Project Report
Online Matching in Sparse Random Graphs: Non-Asymptotic Performances of Greedy Algorithm

Abdessalam Ed-dib*
abdessalam.ed-dib@ip-paris.fr
Ecole Polytechnique
France

Omar Ben Said*
omar.bensaid@ip-paris.fr
Ecole Polytechnique
France

Mohamed Khalil Braham*
khalil.braham@telecom-paris.fr
Télécom Paris
France

ABSTRACT

Online matching, a crucial aspect of internet advertising, consists on immediately allocating ad slots to ads as they arrive sequentially. We categorize this problem into adversarial, random order, and stochastic frameworks, each presenting distinct challenges. While the RANKING algorithm typically outperforms GREEDY in terms of competitive ratio, the latter demonstrates superior performance in specific scenarios, such as 2-regular graphs. Our report aims to explore the efficiency of the GREEDY algorithm in addressing the online matching problem within sparse random graphs with fixed degree distributions.

CCS CONCEPTS

• **Theory of computation → Online algorithms**.

KEYWORDS

Online Matching Algorithms, Sparse Random Graphs, Competitive Ratios Analysis

1 INTRODUCTION

Online algorithms are evident in various real-world scenarios where decisions need to be made without having access to the entire input all at once, as it arrives sequentially. A classic example is the online matching problem, a generalization of bipartite graph matching. This problem has gained considerable attention due to the rise of internet advertising and the critical applications of ad allocations. Essentially, the online matching problem involves promptly matching a, ad slot to an ad as soon as it arrives, ensuring that once a match is made, it cannot be changed. Ultimately, we aim at maximizing the size of the matching, which means having as many successful ad placements as possible, thereby optimizing the use of available ad slots and improving the effectiveness of advertising campaigns.

Depending on the order in which ad slots arrive, we can classify the problem into three main frameworks: the adversarial setting, the random order framework, and the stochastic version. In the adversarial setting, ad slots can arrive in any order. On the other hand, they arrive at random in the random order setting. Finally, in the stochastic version, ad slots are drawn independently and identically distributed from a given distribution. That being said, two main algorithms are commonly employed to solve the studied problem: the GREEDY algorithm and the RANKING algorithm. The greedy version is the simplest approach, involving the straightforward matching of arriving ad slots to any available ad. On the other hand, the RANKING algorithm consists on permuting the set of ads then assigning the ad slot to the ad with the lowest ranked free ad.

While the ranking algorithm generally outperforms the greedy version in terms of performance, there are intriguing nuances to consider. Interestingly, the GREEDY algorithm may exhibit superior performance in specific scenarios. For instance, it outperforms the ranking algorithm on the 2-regular graph.

In this report, we investigate online matching problems in sparse random graphs with fixed degree distributions, focusing on the GREEDY algorithm's performance in maximizing matching size. To do so, we analyze the competitive ratio of GREEDY in capacity-less scenarios and compared its efficiency with RANKING.

Report Structure. Following this introduction, the report is structured as follows. Section 2 introduces the problem setting and discusses preliminary concepts. Moving forward, Section 3 outlines our proposed methodology to handle the specified task. lastly, in Section 4, we showcase the potential held by this approach through comprehensive experimentation and comparative analysis between the two proposed algorithms. The code used for the experimental part has been provided along with the report.

Notations. To establish a consistent notation framework for our report, we adopt the notation used by Noiry et al. [2021]. Let $G = (U, V, E, \omega)$ denotes the bipartite graph with capacities, where $U = \{1, \ldots, N\}$ and $V = \{1, \ldots, T\}$ represent finite sets of vertices, E is the set of edges, and ω is the capacity function mapping vertices in U to non-negative integers. A matching M on G is a subset of edges where each vertex in V is the endpoint of at most one edge, and each vertex in U is the endpoint of at most ωu edges in M. The optimal matching is denoted as M^*, representing the matching with the highest cardinality. Finally, let $N(v) \in U$ be the set of available vertices that v can still be matched with.

2 BACKGROUND AND RELEVANT LITERATURE

2.1 Problem Statement

The online matching problem involves constructing matchings sequentially in a graph G with specified capacities ω. At each stage $t \in \mathbb{N}$, a new vertex v_t arrives with associated edges $\{(u, v_t) : u \in E\}$. The goal is to incrementally build a matching M_t by adding edges (u_t, v_t) to the existing matching M_{t-1}, subject to the constraint that $M_t = M_{t-1} \cup \{(u_t, v_t)\}$ remains a matching. The objective is to maximize the size of the final matching M_T.

The competitive ratio, defined as $CR = |M_T|/|M^*| \in [0, 1]$, serves as a measure of algorithmic performance.

*All authors contributed equally to this report.

2.2 Classic Online Matching Algorithms

In this section, we present and examine two algorithms aimed at solving the online bipartite matching problem:

2.2.1 GREEDY Algorithm.
As previously stated, the greedy algorithm offers a straightforward solution to the online matching problem. Despite its simplicity, it demonstrates reasonable competitiveness and is optimal in deterministic scenarios. When a vertex v is received from the set V, the greedy algorithm matches it to any vertex u in its neighborhood $N(v)$ if $N(v) \neq \emptyset$. Otherwise, it leaves v unmatched.

Algorithm 1: GREEDY

```
for t = 1, . . . , T do
  Match v_t to any free neighbor
```

2.2.2 RANKING Algorithm.
In order to improve the competitiveness of the aforementioned algorithm, Flan. introduces randomness and refines the greedy algorithm to eliminate arbitrary choices.

The Ranking algorithm initiates with an initialization phase involving the random permutation of vertices in U, which defines the ranking function $\pi(u)$ for all vertices $u \in U$ returning the index of u in the permutation. Subsequently, upon receiving each vertex $v \in V$, the algorithm matches v to the vertex $u \in N(v)$ with the lowest rank $\pi(u)$, provided u has not already been matched. If all vertices adjacent to v are already matched (i.e., $N(v) = \emptyset$), the algorithm refrains from matching v.

Algorithm 2: RANKING

```
Initialization Draw a random partition π
for i = 1, . . . , N do
  Assign to u_i rank π(i)
for t = 1, . . . , T do
  Match v_t to its lowest ranked free neighbor
```

2.2.3 Adversarial Setting: Comparative Analysis.
In the classical adversarial setting, where the order of arrival of vertices v in V is unknown, it has been proven that the ranking algorithm outperforms the greedy algorithm in terms of the competitive ratio. Indeed, it can be shown that:

$$\mathrm{CR}(\texttt{GREEDY}) \geq \frac{1}{2} \tag{1}$$

and

$$\mathrm{CR}(\texttt{RANKING}) \geq 1 - \frac{1}{e} \tag{2}$$

However, it is essential to note that this is not universally applicable. Indeed, Noiry et al. [2021] demonstrate theoretically that the greedy version outperforms the ranking algorithm when dealing with 2-regular graphs. They further extend this result empirically to d-regular graphs.

2.3 Configuration Model

In this section, we introduce the necessary framework to evaluate the efficiency of the greedy approach compared to the ranking algorithm when addressing sparse random graphs with fixed degree distribution.

2.3.1 Intuition.
The configuration model serves as a suitable random graph model that mimics several properties of real-world complex networks while remaining analytically tractable. For instance, by selecting power-law distributions for the degrees, it can exhibit the so-called scale-free property, as demonstrated in practical contexts such as the web by Faloutsos et al. [1999]. Additionally, the configuration model displays the "small-world phenomenon," observed, for example, in the Facebook graph by Backstrom et al. [2012], as its diameter is of logarithmic order. This mathematical framework allows us to generate random graphs with specified degree distributions. In this model, vertices are assigned degrees according to given distributions, and edges are randomly connected between vertices while preserving these degrees. Thus, it offers a structured approach to analyze graph properties and algorithms in scenarios where the degree distribution of vertices is known.

2.3.2 Construction of configuration models.
We make the assumption that the degree distributions of U and V, denoted as π_U and π_V respectively, follow sub-Gaussian distributions. These distributions are characterized by means μ_U and μ_V, and proxy-variances σ_U^2 and σ_V^2 respectively. Moreover, we assume that the means of these distributions verify the compatibility condition ($T\mu_V = N\mu_U$). Under these assumptions, we construct the corresponding configuration model, denoted by $\mathrm{CM}(d^U, d^V)$, by matching half-edges of U and half-edges of V. Here, d^U and d^V denote the degree sequences corresponding to vertices in U and V.

2.3.3 Greedy Algorithm for configuration model without capacities.

The paper raises an interesting observation that for the case of generating graphs using a configuration model, we can sequentially match the half-edges coming from V and concurrently construct the corresponding GREEDY matching. In fact, the procedural steps involve the utilization of non-negative integer sequences denoted as $\mathrm{d}^{\mathcal{U}}$ and $\mathrm{d}^{\mathcal{V}} \cup \left\{d^{\mathcal{V}}_{T+1}\right\}$, representing the number of half-edges emanating from each vertex, to simultaneously construct the graph and the matching as new vertices v_t get added to the graph. A pseudocode of this algorithm is shown in Algorithm 3. Two important factors are to consider here:

- Since each pairing of half-edges is done at random then the graph is sampled of distribution the bipartite configuration model.
- The matching thus obtained is indeed the matching computed by GREEDY.

Algorithm 3: Greedy Matching Configuration Model without Capacities

Input : $du = (d_u, \ldots, d_{|U|})$ and $dv = (d_v, \ldots, d_{|V|})$
Initialization $M_0 \leftarrow \emptyset, \mathcal{E}_0 \leftarrow \emptyset$ *and* $H_0^{\mathcal{U}} \leftarrow \{$ *half-edges of* $\mathcal{U}\}$
for $t = 1, \ldots, T$ **do**
  Order uniformly at random the edges emanating from $v_t : e_1^t, \ldots, e_{k_t}^t$
  for $i = 1, \ldots, |dv|$ **do**
    Choose uniformly an half-edge $e_i^{\mathcal{U}} \in H^{\mathcal{U}}$
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{u(e_i^{\mathcal{U}}), v_t\}$; /* Create an edge between e_i^t and $e_i^{\mathcal{U}}$ */
    $H^{\mathcal{U}} \leftarrow H^{\mathcal{U}} \setminus \{e_i^{\mathcal{U}}\}$; /* Remove the half-edge */
    if v_t *and* $u(e_i^{\mathcal{U}})$ *are unmatched* **then**
      $M_t \leftarrow M_{t-1} \cup \{u(e_i^{\mathcal{U}}), v_t\}$; /* v_t is matched */
$\mathbf{CM}(\mathbf{d}^{\mathcal{U}}, \mathbf{d}^{\mathcal{V}}) \leftarrow (\mathcal{U}, \mathcal{V}, \mathcal{E})$
Output: Bipartie configuration model $\mathbf{CM}(\mathbf{d}^{\mathcal{U}}, \mathbf{d}^{\mathcal{V}})$ and matching M_T on it

2.3.4 Examples.
In this section, we provide a detailed explanation of the construction of three typical random graphs using the configuration model framework.

***d*-Regular Graphs.** In a d-regular graph, each vertex has exactly the same degree, denoted by d. Therefore, for a fixed d each vertex of U is initialized to d half-edges. Afterwards, for each time step t, as a new vertex v_t is observed we assign to it d half-edges that are then matched uniformly to the available half-edges in U.

Erdos-Renyi Graphs. In the Erdos-Renyi graph, two vertices u and v are connected with probability p. For a given N number of verticies, the degree distribution is a binomial distribution that approaches a Poisson of parameter $Np = c$ as $N \longrightarrow \infty$. This means that to generate a graph, as was the case with the d-Regular graph, we generate a degree list d^U and d^V using two Poisson distributions π_U and π_V. Since $\sum_i d_i^U$ is not necessarily equal to $\sum_i d_i^V$, we add two virtual nodes u_{N+1} and v_{T+1} that connect to the loose half-edges such that.

$$d_{T+1}^{\mathcal{V}} = \max\left\{\sum_{i=1}^{N} d_i^{\mathcal{U}} - \sum_{j=1}^{T} d_j^{\mathcal{V}}, 0\right\}$$

and

$$d_{N+1}^{\mathcal{U}} = \max\left\{\sum_{j=1}^{T} d_j^{\mathcal{V}} - \sum_{i=1}^{N} d_i^{\mathcal{U}}, 0\right\}$$

We then proceed to match the half-edges as we did in the d-regular graph explained above.

3 MAIN RESULTS

In this section, we summarize the main theoretical results established by Noiry et al. [2021] to measure the performance of the greedy algorithm within structured online models using configuration models. It is worth noting that we limit our analysis to the simple case, specifically the capacity-less case:

$$\forall u \in U, \quad \omega(u) = 1$$

3.1 Performances of GREEDY in the capacity-less case

Theorem 1. Given $N \geq 1$ and $T = \frac{\mu_U}{\mu_V} N$, let M_T be the matching built by GREEDY on $CM(d_U, d_V)$. Then, the following convergence in probability holds:

$$\frac{|M_T|}{N} \xrightarrow{P} 1 - \phi_U(1 - G(1)),$$

where G is the unique solution of the ordinary differential equation:

$$G'(s) = \frac{1 - \phi_V\left(1 - \frac{1}{\mu_U}\phi_U'(1 - G(s))\right)}{\frac{\mu_V}{\mu_U}\phi_U'(1 - G(s))}; \quad G(0) = 0$$

s.t.

- $\phi_U(x) = \sum_{k \geq 0} \pi_U(k) x^k$
- $\phi_V(x) = \sum_{k \geq 0} \pi_V(k) x^k$

Moreover we also have a formula for the intermediate state of online matching. In fact, for any $s \in [0, 1]$ we denote $M_T(s)$ the matching obtained through the greedy algorithm then:

$$\frac{|M_T(s)|}{N} \xrightarrow{P} 1 - \phi_U(1 - G(s))$$

for $s = \frac{k}{N}$ we observe that the above formula gives the competitive ration of greedy algorithm at step k. Finally, we have explicit convergence rates: with probability greater than $1 - \zeta N \exp\left(-\xi N^{c/2}\right)$

$$\sup_{s \in [0,1]} \left| \frac{|M_T(s)|}{N} - (1 - \phi_U(1 - G(s))) \right| \leq \kappa N^{-c}$$

where ζ, ξ, κ are constants that depend on the first two moments of μ_U and μ_V.

3.2 Application on d-regular graphs

The first example discussed in the paper pertains to the d-regular graph case. This type of graph is characterized by $\mu_U = \mu_V = \delta_d$, representing the Dirac mass at d. Expliciting the equation derived from Theorem 3.1 yields:

$$\frac{(1 - G(s))^{d-1}}{1 - \left(1 - (1 - G(s))^{d-1}\right)^d} G'(s) = \frac{1}{d}$$

An interesting scenario arises when $d = 2$ since it can be solved explicitly, resulting in a closed-form expression of G as follows:

$$G(s) = \exp(\frac{s}{2}) - 1$$

This solution provides a competitive ratio of around 0.877, which significantly exceeds the classic competitive ratio $1 - \frac{1}{e}$ (approximately 0.632). This observation offers initial insight into one of the propositions discussed in the paper: "GREEDY outperforms RANKING on the 2-regular graph".

3.3 Application on Erdos-Renyi graphs

The second example concerns the Erdos-Renyi graph and its behavior as the number of vertices, N, tends to infinity. In the Erdos-Renyi graph, the probability of an edge existing between any two vertices is given by $p = \frac{c}{N}$. As N approaches infinity, the degree distributions μ_U and μ_V converge to a Poisson distribution with parameter c. Hence, the equation from Theorem 3.1 becomes:

$$\frac{cG'(s)e^{-cG(s)}}{1 - e^{-ce^{-cG(s)}}} = 1$$

Interestingly, this equation can be solved analytically, yielding:

$$G(s) = \frac{1}{c} \log \left(\frac{c}{\log\left(e^{k-cs} + 1\right)} \right)$$

Hence using this closed-form expression for $G(s)$, we can compute the competitive ratio of GREEDY, which can be verified numerically as we are going to see afterwards.

$$\frac{|M_T|}{N} \xrightarrow{P} 1 - \phi_{\mathcal{U}}(1 - G(1)) = 1 - \frac{\log\left(2 - e^{-c}\right)}{c}$$

3.4 Application on geometric degree distribution based graphs

While Erdos-Renyi and d-regular graphs are useful for theoretical modeling of random graphs, they often fail to capture properties commonly observed in real-world networks, such as scale-free behavior. A more practical alternative, the Barabasi-Albert model, relies on a power-law distribution, which aligns better with real-world network characteristics [Barabasi and Albert 1999]. However, in our specific case, the Barabasi-Albert model doesn't meet the required hypotheses outlined in Theorem 3.1. Therefore, we considered using the geometric distribution as an approximation of this model. The geometric distribution is suitable because it describes networks where nodes generally have a relatively low average degree, and the likelihood of having more connections decreases exponentially as the degree increases. Additionally, it offers a simple and tractable model that satisfies the necessary assumptions for applying Theorem 3.1.

3.4.1 *Verification of Hypotheses.*

To simplify, let's assume that both degrees U and V follow identical geometric distributions with parameter p. Therefore, $T = N$, implying that the probability of a node having k neighbors is expressed as:

$$\mathcal{P}(k) = p(1 - p)^{k-1}$$

That being said, it is true that the geometric distribution doesn't meet the subgaussianity assumption given its exponentially decaying tails. Indeed, it can be proven that Theorem 3.1 still holds for sub-exponential distributions as the main result used to prove Lemma 2 in the original paper is also valid for sub-exponential distributions.

3.4.2 Application of Theorem 1. Let's recall that a geometric distribution with parameter p has a mean equal to $\mu_U = \mu_V = \frac{1}{p}$. With this in mind, and having verified that the geometric distribution satisfies the necessary assumptions, we can now apply Theorem **??**.

Thus, the differential equation to solve can be expressed as:

$$G'(x) = \frac{1 - \phi_U(1 - p\phi'_U(1 - G(x)))}{\phi'_V(1 - G(x))}$$

where:

$$\phi_V(x) = \phi_U(x) = \frac{px}{1 - (1 - p)x}$$

and

$$\phi'_V(x) = \phi'_U(x) = \frac{p}{(1 - (1 - p)x)^2}$$

Given the complexity of the following equation, analytical solutions may be fastidious. Therefore, we will resort to numerical methods to solve it. This approach will allow us to verify the theorem and compare the performance of the GREEDY algorithm against the RANKING algorithm.

4 EXPERIMENTS

In this section, we begin by replicating the experiments conducted in the original paper to numerically verify the results of Theorem 1 and to compare the performance of the greedy and ranking algorithms in terms of competitive ratio. Subsequently, we shift our focus to the case of geometric degree distribution, where we undertake similar analyses. Also, we investigate the impact of the parameter p of the geometric distribution on the competitive ratio. Finally, we delve into the capacity case to gain insights into the performance of both algorithms.

All experiments are performed on a system equipped with four cores of an `11th Gen Intel(R) Core(TM) i7-11800H (16 CPUs)` running at `2.30 GHz`. The simulations are implemented using `Python v3.9.7`.

4.1 Reproducing results

In this section, we begin by reproducing the results presented in the original paper for both types of graphs discussed therein.

4.1.1 *d-Regular Graphs.*

The experiments carried out for d-regular graphs in the paper were designed to assess the effectiveness of the Greedy Algorithm in online matching scenarios and to confirm the validity of the theoretical predictions derived from the Ordinary Differential Equation (ODE) solutions. The researchers conducted numerical experiments and simulations to compare the expected theoretical performance of Greedy with its actual performance on d-regular graphs.

Theoretical Guarantees. As illustrated in Figure 1, we observe that the theoretical guarantees align well with the expected competitive ratio computed numerically. This convergence between theoretical predictions and numerical results confirms the validity of Theorem 3.1 for d-regular graphs.

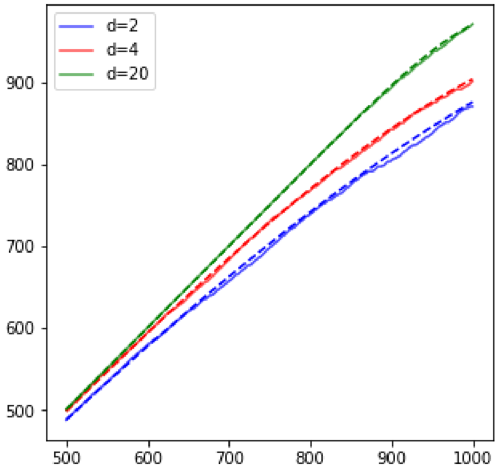

Figure 1: Expected theoretical performance of the GREEDY algorithm (dashed line) along with the simulated performance (full line) for various values of d.

***Comparison between* GREEDY *and* RANKING.** In Figure 2, we present a comparison of the competitive ratios of GREEDY and RANKING across various values of d in the d-regular graph configuration model. Consistent with the findings reported in the original paper, we reaffirm that GREEDY outperforms RANKING in this configuration. Moreover, our analysis reveals that the performance gap between the two algorithms diminishes as d increases. This observation underscores the nuanced relationship between graph sparsity and algorithmic performance.

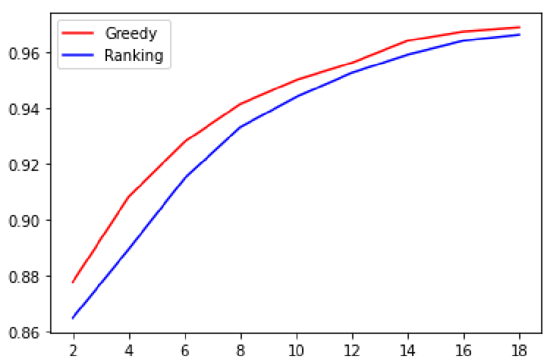

Figure 2: Comparison of RANKING and GREEDY algorithms for different values of d for the d-regular graph

***Impact of* N.** To assess the influence of N on convergence, we illustrate, in Figure 3, the disparity between theoretical and simulated performances across 10 runs for N values of 100, 1000, and 10000. Notably, we observed that the standard deviation of the results escalates sub-linearly, following $\sqrt{N}$, as anticipated.

4.1.2 Erdos-Renyi Graphs.

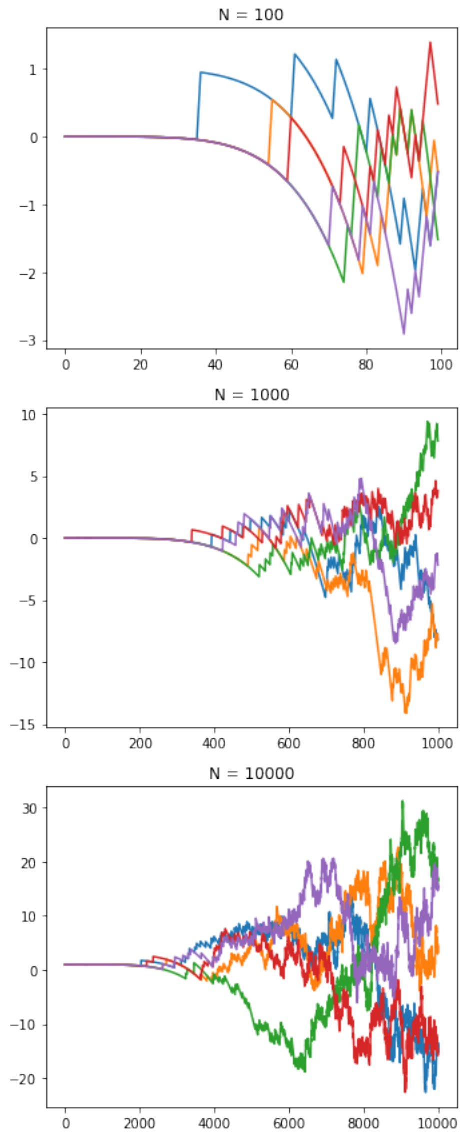

Figure 3: Difference between the theoretical performances and simulated performances of the GREEDY algorithm on the d-regular graph ($d = 20$) on 10 independent runs, with $N =$ 100, 1000, 10000.

Similar experiments were carried out for the Erdos-Renyi case, validating the theoretical guarantees through numerical analysis. Interestingly, in this specific scenario, the greedy algorithm exhibits slightly better performance than the ranking algorithm, although the difference is on the order of 10^{-2}. Additional figures illustrating these results are included in the notebook accompanying the report.

4.2 Geometric degree distribution based graphs

In the subsequent analysis, we perform a similar evaluation to validate the theoretical guarantees and evaluate the performance of both matching algorithms in the context of graphs characterized by a geometric degree distribution.

In Figure 4, we present a comparison between the scores predicted by the numerical solutions of the ordinary differential equation (ODE) for geometric degree distribution based graphs with parameter $p = 0.5$ and the simulated performance of the GREEDY algorithm for various values of N. As anticipated, the deviations of the simulated trajectories remain within $O(\sqrt{N})$ of the expected theoretical trajectory.

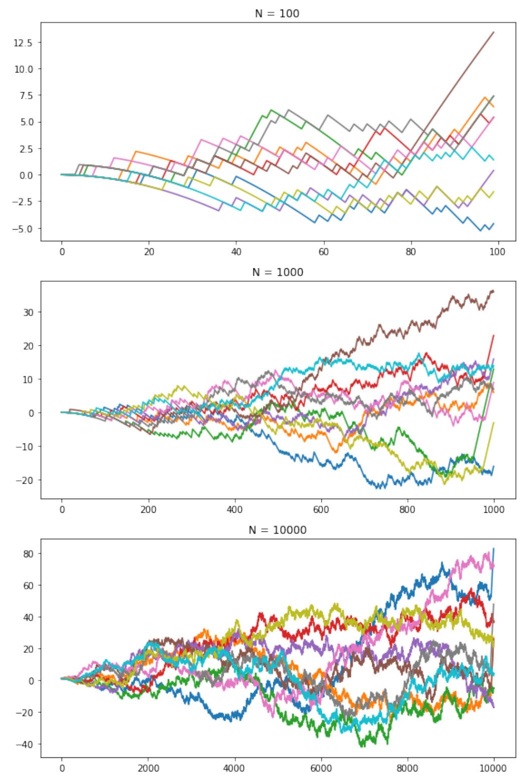

Figure 4: Difference between the theoretical performances and simulated performances of the GREEDY algorithm on the geometric degree distribution based graph ($p = 0.5$) on 10 independent runs, with $N = 100, 1000, 10000$.

The next step involves evaluating the impact of the chosen parameter p on the competitive ratio of the GREEDY algorithm obtained by solving the ODE, as well as the difference between the theoretical performance and the simulated one.

In Figure 5, we observe that the competitive ratio of GREEDY increases with the value of p. Indeed, as p increases, the corresponding graphs become sparser since nodes are less likely to have a high number of neighbors. Consequently, this observation aligns with the fact that the greedy algorithm performs well on sparser graphs.

On the other hand, we can observe in Figure 6 that the theoretical performance matches the experimental results as predicted by Theorem 3.1, regardless of the value of p chosen. It's interesting to note that the convergence is faster for higher values of p, corresponding to sparser graphs. Furthermore, we can notice that the performance curve exhibits a convex shape. This is because, at higher values of p, the graph tends to be sparse, allowing the algorithm to readily identify a satisfactory matching. Conversely, in denser graphs where most vertices have high degrees (approximately 10 on average), the algorithm can still uncover relatively good matches. However, in situations of intermediate density, there's a decline in performance since neither extreme case applies effectively.

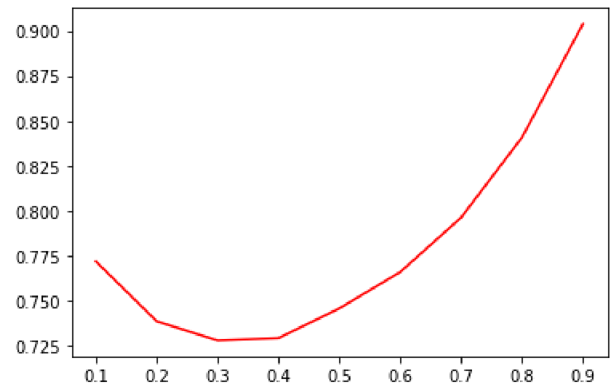

Figure 5: Expected competitive ratio of GREEDY on the geometric degree distribution based graph as a function of p.

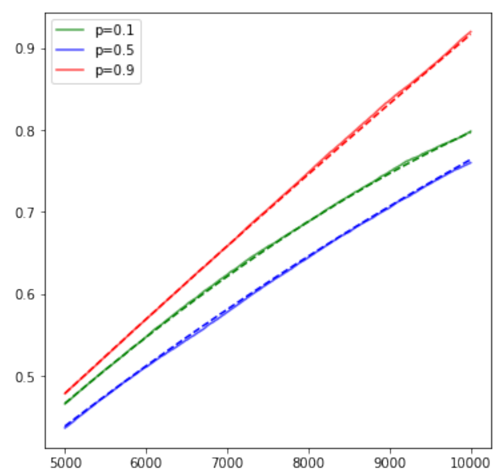

Figure 6: Expected theoretical performance of the GREEDY algorithm (dashed line) along with the simulated performance (full line) for various values of p.

Finally, let's compare both RANKING and GREEDY in terms of competitive ratios for different values of p.

As the sparsity of graphs increases with higher values of p, the performance difference between the GREEDY and RANKING algorithms diminishes, as expected. However, there's an intriguing observation around $p = 0.5$ where GREEDY unexpectedly outperforms RANKING. This anomaly can be attributed to the graph's intermediate density, where neither algorithm consistently excels: it's not

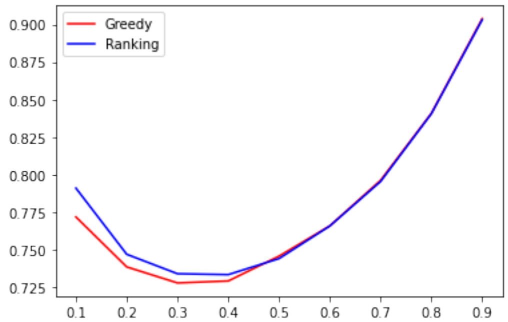

Figure 7: Comparison of RANKING and GREEDY algorithms for different values of p.

sparse enough for either algorithm to easily find good matches, nor dense enough for the GREEDY algorithm to fail and RANKING to succeed.

5 CONCLUSION

In conclusion, this report investigates the GREEDY algorithm's performance in maximizing matching size within sparse random graphs with fixed degree distributions. Despite the general superiority of the RANKING algorithm in several cases such as the adversarial paradigm, intriguing nuances emerge, such as the GREEDY algorithm's occasional outperformance, notably on 2-regular graphs. Through numerical validation and experimental analysis, we also notice that the performance gap between both algorithms closes as the sparsity of the graphs increases. Not to mention, the possible extension of Theorem 3.1 to sub-exponential distributions. Finally, another avenue for exploration involves identifying suitable algorithms for preference attachment models and adapting them to Barabási-Albert graphs, which do not adhere to the assumptions discussed earlier. This extension could offer valuable insights into matching algorithms' applicability in more diverse network structures, broadening the scope of research in this field.

REFERENCES

Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. 2012. Four degrees of separation. In *Proceedings of the 4th Annual ACM Web Science Conference* (Evanston, Illinois) *(WebSci '12)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/2380718.2380723

Albert-Laszlo Barabasi and Reka Albert. 1999. Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509-512. *Science (New York, N.Y.)* 286 (11 1999), 509–12. https://doi.org/10.1126/science.286.5439.509

Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. 1999. On power-law relationships of the Internet topology. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication* (Cambridge, Massachusetts, USA) *(SIGCOMM '99)*. Association for Computing Machinery, New York, NY, USA, 251–262. https://doi.org/10.1145/316188.316229

Nathan Noiry, Flore Sentenac, and Vianney Perchet. 2021. Online Matching in Sparse Random Graphs: Non-Asymptotic Performances of Greedy Algorithm. arXiv:2107.00995 [cs.DS]