Greedy Online Bipartite Matching on Random Graphs*

Andrew Mastin[†], Patrick Jaillet[‡]

November 8, 2013

Abstract

We study the average performance of online greedy matching algorithms on $G(n, n, p)$, the random bipartite graph with n vertices on each side and edges occurring independently with probability $p = p(n)$. In the online model, vertices on one side of the graph are given up front while vertices on the other side arrive sequentially; when a vertex arrives its edges are revealed and it must be immediately matched or dropped. We begin by analyzing the OBLIVIOUS algorithm, which tries to match each arriving vertex to a random neighbor, even if the neighbor has already been matched. The algorithm is shown to have a performance ratio of at least $1 - 1/e$ for all monotonic functions $p(n)$, where the performance ratio is defined asymptotically as the ratio of the expected matching size given by the algorithm to the expected maximum matching size. Next we show that the conventional GREEDY algorithm, which assigns each vertex to a random unmatched neighbor, has a performance ratio of at least 0.837 for all monotonic functions $p(n)$. Under the $G(n, n, p)$ model, the performance of GREEDY is equivalent to the performance of the well known RANKING algorithm, so our results show that RANKING has a performance ratio of at least 0.837. We finally consider vertex-weighted bipartite matching. Our proofs are based on simple differential equations that describe the evolution of the matching process.

2010 *Mathematics subject classification*: Primary 05C70
Secondary 05C80, 05C85, 68W40

1 Introduction

In the online bipartite matching problem, we are given a bipartite graph $G = (I, J, E)$ where I is a set of n bins and J is a set of n balls. Balls arrive in an online fashion; when a ball $j \in J$ arrives, its edges are revealed and it must either be matched with an unmatched neighboring bin or dropped (left unmatched). Each bin may be matched to at most one ball and decisions are irrevocable. The goal is to maximize the number of matched balls.

The problem has received significant attention due to applications in Internet advertising as well as under streaming models of computation, which places limits on memory utilization for processing large datasets [16, 19, 9] . From a worst-case perspective, it is well known that the GREEDY algorithm, which matches each ball to a random unmatched neighboring bin (if possible), always achieves a matching size that is at least as large as 1/2 of the size of the maximum matching. The RANKING algorithm of Karp, Vazirani, and Vazirani picks a random permutation of bins up front and matches each ball to its unmatched neighboring bin that is ranked highest in the permutation [13]. RANKING guarantees a

*Research supported in part by NSF grant 1029603 and ONR grant N00014-12-1-0033. The first author is supported in part by a NSF graduate research fellowship.

[†]Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; mastin@mit.edu

[‡]Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science and Operations Research Department, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; jaillet@mit.edu

matching size at least $1 - 1/e$ of the size of the maximum matching in expectation, which is the best possible worst-case guarantee.

Surprisingly, the average-case performance of these algorithms has been largely overlooked. In this paper we study the performance of greedy-type algorithms under the conventional $G(n, n, p)$ model, where we are given a bipartite graph with n vertices on each side and where each potential edge $(i, j) \in I \times J$ occurs independently with probability p. We are interested in the asymptotic behavior of the matching sizes obtained by these algorithms for monotonic functions $p = p(n)$. We will focus on functions of the form $p(n) = c/n$ for some constant $c > 0$, which we will show is the region where greedy algorithms are most sensitive to $p(n)$.

We start by analyzing the OBLIVIOUS algorithm, which has knowledge of a ball's edges when it arrives but does not know which neighboring bins are occupied. The algorithm picks a random neighboring bin; if the bin is unmatched, then the ball is matched, otherwise it is dropped. This rule models load balancing scenarios where a central server knows which machines can process a given job, but is not aware of the machine availability. Next we analyze GREEDY and RANKING, showing that under the $G(n, n, p)$ model, the performance of the two algorithms is equivalent. Finally, we extend our analysis to the vertex-weighted matching problem, where each bin has a weight and the greedy algorithm matches each ball to the available neighboring bin with largest weight.

Our proofs employ differential equations that describe the evolution of the number of matched bins throughout the arrival process. The differential equations are easily obtained by calculating the expected change in number of occupied bins upon the arrival of each ball. As $n \to \infty$, this gives a deterministic differential equation and solution. The solution is shown to closely model the stochastic behavior due to a general result of Wormald [21]. We believe that this is one of the most simple applications of Wormald's theorem. In fact, our proofs are nearly as simple as the worst-case proof for RANKING of Devanur, Jain, and Kleinberg [5].

In the remainder of this section we briefly state our main results for the OBLIVIOUS and GREEDY algorithms and discuss related work. The analysis of OBLIVIOUS and GREEDY is given in Section 2. We extend the analysis to vertex-weighted matching in Section 3 and a conclusion is given in Section 4.

1.1 Results. Let $\mu_{\mathrm{A}}(n, n, p(n))$ denote the matching size (number of matched balls) obtained by some algorithm A on the graph $G(n, n, p)$ where $p = p(n)$; likewise define the maximum matching size as $\mu_*(n, n, p(n))$. Define a *valid* function $p(n)$ as a function that for all $n > 0$, is monotonic and satisfies $0 < p(n) < 1$ (thus implying that $0 \leq \lim_{n\to\infty} p(n) \leq 1$). We show expressions for the matching sizes obtained by the algorithms that hold asymptotically almost surely (a.a.s., meaning with probability $1 - o(1)$), as well as bounds on the performance ratio. Define the performance ratio for algorithm A given $p(n)$ as

$$\mathcal{R}_{\mathrm{A}}(p(n)) := \lim_{n\to\infty} \frac{\mathbb{E}[\mu_{\mathrm{A}}(n, n, p(n))]}{\mathbb{E}[\mu_*(n, n, p(n))]}. \tag{1.1}$$

The performance ratio is essentially the analog of the competitive ratio adapted for average-case analysis. Throughout the rest of the paper, we define $\gamma_* = \gamma_*(c)$ as the smallest root of the equation $x = c\exp(-ce^{-x})$, and $\gamma^* = \gamma^*(c) = ce^{-\gamma_*}$. Our results for OBLIVIOUS and GREEDY are listed as follows.

THEOREM 1. *Let $\mu_{\mathrm{O}}(n, n, c/n)$ denote the matching size obtained by the* OBLIVIOUS *algorithm on the graph $G(n, n, p)$, where $p = c/n$ and $c > 0$ is a constant. Then a.a.s.,*

$$\frac{\mu_{\mathrm{O}}(n, n, c/n)}{n} = 1 - e^{(e^{-c}-1)} + o(1). \tag{1.2}$$

THEOREM 2. *The performance ratio $\mathcal{R}_{\mathrm{O}}(p(n))$ of* OBLIVIOUS *on $G(n,n,p)$, for all valid functions $p = p(n)$, satisfies*

$$\begin{aligned} \mathcal{R}_{\mathrm{O}}(p(n)) &= 1, & p(n) &= o(1/n), \\ \mathcal{R}_{\mathrm{O}}(p(n)) &\geq \frac{c - ce^{(e^{-c}-1)}}{2c - (\gamma^* + \gamma_* + \gamma^*\gamma_*)}, & p(n) &= c/n, \\ \mathcal{R}_{\mathrm{O}}(p(n)) &\geq 1 - \tfrac{1}{e}, & p(n) &= \omega(1/n). \end{aligned} \tag{1.3}$$

COROLLARY 1. *The performance ratio of* OBLIVIOUS *on $G(n,n,p)$ for all valid functions $p = p(n)$ satisfies*

$$\mathcal{R}_{\mathrm{O}}(p(n)) \geq 1 - \frac{1}{e}. \tag{1.4}$$

THEOREM 3. *Let $\mu_{\mathrm{G}}(n,n,c/n)$ denote the matching size obtained by the* GREEDY *algorithm on the graph $G(n,n,p)$, where $p = c/n$ and $c > 0$ is a constant. Then a.a.s.,*

$$\frac{\mu_{\mathrm{G}}(n,n,c/n)}{n} = 1 - \frac{\log(2 - e^{-c})}{c} + o(1). \tag{1.5}$$

THEOREM 4. *The performance ratio $\mathcal{R}_{\mathrm{G}}(p(n))$ of* GREEDY *on $G(n,n,p)$, for all valid functions $p = p(n)$, satisfies*

$$\begin{aligned} \mathcal{R}_{\mathrm{G}}(p(n)) &= 1, & p(n) &= o(1/n), \\ \mathcal{R}_{\mathrm{G}}(p(n)) &\geq \frac{c - \log(2 - e^{-c})}{2c - (\gamma^* + \gamma_* + \gamma^*\gamma_*)}, & p(n) &= c/n, \\ \mathcal{R}_{\mathrm{G}}(p(n)) &= 1, & p(n) &= \omega(1/n). \end{aligned} \tag{1.6}$$

COROLLARY 2. *The performance ratio of* GREEDY *on $G(n,n,p)$ for all valid functions $p = p(n)$ satisfies*

$$\mathcal{R}_{\mathrm{G}}(p(n)) \geq 0.837. \tag{1.7}$$

1.2 Related work. The online matching problem was originally analyzed by Karp, Vazirani and Vazirani, who introduced the RANKING algorithm and showed that it obtains a competitive ratio of $1 - 1/e$ [13]. Simpler proofs of the RANKING algorithm have since been found [5, 3]. A $1 - 1/e$ competitive algorithm is also known for vertex-weighted online bipartite matching, which was given by Aggarwal, Goel, Karande, and Mehta [1].

The use of deterministic differential equations to model random processes was first studied by Kurtz, who gave a general purpose theorem for continuous-time jump Markov processes [15]. A discrete-time theorem tailored for random graphs was given by Wormald, which we use in this paper [20, 21]. The differential equation method has been used to study a variety of graph properties including k-cores, independent sets, and greedy packing on hypergraphs [18, 20, 21]. It was also used to analyze a load balancing scenario similar to ours by Mitzenmacher [17].

Early studies of matchings on random graphs focused on determining the existence of perfect matchings; Erdős and Rényi showed that the threshold for the existence of a perfect matching occurs when the graph is likely to have a minimum degree of one [7, 8]. One of the first studies of greedy

matchings on random graphs, as well as the first to employ the differential equation method (specifically via Kurtz's theorem), was the work of Karp and Sipser [12, 15]. They considered ordinary sparse graphs, specifically the $G(n,p)$ model[1] with $p = c/n$. The Karp-Sipser algorithm first matches all vertices with degree one until there are no such vertices remaining, and then obtains matches by selecting random edges. This results in a matching size that is within $1 - o(1)$ of the maximum matching. The algorithm was studied more in depth by Aronson, Frieze, and Pittel, who gave sharper error bounds on performance [2].

Simpler greedy matching algorithms for ordinary graphs were studied by Dyer, Frieze, and Pittel [6]. Again for the $G(n,p)$ model with $p = c/n$, they looked at the greedy algorithm that simply picks random edges, as well as what they refer to as MODIFIED GREEDY, which first picks a random vertex and then selects a random connected edge. They showed that as $n \to \infty$, the matching sizes obtained by the two algorithms follow a normal distribution with mean and variance determined by c. It is interesting to note that the expected fraction of matched vertices for their MODIFIED GREEDY algorithm is exactly the result of our GREEDY algorithm with a multiplicative factor of 1/2. Indeed, MODIFIED GREEDY is a natural analog of GREEDY for ordinary graphs, although their result was obtained using a combinatorial approach completely different from our method.

2 Analysis

Let the graph $G(n,n,p)$ be generated by the following discrete random graph process. Initially at $t = 0$, we are given the set of bin vertices I, where $|I| = n$. At each time step $t > 0$, a ball vertex $j \in J$ arrives and all of its neighboring edges are revealed. Each neighboring edge occurs with a probability $p(n)$ independently of all other edges. After n steps an instance of the graph $G(n,n,p)$ is obtained.

We introduce Wormald's theorem using notation consistent with [21]. Consider a discrete-time process with the probability space Ω, which has elements $(q_0, q_1, \ldots, q_T)$, where each q_k, $0 \le k \le T$, takes values in a set S. A sequence $h_t = (q_0, q_1, \ldots, q_t)$ up to time t is referred to as the history of the process; the random variable for the history is denoted by H_t. Define a sequence of random processes over $n = 1, 2, \ldots$, so that we have $S = S^{(n)}$ and $q_t = q_t^{(n)}$. Let $S^{(n)+}$ denote the set of all histories $h_t = (q_0, \ldots q_t)$, $0 \le t \le T$, where $q_k \in S^{(n)}$, $0 \le k \le t$.

The following is the general theorem of Wormald for justifying the use of differential equations on random graph processes [21]. The theorem is stated for a multidimensional process (i.e. $a > 1$), but we will need to go beyond one dimension only for the analysis of vertex-weighted matching. Note that in part (iii), a Lipschitz condition is satisfied for a function $f(u)$ on the domain D if there exists a constant $L > 0$ satisfying

$$|f(u) - f(v)| \le L||u - v||_\infty \tag{2.8}$$

for all $u, v \in D$.

THEOREM 5. (WORMALD [21]) *For $1 \le \ell \le a$, where a is fixed, let $y^{(\ell)} : S^{(n)+} \to \mathbb{R}$ and $f_\ell : \mathbb{R}^{(a+1)} \to \mathbb{R}$, such that for some constant C_0 and all ℓ, $|y^{(\ell)}(h_t)| < C_0 n$ for all $h_t \in S^{(n)+}$ for all n. Let $Y_\ell(t)$ denote the random variable for $y_\ell(H_t)$. Assume the following three conditions hold, where D is some bounded connected open set containing the closure of*

$$\{(0, z_1, \ldots, z_a) : \mathbb{P}(Y_\ell(0) = z_\ell n, 1 \le \ell \le a) \ne 0 \text{ for some } n\}, \tag{2.9}$$

and T_D is a stopping time for the minimum t such that $(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) \notin D$.

[1] The $G(n,p)$ random graph has n vertices and an edge occurring between each pair of vertices independently with probability p.

(i) (Boundedness hypothesis.) For some function $\beta = \beta(n) \geq 1$

$$\max_{1 \leq \ell \leq a} |Y_\ell(t+1) - Y_\ell(t)| \leq \beta, \tag{2.10}$$

for $t < T_D$.

(ii) (Trend hypothesis.) For some function $\lambda_1 = \lambda_1(n) = o(1)$, for all $1 \leq \ell \leq a$

$$|\mathbb{E}[Y_\ell(t+1) - Y_\ell(t)|H_t] - f_\ell(t/n, Y_1(t)/n, \ldots, Y_a(t)/n)| \leq \lambda_1 \tag{2.11}$$

for $t < T_D$.

(iii) (Lipschitz hypothesis.) Each function f_ℓ is continuous, and satisfies a Lipschitz condition, on

$$D \cap \{(t, z_1, \ldots, z_a) : t \geq 0\}, \tag{2.12}$$

with the same Lipschitz constant for each ℓ.

Then the following are true.

(a) For $(0, \hat{z}_1, \ldots, \hat{z}_a) \in D$ the system of differential equations

$$\frac{\partial z_\ell}{\partial \tau} = f_\ell(\tau, z_1, \ldots, z_a), \quad 1 \leq \ell \leq a \tag{2.13}$$

has a unique solution in D for $z_\ell : \mathbb{R} \to \mathbb{R}$ passing through

$$z_\ell(0) = \hat{z}_\ell, \quad 1 \leq \ell \leq a \tag{2.14}$$

and which extends to points arbitrarily close to the boundary of D;

(b) Let $\lambda > \lambda_1$ with $\lambda = o(1)$. For a sufficiently large constant C, with probability $1 - O(\frac{\beta}{\lambda} \exp(-\frac{n\lambda^3}{\beta^3}))$,

$$Y_t^{(\ell)} = n z_\ell(t/n) + O(\lambda n) \tag{2.15}$$

uniformly for $0 \leq t \leq \sigma n$ and for each ℓ, where $z_\ell(\tau)$ is the solution in (a) with $\hat{z}_\ell = \frac{1}{n} Y_\ell(0)$, and $\sigma = \sigma(n)$ is the supremum of those τ to which the solution can be extended before reaching within ℓ^∞-distance $C\lambda$ of the boundary of D.

The stopping time T_D is needed for situations where at some point the Lipschitz condition fails, as often happens near the end of some graph processes. This problem does not arise in our application.

We will use the following identity to satisfy the trend hypothesis of Wormald's theorem.

LEMMA 1. *For $n > 0$, $c \leq n/2$, and $0 \leq w \leq 1$,*

$$0 \leq e^{-cw} - \left(1 - \frac{c}{n}\right)^{nw} \leq \frac{c}{ne}. \tag{2.16}$$

Proof. Using the inequalities $1 - x \geq e^{-x-x^2}$ for $x \leq 1/2$ and $1 - x \leq e^{-x}$ for $x \geq 0$, we obtain $e^{-cw}\left(1 - \frac{c^2 w}{n}\right) \leq \left(1 - \frac{c}{n}\right)^{nw} \leq e^{-cw}$. The result follows by rearranging terms and using $cwe^{-cw} \leq 1/e$. □

For bounding the maximum matching size, we use a result from Bollobás and Brightwell [4]. Their result (specifically Theorem 14 in [4]) is stated in terms of the size of the largest independent set for a bipartite graph, which by König's theorem bounds the maximum matching size [14].

THEOREM 6. (BOLLOBÁS AND BRIGHTWELL [4]) *Let $\mu^*(n, n, c/n)$ denote the size of the maximum matching on the graph $G(n, n, c/n)$ where $p = c/n$. Then a.a.s.,*

$$\frac{\mu^*(n, n, c/n)}{n} \leq 2 - \frac{\gamma^* + \gamma_* + \gamma^*\gamma_*}{c} + o(1), \tag{2.17}$$

where γ_* is the smallest root of the equation $x = c\exp(-ce^{-x})$ and $\gamma^* = ce^{-\gamma_*}$.

The bound is known to be tight for $c \leq e$. Remarkably, this bound differs from the expression for the maximum matching size in the $G(n, p)$ model obtained in [2] by a factor of two.

2.1 Oblivious algorithm. The oblivious algorithm performs a "one shot" trial for each ball j, where it attempts to match j to a random neighbor. The algorithm is unaware of which bins are already matched, so an attempted match to an already occupied bin means that ball j is dropped. This is shown in Algorithm 1. We use $N(j)$ to denote the set of neighboring bins of ball j.

Algorithm 1: OBLIVIOUS

```
foreach j ∈ J do
    When ball j arrives, let N(j) denote the set of neighboring bins
    if |N(j)| > 0 then
        Select a random neighboring bin i ∈ N(j)
        if bin i is unmatched then
            Match j to i
        else
            j is dropped
```

Intuitively, the algorithm is expected to do well on extremely sparse graphs, specifically those where balls are likely to have at most one neighbor. On the other hand, the performance on dense graphs should be suboptimal since each ball has a variety of neighboring bins that are not utilized by the algorithm. The following two lemmas confirm this behavior.

LEMMA 2. *For all valid functions $p(n)$ where $p(n) = o(1/n)$, the performance of* OBLIVIOUS *on $G(n, n, p)$ with $p = p(n)$ satisfies $\mathcal{R}_{\mathrm{G}}(p(n)) = 1$.*

Proof. We consider the number of isolated edges in $G(n, n, p)$, where an edge (i, j) is isolated if $N(i) = \{j\}$ and $N(j) = \{i\}$. For any bin and ball pair (i, j), the probability of an isolated edge occurring between them is

$$\mathbb{P}((i, j) \text{ is isolated edge}) = p(1 - p)^{(2n-2)}. \tag{2.18}$$

Every isolated edge is matched by OBLIVIOUS, so

$$\mathbb{E}[\mu_{\mathrm{O}}] \geq n^2 p(1 - p)^{(2n-2)}, \tag{2.19}$$

where μ_{O} is the size of the OBLIVIOUS matching. The maximum matching size, denoted by μ^*, is upper bounded in expectation by the expected number of edges, n^2p, so

$$\frac{\mathbb{E}[\mu_{\mathrm{O}}]}{\mathbb{E}[\mu^*]} \geq (1-p)^{(2n-2)} = (e^{-o(\frac{1}{n})})^{(2n-2)} = e^{-o(1)} \to 1. \tag{2.20}$$

□

LEMMA 3. *For all valid functions $p(n)$ where $p(n) = \omega(1/n)$, the performance of* OBLIVIOUS *on $G(n,n,p)$ with $p = p(n)$ satisfies $\mathcal{R}_{\mathrm{G}}(p(n)) \geq 1 - 1/e$.*

Proof. Fix a bin i. For each ball j, there is an attempted assignment of j to i with probability $1/n$ if j has at least one neighbor. Thus,

$$\mathbb{P}(\text{no attempted match of } j \text{ to } i) = 1 - \left(\frac{1-(1-p)^n}{n}\right). \tag{2.21}$$

Considering all balls and using $1 - x \leq e^{-x}$,

$$\begin{aligned}\mathbb{P}(\text{no attempted match to } i) &\leq \left(1 - \left(\frac{1-e^{-pn}}{n}\right)\right)^n \\ &= \left(1 - \left(\frac{1-e^{-\omega(1)}}{n}\right)\right)^n \to \left(1 - \frac{1}{n}\right)^n \to e^{-1}.\end{aligned} \tag{2.22}$$

Accordingly, the expected number of bins matched by OBLIVIOUS is at least $n(1-1/e)$. The maximum matching size is at most n, so the result follows. The bound is shown to be tight by simply setting $p(n) = 1 - \epsilon$ where $\epsilon > 0$ and letting $\epsilon \to 0$. □

We are left to analyze the region $p(n) = c/n$, where we observe the transition from a performance ratio of 1 to $1 - 1/e$. The proof of Theorem 1 is shown below; the proofs of Theorem 2 and Corollary 1 then follow easily. Theorem 1 can be proved using a linearity of expectations argument; we instead use the approach below to introduce the differential equation method.

Proof of Theorem 1. Let $Y(t)$ denote the number of occupied bins immediately before the tth ball arrives, where $Y(1) = 0$. Slightly abusing notation, we use $Y(t)$ to denote both the random variable and instances of the random variable. Given $Y(t)$, the tth ball is dropped if it is isolated or if its selected neighbor is already matched. Thus,

$$\begin{aligned}\mathbb{P}(\text{ball } t \text{ is matched}|Y(t)) &= (1-(1-p)^n)\left(1 - \frac{Y(t)}{n}\right) \\ &= \mathbb{E}[Y(t+1) - Y(t)|Y(t)].\end{aligned} \tag{2.23}$$

Define the normalized random variable

$$Z(\tau) = \frac{Y(n\tau)}{n}, \quad 0 \leq \tau \leq 1, \tag{2.24}$$

so that $Z(\tau)$ indicates the fraction of occupied bins after $n\tau$ of the arrivals have occurred. We have

$$\begin{aligned}\frac{\mathbb{E}[Z(\tau+1/n) - Z(\tau)|Z(\tau)]}{1/n} &= (1-(1-p)^n)(1-Z(\tau)) \\ &= \left(1 - \left(1 - \frac{c}{n}\right)^n\right)(1-Z(\tau)) \\ &= (1-e^{-c})(1-Z(\tau)) + o(1).\end{aligned} \tag{2.25}$$

As $n \to \infty$, we arrive at the differential equation

$$\frac{dz(\tau)}{d\tau} = (1 - e^{-c})(1 - z(\tau)), \tag{2.26}$$

where $z(\cdot)$ is deterministic and replaces $Z(\cdot)$. Integrating and using $z(0) = 0$ gives

$$z(\tau) = 1 - e^{(e^{-c}-1)\tau}, \quad 0 \le \tau \le 1. \tag{2.27}$$

Applying Theorem 5, we choose the domain D defined by $-\epsilon < \tau < 1 + \epsilon$ and $-\epsilon < z < 1 + \epsilon$, for $\epsilon > 0$. Clearly we have $C_0 = 1$ and $\beta = 1$ by the nature of the matching process. Let $\lambda_1 = c/(en)$ for the trend hypothesis, which is satisfied according to Lemma 1. The Lipschitz hypothesis is satisfied with a Lipschitz constant $L = (1 + \epsilon)(1 - e^{-c})$. Setting $\tau = 1$ and choosing $\lambda = cn^{-1/4}$, Theorem 5 (b) gives that with probability $1 - O(n^{1/4}e^{-c^3 n^{1/4}})$,

$$\mu_{\mathrm{O}}(n, n, p) = n(1 - e^{(e^{-c}-1)}) + O(n^{3/4}). \tag{2.28}$$

The extra ϵ portion of the domain ensures that $z(\tau)$ remains within distance $C\lambda$ of the boundary of D. □

Proof of Theorem 2. The cases of $p(n) = o(1/n)$ and $p(n) = \omega(1/n)$ are given by Lemma 2 and Lemma 3. For the regime $p = c/n$, define the normalized random variables

$$\widetilde{\mu}_{\mathrm{O}}(n, n, c/n) \;:=\; \frac{\mu_{\mathrm{O}}(n, n, c/n)}{n}, \quad \widetilde{\mu}^*(n, n, c/n) := \frac{\mu^*(n, n, c/n)}{n}. \tag{2.29}$$

By Theorem 1, $\widetilde{\mu}_{\mathrm{O}}$ converges in probability. Additionally, $\widetilde{\mu}_{\mathrm{O}}$ is bounded and thus uniformly integrable, so convergence in probability implies convergence in mean:

$$\lim_{n\to\infty} \mathbb{E}[\widetilde{\mu}_{\mathrm{O}}(n, n, c/n)] = 1 - e^{(e^{-c}-1)}. \tag{2.30}$$

Similarly, $\widetilde{\mu}^*$ must satisfy

$$\lim_{n\to\infty} \mathbb{E}[\widetilde{\mu}^*(n, n, c/n)] \ge 2 - \frac{\gamma^* + \gamma_* + \gamma^*\gamma_*}{c}. \tag{2.31}$$

□

Proof of Corollary 1. The expected maximum matching size is no greater than the expected number of non-isolated vertices on one side of the graph, so $\mathcal{R}_{\mathrm{O}}(c/n) \ge \frac{1-e^{(e^{-c}-1)}}{1-e^{-c}} \ge 1 - 1/e$. □

2.2 Greedy algorithm. The GREEDY algorithm is shown in Algorithm 2. Upon the arrival of each ball, GREEDY assigns the ball to a randomly selected neighboring bin that is unmatched. If no such bin exists, the ball is dropped. Recall that the RANKING algorithm instead picks an initial ranking of bins and always assigns each ball to the neighboring unmatched bin with highest rank. We begin by showing that RANKING and GREEDY perform equivalently on $G(n, n, p)$. Let $U(j)$ denote the set of unmatched neighboring bins of ball j when it arrives.

LEMMA 4. *The performance of* GREEDY *and* RANKING *are equivalent on the graph* $G(n, n, p)$.

Proof. Consider the evolution of an instance of the RANKING algorithm. Upon the arrival of each ball j, a match does not occur only if the bins $N(j)$ are all occupied or if $N(j) = \emptyset$. Immediately before the edges for j are revealed, the probability of such an event depends only on the number occupied bins and is independent of the bin ranking, as edges are generated independently of the existing graph. It follows that the matches obtained by GREEDY and RANKING are determined only as a function of the number of available bins. □

Algorithm 2: GREEDY

```
foreach j ∈ J do
    When ball j arrives, let U(j) denote the set of unmatched neighboring bins
    if |U(j)| > 0 then
        Match j to a random bin i ∈ U(j)
    else
        j is dropped
```

It is clear that on extremely sparse graphs, GREEDY should perform near optimally, as OBLIVIOUS does. The GREEDY algorithm should also perform well on very dense graphs because many bins are available to each ball. We formalize this behavior with the following two lemmas.

LEMMA 5. *For all valid functions $p(n)$ where $p(n) = o(1/n)$, the performance of* GREEDY *on $G(n, n, p)$ with $p = p(n)$ satisfies $\mathcal{R}_{\mathrm{G}}(p(n)) = 1$.*

Proof. The proof for Lemma 2 holds. □

LEMMA 6. *For all valid functions $p(n)$ where $p(n) = \omega(1/n)$, the performance of* GREEDY *on $G(n, n, p)$ with $p = p(n)$ satisfies $\mathcal{R}_{\mathrm{G}}(p(n)) = 1$.*

Proof. We use a crude lower bound for the probability that each ball is matched. When ball t arrives, at most $t - 1$ bins are occupied, so

$$\mathbb{P}(\text{ball } t \text{ is matched}) \geq 1 - (1-p)^{(n-t+1)}. \tag{2.32}$$

Let μ_{G} denote the matching size obtained by GREEDY. Then

$$\mathbb{E}[\mu_{\mathrm{G}}] \geq n - \sum_{t=1}^{n} (1-p)^{(n-t+1)} = n - \frac{(1-(1-p)^n)(1-p)}{p} \geq n - \frac{1}{p}. \tag{2.33}$$

The maximum matching size is at most n, so

$$\frac{\mathbb{E}[\mu_{\mathrm{G}}]}{\mathbb{E}[\mu^*]} \geq 1 - \frac{1}{np(n)} = 1 - \frac{1}{\omega(1)} \to 1. \tag{2.34}$$

□

We now prove the result for the matching size obtained by GREEDY in the case $p = c/n$ (Theorem 3) and follow with proofs of Theorem 4 and Corollary 2.

Proof of Theorem 3. Again let $Y(t)$ denote the number of occupied bins immediately before the tth arrival. Conditioning on $Y(t)$, the tth ball cannot be assigned only if edges connecting the ball to the $n - Y(t)$ neighboring bins are not present. The probability of a match is then

$$\begin{aligned} \mathbb{P}(\text{ball } t \text{ is matched} | Y_t) &= 1 - (1-p)^{n\left(1 - \frac{Y(t)}{n}\right)} \\ &= \mathbb{E}[Y(t+1) - Y(t) | Y(t)]. \end{aligned} \tag{2.35}$$

Normalizing as before, we obtain

$$\begin{aligned}\frac{\mathbb{E}[Z(\tau+1/n)-Z(\tau)|Z(\tau)]}{1/n} &= 1-\left(1-\frac{c}{n}\right)^{n(1-Z(\tau))} \\ &= 1-e^{-c(1-Z(\tau))}+o(1). \end{aligned} \tag{2.36}$$

The corresponding differential equation for $n \to \infty$ is

$$\frac{dz(\tau)}{d\tau} = 1-e^{-c(1-z(\tau))}. \tag{2.37}$$

By integration and the initial value for $z(\tau)$,

$$z(\tau) = 1-\frac{\log\left(1+e^{-c\tau}(e^c-1)\right)}{c}, \quad 0 \le \tau \le 1. \tag{2.38}$$

The application of Wormald's theorem is the same as the proof for OBLIVIOUS but with Lipschitz constant $L = ce^{c\epsilon}$. □

Proof of Theorem 4. The proof follows the same approach as the proof of Theorem 2. □

Proof of Corollary 2. The approach for Corollary 1 can be used to show that the property holds for small c values (e.g. $c < 1/2$) and large c values (e.g. $c > 5$). For the remaining region, the exact expression for the performance ratio can be minimized numerically; the minimum is obtained at $c^* = 3.1685009$ and $\mathcal{R}_{\mathrm{G}}(c^*) = 0.8370875$. □

3 Vertex-Weighted Matching

In this section we modify the setting so that each bin i has a *rank*, $r_i \in \{1, 2, \ldots, m\}$ for some constant m, and that matching balls to lower ranked bins is preferable. This is equivalent to allowing each bin to have one of m distinct weights $w_i \in \mathbb{R}$, where a larger weighted bin has a lower rank. The VERTEX-WEIGHTED GREEDY algorithm matches each ball to its neighboring bin with smallest rank, as shown in Algorithm 3.

Algorithm 3: VERTEX-WEIGHTED GREEDY

```
foreach j ∈ J do
    When ball j arrives, let U(j) denote the set of unmatched neighboring bins
    if |U(j)| > 0 then
        Match j to a neighboring bin i ∈ argmin_{i∈N(j)} r_i
    else
        j is dropped
```

Let n_r denote the total number of bins with rank r, where $\sum_r n_r = n$, and let $g_r := n_r/n$. The result of this section is as follows.

THEOREM 7. *Let $\mu^r_{\mathrm{W}}(n,n,c/n)$ denote the number of bins with rank r that are matched by* VERTEX-WEIGHTED GREEDY *on the graph $G(n,n,p)$, where $p = c/n$ and $c > 0$ is a constant. Then a.a.s.,*

$$\frac{\mu^r_{\mathrm{W}}(n,n,c/n)}{n} = g_r - \frac{1}{c}\log\left(\frac{1+e^{-c\tau}(e^{c\sum_{k=1}^{r} g_k}-1)}{1+e^{-c\tau}(e^{c\sum_{k=1}^{r-1} g_k}-1)}\right)+o(1), \quad 1 \le r \le m. \tag{3.39}$$

Proof. We use the notation $Y_r(t),\ 1 \le r \le m$, to denote the number of rank r bins that are occupied immediately prior to the arrival of ball t. Consider the probability that a given ball t is matched to a rank two bin. This occurs if edges connecting to the $n_1 - Y_1(t)$ are not present and there is at least one neighboring bin with rank two, so

$$\mathbb{E}[Y_2(t+1) - Y_2(t)|\mathbf{Y}(t)] = \left((1-p)^{n\left(g_1 - \frac{Y_1(t)}{n}\right)}\right)\left(1 - (1-p)^{n\left(g_2 - \frac{Y_2(t)}{n}\right)}\right), \tag{3.40}$$

where $\mathbf{Y}(t) = (Y_1(t), Y_2(t), \dots Y_a(t))$. Generalizing, for $1 \le r \le m$,

$$\mathbb{E}[Y_r(t+1) - Y_r(t)|\mathbf{Y}(t)] \;=\; \left(1 - (1-p)^{n\left(g_r - \frac{Y_r(t)}{n}\right)}\right)\prod_{k=1}^{r-1}\left((1-p)^{n\left(g_k - \frac{Y_k(t)}{n}\right)}\right). \tag{3.41}$$

After normalizing and substituting $p = c/n$,

$$\frac{\mathbb{E}[Z_r(\tau + 1/n) - Z_r(\tau)|\mathbf{Z}(\tau)]}{1/n} \;=\; \left(1 - e^{-c(g_r - Z_r(\tau))}\right) e^{-c\sum_{k=1}^{r-1}(g_k - Z_k(\tau))} + o(1). \tag{3.42}$$

We arrive at the following system of differential equations for $n \to \infty$:

$$\frac{dz_r(\tau)}{d\tau} = \left(1 - e^{-c(g_r - z_r(\tau))}\right) e^{-c\sum_{k=1}^{r-1}(g_k - z_k(\tau))}, \qquad 1 \le r \le m. \tag{3.43}$$

It can be verified that the solution to the system of differential equations with initial conditions $z_r(0) = g_r$, $1 \le r \le m$, is

$$z_r(\tau) = g_r - \frac{1}{c}\log\left(\frac{1 + e^{-c\tau}(e^{c\sum_{k=1}^{r} g_k} - 1)}{1 + e^{-c\tau}(e^{c\sum_{k=1}^{r-1} g_k} - 1)}\right), \quad 0 \le \tau \le 1, \tag{3.44}$$

for $1 \le r \le m$. The application of Wormald's theorem is the same as the application of OBLIVIOUS and GREEDY, but with Lipschitz constant $L = (a-1)(1 - e^{-c(1+\epsilon)})ce^{(a-1)c\epsilon} + ce^{ac\epsilon}$. □

4 Conclusion

We have determined lower bounds on the performance of greedy matching algorithms on $G(n, n, p(n))$ that hold across all monotone functions $p(n)$, and shown that a clear phase transition occurs at $p(n) \sim 1/n$. In particular, $p(n) \sim 1/n$ is where the performance ratio of the OBLIVIOUS algorithm transitions from 1 to $1 - 1/e$ and the performance ratio of GREEDY passes through its global minimum.

The greedy algorithms perform relatively well: it is interesting that OBLIVIOUS, which is clearly inferior to RANKING, has a minimum performance ratio that is equal to the best possible worst-case competitive ratio of any online matching algorithm (i.e. that of RANKING). Likewise, the lower bound of 0.837 on the performance ratio of GREEDY is surprisingly high. Our analysis is tight inasmuch as the bound on the expected maximum matching size is tight (specifically Theorem 6); we conjecture that this theorem is in fact tight for all $c > 0$. Even if this is not the case, the lower bound on the performance of GREEDY cannot be any greater than 0.845, where the bound is tight for $c = e$.

There appears to be a close relationship between matching behavior on $G(n, n, p)$ and $G(n, p)$, where results differ by a mere factor of two. As we have mentioned, this relationship occurs with our asymptotic result for GREEDY (Theorem 3) and the result of MODIFIED GREEDY given in [6]. If Theorem 6 is in fact tight for all $c > 0$, the same can be said regarding a result given in [2]. A similar observation was also made by Frieze with respect to differential equations for the Karp-Sipser algorithm on bipartite

and ordinary graphs [10]. Note that it is not sufficient to simply argue that these properties hold for the mere fact that bipartite graphs have twice as many vertices; there are indeed important structural differences between ordinary and bipartite graphs.

A variety of possibilities exist for further research. We have allowed the existence of isolated vertices in our analysis, which is unlikely to be realistic for many applications. This could be resolved by imposing a restriction on the minimum degree of vertices, as was done in [11]. Unbalanced bipartite graphs (i.e. with more vertices on one side) are likely to be encountered in practice – our approach can be used in this situation, but less is known about expected maximum matching size here. Stepping away from the online scenario, one could consider multiple pass algorithms, which are highly relevant to streaming models.

References

[1] G. Aggarwal, G. Goel, C. Karande, and A. Mehta, *Online vertex-weighted bipartite matching and single-bid budgeted allocations*, in Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2011, pp. 1253–1264.

[2] J. Aronson, A. Frieze, and B. G. Pittel, *Maximum matchings in sparse random graphs: Karp-Sipser revisited*, Random Structures and Algorithms, 12 (1998), pp. 111–177.

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

[4] B. Bollobás and G. Brightwell, *The width of random graph orders*, Mathematical Scientist, 20 (1995), pp. 69–90.

[5] N. R. Devanur, K. Jain, and R. D. Kleinberg, *Randomized primal-dual analysis of ranking for online bipartite matching*, in SODA, 2013, pp. 101–107.

[6] M. Dyer, A. Frieze, and B. Pittel, *The average performance of the greedy matching algorithm*, The Annals of Applied Probability, (1993), pp. 526–552.

[7] P. Erdős and A. Rényi, *On random matrices*, Magyar Tud. Akad. Mat. Kutató Int. Közl, 8 (1964), p. 1964.

[8] ——, *On the existence of a factor of degree one of a connected random graph*, Acta Mathematica Hungarica, 17 (1966), pp. 359–368.

[9] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang, *On graph problems in a semi-streaming model*, Theoretical Computer Science, 348 (2005), pp. 207–216.

[10] A. Frieze, *Perfect matchings in random bipartite graphs with minimal degree at least 2*, Random Structures & Algorithms, 26 (2005), pp. 319–358.

[11] A. Frieze and B. Pittelt, *Perfect matchings in random graphs with prescribed minimal degree*, in Mathematics and Computer Science III, Springer, 2004, pp. 95–132.

[12] R. M. Karp and M. Sipser, *Maximum matching in sparse random graphs*, Foundations of Computer Science, IEEE Annual Symposium on, 0 (1981), pp. 364–375.

[13] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, *An optimal algorithm for on-line bipartite matching*, in Proceedings of the twenty-second annual ACM symposium on Theory of computing, ACM, 1990, pp. 352–358.

[14] D. König, *Über graphen und ihre anwendung auf determinantentheorie und mengenlehre*, Mathematische Annalen, 77 (1916), pp. 453–465.

[15] T. G. Kurtz, *Solutions of ordinary differential equations as limits of pure jump Markov processes*, Journal of Applied Probability, 7 (1970), pp. 49–58.

[16] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, *Adwords and generalized online matching*, Journal of the ACM (JACM), 54 (2007), p. 22.

[17] M. Mitzenmacher, *Studying balanced allocations with differential equations*, Combinatorics Probability and Computing, 8 (1999), pp. 473–482.

[18] B. Pittel, J. Spencer, and N. Wormald, *Sudden emergence of a giant k-core in a random graph*, Journal of Combinatorial Theory, Series B, 67 (1996), pp. 111–151.

[19] M. R. H. P. Raghavan, *Computing on data streams*, in External Memory Algorithms: DIMACS Workshop External Memory Algorithms and Visualization, May 20-22, 1998, vol. 50, AMS Bookstore, 1999, p. 107.

[20] N. C. Wormald, *Differential equations for random processes and random graphs*, The annals of applied probability, (1995), pp. 1217–1235.

[21] ——, *The differential equation method for random graph processes and greedy algorithms*, Lectures on approximation and randomized algorithms, (1999), pp. 73–155.