



Online Matching in Regular Bipartite Graphs with Randomized Adversary

Josep Fàbrega and Xavier Muñoz^(✉)

Mathematics Department, Universitat Politècnica de Catalunya, Barcelona, Spain
{josep.fabrega,xavier.munoz}@upc.edu

Abstract. This paper deals with the problem of finding matchings in bipartite regular graphs in the online model with randomized adversary. We determine the competitive ratio for 2-regular graphs for any number of random decisions made by the adversary. We also give upper and lower bounds for other values of the degree.

1 Introduction

An online algorithm solves a computational problem piecewise. Its input arrives as a sequence and each piece of the sequence is a request. The algorithm is required to compute a partial output after each time step, before the next request arrives, i.e., the partial output only depends on the previous requests. This partial output may not be changed. Commonly, competitive analysis is used for analyzing the output quality of online algorithms.

1.1 Different Models for Online Algorithms Used in This Paper

Competitive analysis was introduced in 1985 by Sleator and Tarjan [17] and is a powerful tool to measure the performance of online algorithms. For an overview on competitive analysis and online algorithms, we refer to the standard literature [3, 8, 10, 13].

Let us consider an online maximization problem p , and a deterministic algorithm A that solves that problem. Let I be an adversarial instance for the problem p presented piecewise to A . We name $\text{Cost}_p(A, I)$ the cost of the solution computed by A on the instance I of p . We measure the performance of A on I comparing its cost to the cost of an algorithm performing optimally on I . We name such an algorithm Opt and the cost of its solution $\text{Cost}_p(\text{Opt}, I)$. The *strict competitive ratio* of an algorithm A on the instance I is then

$$r_p(A, I) = \frac{\text{Cost}_p(\text{Opt}, I)}{\text{Cost}_p(A, I)}.$$

Observe that, as p is a maximization problem, the cost of A is always going to be smaller than the cost of Opt on any instance. This means that $r_p(A, I) \geq 1$.

Now, the competitive ratio of an algorithm A for any possible instance $I \in \mathcal{I}$ is defined in a worst-case manner as

$$r_p(A) = \sup_{I \in \mathcal{I}} (r_p(A, I)).$$

Finally, we define the competitive ratio of a problem p as the best case, i.e.,

$$r_p = \inf_{A \in \mathcal{A}} (r_p(A))$$

where \mathcal{A} is the set of deterministic online algorithms solving problem p .

Let RA be a randomized online algorithm that solves p . It can be modeled as a set $RA = \{A_1, A_2, \dots, A_k\}$ of deterministic algorithms one of which will be picked uniformly at random. The competitive ratio of a randomized algorithm RA on the instance I is defined as

$$\bar{r}_p(RA, I) = \frac{1}{\mathbb{E}(\frac{1}{r_p(RA, I)})},$$

and the competitive ratio of a randomized algorithm on p is defined analogously as

$$\bar{r}_p(RA) = \sup_{I \in \mathcal{I}} \{\bar{r}_p(RA, I)\}$$

and

$$\bar{r}_p = \inf_{RA \in \mathcal{RA}} \{\bar{r}_p(RA)\},$$

respectively.

The definition of the competitive ratio for a randomized algorithm may sound a little odd. The reason of such a definition is that the competitive ratio is typically defined for minimization problems instead of maximization problems. In case of minimization the competitive ratio is defined as $r_p(A, I) = \text{Cost}_p(A, I) / \text{Cost}_p(\text{Opt}, I)$ (i.e., the inverse of the competitive ratio for maximization). In order to have similar behaviors in minimization and maximization problems with randomization, it is more convenient to use that definition.

The randomized adversary model we use here was introduced by Hromkovič (see [18]) and further investigated by Burjons et al. [4] and Dobrev et al. [7].

In this model, the adversary has access to s random bits, and uses them to pick uniformly at random an instance from the set $RI = \{I_1, I_2, \dots, I_{2^s}\}$ for the problem p . The online algorithm A is deterministic and knows in advance all the information from the adversary by means of an oracle, except the outcome of the random bits used by the adversary. This means that the algorithm knows the full set of instances RI but does not know which of the instances will be chosen by the adversary. The competitive ratio of A with a randomized adversary is defined as

$$r_p^*(A, RI) = \frac{1}{\mathbb{E}(\frac{1}{r_p(A, RI)})} = \frac{2^s}{\sum_i \frac{1}{r_p(A, I_i)}}.$$

Analogously to the previous models, the competitive ratio for a specific algorithm A in the class of instance sets \mathcal{RI}_s using no more than s random bits

and the competitive ratio for the problem \mathbf{p} in the same set of instances can be defined as

$$\mathbf{r}_p^*(\mathbf{A}, s) = \sup_{\mathbf{RI} \in \mathcal{R}\mathcal{I}_s} \{\mathbf{r}_p^*(\mathbf{A}, \mathbf{RI})\}$$

and

$$\mathbf{r}_p^*(s) = \inf_{\mathbf{A} \in \mathcal{A}} \{\mathbf{r}_p^*(\mathbf{A}, s)\},$$

respectively.

Using the introduced notation, Yao’s principle for online algorithms reads as follows:

Theorem 1 (Yao [19]). *Let \mathbf{p} be an online maximization problem. Then, for any $s \in \mathbb{N}$,*

$$\mathbf{r}_p^*(s) \leq \bar{\mathbf{r}}_p.$$

This theorem states that the competitive ratio of a problem \mathbf{p} solved using any randomized algorithm against instances of finite length is bounded from below by the competitive ratio of a problem with a randomized adversary using any finite number of random bits s to select amongst finite instances.

In order to simplify the expressions and the formulae throughout the paper, let us define the *inverse competitive ratio* ρ as

$$\rho_p(\mathbf{A}, \mathbf{I}) = \frac{1}{\mathbf{r}_p(\mathbf{A}, \mathbf{I})}.$$

It is easy to check that, with this parameter,

$$\rho_p^*(\mathbf{A}, \mathbf{RI}) = \mathbb{E}(\rho_p(\mathbf{A}, \mathbf{RI})) = \frac{\sum_i \rho_p(\mathbf{A}, \mathbf{I}_i)}{2^s}.$$

1.2 Online Matching in Regular Graphs with Randomized Adversary

In the online bipartite matching problem, we are given a bipartite graph, $G = (U \cup W, E)$. The online algorithm processes the input as follows. The vertices in U are *known*; the vertices in W arrive *one at a time*, together with their *list of adjacent vertices*, and each edge must be matched (or not matched) exactly at the time it arrives. The goal is to obtain a matching as close as possible to the optimum one.

First results on this problem are due to the seminal paper by Karp, Vazirani, and Vazirani [12]. Since then, several variants of the problem have been studied, some of them motivated by the following applications: adwords [14, 16], matching in metric space [15], weighted matching in metric spaces [11], matching on the real line [9].

The competitive ratio of deterministic algorithms for the online problem is 2 [12], as any algorithm that always matches a vertex if possible constructs a maximal matching. Also, given any deterministic algorithm, it is easy to construct an input that forces the algorithm to find a matching of size no more than

half of the optimum. It is also proven in [12] that choosing an edge uniformly at random every time a vertex arrives does not provide significant improvement. Indeed, the expected size of the matching is bounded by $\frac{n}{2} + O(\log n)$.

Several randomized algorithms for online bipartite matching have been proposed. The first known is an algorithm called RANKING [12]. It initially ranks the known vertices, so that, each time a new vertex arrives, it is matched to the highest ranked available vertex that is adjacent to it. The competitive ratio is proven to be $1 + \frac{1}{e-1}$. In [2], the authors revisit the proof and give a simpler one.

The online matching problem for the case of regular bipartite graphs has recently been studied in the deterministic model [1, 16] and also in the advice complexity model [1].

In this paper, we deal with the online bipartite matching problem in regular graphs under the randomized adversary model.

Recall that any regular bipartite graph, $G(U \cup W, E)$, has a perfect matching (see [6, Chap. 2]), and therefore an optimum matching will contain $n = |U| = |W|$ edges.

Along this paper, in order to count the size of a matching, we may sometimes refer to a *matched vertex* $u \in U$ standing for a matched edge incident to u .

2 Online Bipartite Matchings in 2-Regular Bipartite Graphs

In this section, we determine the competitive ratio for the online matching problem for the case of 2-regular graphs with any number of random bits used by the adversary.

Proposition 1. *Let μ_2 be the problem of finding a matching in a 2-regular bipartite graph. Then, for any number of random bits s , $r_{\mu_2}^*(s) \geq 8/7$.*

Proof. Take first $s = 1$ and consider the set \mathcal{RI} constituted by the following two bipartite graphs with eight vertices (see Fig. 1), $G_1 = (U \cup W, E_1)$ and $G_2 = (U \cup W, E_2)$, where $U = \{u_1, u_2, u_3, u_4\}$, $W = \{w_1, w_2, w_3, w_4\}$,

$$E_1 = \{u_1w_1, u_1w_3, u_2w_1, u_2w_4, u_3w_2, u_3w_3, u_4w_2, u_4w_4\}$$

and

$$E_2 = \{u_1w_1, u_1w_3, u_2w_1, u_2w_4, u_3w_2, u_3w_4, u_4w_2, u_4w_3\}.$$

The adversary selects at random one of the two graphs and presents the vertices of W in the order w_1, w_2, w_3, w_4 . Notice that, before presenting vertex w_3 , both instances are indistinguishable.

It is easy to check that, for any choice of the algorithm in the first two rounds, there will always be exactly one round, either in G_1 or in G_2 , in which no edge can be selected. Therefore, the competitive ratio of the algorithm is $4/3$ for one of the instances and 1 for the other. This proves that, for any online algorithm A ,

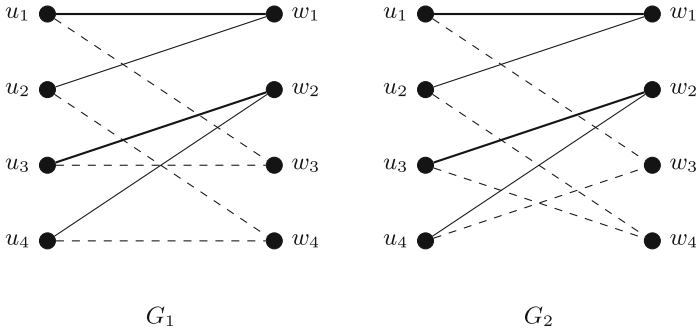


Fig. 1. The two adversarial instances in the set \mathcal{RI} in the lower bound for $s = 1$ with the two first edges selected.

$$r_{\mu_2}^*(A, s = 1) \geq r_{\mu_2}^*(s = 1) = \frac{2}{3/4 + 1} = \frac{8}{7},$$

as claimed.

For a general value of the number of random bits, $s \in \mathbb{N}$, let \mathcal{RI}' be a set of size 2^s from which the adversary chooses the instance to be presented, and assume that \mathcal{RI}' is formed by 2^{s-1} copies of G_1 and 2^{s-1} copies of G_2 . Then,

$$r_{\mu_2}^*(A, s) \geq r_{\mu_2}^*(s = 1) \geq \frac{8}{7},$$

which finishes the proof. □

Proposition 2. *The competitive ratio for the bipartite matching problem for 2-regular graphs, μ_2 , in the randomized algorithm model is $\bar{r}_{\mu_2} \leq \frac{8}{7}$.*

Proof. Let us consider an instance $I(G)$ presented by the adversary, with $G = (U \cup W, E)$. Every new vertex $w \in W$ shown by the adversary appears together with two edges u_1w and u_2w and the algorithm selects one of them for the matching whenever it is possible. We can assume that the algorithm is greedy, since not choosing an edge when it is possible will never produce a better solution than choosing a wrong edge (a wrongly chosen edge will produce at most one error).

To make the proof clearer, we transform the problem as follows: The instance $I(G)$ will be represented by $I'(G')$ with $G' = (U, E')$. Every new vertex $w \in W$ appearing in the original problem together with edges $u_1w, u_2w \in E(G)$ will be represented by a new edge $u_1u_2 \in E'(G')$. It is clear that, if G is a 2-regular bipartite graph, G' will be a set of disjoint cycles.

The algorithm choosing one of the edges $u_1w, u_2w \in E(G)$ when vertex $w \in G$ appears is translated to the assignment of a direction to the edge $u_1u_2 \in E'(G')$. The new arc will be adjacent to $u_1 \in U(G')$ if the selected edge in the original problem is $u_1w \in E(G)$ (see Fig. 2). The algorithm should assign a direction to every new edge, in such a way that the newly built directed graph maintains an in-degree not larger than 1.

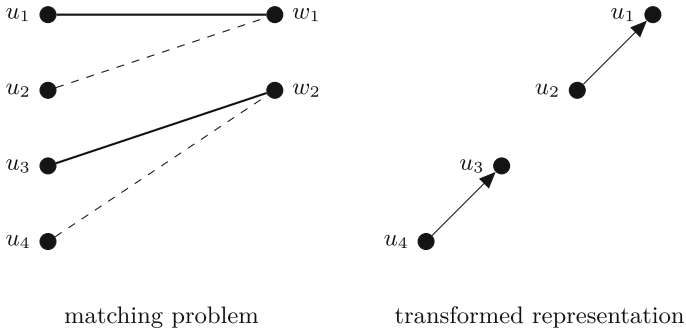


Fig. 2. Translation of the matching problem for $\delta = 2$

Notice that, if the adversary shows a new edge incident to an already shown path, the algorithm can always assign to that edge the direction of the path and no error will be produced. Therefore, we can assume that the adversary will always present disjoint edges as long as it is possible, and this will be the worst-case adversary.

We can further assume (for the sake of simplicity in the analysis) that the adversary will consist of two phases: In the first one, a set of disjoint edges will be presented. In the second phase, the edges presented will be incident to vertices that have already appeared in the first phase, closing cycles of even length, say $2m$.

Any randomized algorithm RA will choose at random the directions of the edges in the first phase. In the second phase, no decision has to be taken: If an edge u_1u_2 appears in the second phase, either u_1 and u_2 have already in-degree 1, and no direction can be assigned (producing an error), or at least one of them has in-degree 0, and it is possible to assign the direction towards that vertex.

Therefore, all mistakes are produced during the random choices in the first phase. This problem can be translated again into the following game: Given a cycle C_m , let the algorithm choose at random the direction of each edge in the cycle. The number of errors is the number of vertices with in-degree 2. That number of errors is equal to the number of edges missing in the matching in a bipartite graph with $2m$ vertices in each vertex set.

We claim that the expected number of vertices with in-degree 2 in a digraph obtained by orienting uniformly at random the edges of the cycle C_m is $m/4$: Indeed, let v_1, \dots, v_m be the vertices of C_m and let N be the number of vertices with in-degree 2 in a random orientation of its edges. Then $N = \sum_{i=1}^m I_i$, where I_i is the indicator random variable of the event $\delta^+(v_i) = 2$. Clearly, $\mathbb{E}(I_i) = \mathbb{P}(I_i = 1) = 1/4$. Hence, $\mathbb{E}(N) = \sum_{i=1}^m \mathbb{E}(I_i) = m/4$, as claimed.

The back-translation of this result into the original matching problem states that the number of missed edges in expectation in the matching is $m/4$ out of $2m$ possible, and therefore $\bar{\rho}_{\mu_2}(\text{RA}) = \frac{2m - m/4}{2m} = \frac{7}{8}$. Finally, $\bar{\mathbf{r}}_{\mu_2} \leq \frac{1}{\bar{\rho}_{\mu_2}(\text{RA})} = \frac{8}{7}$. \square

As a consequence of Propositions 1 and 2 and Theorem 1, we conclude:

Theorem 2. *The competitive ratio for the bipartite matching problem μ_2 for 2-regular graphs with randomized adversary for any number of random bits s is given by $\mathbf{r}_{\mu_2}^*(s) = 8/7$.*

3 Adversary with One Random Bit for Other Values of δ

3.1 Lower Bounds on the Competitive Ratio

Proposition 3. *Let μ_3 be the problem of finding a matching in a 3-regular bipartite graph with a randomized adversary. Then $\mathbf{r}_{\mu_3}^*(s = 1) \geq 18/17$.*

Proof. We will prove that there exists an adversary using one random bit for which, for any online algorithm, $\mathbf{r}_{\mu_3}^*(\mathbf{A}, s = 1) \geq 18/17$. In order to define such an adversary, let us use a result borrowed from the theory of combinatorial designs. A *balanced incomplete block design* (BIBD for short) is a pair (V, \mathcal{B}) where V is a v -set of points, and \mathcal{B} is a collection of b k -subsets of V (blocks) such that each element of V is contained in exactly r blocks and any 2-subset of V is contained in exactly λ blocks. The numbers v, b, r, k , and λ are parameters of the BIBD. A *parallel class* or *resolution class* in a design is a set of blocks that partition the point set. A BIBD is said to be *resolvable* if its blocks can be partitioned into parallel classes. The notation RBIBD(v, k, λ) is commonly used for a resolvable balanced incomplete block design (see [5] for more information on resolvable designs).

Let us consider the resolvable design RBIBD(9, 3, 1) given by

$$\begin{array}{cccc}
 I & II & III & IV \\
 \{1, 2, 3\} & \{1, 4, 7\} & \{1, 5, 9\} & \{1, 6, 8\} \\
 \{4, 5, 6\} & \{2, 5, 8\} & \{2, 6, 7\} & \{2, 4, 9\} \\
 \{7, 8, 9\} & \{3, 6, 9\} & \{3, 4, 8\} & \{3, 5, 7\}.
 \end{array}$$

Observe that each column is a parallel class, i.e., a partition of the set $\{1, \dots, 9\}$.

Let the adversary present one instance out of two possible bipartite graphs $G_0 = (U \cup W, E_0)$ and $G_1 = (U \cup W, E_1)$, with $U = \{1, \dots, 9\}$ and $W = \{w_1, \dots, w_9\}$. The edge sets E_0 and E_1 are given by the blocks in the design shown above as follows: Let the blocks in the parallel classes I and II be the neighbors of vertices w_1, \dots, w_6 , respectively, in both graphs, and let the neighbors of vertices w_7, w_8 and w_9 be the blocks in class III in G_0 , and the blocks in class IV in G_1 . Let the adversary present the vertices w_1, \dots, w_9 in increasing order.

In order to analyze the performance of this algorithm, let us consider two phases:

- **Phase I.** During this phase, vertices w_1, \dots, w_6 are presented. At the end of this phase, 6 edges incident to different vertices in U have been selected for the matching. Notice that both instances are indistinguishable at this point.

- **Phase II.** During this phase, the adversary presents vertices w_7, w_8 and w_9 . At this point, the algorithm knows which of both instances was selected by the adversary, and the algorithm should select three edges incident to the three vertices in U not used during the first phase. Let U_m be the set of those three edges.

Let us show that it is not possible to match all vertices in U_m during the second phase in both instances. Let us recall that U_m cannot be a block in classes I or II (if it were, no edge would have been selected from that block). Therefore there are at least two vertices in U_m that are not in any block in classes I or II . Because of the properties of block designs, every pair of elements belongs to exactly λ blocks (in our case $\lambda = 1$), and therefore these two vertices belong to the same block either in class III or IV . Due to the pigeonhole principle, it will not be possible to select all three vertices in the instance corresponding to the class where the pair is in the same block. Therefore, at least one edge is missed in one of the instances. The bound for the inverse competitive ratio follows straightforwardly. \square

Notice that this adversary is a generalization of the adversary shown for $\delta = 2$ (the adversary in the proof for $\delta = 2$ is actually an RBIBD(4, 2, 1)).

The next proposition shows a generalization of the adversary for $\delta = 2$ to adversaries for any even degree.

Proposition 4. *Let $\mu_{2\delta}$ be the problem of finding a matching in a 2δ -regular bipartite graph. Then $r_{\mu_{2\delta}}^*(s = 1) \geq \frac{8\delta}{8\delta - 1}$.*

Proof. Let us consider the graphs G_1 and G_2 used in the case $\delta = 2$ (Fig. 1) and let us construct the Cartesian product $G'_i = G_i \times \{1, \dots, \delta\}$, i.e., $V(G'_i) = \{(v, j) \mid v \in G_i \text{ and } j \in \{1 \dots \delta\}\}$ and (v_a, j_1) is adjacent to (v_b, j_2) in G'_i whenever $v_a v_b$ is an edge in G_i . Two instances are given by presenting the vertices of the graphs G'_1 and G'_2 in the order

$$(w_1, 1), \dots, (w_1, \delta), (w_2, 1), \dots, (w_2, \delta), (w_3, 1), \dots, (w_3, \delta), (w_4, 1), \dots, (w_4, \delta).$$

It is a simple exercise to check that the claimed bound holds by using the same reasoning as in the proof of Proposition 1. \square

3.2 Upper Bounds on the Competitive Ratio

Proposition 5. *Let μ_δ be the problem of finding a matching in a δ -regular bipartite graph with a randomized adversary using a single random bit. Then $r_{\mu_\delta}^*(s = 1) \leq \frac{4\delta}{3\delta + 1}$.*

Proof. We prove that, for any adversary using one random bit, there exists an online algorithm A solving μ_δ for a δ -regular bipartite graph, for which the inverse competitive ratio satisfies $\rho_A(s = 1) \geq \frac{3\delta + 1}{4\delta}$.

Since the adversary is using one random bit, there are two possible instances I_0, I_1 that can each be presented with probability $\frac{1}{2}$. Let the algorithm start

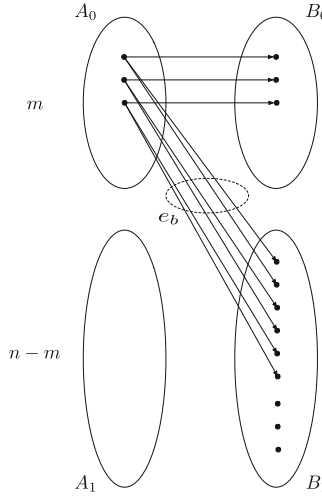


Fig. 3. Counting the number of edges when an error is detected

taking edges in the matching assuming it is one of them (let us suppose it is I_0). If the presented instance is I_0 , then the algorithm will achieve a perfect matching. If the presented instance is I_1 , let us assume that the algorithm will realize the wrong choice after m vertices have appeared. Since no error was detected before, it can be guaranteed that the algorithm will have selected m edges in the matching at this point.

Let $G_1 = (U \cup W, E)$ be the graph presented by the adversary in instance I_1 . Let $A_0 \subset W$ be the subset of vertices in W that have been shown at time step m , and let $B_0 \subset U$ the subset of vertices in U that have been chosen in the matching. $A_1 \subset W$ and $B_1 \subset U$ contain the rest of the vertices of the graph. Let e_b be the number of edges that have already appeared between sets A_0 and B_1 . This situation is illustrated in Fig. 3.

Let us point out the following facts:

- $e_b \leq (n-m)(\delta-1)$, since the vertices of B_1 have degree δ , but there is at least one edge per vertex which is not incident to A_0 . Otherwise an error would have been detected before.
- $e_b \leq (\delta-1)m$, since the number of edges going from A_0 to B_0 is at least m .
- The number of edges going from A_1 to B_0 is also e_b .

At this point, the algorithm knows exactly what is the graph that will be presented by the adversary, and therefore the algorithm can proceed as in an off-line model (since the algorithm discovered that the instance was not I_0 , and then it must be I_1).

Let h be the number of edges that the algorithm may miss to put in the matching. It is easy to see that $h \leq e_b/\delta$: Take the subgraph induced by the vertex sets A_1 and B_1 and add e_b edges so that it becomes δ -regular. Since it is

a δ -regular bipartite graph, it can be edge-colored with δ colors. This coloring is equivalent to the partition of the edge set into δ perfect matchings. Therefore, at least one of those matchings will contain at most e_b/δ edges among the added ones.

Thus, $h \leq \frac{e_b}{\delta} \leq \min\{\frac{\delta-1}{\delta}m, \frac{\delta-1}{\delta}(n-m)\}$. The maximum value of that minimum is obtained when $m = \frac{n}{2}$. For that value of m , we have $h = n - m$, and therefore the number of edges in the matching is $n - h$.

Finally, the inverse competitive ratio in expectation is $\rho \geq \frac{1 + \frac{\delta+1}{2\delta}}{2} = \frac{3\delta+1}{4\delta}$. \square

4 Adversaries with a Large Number of Random Bits

Let us consider the set $\text{RI}_{k\delta}$ formed by the instances $\text{I}_i(G_i)$ in which $G_i = (U \cup W, E)$ is a δ -regular bipartite graph with $n = |U| = |W| = k\delta$. For any instance $\text{I}_i(G_i)$, let the adversary present the vertices $w \in W$ in δ phases. In phase i , $1 \leq i \leq \delta$, a subset of k vertices $F_i, F_i \subset W$, is presented in such a way that $\Gamma(w) \cap \Gamma(w') = \emptyset$, for any two distinct vertices $w, w' \in F_i$, where $\Gamma(v)$ stands for the set of neighbors of vertex v .

In this section, we want to compute the competitive ratio $\mathbf{r}_{\text{RI}_{k\delta}}^*$ for the randomized adversary given by the set $\text{RI}_{k\delta}$. Notice that, with such instance set, all deterministic greedy algorithms will have the same behavior, since at each time step all edges will look the same and any deterministic choice will have the same probability of being a good choice.

Theorem 3. *The competitive ratio for the randomized adversary given by the set $\text{RI}_{k\delta}$ for large values of k is $\mathbf{r}_{\text{RI}_{k\delta}}^* = \frac{\delta}{\delta - g(\delta)}$ with $g(\delta) = \sum_{i=1}^{\delta} (\frac{i-1}{\delta})^{\delta}$.*

Proof. Let $G = (U \cup W, E)$ be the graph presented by the adversary. Let us recall that G is a δ -regular bipartite graph with $n = |U| = |W| = k\delta$. Let the adversary present the vertices $w \in W$ in δ phases. In phase i , $1 \leq i \leq \delta$, a subset of k vertices $F_i, F_i \subset W$, is presented in such a way that $\Gamma(w) \cap \Gamma(w') = \emptyset$, for any two distinct vertices $w, w' \in F_i$.

For every $w \in F_i$, the algorithm tries to match w with a vertex of $\Gamma(w)$. From the point of view of the algorithm the vertices of F_i are selected at random, and each vertex $w \in F_i$ with at least one available neighbor in $\Gamma(w)$ will produce a new matched vertex in U that will not be used in the following phases.

Let us say that a vertex of U is marked if it has already been used by the algorithm to construct the matching. In phase i the algorithm will mark $k - N_i$ vertices of U , with N_i being the random variable that counts the number of vertices $w \in F_i$ such that all the vertices of $\Gamma(w)$ are already marked. Notice that $N_1 = 0$ because k vertices are always marked in the first phase. Moreover, since the total number of vertices of U that have been marked previous to phase i is $\sum_{j=1}^{i-1} (k - N_j)$, $1 \leq i \leq \delta$, we have

$$N_i \leq \left\lfloor \frac{\sum_{j=1}^{i-1} (k - N_j)}{\delta} \right\rfloor. \tag{1}$$

Moreover, $N_i = \sum_{w \in F_i} I_w$, where I_w is the indicator random variable of the event that all the vertices of $\Gamma(w)$ have been previously marked. If $M_i = \sum_{j=1}^{i-1} N_j$, $1 \leq i \leq \delta$, then

$$\mathbb{P}(I_w = 1 \mid M_i = m_i) = \frac{\binom{(i-1)k - m_i}{\delta}}{\binom{k\delta}{\delta}}. \tag{2}$$

(For $i = 1$ one has $M_1 = 0$ and $\mathbb{P}(I_w = 1) = 0$.) Therefore, the expected number of edges that will be missed during phase i , conditioned by that during previous phases (1 to $i - 1$) the number of missed edges is m_i , is given by

$$\begin{aligned} \mathbb{E}(N_i \mid M_i = m_i) &= \sum_{w \in F_i} \mathbb{E}(I_w \mid M_i = m_i) \\ &= \sum_{w \in F_i} \mathbb{P}(I_w = 1 \mid M_i = m_i) = k \frac{\binom{(i-1)k - m_i}{\delta}}{\binom{k\delta}{\delta}}. \end{aligned} \tag{3}$$

Notice that, for $2 \leq i \leq \delta$, the binomial coefficient $\binom{(i-1)k - m_i}{\delta}$ will give a positive contribution only if $m_i \leq (i - 1)k - \delta$, and that, for large values of k , the probability $\mathbb{P}(I_w = 1 \mid M_i = m_i)$ is asymptotically $((i - 1)/\delta)^\delta$, because

$$\frac{\binom{(i-1)k - m_i}{\delta}}{\binom{k\delta}{\delta}} = \prod_{j=1}^{\delta} \frac{(i-1)k - m_i - j + 1}{k\delta - j + 1} = \left(\frac{i-1}{\delta}\right)^\delta \cdot \prod_{j=1}^{\delta} \frac{1 - \frac{m_i + j - 1}{(i-1)k}}{1 - \frac{j-1}{k\delta}}. \tag{4}$$

When the δ phases of the algorithm have been accomplished, the total number of matched vertices is $\sum_{i=1}^{\delta} (k - N_i)$ and the inverse competitive ratio can be expressed as

$$\rho = \frac{\sum_{i=1}^{\delta} (k - N_i)}{k\delta} = 1 - \frac{1}{k\delta} \sum_{i=1}^{\delta} N_i.$$

The inverse competitive ratio ρ is a random variable and we are interested in estimating its expected value

$$\mathbb{E}(\rho) = 1 - \frac{1}{k\delta} \sum_{i=1}^{\delta} \mathbb{E}(N_i).$$

A more detailed analysis of expressions (2) to (4) allows us to write (for $2 \leq i \leq \delta$)

$$\left(1 - \frac{m_i + \delta - 1}{(i-1)k}\right)^\delta \leq \frac{\mathbb{P}(I_w = 1 \mid M_i = m_i)}{((i-1)/\delta)^\delta} \leq \left(\frac{1 - m_i / ((i-1)k)}{1 - (\delta - 1) / (k\delta)}\right)^\delta$$

and

$$\left(1 - \frac{m_i + \delta - 1}{(i-1)k}\right)^\delta \leq \frac{\mathbb{E}(N_i \mid M_i = m_i)}{k((i-1)/\delta)^\delta} \leq \left(\frac{1 - m_i / ((i-1)k)}{1 - (\delta - 1) / (k\delta)}\right)^\delta.$$

Hence, the expression

$$\frac{1}{k\delta} \sum_{i=1}^{\delta} \mathbb{E}(N_i \mid M_1 = 0, M_2 = m_2, \dots, M_{\delta} = m_{\delta}) = \frac{1}{k\delta} \sum_{i=1}^{\delta} \mathbb{E}(N_i \mid M_i = m_i)$$

is bounded from below by

$$a(k, \delta) = \frac{1}{\delta} \sum_{i=2}^{\delta} \left(\frac{i-1}{\delta}\right)^{\delta} \left(1 - \frac{m_i + \delta - 1}{(i-1)k}\right)^{\delta}$$

and bounded from above by

$$b(k, \delta) = \frac{1}{\delta} \sum_{i=2}^{\delta} \left(\frac{i-1}{\delta}\right)^d \left(\frac{1 - m_i / ((i-1)k)}{1 - (\delta - 1) / (k\delta)}\right)^{\delta}.$$

The expected inverse competitive ratio can be calculated as

$$\mathbb{E}(\rho) = 1 - \frac{1}{k\delta} \sum_{i=1}^{\delta} \mathbb{E}(N_i) = 1 - \mathbb{E}\left(\frac{1}{k\delta} \sum_{i=1}^{\delta} \mathbb{E}(N_i \mid M_i)\right).$$

Therefore,

$$l(k, \delta) \leq \mathbb{E}(\rho) \leq u(k, \delta)$$

where

$$l(k, \delta) = 1 - \frac{1}{\delta} \sum_{i=2}^{\delta} \left(\frac{i-1}{\delta}\right)^{\delta} \mathbb{E}\left(\left(\frac{1 - M_i / ((i-1)k)}{1 - (\delta - 1) / (k\delta)}\right)^{\delta}\right)$$

and

$$u(k, \delta) = 1 - \frac{1}{\delta} \sum_{i=2}^{\delta} \left(\frac{i-1}{\delta}\right)^{\delta} \mathbb{E}\left(\left(1 - \frac{M_i + \delta - 1}{(i-1)k}\right)^{\delta}\right).$$

From (1), we deduce that

$$M_i = \sum_{j=1}^{i-1} N_j \leq \sum_{j=1}^{i-1} \frac{(j-1)k}{\delta} = \frac{(i-1)(i-2)}{2\delta}.$$

Therefore, $u(k, \delta) \leq u'(k, \delta)$ where

$$u'(k, \delta) = 1 - \frac{1}{\delta} \sum_{i=2}^{\delta} \left(\frac{i-1}{\delta}\right)^{\delta} \left(1 - \frac{(i-1)(i-2)/2\delta + \delta - 1}{(i-1)k}\right)^{\delta}$$

and

$$u'(k, \delta) \longrightarrow 1 - \frac{1}{\delta} \sum_{i=1}^{\delta} \left(\frac{i-1}{\delta}\right)^{\delta} \text{ as } k \longrightarrow \infty.$$

Similarly, since $M_i \geq 0$, we have $l(k, \delta) \geq l'(k, \delta)$ where

$$l'(k, \delta) = 1 - \frac{1}{\delta} \sum_{i=2}^{\delta} \left(\frac{i-1}{\delta}\right)^{\delta} \left(\frac{1}{1 - (\delta-1)/(k\delta)}\right)^{\delta},$$

and, also,

$$l'(k, \delta) \longrightarrow 1 - \frac{1}{\delta} \sum_{i=1}^{\delta} \left(\frac{i-1}{\delta}\right)^{\delta} \text{ as } k \longrightarrow \infty.$$

Finally, since $l'(k, \delta) \leq \mathbb{E}(\rho) \leq u'(k, \delta)$, we conclude that

$$\lim_{k \rightarrow \infty} \mathbb{E}(\rho) = 1 - \frac{g(\delta)}{\delta}$$

with

$$g(\delta) = \sum_{i=1}^{\delta} \left(\frac{i-1}{\delta}\right)^{\delta}.$$

The value for the competitive ratio $\mathbf{r}_{\text{RI}_{k\delta}}^*$ is obtained by taking into account that $\mathbf{r}_{\text{RI}_{k\delta}}^* = \frac{1}{\mathbb{E}(\rho)}$. □

Theorem 3 shows the competitive ratio for the online bipartite matching problem for regular graphs with a given randomized adversary choosing between $O(n!^2)$ instances. This is therefore a lower bound on the competitive ratio of the matching problem for regular bipartite graphs for any randomized adversary.

It can be shown that $g(m)$ is a monotonically increasing function and that

$$\lim_{m \rightarrow \infty} g(m) = \lim_{m \rightarrow \infty} \sum_{i=1}^m \left(\frac{i-1}{m}\right)^m = \frac{1}{e-1}.$$

5 Conclusions

We have studied the online bipartite matching problem for regular graphs with a randomized adversary. We have determined the competitive ratio for the case $\delta = 2$ for any number of random bits. This turns out to be an interesting example of a problem in which one random bit used by the adversary makes the problem as hard as for many random bits. It is also an example in which the competitive ratio for the randomized adversary model is equal to the ratio for the randomized algorithm model. We also gave upper and lower bounds for other values of the degree for adversaries using one random bit.

Finally, we studied the case for adversaries with a large number of random bits and found the competitive ratio for a given class of instances. We were not able to find an upper bound for the competitive ratio in the general case, but we are convinced that the exact value for the competitive ratio should not be far from the given result.

References

1. Barrière, L., Muñoz, X., Fuchs, J., Unger, W.: Online matching in regular bipartite graphs. *Parallel Process. Lett.* **28**(2) (2018). <https://doi.org/10.1142/S0129626418500081>
2. Birnbaum, B., Mathieu, C.: On-line bipartite matching made simple. *ACM SIGACT News* **39**(1), 80 (2008)
3. Borodin, A., El-Yaniv, R.: *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge (1998)
4. Burjons, E., Hromkovič, J., Muñoz, X., Unger, W.: Online graph coloring with advice and randomized adversary (extended abstract). In: Freivalds, R.M., Engels, G., Catania, B. (eds.) *SOFSEM 2016*. LNCS, vol. 9587, pp. 229–240. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49192-8_19
5. Colbourn, Ch., Dinitz, J.: *Handbook on Combinatorial Designs*, 2nd edn. Chapman & Hall/CRC, London (2007)
6. Diestel, R.: *Graph Theory*, 3rd edn. Springer, Heidelberg (2005)
7. Dobrev, S., Hromkovič, J., Komm, D., Kráľovič, R., Kráľovič, R., Mömke, T.: The complexity of paging against a probabilistic adversary. In: Freivalds, R.M., Engels, G., Catania, B. (eds.) *SOFSEM 2016*. LNCS, vol. 9587, pp. 265–276. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49192-8_22
8. Fiat, A., Woeginger, G.J. (eds.): *Online Algorithms – The State of the Art*. LNCS, vol. 1442. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0029561>
9. Fuchs, B., Hochstättler, W., Kern, W.: Online matching on a line. *Electron. Notes Discret. Math.* **13**, 49–51 (2003)
10. Irani, S., Karlin, A.R.: On online computation. In: Hochbaum, D. (ed.) *Approximation Algorithms for NP-Hard Problems*, Chap. 13, pp. 521–564. PWS Publishing Company, Boston (1997)
11. Kalyanasundaram, B., Pruhs, K.: Online weighted matching. *J. Algorithms* **14**(3), 478–488 (1993)
12. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 352–358 (1990)
13. Komm, D.: *An Introduction to Online Computation - Determinism, Randomization, Advice*. Springer, Berlin (2016). <https://doi.org/10.1007/978-3-319-42749-2>
14. Mehta, A., Saberi, A., Vazirani, U.V., Vazirani, V.V.: AdWords and generalized online matching. *J. ACM* **54**(5), 1–19 (2007)
15. Meyerson, A., Nanavati, A., Poplawski, L.: Randomized online algorithms for minimum metric bipartite matching. In: *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 954–959 (2006)
16. Naor, J., Wajc, D.: Near-optimum online ad allocation for targeted advertising. In: *Proceedings of the Sixteenth ACM Conference on Economics and Computation (EC 2015)*, pp. 131–148 (2015)
17. Sleator, D.D., Tarjan, R.E.: Amortized efficiency of list update and paging rules. *Commun. ACM* **28**(2), 202–208 (1985)
18. Smula, J.: *Information content of online problems - advice versus determinism and randomization*. Ph.D. thesis, ETH Zürich (2015)
19. Yao, A.: Probabilistic computations: toward a unified measure of complexity. In: *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 222–227 (1977)