y_v. In either case, $y_u + y_v = 2 + \epsilon$. The key fact is that e_1 could not have assigned 2ϵ to a lower ranked vertex. Since, then, by Lemma 1, e cannot be a bad edge.

Case 3. Suppose e is not a bad edge, and it belongs to a single matching.
Case 3(a). $i = 0$. There are two sub cases.

- There is an edge e_1 from some rank $j - 1$ vertex to v which belongs to 2 matchings, or there are two other edges e_2 and e_3 from some lower ranked vertices to v, each belonging to separate matchings. The edge e assigns a charge of 1 to y_v. Either e_1 assigns a charge of at least $2 - 3\epsilon$ to y_v, or e_2 and e_3 assign a charge of at least $1 - 2\epsilon$ each, to y_v. In either case, $y_u + y_v \geq 3 - 4\epsilon \geq 2 + \epsilon$.
- There is one edge e_1, from a rank $j - 1$ vertex to v, which belongs to a single matching, and there is one edge e_2, from v to a higher ranked vertex, which belongs to 2 matchings. The edge e assigns a charge of 1 to y_v. If e_1 assigns a charge of 1 (or $1 - \epsilon$ or $1 - 2\epsilon$) to y_v, then e_2 assigns ϵ (or 2ϵ or 3ϵ respectively) to y_v. In either case, $y_u + y_v = 2 + \epsilon$.

Case 3(b). $i > 0$. There are two sub cases.

- There are at least two edges e_1 and e_2 from lower ranked vertices to u, and one edge e_3 from v to a higher ranked vertex. Each of these edges are in one matching only (not necessarily the same matching).
- There is one edge e_4 from a vertex of lower rank to u, at least one edge e_5 from a lower ranked vertex to v, and one edge e_6 from v to a vertex of higher rank. All these edges belong to a single matching (not necessarily the same).

The edge e assigns a charge of 1 among y_u and y_v. If e_1 and e_2 assign a charge of at least $1-2\epsilon$ each, to y_u, then $y_u + y_v \geq 3 - 4\epsilon \geq 2 + \epsilon$. Similarly, if e_4 assigns a charge of at least $1-2\epsilon$ to y_u, and e_5 assigns a charge of at least $1-2\epsilon$ to y_v, then $y_u + y_v \geq 3 - 4\epsilon \geq 2 + \epsilon$.

Case 4. Suppose e is a bad edge that belongs to two matchings. Then $i = 0$. This implies that there is an edge e_1, from v to a vertex of higher rank which belongs to a single matching. The edge e assigns a charge of 2 to y_v, and the edge e_1 assigns a charge of ϵ to y_v. Thus, $y_u + y_v = 2 + \epsilon$.

Case 5. Suppose e is not a bad edge and it belongs to two matchings. This means that either there is an edge e_1 from a lower ranked vertex to u, which belongs to at least one matching, or there is an edge from some lower ranked vertex to v that belongs to at least one matching, or there is an edge from v to some higher ranked vertex which belongs to two matchings. The edge e assigns a charge of 2 among y_u and y_v. The neighboring edges assign a charge of ϵ to y_u or y_v (depending on which vertex it is incident), to give $y_u + y_v \geq 2 + \epsilon$.

Case 6. Suppose, e belongs to 3 or 4 matchings, then trivially $y_u + y_v \geq 2 + \epsilon$. From the above conditions, the best value for the competitive ratio is obtained when $\epsilon = \frac{1}{7}$, yielding $\mathbb{E}[y_u + y_v] \geq \frac{15}{28}$. □

Lemma 2 implies that the competitive ratio of the algorithm is at most $\frac{28}{15}$.

3 Lower Bounds

3.1 Lower Bound for MWM

In this section, we prove a lower bound on the competitive ratio of a natural class of randomized algorithms in the online preemptive model for MWM. The algorithms in this class, which we call *local* algorithms, have the property that their decision to accept or to reject a new edge is completely determined by the weights of the new edge and the conflicting edges in the matching maintained by the algorithm. Indeed, the optimal deterministic algorithm by McGregor [7] is a local algorithm. The notion of locality can be extended to randomized algorithms as well. In case of *randomized local algorithms*, the event that a new edge is accepted is independent of all such previous events, given the current matching maintained by the algorithm. Furthermore, the probability of this event is completely determined by the weight of the new edge and the conflicting edges in the matching maintained by the algorithm. Given that the optimal $(3+2\sqrt{2})$-competitive deterministic algorithm for MWM is a local algorithm, it is natural to ask whether randomized local algorithms can beat the deterministic lower bound of $(3 + 2\sqrt{2})$ by Varadaraja [9]. We answer this question in the negative, and prove the following theorem.

Theorem 2. *No randomized local algorithm for the MWM problem can have a competitive ratio less than* $\alpha = 3 + 2\sqrt{2} \approx 5.828$.

Note that the randomized algorithm by Epstein et al. [4] does not fall in this category, since the decision of accepting or rejecting a new edge is also dependent on the outcome of the coins tossed at the beginning of the run of the algorithm. (For details, see Section 3 of [4].) In order to prove Theorem 2, we will crucially use the following lemma, which is a consequence of Section 4 of [9].

Lemma 3. *If there exists an infinite sequence $(x_n)_{n\in\mathbb{N}}$ of positive real numbers such that for all n, $\beta x_n \geq \sum_{i=1}^{n+1} x_i + x_{n+1}$, then $\beta \geq 3 + 2\sqrt{2}$.*

Characterization of Local Randomized Algorithms. Suppose, for a contradiction, that there exists a randomized local algorithm $\mathcal{A}$ with a competitive ratio $\beta < \alpha = 3 + 2\sqrt{2}$, $\beta \geq 1$. Define the constant γ to be

$$\gamma = \frac{\beta\left(1 - \frac{1}{\alpha}\right)}{\left(1 - \frac{\beta}{\alpha}\right)} = \frac{\beta(\alpha - 1)}{\alpha - \beta} \geq 1 > \frac{1}{\alpha}$$

For $i = 0, 1, 2$, if w is the weight of a new edge and it has i conflicting edges, in the current matching, of weights $w_1, \ldots, w_i$, then $f_i(w_1, \ldots, w_i, w)$ gives the probability of switching to the new edge. The behavior of $\mathcal{A}$ is completely described by these three functions. We need the following key lemma to state our construction of the adversarial input.

The lemma states (informally) that given an edge of weight w_1, there exists weights x and y, close to each other such that if an edge of weight x (respective y) is adjacent to an edge of weight w_1, the probability of switching is at most (respectively at least) δ.

Lemma 4. *For every $\delta \in (0, 1/\alpha)$, $\epsilon > 0$, and w_1, there exist x and y such that $f_1(w_1, x) \geq \delta$, $f_1(w_1, y) \leq \delta$, $x - y \leq \epsilon$, and $w_1/\alpha \leq y \leq x \leq \gamma w_1$.*

The proof of this lemma can be found in Appendix C in [3].

The Adversarial Input. The adversarial input is parameterized by four parameters: $\delta \in (0, 1/\alpha)$, $\epsilon > 0$, m, and n, where m and n determine the graph and δ and ϵ determine the weights of its edges.

Define the infinite sequences $(x_i)_{i\in\mathbb{N}}$ and $(y_i)_{i\in\mathbb{N}}$, as functions of ϵ and δ, as follows. $x_1 = 1$, and for all i, having defined x_i, let x_{i+1} and y_i be such that $f_1(x_i, x_{i+1}) \geq \delta$, $f_1(x_i, y_i) \leq \delta$, $x_{i+1} - y_i \leq \epsilon$, and $x_i/\alpha \leq y_i \leq x_{i+1} \leq \gamma x_i$. Lemma 4 ensures that such x_{i+1} and y_i exist. Furthermore, by induction on i, it is easy to see that for all i,

$$1/\alpha^i \leq y_i \leq x_{i+1} \leq \gamma^i \tag{1}$$

These sequences will be the weights of the edges in the input graph.

Given m and n, the input graph contains several layers of vertices, namely $A_1, A_2, \ldots, A_{n+1}, A_{n+2}$ and $B_1, B_2, \ldots, B_{n+1}$; each layer containing m vertices. The vertices in the layer A_i are named $a_1^i, a_2^i, \ldots, a_m^i$, and those in layer B_i are named analogously. We have a complete bipartite graph J_i between layer A_i

and A_{i+1} and an edge between a^i_j and b^i_j for every i, j (that is, a matching M_i between A_i and B_i).

For $i = 1$ to n, the edges $\{(a^i_j, a^{i+1}_{j'})|1 \leq j, j' \leq m\}$, in the complete bipartite graph between A_i and A_{i+1}, have weight x_i, and the edges $\{(a^i_j, b^i_j)|1 \leq j \leq m\}$, in the matching between A_i and B_i, have weight y_i. The edges in the complete graph J_{n+1} have weight x_n, and those in the matching M_{n+1} have weight y_n. Note that weights x_i and y_i depend on ϵ and δ, but are independent of m and n. Clearly, the weight of the maximum weight matching in this graph is bounded from below by the weight of the matching $\bigcup_{i=1}^{n+1} M_i$. Since $y_i \geq x_{i+1} - \varepsilon$, we have

$$\text{OPT} \geq m \left(\sum_{i=1}^{n} y_i + y_n \right) \geq m \left(\sum_{i=2}^{n+1} x_i + x_{n+1} - (n+1)\epsilon \right) \tag{2}$$

The edges of the graph are revealed in $n+1$ phases. In the i^{th} phase, the edges in $J_i \cup M_i$ are revealed as follows. The phase is divided into m sub phases. In the j^{th} sub phase of the i^{th} phase, edges incident on a^i_j are revealed, in the order $(a^i_j, a^{i+1}_1), (a^i_j, a^{i+1}_2), \ldots, (a^i_j, a^{i+1}_m), (a^i_j, b^i_j)$.

Analysis of the Lower Bound. The overall idea of bounding the weight of the algorithm's matching is as follows. In each phase i, we will prove that as many as $m - O(1)$ edges of J_i and only $\delta m + O(1)$ edges of M_i are picked by the algorithm. Furthermore, in the $i + 1^{\text{th}}$ phase, since $m - O(1)$ edges from J_{i+1} are picked, all but $O(1)$ edges of the edges picked from J_i are discarded. Thus, the algorithm ends up with $\delta m + O(1)$ edges from each M_i, and $O(1)$ edges from each J_i, except possibly J_n and J_{n+1}. The algorithm can end up with at most m edges from $J_n \cup J_{n+1}$, since the size of the maximum matching in $J_n \cup J_{n+1}$ is m. Thus, the weight of the algorithm's matching is at most mx_n plus a quantity that can be neglected for large m and small δ.

Let X_i (resp. Y_i) be the set of edges of J_i (resp. M_i) held by the algorithm at the end of input. Then we have,

Lemma 5. *For all $i = 1$ to n*

$$E[|Y_i|] \leq \delta m + \frac{1-\delta}{\delta}$$

Lemma 6. *For all $i = 1$ to $n - 1$*

$$E[|X_i|] \leq \frac{1-\delta}{\delta}$$

Lemma 7.

$$E[|Y_{n+1}|] \leq \delta m + \frac{1-\delta}{\delta}$$

The proof of the above lemmas can be found in Appendix C in [3].

We are now ready to prove Theorem 2. The expected weight of the matching held by $\mathcal{A}$ is

$$E[\text{ALG}] \leq \sum_{i=1}^{n} y_i E[|Y_i|] + y_n E[|Y_{n+1}|] + \sum_{i=1}^{n-1} x_i E[|X_i|] + x_n E[|X_n \cup X_{n+1}|]$$

Using Lemmas 5, 7, 6, and the facts that $y_i \leq x_{i+1}$ for all i and $E[|X_n \cup X_{n+1}|] \leq m$ (since $X_n \cup X_{n+1}$ is a matching in $J_n \cup J_{n+1}$), we have

$$E[\text{ALG}] \leq \left(\delta m + \frac{1-\delta}{\delta}\right)\left(\sum_{i=2}^{n+1} x_i + x_{n+1}\right) + \frac{1-\delta}{\delta}\sum_{i=1}^{n-1} x_i + m x_n$$

Since the algorithm is β-competitive, for all n, m, δ and ϵ we must have $E[\text{ALG}] \geq \text{OPT}/\beta$. From the above and equation (2), we must have

$$\begin{aligned}\left(\delta m + \tfrac{1-\delta}{\delta}\right)\left(\textstyle\sum_{i=2}^{n+1} x_i + x_{n+1}\right) \\ + \tfrac{1-\delta}{\delta}\textstyle\sum_{i=1}^{n-1} x_i + m x_n\end{aligned} \geq \frac{m}{\beta}\left(\sum_{i=2}^{n+1} x_i + x_{n+1} - (n+1)\epsilon\right)$$

Since the above holds for arbitrarily large m, ignoring the terms independent of m (recall that x_i's are functions of ϵ and δ only), we have for all δ and ϵ,

$$\delta\left(\sum_{i=2}^{n+1} x_i + x_{n+1}\right) + x_n \geq \frac{1}{\beta}\left(\sum_{i=2}^{n+1} x_i + x_{n+1} - (n+1)\epsilon\right)$$

that is,

$$x_n \geq \frac{1}{\beta}\left(\sum_{i=2}^{n+1} x_i + x_{n+1} - (n+1)\epsilon\right) - \delta\left(\sum_{i=2}^{n+1} x_i + x_{n+1}\right)$$

Taking limit inferior as $\delta \to 0$ in the above inequality, and noting that limit inferior is super-additive we get for all ϵ,

$$\begin{aligned}\liminf_{\delta\to 0} x_n \geq \tfrac{1}{\beta}\left(\textstyle\sum_{i=2}^{n+1} \liminf_{\delta\to 0} x_i + \liminf_{\delta\to 0} x_{n+1} - (n+1)\epsilon\right) \\ - \limsup_{\delta\to 0} \delta\left(\textstyle\sum_{i=2}^{n+1} x_i + x_{n+1}\right)\end{aligned}$$

Recall that x_i's are functions of ϵ and δ, and that from equation (1), $1/\alpha^i \leq x_{i+1} \leq \gamma^i$, where the bounds are independent of δ. Thus, all the limits in the above inequality exist. Moreover, $\lim_{\delta\to 0} \delta\left(\sum_{i=2}^{n+1} x_i + x_{n+1}\right)$ exists and is 0, for all ϵ. This implies $\limsup_{\delta\to 0} \delta\left(\sum_{i=2}^{n+1} x_i + x_{n+1}\right) = 0$ and we get for all ε,

$$\liminf_{\delta\to 0} x_n \geq \frac{1}{\beta}\left(\sum_{i=2}^{n+1} \liminf_{\delta\to 0} x_i + \liminf_{\delta\to 0} x_{n+1} - (n+1)\epsilon\right)$$

Again, taking limit inferior as $\epsilon \to 0$, and using super-additivity,

$$\liminf_{\epsilon\to 0}\liminf_{\delta\to 0} x_n \geq \frac{1}{\beta}\left(\sum_{i=2}^{n+1} \liminf_{\epsilon\to 0}\liminf_{\delta\to 0} x_i + \liminf_{\epsilon\to 0}\liminf_{\delta\to 0} x_{n+1}\right)$$

Note that the above holds for all n. Finally, let $\overline{x_n} = \liminf_{\epsilon \to 0} \liminf_{\delta \to 0} x_{n+1}$. Then we have the infinite sequence $(\overline{x_n})_{n \in \mathbb{N}}$ such that for all n, $\beta \overline{x_n} \geq \sum_{i=1}^{n+1} \overline{x_i} + \overline{x_{n+1}}$. Thus, by Lemma 3, we have $\beta \geq 3 + 2\sqrt{2}$.

3.2 Lower Bound for θ Structured Graphs

Recall that an edge weighted graph is said to be θ-structured if the weights of the edges are powers of θ. The following bound applies to any deterministic algorithm for MWM on θ-structured graphs.

Theorem 3. *No deterministic algorithm can have a competitive ratio less than* $2 + \frac{2}{\theta - 2}$ *for MWM on* θ*-structured graphs, for* $\theta \geq 4$.

The proof of the above theorem can be found in Appendix D in [3].

4 Randomized Algorithm for Paths

When the input graph is restricted to be a collection of paths, then every new edge that arrives connects two (possibly empty) paths. Our algorithm consists of several cases, depending on the lengths of the two paths.

Algorithm 2. Randomized Algorithm for Paths

$M = \emptyset$. $\{M$ is the matching stored by the algorithm.$\}$
for each new edge e **do**
  Let $L_1 \geq L_2$ be the lengths of the two (possibly empty) paths P_1, P_2 that e connects.
  If $L_1 > 0$ (resp. $L_2 > 0$), let e_1 (resp. e_2) be the edge on P_1 (resp. P_2) adjacent to e.
  if e is a disjoint edge $\{L_1 = L_2 = 0\}$ **then**
    $M = M \cup \{e\}$.
  else if e is revealed on a disjoint edge e_1 $\{L_1 = 1, L_2 = 0.\ e_1 \in M\}$ **then**
    with probability $\frac{1}{2}$, $M = M \setminus \{e_1\} \cup \{e\}$.
  else if e is revealed on a end point of path of length > 1 $\{L_1 > 1, L_2 = 0\}$ **then**
    if $e_1 \notin M$, $M = M \cup \{e\}$.
  else if e joins two disjoint edges $\{L_1 = L_2 = 1.\ e_1, e_2 \in M\}$ **then**
    with probability $\frac{1}{2}$, $M = M \setminus \{e_1, e_2\} \cup \{e\}$.
  else if e joins a path and a disjoint edge $\{L_1 > 1, L_2 = 1.\ e_2 \in M\}$ **then**
    if $e_1 \notin M$, $M = M \setminus \{e_2\} \cup \{e\}$.
  else if e joins two paths of length $> 1 \{L_1 > 1, L_2 > 1\}$ **then**
    if $e_1 \notin M$ and $e_2 \notin M$, $M = M \cup \{e\}$.
  end if
  Output M.
end for

The following simple observations can be made by looking at the algorithm:
– All isolated edges belong to M with probability one.

– The end vertex of any path of *length* > 1 is covered by M with probability $\frac{1}{2}$, and this is independent of the end vertex of any other path being covered.
– For a path of length $2, 3$, or 4, each maximal matching is present in M with probability $\frac{1}{2}$.

Theorem 4. *The randomized algorithm for finding MCM on path graphs is* $\frac{4}{3}$*-competitive.*

The proof of above theorem can be found in Appendix E in [3].

References

1. Buchbinder, N., Naor, J.: The Design of Competitive Online Algorithms via a Primal-Dual Approach. Foundations and Trends in Theoretical Computer Science 3(2-3), 93–263 (2009)
2. Chan, T.H., Chen, F., Wu, X., Zhao, Z.: Ranking on Arbitrary Graphs: Rematch via Continuous LP with Monotone and Boundary Condition Constraints. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, pp. 1112–1122 (2014)
3. Chiplunkar, A., Tirodkar, S., Vishwanathan, S.: On Randomized Algorithms for Matching in the Online Preemptive Model. CoRR abs/1412.8615 (2014). http://arxiv.org/abs/1412.8615
4. Epstein, L., Levin, A., Segev, D., Weimann, O.: Improved Bounds for Online Preemptive Matching. In: 30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, Kiel, Germany, pp. 389–399 (2013)
5. Feigenbaum, J., Kannan, S., McGregor, A., Suri, S., Zhang, J.: On Graph Problems in a Semi-streaming Model. Theor. Comput. Sci. 348(2), 207–216 (2005)
6. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An Optimal Algorithm for On-line Bipartite Matching. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, STOC 1990, pp. 352–358. ACM, New York (1990)
7. McGregor, A.: Finding Graph Matchings in Data Streams. In: Chekuri, C., Jansen, K., Rolim, J.D.P., Trevisan, L. (eds.) APPROX 2005 and RANDOM 2005. LNCS, vol. 3624, pp. 170–181. Springer, Heidelberg (2005)
8. Poloczek, M., Szegedy, M.: Randomized Greedy Algorithms for the Maximum Matching Problem with New Analysis. In: 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, pp. 708–717 (2012)
9. Badanidiyuru Varadaraja, A.: Buyback Problem - Approximate Matroid Intersection with Cancellation Costs. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 379–390. Springer, Heidelberg (2011)
10. Yao, A.C.C.: Probabilistic computations: Toward a unified measure of complexity. In: 18th Annual Symposium on Foundations of Computer Science, FOCS 1977, pp. 222–227 (October 1977)