Social Network De-Anonymization Under Scale-Free User Relations

Carla-Fabiana Chiasserini, *Senior Member, IEEE*, Michele Garetto, *Member, IEEE*, and Emilio Leonardi, *Senior Member, IEEE*

***Abstract*— We tackle the problem of user de-anonymization in social networks characterized by scale-free relationships between users. The network is modeled as a graph capturing the impact of power-law node degree distribution, which is a fundamental and quite common feature of social networks. Using this model, we present a de-anonymization algorithm that exploits an initial set of users, called seeds, that are known *a priori*. By employing the bootstrap percolation theory and a novel graph slicing technique, we develop a rigorous analysis of the proposed algorithm under asymptotic conditions. Our analysis shows that large inhomogeneities in the node degree lead to a dramatic reduction in the size of the seed set that is necessary to successfully identify all the other users. We characterize this set size when seeds are properly selected based on the node degree as well as when seeds are uniformly distributed. We prove that, given n nodes, the number of seeds required for network de-anonymization can be as small as n^{ϵ}, for any small $\epsilon > 0$. In addition, we discuss the complexity of our de-anonymization algorithm and validate our results through numerical experiments on a real social network graph.**

***Index Terms*— Computer networks, online social networks, user de-anonymization.**

I. Introduction

THE INCREASING availability of always-on connectivity on affordable portable devices, coupled with the proliferation of services and online social platforms, has provided unprecedented opportunities to interact and exchange information among people. At the same time, electronic traces of our communications, searches and mobility patterns, specifically their collection and analysis by service providers and unintended third parties, are posing serious treats to user privacy, raising a number of well known and hotly debated issues which have recently caused quite a stir in the media.

A distinctive feature of this trend is the uncontrolled proliferation of different accounts/identities associated to each individual: most of us have more than one mobile subscription, more than one email address, and a plethora of accounts on popular platforms such as Facebook, Twitter, LinkedIn, etc.

A specific issue that naturally arises in this context is the identification of the different identities/accounts belonging to the same individual. This problem, which has strong implications with user privacy, is known in the scientific literature as social network de-anonymization (or reconciliation). The two most frequently cited reasons why companies/organizations are interested in network de-anonymization are user profiling (for targeted advertising and marketing research) and national security (i.e., the prevention of terrorism and other forms of criminal activity).

It is fundamental to notice that privacy concerns related to de-anonymization are very subjective: some people do not care at all about providing "personally identifiable information" (PII) in their service registrations, explicitly linking their accounts "for-free".[1] As we will see, such users play a fundamental role in network de-anonymization, acting as "seeds" to identify other users. On the other extreme, some people are totally obsessed by the idea of Big Brother spying into their life and compiling tons of personal information on all of us. Such users try to hide themselves behind anonymous identities containing the minimum possible amount of personal data and linkage information with other identities. In the worst case (for the entity trying to de-anonymize) an identity consists just of a random identifier (e.g., a code or label).[2]

One recent [1]–[4], dramatic discovery in the network security field is the following: user privacy (in terms of anonymity) cannot be guaranteed by just resorting to anonymous identifiers. In particular, the identities used by a user across different systems can be matched together by using only the network structure of the communications made by users (i.e., electronic traces of who has come in contact with whom). More formally, considering just the simple case of two systems, the (disordered) vertices of two social network graphs, whose edges represent the observed contacts among users in the two systems, can be perfectly matched under very mild conditions on the graph structures.

In particular, as anticipated, the complexity of the network de-anonymization problem can be greatly reduced by having an initial (even small) number of user identifiers already correctly matched (the seeds). Such initial side information is

Manuscript received June 30, 2015; revised January 25, 2016; accepted March 24, 2016; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Wierman. Date of publication April 27, 2016; date of current version December 15, 2016. Preliminary results were presented at the International Conference on Computer Communications (INFOCOM) 2015.

C.-F. Chiasserini and E. Leonardi are with the Politecnico di Torino, Turin 10129, Italy, and also with the Institute of Electronics, Computer and Telecommunication Engineering of the National Research Council of Italy, Turin 10129, Italy (e-mail: chiasserini@polito.it; emilio.leonardi@polito.it).

M. Garetto is with the University of Torino, Turin 10124, Italy (e-mail: michele.garetto@unito.it).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. This consists of a 327-kB PDF containing proofs.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2016.2553843

[1] Many systems and applications strongly encourage (or even enforce) users to provide information such as email address/cell-phone number as part of account creation and maintenance, or for the purpose of backup and synchronization with cloud services.

[2] In our work, we will assume that each random identifier is at least guaranteed to be uniquely and statically assigned to a single user of a system/application.

often available, thanks to users who have explicitly linked their accounts, to the presence of compromised or fake users, as well as other forms of external information providing total or partial correlations among identities. Starting from the seeds, one can design clever algorithms to progressively expand the set of matched user identifiers, incurring only negligible error probability [5].

In previous work [6] the number of seeds that allow almost perfect de-anonymization of two networks has been characterized for the case of Erdös–Rényi random graphs, adopting a convenient probabilistic model to generate the two graphs from the (unknown) graph describing the complete set of human relationships among people. By reducing the graph matching problem to a bootstrap percolation problem, the authors of [6] identify a phase transition in the number of seeds required by their algorithm. In particular, in the case of a sparse network with n nodes and average node degree $\Theta(\log n)$, the number of seeds that are provably sufficient to match all nodes scales as $\Theta(\frac{n}{\log^{4/3} n})$, which is only a poly-log factor less than n. One obvious limitation of the results in [6] is that they apply only to Erdös–Rényi random graphs, which are a poor representation of real social networks (see Section IX for a more detailed discussion of previous work).

In this work we address the problem of user de-anonymization in the case of graphs that account for a relevant characteristic of social networks, namely, scale-free relationships between users. Our main contributions are as follows.

- We consider the underlying social network to be represented by a graph with power-law node degree, specifically, a Chung-Lu random graph [7] with constant average node degree. We then propose a novel algorithm for graph matching, hereinafter referred to as degree-driven graph matching (DDM), and show that DDM successfully matches a large fraction of the nodes.
- Similarly to [6], we are interested in the scaling law of the number of seeds that are needed to make the nodes' identification process 'percolate', i.e., to propagate over the entire set of nodes. However, our results mark a striking difference with respect to those obtained for Erdös–Rényi graphs. In particular, when initial seeds are uniformly distributed among the vertices, we show that order of $n^{\frac{1}{2}+\epsilon}$ seeds (for an an arbitrarily small ϵ) are sufficient to match almost all vertices, even in the case of constant average node degree.
- We obtain even more remarkable results when initial seeds can be chosen (e.g., by the attacker) considering their degree: in this case, as few as n^{ϵ} seeds are sufficient. The implications of these results are clear: scale-free social networks can be surprisingly simple to match (i.e., de-anonymize), especially when initial seeds are properly selected among the population.
- We empirically validate our findings running the DDM algorithm on a realistic data set (e.g., a Facebook snapshot). This validation is important because our model captures, in isolation, only the impact of power-law degree, without jointly accounting for other salient features of real social networks such as clustering, community structure and so on. Our experimental results confirm that real social networks can be de-anonymized starting from very limited side information.

The rest of the paper is organized as follows. Section II provides some definitions, describes the problem under study and outlines the proposed DDM algorithm. Section III presents preliminary results on Erdös–Rényi and Chung-Lu graphs. Section IV details the DDM algorithm, whose properties are then analysed in Section V, when the initial seeds can be chosen based on their degree. The analysis is extended to the case where seeds are uniformly distributed in Section VI. The complexity of the DDM algorithm is discussed in Section VII. Section VIII validates our theoretical results using a real-world data set. In Section, IX we discuss previous work, highlighting the novelty of our contribution. Finally, Section X draws some conclusions.

II. Model and Matching Algorithm

A. Basic Assumptions

We study the network de-anonymization problem in the case of two social networks represented by graphs $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$, respectively. However, our model and analysis can be extended to the case in which more than two networks are available. $\mathcal{G}_1$ and $\mathcal{G}_2$ can be considered to be sub-graphs of a larger, inaccessible graph, $\mathcal{G}_{\rm g}(\mathcal{V}, \mathcal{E})$, representing the ground-truth, i.e., the underlying social relationships between individuals. We will assume for simplicity that all the above graphs have the same set of vertices $\mathcal{V}$ with cardinality $|\mathcal{V}| = n$, i.e., $\mathcal{V}_1 = \mathcal{V}_2 = \mathcal{V}$, although this assumption can be easily removed by seeking to match only the intersection of vertices belonging to $\mathcal{G}_1$ and $\mathcal{G}_2$ (see [8] for an analysis with no-coincident node sets in Erdös–Rényi graphs). We remark that $\mathcal{G}_1$ and $\mathcal{G}_2$ do not necessarily represent subsets of social relationships as observed in totally different systems (e.g., Facebook and Twitter). They could also be obtained within the same communication system (i.e., from traces of emails, or from traces of phone calls), due to the fact that users employ two ID's in the same system (i.e., two email addresses, or two SIM cards).

We need a mathematical model describing how edges $\mathcal{E}_1$ and $\mathcal{E}_2$ are selected from the ground-truth set of edges $\mathcal{E}$. Any such model will necessarily be an imperfect representation of reality, since a large variety of different situations can occur. A user might employ either of her ID's to exchange messages with a friend, or use only one of them to communicate with a given subset of friends. General, realistic models trying to capture possibly heterogeneous correlations (positive or negative) in the set of neighbors of a vertex as seen in $\mathcal{G}_1$ and $\mathcal{G}_2$ become inevitably mathematically intractable. We therefore resort to the same assumption adopted in previous mathematical work [2], [4]–[6]: each edge in $\mathcal{E}$ is retained in $\mathcal{G}_1$ (or $\mathcal{G}_2$) with a fixed probability s, independently between $\mathcal{G}_1$ and $\mathcal{G}_2$, and independently of all other edges.[3] This model serves as a reasonable, first-step approximation of real systems, which

[3]Two different sampling probabilities s_1 and s_2, respectively for $\mathcal{G}_1$ and $\mathcal{G}_2$, could be considered as well.

permits obtaining fundamental analytical insights. Moreover, authors in [2] and [4] have experimentally found that the above assumption is largely acceptable in their scenario.

Another key element is the model for the underlying social graph $\mathcal{G}_{\rm g}$. To understand the impact of the power-law distribution of vertex degree, which characterizes realistic social networks, we have chosen a simple model known in the literature as Chung-Lu random graph [7]. In contrast to the classic model of Erdös–Rényi, Chung-Lu graphs permit considering a fairly general vertex degree distribution while preserving the nice property of independence among edge probabilities, which is of paramount importance in the analysis.

Definition 1: A Chung-Lu graph is a random graph of n vertices where each vertex i is associated with a positive weight w_i. Let $\bar{w} = \frac{1}{n}\sum_n w_i$ be the average weight. Given two vertices $i, j \in \mathcal{V}$, with $i \neq j$, the undirected edge (i, j) is included in the graph with probability

$$p_{ij} = \min\left\{\frac{w_i w_j}{n\bar{w}}, 1\right\}, \tag{1}$$

independently of the inclusion of any other edge in $\mathcal{E}$.

To avoid pathological behavior, it is customary in the Chung-Lu model to assume that the maximum vertex weight is $O(n^{1/2})$. Doing so, weight w_i essentially coincides with the average degree of vertex i, i.e., $p_{ij} = w_i w_j/(n\bar{w})$. For simplicity, in our work we will assume that weights are deterministic[4] (but note that they depend on n, albeit we avoid explicitly indicating this). A suitable way to obtain a power-law degree sequence with exponent β (with $2 < \beta < 3$, as typically observed in real systems) is to set $w_i = \bar{w}\frac{\beta-2}{\beta-1}(\frac{n}{i+i_0})^{1/(\beta-1)}$ where i_0 can be chosen such that the maximum degree is $O(n^{1/2})$. Also, we will assume $\bar{w}$ to be a finite constant, although our analysis can be easily extended to the more general case in which $\bar{w}$ scales with n.

B. Problem Definition

The network de-anonymization problem under study can be formulated as follows. We assume the underlying social network graph $\mathcal{G}_{\rm g}(\mathcal{V}, \mathcal{E})$ to be a known instance of a Chung-Lu graph having power-law degree distribution with exponent β (with $2 < \beta < 3$). However, we cannot access its edge set $\mathcal{E}$. Instead, we know the complete structure of two sub-graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ obtained by independently sampling each edge of $\mathcal{E}$ with probability s. In other words, each edge in $\mathcal{E}$ is assumed to be (independently) sampled twice, the first time to determine its presence in $\mathcal{E}_1$, the second time to determine its presence in $\mathcal{E}_2$. Note that sets $\mathcal{V}_1$ and $\mathcal{V}_2$ include the same vertices but the correspondence between vertices in the two graphs is unknown. The objective is therefore to find the correct match among them, i.e., to identify all pairs of vertices $[i_1, i_2] \in \mathcal{V}_1 \times \mathcal{V}_2$ such that i_1 and i_2 correspond to the same vertex $i \in \mathcal{V}$.

We define the graph of all possible vertex pairs as the pairs graph $\mathcal{P}(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$, with $\boldsymbol{\mathcal{V}} = \mathcal{V}_1 \times \mathcal{V}_2$ and $\boldsymbol{\mathcal{E}} = \mathcal{E}_1 \times \mathcal{E}_2$. In $\mathcal{P}(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$ there exists an edge connecting $[i_1, j_2]$ with $[k_1, l_2]$ iff edge $(i_1, k_1) \in \mathcal{E}_1$ and edge $(j_2, l_2) \in \mathcal{E}_2$. We will slightly abuse the notation and denote the pairs graph referring to $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ by $\mathcal{P}(\mathcal{G}_{\rm g})$. We will refer to pairs $[i_1, i_2]$, whose vertices correspond to the same vertex $i \in \mathcal{G}_{\rm g}$ as good pairs, and to all others (e.g., $[i_1, j_2]$) as bad pairs. Also, we define pairs such as $([i_1, j_2], [i_1, k_2])$ or $([i_1, j_2], [k_1, j_2])$ as conflicting pairs, and pairs that are adjacent on $\mathcal{P}(\mathcal{G}_{\rm g})$ as neighbors. The generic pair will be denoted by $[*_1, *_2]$.

As mentioned, [2]–[4] showed that network graph de-anonymization can be achieved even in the case where no a-priori matched vertices, i.e., the so-called *seed* set,[5] are available. However, in this case the de-anonymization procedure does not scale with the number of nodes, due to its exponential complexity. Thus, in this work we consider that de-anonymization is performed with the help of a seed set denoted by $\mathcal{A}_0(n)$ and with cardinality a_0. In particular, we will consider two variants of the problem which differs in the way seeds are assumed to be selected among the n vertices. In the first variant, they are assumed to be selected at wish, using just information on the vertex degree. In the second, we assume that they are selected uniformly at random among all vertices.

C. Overview of the DDM Algorithm

Before providing a high-level description of our matching algorithm (DDM), we remark that we limit our study to de-anonymization procedures that assume the availability of a seed set and can be mapped into bootstrap percolation processes, such as the PGM (percolation graph matching) procedure in [6]. Such procedure has been shown to be successful in the case of Erdös–Rényi graphs. In essence, PGM maintains a mark counter, initialized to zero, for any pair $[*_1, *_2] \in \mathcal{P}(\mathcal{G}_{\rm g})$ that can still potentially be matched. The counter is increased by one whenever the candidate pair becomes *neighbor* of an already matched pair. Among the candidate pairs whose counter is larger than or equal to a fixed threshold r, the algorithm selects one uniformly at random, adding it to the set of matched pairs. After this, counters are updated. Note that some candidate pairs might have to be permanently discarded because they are conflicting with previously matched pairs. The algorithm proceeds until no more pairs can be matched. Of course seeds will be matched irrespective of their mark counter. The PGM algorithm, although potentially suboptimal, has linear complexity and is simple enough that its performance can be predicted using known results from bootstrap percolation [9], establishing a lower bound on the number of seeds required to correctly match almost all vertices. A more formal description of the PGM algorithm is given in Alg. 1, where:

- $\mathcal{B}_t(\mathcal{G}_{\rm g})$ is the set of pairs in $\mathcal{P}(\mathcal{G}_{\rm g})$ that at time step t have already collected at least r marks. It is composed of $\mathcal{B}'_t(\mathcal{G}_{\rm g})$ and $\mathcal{B}''_t(\mathcal{G}_{\rm g})$, comprising good and bad pairs, respectively.[6]

[4]Our results generalize to the case of weights being random variables.

[5]We will refer to the seed set as a subset of vertices, or, equivalently, of good vertex pairs, that have been identified a-priori.

[6]The dependency of the sets $\mathcal{A}_t(\mathcal{P}(\mathcal{G}_{\rm g}))$, $\mathcal{B}_t(\mathcal{P}(\mathcal{G}_{\rm g}))$, $\mathcal{Z}_t(\mathcal{P}(\mathcal{G}_{\rm g}))$ on the generic pairs graph $\mathcal{P}(\mathcal{G}_{\rm g})$ is simply indicated by $\mathcal{A}_t(\mathcal{G}_{\rm g})$, $\mathcal{B}_t(\mathcal{G}_{\rm g})$, $\mathcal{Z}_t(\mathcal{G}_{\rm g})$, respectively, and it is dropped whenever not strictly necessary.

Algorithm 1 The PGM Algorithm

$\mathcal{A}_0 = \mathcal{B}_0 = \mathcal{A}_0(n)$, $\mathcal{Z}_0 = \emptyset$, $t = 0$
while $\mathcal{A}_t \setminus \mathcal{Z}_t \neq \emptyset$ **do**
  $t = t + 1$
  Randomly select a pair $[*_1, *_2] \in \mathcal{A}_{t-1} \setminus \mathcal{Z}_{t-1}$ and add one mark to all neighbor pairs of $[*_1, *_2]$ in $\mathcal{P}(\mathcal{G}_\mathrm{g})$.
  Let $\Delta\mathcal{B}_t$ be the set of all neighbor pairs of $[*_1, *_2]$ in $\mathcal{P}(\mathcal{G}_\mathrm{g})$ whose mark counter has reached threshold r at time t.
  Construct set $\Delta\mathcal{A}_t \subseteq \Delta\mathcal{B}_t$ as follows. Order the pairs in $\Delta\mathcal{B}_t$ in an arbitrary way, select them sequentially and test them for inclusion in ΔA_t:
  if the selected pair in $\Delta\mathcal{B}_t$ has no conflicting pair in $\mathcal{A}_{t-1}$ or $\Delta\mathcal{A}_t$ **then**
    Insert the pair in $\Delta\mathcal{A}_t$
  else
    Discard it
  $\mathcal{Z}_t = \mathcal{Z}_{t-1} \cup [*_1, *_2]$, $\mathcal{B}_t = \mathcal{B}_{t-1} \cup \Delta\mathcal{B}_t$, $\mathcal{A}_t = \mathcal{A}_{t-1} \cup \Delta\mathcal{A}_t$
return $T = t$, $\mathcal{Z}_T = \mathcal{A}_T$

- $\mathcal{A}_t(\mathcal{G}_\mathrm{g})$ is the set of matchable pairs at time t. In general, $\mathcal{A}_t(\mathcal{G}_\mathrm{g})$ and $\mathcal{B}_t(\mathcal{G}_\mathrm{g})$ do not coincide as $\mathcal{B}_t(\mathcal{G}_\mathrm{g})$ may include conflicting pairs that are not present in $\mathcal{A}_t(\mathcal{G}_\mathrm{g})$. $\mathcal{A}_t(\mathcal{G}_\mathrm{g})$ includes two subsets of good and bad pairs, denoted by $\mathcal{A}'_t(\mathcal{G}_\mathrm{g})$ and $\mathcal{A}''_t(\mathcal{G}_\mathrm{g})$, respectively.
- $\mathcal{Z}_t(\mathcal{G}_\mathrm{g})$ is the set of pairs in $\mathcal{A}_{t-1}(\mathcal{G}_\mathrm{g})$ that have been matched at time t. By construction, $|\mathcal{Z}_t(\mathcal{G}_\mathrm{g})| = t \ \forall t$.

Let T be the time step at which the algorithm terminates. Note that, since $|\mathcal{Z}_t(\mathcal{G}_\mathrm{g})| = t \ \forall t$, $|\mathcal{A}_T(\mathcal{G}_\mathrm{g})| = |\mathcal{Z}_T(\mathcal{G}_\mathrm{g})| = T$.

In our work, since we want to establish the minimum number of required seeds by means of bootstrap percolation theory, we keep the simplicity of the PGM algorithm, adding some fundamental improvements to exploit the heterogeneity of vertex degrees. Before explaining our approach, we make the following observations on the PGM algorithm described above for Erdös–Rényi graphs. First, in PGM pairs are selected irrespective of the degree of their constituting vertices. Indeed, in Erdös–Rényi graphs this is not so important since vertices degree (which is binomially distributed) is highly concentrated around the mean, and all matchable pairs are essentially equivalent. Second, there exists a unique threshold r, common to all pairs, which is a fixed parameter of the algorithm subject to the constraint $r \geq 4$.

Our DDM algorithm for power-law graphs is based instead on partitioning the pairs in $\mathcal{P}(\mathcal{G}_\mathrm{g})$ based on the degree of their constituent vertices. The sub-graph corresponding to a partition is called *slice*. Then the algorithm employs a careful expansion of the set of matched pairs through the various slices, using different thresholds r and seed sets at the various stages of the process.

In particular, we first isolate a specific slice $\mathcal{P}_1$ of $\mathcal{P}(\mathcal{G}_\mathrm{g})$, induced by vertices having large (but not too large) degree. $\mathcal{P}_1$ includes pairs whose vertices have weights between $\alpha_1 = n^\gamma$ and $\alpha_2 = n^\gamma/2$, where γ is a constant (slightly) smaller than $1/2$. This slice is somehow the crucial one: we show that its percolation triggers the entire matching process, as the identification of all other vertices in the network follows easily after we correctly match all pairs in $\mathcal{P}_1$. Note that degrees of vertices in $\mathcal{P}_1$ are fairly homogeneous (a constant factor of difference), so that the results for Erdös–Rényi graphs can be applied to this slice, using a proper threshold r.

Vertices having degree smaller than those in $\mathcal{P}_1$ are partitioned in geometric slices $\mathcal{P}_k$ including vertex pairs with weights between α_k and $\alpha_{k+1} = \alpha_k/2$, with $k \geq 2$. Then, a top-down cascading process is unrolled starting from $\mathcal{P}_1$ and using a proper threshold r, where matched pairs in a slice are used as seeds to identify the good pairs in the slice below, and so on. Vertices with very large degree are identified at the end, using as seed set a properly defined subset of previously matched pairs with relatively small degree.

Here we have provided just the basic idea of our DDM algorithm: many subtleties must be addressed to show its correctness. Among them, we emphasize the problem that the DDM algorithm has no direct access to vertex weights (i.e., it does not know the original degree of a vertex in $\mathcal{G}_\mathrm{g}$), and can only make use of the observable vertex degrees in $\mathcal{G}_1$ and $\mathcal{G}_2$. As a consequence, in our analysis we need to show that our matching algorithm is sufficiently robust also in the presence of imperfect (estimated) vertex partitioning.

At last, we remark that, when $\bar{w}$ is finite, a non-negligible fraction of good pairs cannot be identified, no matter which matching algorithm is used. This fact can be immediately grasped by observing that any good pair $[i_1, i_2]$ can be identified only if both i_1 and i_2 have at least r neighbors in $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. Clearly, since the average node degree is assumed to be constant, a non-vanishing fraction of vertices in $\mathcal{G}_\mathrm{g}$ gives origin to vertices with degree smaller than the required r in either $\mathcal{G}_1$ or $\mathcal{G}_2$.

III. Preliminary Results

In this section we first introduce some useful properties of the PGM algorithm applied to Erdös-Rényi graphs, extending in part results already derived in [6]. We then present some related, preliminary results that will be used in the analysis of our DDM algorithm for Chung-Lu graphs. Specifically, we consider a subgraph of a Chung-Lu graph whose vertices have similar weight (i.e., a slice) and prove that:

- such slice can be bounded from below and above by two Erdös-Rényi graphs including the same vertices and with properly defined edge probabilities (Proposition 1). This implies that also the pair graph of the Chung-Lu slice is upper and lower bounded by the pair graphs of the corresponding Erdös-Rényi graphs (Proposition 1);
- the PGM algorithm successfully percolates over the slice, matching all good pairs under properly defined conditions on the seed set. This is proven for a slice that includes all and only the pairs whose vertex weights fall within a predefined range (Theorem 2), as well as for a slice whose pairs have been selected based on the estimated weights (Corollary 3).

A. On Erdös-Rényi Graphs

We first recall the result in [6, Th. 1].

Theorem 1: Let the ground-truth graph be an Erdös-Rényi random graph $G(n,p)$. Let $r \geq 4$ and the critical seed set size be defined as:

$$a_c = \left(1 - \frac{1}{r}\right)\left(\frac{(r-1)!}{n(ps^2)^r}\right)^{\frac{1}{r-1}}. \quad (2)$$

For $n^{-1} \ll ps^2 \leq s^2 n^{-\frac{4}{r}}$, we have: that, if $a_o/a_c \to a > 1$, the PGM algorithm matches a number of good pairs equal to $|\mathcal{A}'_T| = n - o(n)$ w.h.p. Furthermore, $\mathcal{A}''_T = \emptyset$ w.h.p.

Observe that under the assumptions of Theorem 1, we have $T = |\mathcal{A}_T| = |\mathcal{A}'_T| = n - o(n)$. The two corollaries below, which can be derived from the arguments presented in [6], strengthen the result in Theorem 1 and will come in handy in the following.

Corollary 1: For any $\epsilon > 0$, define $t_0 = \min\left(T, \frac{n^{-3/r-\epsilon}}{(ps)^2}\right)$. Then, $\mathcal{B}''_{t_0} = \emptyset$ w.h.p.

The proof of this corollary is reported in the Supplemental Material.

When $t_0 = T$, the corollary guarantees that $\mathcal{A}''_T \subseteq \mathcal{B}''_T = \emptyset$, i.e., no bad pairs are matched by the PGM algorithm. When $t_0 < T$ $\left(\text{i.e., for } p \gg \sqrt{\frac{n^{-3/r-\epsilon-1}}{s^2}}\right)$, we complement the above statement with the corollary below.

Corollary 2: Under the conditions of Theorem 1, for $p \gg \sqrt{\frac{n^{-3/r-1}}{s^2}}$, let $t_0 = \frac{n^{-3/r-\epsilon}}{(ps)^2}$ for any $0 < \epsilon < \frac{1}{r}$. Then, $|\mathcal{B}'_{t_0}| = n$ w.h.p.

The fact that, for some $t_0 < T$, $|\mathcal{B}'_{t_0}| = n$ and $\mathcal{B}''_{t_0} = \emptyset$ jointly occur w.h.p. implies that the PGM algorithm matches all the good pairs (i.e., $|\mathcal{A}'_T| = n$ and $\mathcal{A}''_T = \emptyset$) w.h.p. This is because, by construction, $\mathcal{A}'_{t_0} = \mathcal{B}'_{t_0}$. Indeed, $\mathcal{B}'_{t_0}$ contains no conflicting pairs and none of the pairs in $\mathcal{B}'_{t_0}$ can be blocked by previously matched bad pairs since $\mathcal{B}''_{t_0} = \emptyset$.

B. On Chung-Lu Graphs

We now extend the above results to Chung-Lu graphs. First we introduce the key concepts of *perfect slices* and *increasing* property.

Definition 1: A *perfect slice* is a subgraph of $\mathcal{G}_{\rm g}(\mathcal{V},\mathcal{E})$, denoted by ${\mathcal{G}_{\rm g}}^x(\mathcal{V}_x,\mathcal{E}_x)$, that includes vertices with weights exactly in a properly defined range $[w_{\min}, w_{\max}]$. Similarly, define $\mathcal{G}_1^x$ and $\mathcal{G}_2^x$ as perfect slices of, respectively, $\mathcal{G}_1$ and $\mathcal{G}_2$. Finally, define a *perfect slice* of $\mathcal{P}(\mathcal{G}_{\rm g})$, $\mathcal{P}_x$, as the the pairs graph corresponding to ${\mathcal{G}_{\rm g}}^x$.

In the following, with abuse of language, we will use the term perfect slice $\mathcal{P}_x$ also to indicate the pairs in $\mathcal{P}_x$. Furthermore, we will denote by N_x the number of nodes in slice $\mathcal{G}_g^x$, i.e., $N_x = |\mathcal{V}_x|$. We observe that a perfect slice $\mathcal{P}_x$ can also be obtained by directly extracting from $\mathcal{P}(\mathcal{G}_{\rm g})$ the subgraph induced by those pairs whose vertices have weights in the range $[w_{\min}, w_{\max}]$.

Definition 2: Let $\mathcal{H}(\mathcal{V},\mathcal{E}_H)$ and $\mathcal{K}(\mathcal{V},\mathcal{E}_K)$ be two random graphs insisting on the same set of vertices $\mathcal{V}$, where $\mathcal{E}_H \subseteq \mathcal{E}_K$, i.e., $\mathcal{E}_H$ can be obtained by sampling $\mathcal{E}_K$. We define the following partial order relationship: $\mathcal{H}(\mathcal{V},\mathcal{E}_H) \leq_{st} \mathcal{K}(\mathcal{V},\mathcal{E}_K)$. Then, we can define a vertex property $\mathcal{R}$ satisfied by a subset of vertices, and denote with $\mathcal{R}(\mathcal{H}) \subseteq \mathcal{V}$ the set of vertices of $\mathcal{H}$ that satisfy property $\mathcal{R}$. We say that $\mathcal{R}$ is *monotonically increasing with respect to the graph ordering relation "$\leq_{st}$"* if $\mathcal{R}(\mathcal{H}) \subseteq \mathcal{R}(\mathcal{K})$ whenever $\mathcal{H} \leq_{st} \mathcal{K}$.

In our case, for any $0 \leq t \leq T$, sets $\mathcal{B}_t$, $\mathcal{B}'_t$, $\mathcal{B}''_t$ are all monotonic increasing with respect to relationship "$\leq_{\rm st}$" defined on the pairs graph $\mathcal{P}(\mathcal{G}_{\rm g})$. Below, we use this observation and show that, in the case of Chung-Lu graphs, a properly defined perfect slice $\mathcal{P}_x$ can be lower and upper bounded (w.r.t. "$\leq_{\rm st}$" relation) by pairs graphs corresponding to opportunely defined Erdös-Rényi graphs.

Proposition 1: Given a perfect slice ${\mathcal{G}_{\rm g}}^x(\mathcal{V}_x,\mathcal{E}_x)$ obtained from a Chung-Lu graph $\mathcal{G}_{\rm g}$ with $N_x = |\mathcal{V}_x|$, we have $G(N_x, p_{\min}) \leq_{st} {\mathcal{G}_{\rm g}}^x \leq_{st} G(N_x, p_{\max})$ where $G(N_x, p_{\min})$ and $G(N_x, p_{\max})$ are Erdös-Rényi graphs with $p_{\min} = w_{\min}^2/(n\bar{w})$ and $p_{\max} = w_{\max}^2/(n\bar{w})$. Furthermore, considering the corresponding pairs graphs $\mathcal{P}_x$, $\mathcal{P}(G(N_x, p_{\min}))$ and $\mathcal{P}(G(N_x, p_{\max}))$, it holds $\mathcal{P}(G(N_x, p_{\min})) \leq_{st} \mathcal{P}_x \leq_{st} \mathcal{P}(G(N_x, p_{\max}))$.

Proof: Let $X_{i,j}$ be the indicator function that is equal to 1 if vertex $i_x \in \mathcal{V}_x$ has an edge with a generic other vertex $j_x \in \mathcal{V}_x$, and it is equal to 0 otherwise. Also, $\mathbb{E}[X_{i,j}] = p_{i,j}$.

In order to prove the first assertion, we build $G(N_x, p_{\min})$ by independently sampling the edges included in $\mathcal{G}_x(\mathcal{V}_x,\mathcal{E}_x)$ with an appropriate probability value. Similarly, we build $G(N_x, p_{\max})$ by adding to $\mathcal{G}_x(\mathcal{V}_x,\mathcal{E}_x)$ edges with a properly selected probability value.

Specifically, let us focus on a generic pair of nodes (i,j) of the perfect slice $\mathcal{G}_g^x$. We denote by $Y_{\min}$ the indicator function that is equal to 1 if edge (i,j) exists in $G(N_x, p_{\min})$ and it is equal to 0 otherwise. We define $Y_{\min} = X_{i,j} Z_{\min}$ where $Z_{\min}$ is a $\text{Ber}\left(\frac{p_{\min}}{p_{i,j}}\right)$. Then, we have $\mathbb{E}[Y_{\min}] = \mathbb{E}[X_{i,j}]\mathbb{E}[Z_{\min}] = p_{\min}$, i.e., $Y_{\min}$ is $\text{Ber}(p_{\min})$. Furthermore, by construction $Y_{\min} \leq X_{i,j}$. The same rationale holds for any nodes pair in $\mathcal{G}_x(\mathcal{V}_x,\mathcal{E}_x)$, thus $G(N_x, p_{\min})$ will include only a fraction of the edges in $\mathcal{E}_x$ (since $Y_{\min} = 0$ if $X_{i,j} = 0$).

Similarly, let $Y_{\max}$ be the indicator function that is equal to 1 if edge (i,j) exists in $G(N_x, p_{\max})$, and it is equal to 0 otherwise. We define $Y_{\max} = X_{i,j} + (1 - X_{i,j}) Z_{\max}$ where $Z_{\max}$ is a $\text{Ber}\left(\frac{p_{\max} - p_{i,j}}{1 - p_{i,j}}\right)$. Then, we have $\mathbb{E}[Y_{\max}] = p_{\max}$. Again, the same rationale holds for any nodes pair in $\mathcal{G}_x(\mathcal{V}_x,\mathcal{E}_x)$, thus the edges in $\mathcal{E}_x$ will represent a subset of the edges in $G(N_x, p_{\max})$.

The second assertion is an immediate consequence of the fact that, by construction, the "$\leq_{st}$" relationship is transferred from graphs $\mathcal{G}_x(\mathcal{V}_x,\mathcal{E}_x)$, $G(N_x, p_{\min})$ and $G(N_x, p_{\max})$, to the corresponding pairs graphs. ■

Next, we present our first main result, which shows that the PGM algorithm can successfully match all good pairs in a properly defined perfect slice in the case of Chung-Lu graphs.

Theorem 2: Consider ${\mathcal{G}_{\rm g}}^x$ as defined in Proposition 1. Applying the PGM algorithm to $\mathcal{P}_x$ guarantees that $|A_T({\mathcal{G}_{\rm g}}^x)| = N_x$ and $\mathcal{A}''_T({\mathcal{G}_{\rm g}}^x) = \emptyset$ w.h.p., provided that:

(i) $N_x \to \infty$ as $n \to \infty$;

(ii) $p_{\min} = w_{\min}^2/(n\bar{w})$ satisfies: $p_{\min} \gg \sqrt{\frac{N_x^{-3/r-1}}{s^2}}$;

(iii) $p_{\max} = w_{\max}^2/(n\bar{w})$ satisfies: $p_{\max} \leq N_x^{-\frac{4}{r}}$;

(iv) $w_{\max}/w_{\min} = 2$;

(v) $\liminf_{N_x \to \infty} a_o/a_c > 1$, with a_c computed from (2) by setting $p = p_{\min}$.

Under conditions *(i)-(v)*, PGM successfully matches w.h.p. all the good pairs (with no errors) also in any subgraph ${\mathcal{G}_{\rm g}}^{x'}$ of ${\mathcal{G}_{\rm g}}^{x}$ that comprises a finite fraction of vertices of ${\mathcal{G}_{\rm g}}^{x}$ and all the edges between the selected vertices.

Proof: First observe that, if we find $t_0 = o(N_x)$ such that $\mathcal{B}''_{t_0}({\mathcal{G}_{\rm g}}^{x}) = \emptyset$ w.h.p., then we have w.h.p that $\forall t \leq t_0$:

$$|\mathcal{A}_t({\mathcal{G}_{\rm g}}^{x})| = |\mathcal{B}'_t({\mathcal{G}_{\rm g}}^{x})| \overset{(a)}{\geq} |\mathcal{B}'_t(G(N_x, p_{\min}))| \overset{(b)}{=} |\mathcal{A}_t(G(N_x, p_{\min}))| \overset{(c)}{>} t. \quad (3)$$

In (3), inequality (a) holds by monotonicity of sets $\mathcal{B}'_t$ with respect to "$\leq_{\rm st}$", while equality (b) descends from Theorem 1. Inequality (c) descends from the following argument. Denoted by $T_G = \min\{t \, s.t. \, |\mathcal{A}_t(G(N_x, p_{\min})| = t\}$, by Theorem 1 we have $T_G = N_x - o(N_x)$. Since $t_0 = o(N_x)$, $t_0 < T_G$, i.e., $|\mathcal{A}_t(G(N_x, p_{\min}))| > t$ for $t \leq t_0$. From (3), we immediately get $t_0 < T$, with $T = \min\{t \, s.t. \, |\mathcal{A}_t({\mathcal{G}_{\rm g}}^{x})| = t\}$.

Now, let us define, for an arbitrarily small $\epsilon > 0$, $t_0 = \frac{N_x^{-3/r-\epsilon}}{(p_{\max} s)^2}$; observe that by construction $t_0 = o(N_x)$. We prove that $\mathcal{B}''_{t_0}(\mathcal{G}_0) = \emptyset$ exploiting the monotonicity of $\mathcal{B}''_{t_0}$ with respect to "$\leq_{\rm st}$". Indeed $|\mathcal{B}''_{t_0}({\mathcal{G}_{\rm g}}^{x})| \leq |\mathcal{B}''_{t_0}(\mathcal{G}(N_x, p_{\max}))|$ (by Proposition 1), with $\mathcal{B}''_{t_0}(\mathcal{G}(N_x, p_{\max})) = \emptyset$ w.h.p. as immediate consequence of Corollary 1 (recall that $N_x \to \infty$ as $n \to \infty$). Furthermore, by Corollary 2, for an arbitrary $0 < \epsilon < 1/r$, define $t_1 = \frac{N_x^{-3/r-\epsilon}}{(p_{\min} s)^2} = o(n_0)$. We have: $|\mathcal{B}'_{t_1}(\mathcal{G}(N_x, p_{\min}))| = N_x$. Next, by monotonicity, we have $|\mathcal{B}'_{t_1}({\mathcal{G}_{\rm g}}^{x})| \geq |\mathcal{B}'_{t_1}(\mathcal{G}(N_x, p_{\min}))| = n_0$ (by Proposition 1), provided that $t_1 \leq T$.

At last, since $p_{\max}/p_{\min} = 4$, we can always choose an $\epsilon < \epsilon$ such that $T > t_0 = \frac{N_x^{-3/r-\epsilon}}{(p_{\max} s)^2} > \frac{N_x^{-3/r-\epsilon}}{(p_{\min} s)^2} = t_1$. Thus, since $\mathcal{B}'_{t_0}({\mathcal{G}_{\rm g}}^{x})$ is by construction non decreasing with t, we have: $|\mathcal{B}'_{t_0}({\mathcal{G}_{\rm g}}^{x})| \geq |\mathcal{B}'_{t_1}({\mathcal{G}_{\rm g}}^{x})| = N_x$. In conclusion, there exists a $t_0 < T$ such that $|\mathcal{B}'_{t_0}({\mathcal{G}_{\rm g}}^{x})| = N_x$ and $\mathcal{B}''_{t_0}({\mathcal{G}_{\rm g}}^{x}) = \emptyset$. Hence, $|\mathcal{A}'_T({\mathcal{G}_{\rm g}}^{x})| = |\mathcal{A}'_{t_0}({\mathcal{G}_{\rm g}}^{x})| = |\mathcal{B}'_{t_0}({\mathcal{G}_{\rm g}}^{x})| = N_x$ and $|\mathcal{A}''_T({\mathcal{G}_{\rm g}}^{x})| = |\mathcal{B}''_{t_0}({\mathcal{G}_{\rm g}}^{x})| = 0$. The extension of previous results to ${\mathcal{G}_{\rm g}}^{x'}$ is immediate in light of the fact that ${\mathcal{G}_{\rm g}}^{x'}$ inherits all properties of ${\mathcal{G}_{\rm g}}^{x}$. ■

Clearly, to build perfect slices it is necessary to have direct access to vertex weights. In practice, graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be sliced by our algorithm only according to the observable vertices degree. The vertex weight inferred from the observable degree will be referred to as estimated weight. Specifically, given a vertex i_1 in $\mathcal{G}_1$ with observable degree D_1^i, the estimated weight associated to it is $\hat{w}_i^1 = D_1^i/s$. By slicing $\mathcal{G}_1$ on the basis of such estimated weights, it is clear that each slice may include vertices with different weight than expected. A similar observation holds for $\mathcal{G}_2$. We therefore define the generic *imperfect* pair graph slice (or, simply, imperfect slice) $\hat{\mathcal{P}}_x$, as a subgraph of $\mathcal{P}(\mathcal{G}_{\rm g})$ composed of vertices with estimated weights in the range $[w_{\min}, w_{\max}]$. Imperfect slices can be built so that the following three conditions are satisfied:

1) Only pairs formed by vertices whose actual weight is in the interval $[w_{\min}, w_{\max}]$ are included in $\hat{\mathcal{P}}_x$;
2) A finite fraction (bounded away from 0) of good pairs of $\mathcal{P}_x$ is included in $\hat{\mathcal{P}}_x$;
3) The following situation occurs with negligible probability: a bad pair $[i_1, j_2]$ is included in $\hat{\mathcal{P}}_x$ while none of the pairs $[i_1, i_2]$ and $[j_1, j_2]$ are included.

The third condition ensures that every bad pair in $\hat{\mathcal{P}}_x$ conflicts with at least one good pair in $\hat{\mathcal{P}}_x$, thus it will not be matched by the PGM algorithm when it (eventually) reaches the threshold. In the Supplemental Material we show that imperfect slices satisfying the above conditions can be built, provided that:

$$\frac{w_{\max}}{w_{\min}} = 2 \quad \text{and} \quad w_{\max} > 2\frac{65}{(\epsilon')^2} \log n \text{ with } \epsilon > 0. \quad (4)$$

Considering imperfect slices, the following corollary immediately follows from Theorem 2.

Corollary 3: Under the same conditions as in Theorem 2, PGM can be successfully applied to an imperfect slice $\hat{\mathcal{P}}_x \subset \mathcal{P}_x$ comprising a finite fraction of the pairs in $\mathcal{P}_x$ and satisfying the following constraint: a bad pair $[i_1, j_2] \in \mathcal{P}_x$ is included in $\hat{\mathcal{P}}_x$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also in $\hat{\mathcal{P}}_x$.

Proof: Essentially the scheme of Theorem 2 can be repeated to show that there exists $t_1 < T$ such that $\mathcal{B}_{t_1}(\hat{\mathcal{P}}_x)$ comprises all the good pairs in $\hat{\mathcal{P}}_x$ and no bad pairs. First observe that $\hat{\mathcal{P}}_x$ can be always transformed into $\mathcal{P}({\bar{\mathcal{G}}_{\rm g}}^{x})$ for some ${\bar{\mathcal{G}}_{\rm g}}^{x}$ by adding and removing only bad pairs. Second, from Theorem 2 we know that, for $t_1 = \frac{(\bar{N}_x)^{-3/r-\epsilon}}{(p_{\min} s)^2} = o(\bar{N}_x)$, it holds: $\mathcal{B}'_{t_1}(\mathcal{P}({\bar{\mathcal{G}}_{\rm g}}^{x})) = \bar{N}_x$ where $\bar{N}_x$ denotes the number of vertices in ${\bar{\mathcal{G}}_{\rm g}}^{x}$ (equal, by construction, to the number of good pairs in $\hat{\mathcal{P}}_x$). Third, again from Theorem 2, it holds that $\mathcal{B}''_{t_1}(\mathcal{P}({\bar{\mathcal{G}}_{\rm g}}^{x})) = \emptyset$.

Hence, if we prove that $\mathcal{B}''_{t_1}(\hat{\mathcal{P}}_x) = \emptyset$, we can conclude that $\mathcal{B}'_{t_1}(\hat{\mathcal{P}}_x) = \bar{N}_x$ since condition $\mathcal{B}''_{t_1}(\hat{\mathcal{P}}_x) = \emptyset$ necessarily implies $\mathcal{B}'_t(\hat{\mathcal{P}}_x) = \mathcal{B}'_t(\mathcal{P}({\bar{\mathcal{G}}_{\rm g}}^{x}))$ for every $t \leq t_1$. Indeed, by construction, the subgraphs of $\hat{\mathcal{P}}_x$ and $\mathcal{P}({\bar{\mathcal{G}}_{\rm g}}^{x})$ induced by their good pairs are identical. To prove that $\mathcal{B}''_{t_1}(\hat{\mathcal{P}}_x) = \emptyset$, we can repeat the same arguments as in the proof of Corollary 1: we upper-bound the number of marks collected at time t by every bad pair $[i_1, j_2] \in \hat{\mathcal{P}}_x$ with a binomial r.v. $\text{Bi}(t, p_{\max}^2 s^2)$ and, then, proceed exactly as in the proof of Corollary 1 to show that $\mathbb{P}\{\mathcal{B}''_{t_1}(\hat{\mathcal{P}}_x) \neq \emptyset\} \to 0$. The assert then follows from the observation that, given condition 3) reported above, no bad pairs can be matched for $t > t_1$, because they will be necessarily blocked by a previously matched good pair. ■

IV. DDM Algorithm

Here we present an overview and a detailed description of our DDM algorithm.

A. Graph Slicing and Pairs Matching

In order to ensure that the matching process of good pairs successfully percolates without errors, we perform a *logical* partition of the pairs graph into perfect slices, as reported below. Note that such slices are defined based on the order of magnitude of the nodes degree. Indeed, depending on the nodes degree, the matching process evolves differently, i.e., it requires a different number of already matched pairs

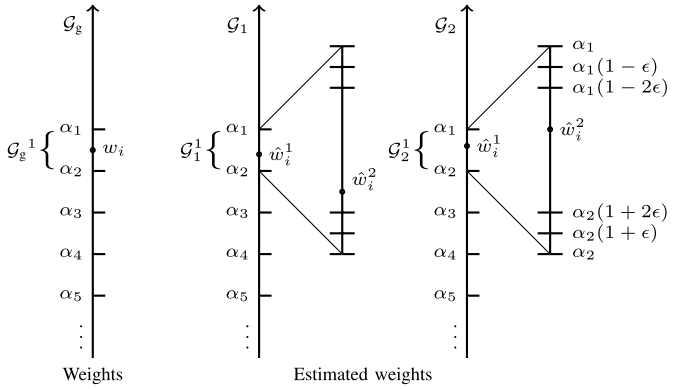

Fig. 1. Graphical representation of slice partitioning. In the case of $\mathcal{G}_1$ and $\mathcal{G}_2$ the figure also highlights the sub-intervals within a slice that are used in the proof given in in the Supplemental Material.

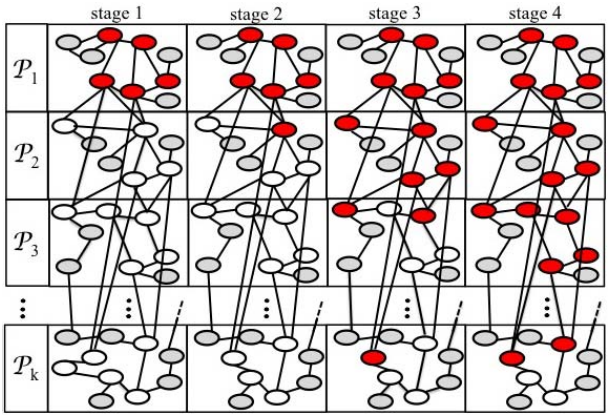

Fig. 2. Example of the percolation process on the pairs graph, for slices $\mathcal{P}_2, \ldots, \mathcal{P}_k$. Columns correspond to different stages of the process. Grey and white pairs denote bad and good pairs, respectively. Red pairs represent the initial seeds in the first stage and matched pairs in the following stages.

and a different expression for the threshold r. The perfect slices partitioning is also depicted for $\mathcal{G}_g$, $\mathcal{G}_1$ and $\mathcal{G}_2$ in Fig. 1.

(I) Slice $\mathcal{P}_0$ including pairs whose vertices have weights between $\alpha_0 = n^{1/2}$ and $\alpha_1 = n^{\gamma}$, with $0 < \gamma < 1/2$ to be determined;

(II) Slice $\mathcal{P}_1$ including pairs whose vertices have weights between $\alpha_1 = n^{\gamma}$ and $\alpha_2 = n^{\gamma}/2$;

(III) Slices $\mathcal{P}_k$ including vertex pairs with weights between α_k and α_{k+1}, with $k \geq 2$, $\alpha_k = \alpha_{k-1}/2$, $\alpha_k > \max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*)$; where $\alpha_k^* = \left(\frac{8\bar{w}\log n}{Cs^2(1-\epsilon)^2}\right)^{\frac{1}{3-\beta}}$ for arbitrarily chosen $0 < \epsilon' < \frac{1}{16}$ and $0 < \epsilon < 1$. We denote with $\mathcal{K}$ the set of all k.

(IV) Slices $\mathcal{P}_h$ including vertex pairs with weights between α_h and α_{h+1}, with $\alpha_h = \alpha_{h-1}/2$ and $\alpha_h \leq \max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*)$ but s.t. $\alpha_h \to \infty$ as $n \to \infty$; We denote with $\mathcal{H}$ the set of all h.

(V) Slices $\mathcal{P}_q$ including vertices with weights between α_q and α_{q+1}, with $\alpha_q = \alpha_{q-1}/2$ and $\lim \alpha_q < \infty$. We denote with $\mathcal{Q}$ the set of all q.

Our algorithm exploits the above logical partition into perfect slices as described below. We remark that, when it is necessary to isolate a specific slice from the rest of the graph, necessarily the algorithm has to be applied to an imperfect slice.

[A)] First, we aim to match good pairs in $\mathcal{P}_1$, assuming that the whole seed set $\mathcal{A}_0$ is included in it. As perfect slices are inaccessible, we extract an imperfect slice $\hat{\mathcal{P}}_1$, which is built by exploiting the procedure described in in the Supplemental Material. As shown there, $\hat{\mathcal{P}}_1$ satisfies conditions 1), 2), 3) stated in Section III. We then run the standard PGM algorithm on $\hat{\mathcal{P}}_1$ using a proper threshold r_1, and show that percolation takes places successfully. At this point, good pairs in $\mathcal{P}_1$ are also matched as a consequence of condition 2 in Section III-B.

[B)] After that, a second imperfect slice, denoted by $\hat{\mathcal{P}}_0$, is defined. Such slice includes pairs of vertices with estimated weight larger than vertices in $\hat{\mathcal{P}}_1$.

[C)] We then consider the subgraph formed by remaining pairs $(\mathcal{P} \setminus \hat{\mathcal{P}}_0 \cup \hat{\mathcal{P}}_1)$ and use the pairs matched in A) as seeds to successfully match the good pairs in slice $\mathcal{P}_2$, by setting a proper threshold r_2. Note that, in this phase of the algorithm, the matching procedure is simpler than PGM, as all pairs with at least r_2 neighboring seeds are matched. Also, besides all good pairs in $\mathcal{P}_2$, some good pairs in the rest of $\mathcal{P} \setminus \hat{\mathcal{P}}_0 \cup \hat{\mathcal{P}}_1$ may be matched as well. The same procedure is iterated, with thresholds r_k properly set at every stage, so that a top-down cascade process is unrolled and all good pairs are matched till slice $\mathcal{P}_k$, with $\alpha_k > \max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*)$. The matching process for slices from $\mathcal{P}_2, \ldots, \mathcal{P}_k$ is sketched in Fig. 2.

[D)] Next, we show all good pairs in $\mathcal{P}_h$ can be matched following a similar approach to that described in C) and using a proper threshold r_h, except for an arbitrarily small fraction of pairs. Likewise, we prove that, in the case of $\mathcal{P}_q$, a fraction of good pairs can be successfully matched.

[E)] Finally, all the remaining unmatched good pairs in $\hat{\mathcal{P}}_0$ can be matched by exploiting the edges with the already matched pairs in $\mathcal{P}_k$ and using a proper threshold r_0.

B. Detailed Description

In more details, our algorithm proceeds as follows. First, it builds $\hat{\mathcal{P}}_1$, which is an imperfect slice.

a) Buildind and deanonymizing slice $\hat{\mathcal{P}}_1$: To this end, let us consider $\mathcal{G}_1$ and $\mathcal{G}_2$, and partition the interval of the nodes weights $[w_{\min}, w_{\max}]$ into three sub-intervals (see Fig. 1). An interval $[w_{\min}(1+2\epsilon), w_{\max}(1-2\epsilon)]$, with $0 < \epsilon < 1/16$, is defined as inner region, the range of values $[w_{\min}(1+\epsilon), w_{\max}(1-\epsilon)] \setminus [w_{\min}(1+2\epsilon), w_{\max}(1-2\epsilon)]$ is defined as outer region, finally the remaining range of values correspond to the forbidden region. The idea is to include in $\hat{\mathcal{P}}_1$ pairs of vertices whose estimated weights fall in either the inner or the outer region, adding the extra constraint that only pairs for which at least one vertex falls in the inner region are included in $\hat{\mathcal{P}}_1$. This expedient implies that $[i_1, j_2]$ is included in $\hat{\mathcal{P}}_1$ only if i_1 (j_2) falls in the inner region and i_2 (j_1) falls in the inner plus outer region. In in the Supplemental Material we show that, through the above procedure, we obtain slices that satisfy conditions 1), 2) and 3) in Section III.

Next, given $\hat{\mathcal{P}}_1$, the DDM algorithm matches all good pairs therein, hence in $\mathcal{P}_1$. This is achieved by applying PGM with threshold $r_1 \geq \frac{4[1+\gamma(1-\beta)]}{1-2\gamma}$, as derived in Proposition 2 in the next section.

Algorithm 2 The DDM Algorithm

Require: $\mathcal{P}(\mathcal{G}_\mathrm{g})$
1: Build imperfect slice $\hat{\mathcal{P}}_1$
2: Apply PGM to $\hat{\mathcal{P}}_1$ with seed set $\mathcal{A}_0$ and properly selected r_1. Obtain a set of matched pairs $\mathcal{R}_1$
3: Build $\hat{\mathcal{P}}_0$
4: **for** $k = 1$ **to** $|\mathcal{K}|$ **do**
5: $\mathcal{S}_0^k = \hat{\mathcal{P}}_{k-1} \cap \mathcal{R}_{k-1}$ and properly select threshold r_k
6: Match all pairs in $\hat{\mathcal{P}}_k$ whose number of neighbors in $\mathcal{S}_0^k$ is greater or equal to r_k. Obtain a set of matched pairs $\mathcal{R}_k$
7: **for** $h = 1$ **to** $|\mathcal{H}|$ **do**
8: $\mathcal{S}_0^h = S_0^{h-1} \cup \mathcal{R}_{h-1}$ and properly select threshold r_h
9: Match all pairs in $\mathcal{P}_h$ whose number of neighbors in $\mathcal{R}_1 \cup \mathcal{R}_2 \ldots \cup \mathcal{R}_{h-1}$ is greater or equal to r_h. Obtain a set of matched pairs $\mathcal{R}_h$
10: **for** $k = 1$ **to** $|\mathcal{Q}|$ **do**
11: $\mathcal{S}_0^q = S_0^{q-1} \cup \mathcal{R}_{q-1}$ and properly select threshold r_q
12: Match all pairs in $\mathcal{P}_q$ whose number of neighbors in $\mathcal{R}_1 \cup \mathcal{R}_2 \ldots \cup \mathcal{R}_{q-1}$ is greater or equal to r_q. Obtain a set of matched pairs $\mathcal{R}_q$
13: $\mathcal{S}_0^0 = \mathcal{S}_0^k$ with k s.t. $\max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*) < \alpha_k < \log^\xi n$, with $\xi < \infty$, and properly select threshold r_0
14: Match all pairs in $\hat{\mathcal{P}}_0$ whose number of neighbors in $\mathcal{S}_0^0$ is greater or equal to r_0. Obtain a set of matched pairs $\mathcal{R}_0$
15: **return** All matched pairs: $\mathcal{R}_0 \cup \mathcal{R}_2 \ldots \cup \mathcal{R}_Q$

b) Buliding slice $\hat{\mathcal{P}}_0$: Then DDM creates a slice that is filled with all the pairs that have not been placed in $\hat{\mathcal{P}}_1$ and for which the estimated weight of at least one vertex exceeds threshold $(\alpha_1 + \alpha_2)/2$.

With abuse of notation, we will refer to this slice as $\hat{\mathcal{P}}_0$. By definition, $\mathcal{P}_0 \subseteq \hat{\mathcal{P}}_0$ and $\mathcal{P}_k \cap \hat{\mathcal{P}}_0 = \emptyset$ for any $k > 2$.

c) Deanonymizing slices $\mathcal{P}_k$: Then, we fix $r_2 = \max\left(\left\lceil \frac{6(4-\beta)}{3-\beta} + 1 \right\rceil, \frac{(\alpha_1)^{4-\beta}}{\sqrt{n}}\right)$ (see Theorem 3) and match all remaining pairs (i.e., pairs in $\mathcal{P}(\mathcal{G}_\mathrm{g}) \setminus (\hat{\mathcal{P}}_1 \cup \hat{\mathcal{P}}_0)$) that have at least r_2 neighbors among the matched pairs in $\hat{\mathcal{P}}_1$. Let us call this set $\mathcal{R}_2$. Note that, in Corollary 4 we show that, given only a finite fraction of matched good pairs in $\mathcal{P}_1$, every good pair in $\mathcal{P}_2$ can be matched, i.e., it belongs to $\mathcal{R}_2$. Furthermore, by monotonicity of the number of neighbors with respect to the vertex weight, it immediately follows that every good pair eventually in $\mathcal{P}_1 \setminus (\hat{\mathcal{P}}_1 \cup \hat{\mathcal{P}}_0)$ also belongs to $\mathcal{R}_2$. Finally, by Theorem 4, we can claim that no bad pair falls in $\mathcal{R}_2$.

Then, using the same procedure described above to build imperfect slices, we "extract" from $\mathcal{R}_2$ a subset $\mathcal{S}_0^2 \subseteq \mathcal{R}_2$ satisfying the following two properties: i) every pair in $\mathcal{S}_0^2$ belongs to $\mathcal{P}_2$, ii) pairs in $\mathcal{S}_0^2$ are a finite fraction of all good pairs in $\mathcal{P}_2$. As the next step, we set r_3 and match all pairs in $\mathcal{P}(\mathcal{G}_\mathrm{g}) \setminus (\mathcal{R}_2 \cup \hat{\mathcal{P}}_1 \cup \hat{\mathcal{P}}_0)$ that have at least r_3 neighbors among the matched pairs in $\mathcal{S}_0^2$. The algorithm is then iterated for every slice $\mathcal{P}_k$, with k such that $\alpha_k > \max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*)$ (see Proposition 3 in the Supplemental Material). So doing, we show that every good pair in $\mathcal{P}_k$ ($k > 1$) is matched, while no bad pairs are matched thanks to Theorem 4.

Observe that the validity of the whole recursion through k is guaranteed again by sub-additivity of probability. Specifically, given that the number of stages is by construction upper bounded by $\frac{\log n}{2}$, for $n > \max(n_{th}^{(1)}/\eta_0, n_{th}^{(2)})$:

$$\mathbb{P}\Big(\exists k \text{ with } \alpha_k > \max\left(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*\right) \Big| \text{ either not all good pairs in } \mathcal{P}_k \text{ are matched or some bad pair is matched}\Big) \leq n^{-1}\log n, \quad (5)$$

which goes to 0 as n grows large.

d) Deanonymizing slices $\mathcal{P}_h$: The DDM algorithm then proceeds by matching the good pairs within slices h. It fixes the threshold to $r_h = \left\lceil \frac{6(4-\beta)}{3-\beta} + 1 \right\rceil$ (see Theorem 5) and starts considering an initial set of matched pairs $\mathcal{S}_0^{h=k^*+1} = \mathcal{R}_{k*}$, where $k^* = \max\{k : \alpha_k > \max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*)\}$. All pairs that have at least r_h neighbors in $\mathcal{S}_0^{h=k^*+1}$ are matched. Let $\mathcal{R}_{h=k^*+1}$ denote the set of matched pairs. By Theorem 5, $\mathcal{R}_{h=k^*+1}$ contains an arbitrarily large fraction of good pairs in $\mathcal{P}_{h=k^*+1}$. Then set $\mathcal{S}_0^h$ is updated according to the recursion: $\mathcal{S}_0^h = \mathcal{S}_0^{h-1} \cup \mathcal{R}_{h-1}$, and, again, the matching procedure is iterated to identify good pairs in the next slice for any h such that $\alpha_h \to \infty$. Theorem 6 guarantees that at no stage of the algorithm any bad pair is matched, while Theorem 5 guarantees that an arbitrarily large fraction of good pairs in $\mathcal{P}_h$ are matched within step h (this because by construction $\mathcal{S}_0^h$ contains an arbitrarily large fraction of good pairs in $\mathcal{P}_{h-1}$).

e) Deanonymizing slices $\mathcal{P}_q$: The same algorithm is then applied to slices q with $q < \log_2 n^\gamma$, in order to identify in each of such slices at least a fraction $f(\alpha_q)$ of good pairs.

f) Deanonymizing slice $\hat{\mathcal{P}}_0$: At last, we set $r_0 = n^{\gamma/2}$ and match pairs in $\hat{\mathcal{P}}_0$ that have at least r_0 neighbors in one of the slices $\mathcal{S}_0^k$, for k satisfying $\max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*) < \alpha_k < \log^\xi n$, with an opportunely defined $\xi < \infty$.

We remark that, at any stage k (with $k > 1$), h and q, the matching procedure can be run on the graph that contains all pairs but $\hat{\mathcal{P}}_0 \cup \hat{\mathcal{P}}_1$, without the necessity to extract from it slices k, h or q. Nevertheless, the procedure to build imperfect slices is adopted at the end of every stage $k \geq 1$ to extract from the set of good pairs matched within stage k a proper subset of pairs to be used as seeds for the following stage.

The simplified DDM pseudo-code is presented in Algorithm 2.

V. Analysis of the Graph De-Anonymization Procedure

Here we analyse the properties of the DDM algorithm and prove the following main results.

- For a sufficiently large seed set, the DDM algorithm successfully matches $\Theta(n)$ good pairs and no bad pairs. Also, it matches all good pairs (except for a negligible fraction) constituted by vertices with sufficiently high weight, i.e., a weight that tends to infinity as $n \to \infty$.
- The above results hold for a seed set as small as n^ϵ (with any arbitrary $\epsilon > 0$) when the seeds can be chosen

based on the vertices observable degree. When, instead, the seeds are uniformly distributed over the graph, $n^{\frac{1}{2}+\epsilon}$ seeds are sufficient.

- In the more general case where seeds are arbitrarily distributed, the key parameter for triggering the good pairs identification process is represented by the size of the set of edges between the seeds and the rest of pairs in the graph.

A. Matching Pairs in $\hat{\mathcal{P}}_1$ and $\mathcal{P}_1$

Our goal is to show that the process of good pair matching percolates on $\hat{\mathcal{P}}_1$ (and thus in $\mathcal{P}_1$ thanks to condition 2 in Section III-B) faster than bad pairs, provided that a sufficient number of seeds ($|\mathcal{A}_0|$) is available in $\hat{\mathcal{P}}_1$. To prove the above statement, we apply Proposition 3.

Proposition 2: Good pairs are successfully matched in $\hat{\mathcal{P}}_1$ if the following conditions are jointly satisfied: a) $\frac{1}{4}-\frac{3}{2r_1} < \gamma < \frac{1}{\beta-1}$, b) $r_1 \geq \frac{4[1+\gamma(1-\beta)]}{1-2\gamma}$ and c) $|\mathcal{A}_0| \gg n^{\frac{(1-2\gamma)r_1+\gamma(\beta-1)-1}{r_1-1}}$.

Proof: We start by showing that conditions *(i)-(v)* of Theorem 2 are met. Then percolation in $\hat{\mathcal{P}}_1$ immediately follows from Corollary 3, in light of the fact that $\hat{\mathcal{P}}_1$ satisfies by hypothesis properties 1) 2) and 3).

First, we compute the number of good pairs in $\mathcal{P}_1$, denoted by N_1, and make sure that N_1 grows to infinity when $n \to \infty$ (as requested by condition *(i)* of Theorem 2).

We have:

$$N_1 = \sum_{i\in\mathcal{V}} \mathbb{1}_{\{w_i\in[\alpha_2,\alpha_1]\}} \approx \int_{\alpha_2}^{\alpha_1} nx^{-\beta}\,\mathrm{d}x = Cn^{1+\gamma(1-\beta)}$$

where C is a proper constant term. Clearly, $N_1 \to \infty$ provided that $1+\gamma(1-\beta) > 0$, i.e., $\gamma < \frac{1}{(\beta-1)}$. Now, probabilities $p_{\min}$ and $p_{\max}$, defined as in Theorem 2, satisfy the following relationship:

$$p_{\min,\max} = \Theta\left(\frac{n^{2\gamma}}{n\bar{w}}\right) = \Theta(n^{2\gamma-1}).$$

To verify condition *(ii)* in Theorem 2, we must enforce: $-\frac{3}{2r_1}-\frac{1}{2} < 2\gamma-1$, thus $\gamma > \frac{1}{4}-\frac{3}{4r_1}$, and to verify condition *(iii)* (i.e., $p_{\max} < N_1^{-\frac{4}{r_1}}$), we must have: $n^{2\gamma-1} \leq n^{[1+\gamma(1-\beta)]4/r_1}$ or, equivalently,

$$r_1 \geq \frac{4[1+\gamma(1-\beta)]}{1-2\gamma}. \tag{6}$$

Next, condition *(iv)* immediately holds by construction. At last, we observe that the critical seed size to match all good pairs in $\hat{\mathcal{P}}_1$ with no errors is given by:

$$\begin{aligned} a_c(N_1) &= \left(1-\frac{1}{r_1}\right)\left(\frac{(r_1-1)!}{N_1 p_{\min}^{r_1}}\right)^{1/(r_1-1)} \\ &= \Theta(n^{\frac{(1-2\gamma)r_1+\gamma(\beta-1)-1}{r_1-1}}). \end{aligned}$$

Thus, condition *(v)* of Theorem 2 is surely satisfied if $|\mathcal{A}_0| \gg n^{\frac{(1-2\gamma)r_1+\gamma(\beta-1)-1}{r_1-1}}$. ∎

We emphasize that the above is one of our main results. Essentially it states that we can choose any $\frac{1}{4} \leq \gamma < \frac{1}{2}$ and determine a minimal r_1 and a minimal $|\mathcal{A}_0|$ for which Proposition 2 holds. Also, if our goal is to minimize $|\mathcal{A}_0|$ (hence, a_c), γ should be chosen as close as possible to $\frac{1}{2}$ (i.e., $\gamma = \frac{1}{2}-\epsilon$ for some small ϵ). Under such condition and for a sufficiently large r_1, we can make the seed set arbitrarily small and still correctly match all pairs.

B. Matching Pairs in $\mathcal{P}_k$

We now consider slices $\{\mathcal{P}_k\}_{k>1}$ and prove that: (i) the process of matching good pairs successfully propagates from one slice to the next and (ii) no errors are made. To this end, we look at the number of edges from every good pair within the k-th slice toward those already matched in the $(k-1)$-th slice and show that the probability that this number is smaller than or equal to a threshold r_k goes to 0 sufficiently fast. At the same time we show that the number of edges from any bad pair within a slice of index greater than $k-1$ toward good pairs in the $(k-1)$-th slice is sufficiently small to guarantee that no bad pairs are matched. At last, we remark that the whole cascading procedure is made slightly more involved by the fact that the matching process of good pairs, starting from the seed set in the $(k-1)$-th slide, cannot be limited to the pairs included in the k-th slide only.

Theorem 3: Consider a generic good pair $[i_1,i_2] \in \mathcal{P}_k$, with vertex weight $w_i \in [\alpha_{k+1},\alpha_k]$ and $1 \leq M < \infty$. Then, for any $\epsilon > 0$, the number of its neighbor good pairs $[l_1,l_2] \in \mathcal{P}_{k-1}$ is greater than $r_k = \max(M, \frac{(\alpha_k)^{4-\beta}}{\sqrt{n}})$ with probability greater than $1-n^{-2}$, as long as $\left(\frac{8\bar{w}\log n}{Cs^2(1-\epsilon)^2}\right)^{\frac{1}{3-\beta}} = \alpha_k^* < \alpha_k < n^{\gamma}$ (with $1/4 < \gamma < 1/2$), and

$$\begin{aligned} n > n_{th}^{(1)} = \max\Bigg\{ &\exp\left[\frac{1}{8}\left(\frac{\bar{w}}{Cs^2}\right)^{2-\beta}\left(\frac{\epsilon}{2M}\right)^{\beta-3}\right], \\ &\left(\frac{2\bar{w}}{Cs^2\epsilon}\right)^{\frac{2}{1-2\gamma}}\Bigg\}. \end{aligned} \tag{7}$$

Also, the above property holds uniformly over the good pairs in $\mathcal{P}_k$ with a probability greater than $1-n^{-1}$, under the same conditions as before on α_k and n.

We remark that it is important to explicitly find the minimum value of n for which the above theorem applies. Indeed, later on we have to show that the same property on the number of neighbors holds uniformly over all considered slices, for sufficiently large n.

Proof: Given two good pairs $[i_1,i_2] \in \mathcal{P}_k$ and $[l_1,l_2] \in \mathcal{P}_{k-1}$, we denote by $\mathbb{1}_{i,l}$ the indicator function associated to the presence of an edge between $[i_1,i_2]$ and $[l_1,l_2]$ in $\mathcal{P}(\mathcal{G}_{\mathrm{g}})$. Note that $\mathbb{E}[\mathbb{1}_{i,l}] \geq \frac{\alpha_{k+1}\alpha_k s^2}{n\bar{w}} = \frac{\alpha_k^2 s^2}{2n\bar{w}} = p_{\min}$, and that $\mathbb{1}_{i,l}$'s are independent r.v.'s Thus, by denoting the number of good pairs in $\mathcal{P}_{k-1}$ by $N_{k-1} = Cn\alpha_k^{(1-\beta)}$, and defining $\mu = N_{k-1}p_{\min} = Cns^2\alpha_k^{1-\beta}\frac{\alpha_k^2}{2n\bar{w}} = \frac{Cs^2\alpha_k^{3-\beta}}{2\bar{w}}$ for any $r_k < \mu$, we have:

$$\begin{aligned} \mathbb{P}\left(\sum_{l\in\mathcal{P}_{k-1}} \mathbb{1}_{i,l} \leq r_k\right) &< \mathbb{P}(\mathbf{Bi}(N_{k-1},p_{\min}) \leq r_k) \\ &\leq \exp(-\delta^2\mu/2) \end{aligned} \tag{8}$$

with $\delta = \frac{\mu-r_k}{\mu}$. In the above derivation, the first inequality descends from the fact that $\sum_{[l_1,l_2]\in\mathcal{P}_{k-1}} \mathbb{1}_{i,l}$ can be stochasti-

cally lower bounded by a sum of N_{k-1} independent Bernoulli r.v.'s with average $p_{\min}$, while the second descends from the Chernoff bound. Now, let us fix $r_k = \max\left(M, \frac{(\alpha_k)^{4-\beta}}{\sqrt{n}}\right) = o(\mu)$. For any $\epsilon > 0$ and choosing $\delta \geq 1-\epsilon$, we have that whenever $r_k = (1-\delta)\mu \leq \epsilon\mu$,

$$\mathbb{P}\left(\sum_{[l_1,l_2]\in\mathcal{P}_{k-1}} \mathbb{1}_{i,l} \leq r_k\right) < \exp(-(1-\epsilon)^2\mu/2).$$

It is straightforward to see that $\exp(-(1-\epsilon)^2\mu/2) < n^{-2}$ provided that $\mu > 4\log n/(1-\epsilon)^2$, which corresponds to $\alpha_k > \left(\frac{8\bar{w}\log n}{Cs^2(1-\epsilon)^2}\right)^{\frac{1}{3-\beta}}$.

Then, we can claim that $\mathbb{P}\left(\sum_{[l_1,l_2]\in\mathcal{P}_{k-1}} \mathbb{1}_{i,l} \leq r_k\right) < n^{-2}$ provided that for some $\epsilon > 0$ jointly $\alpha_k > \alpha_k^* = \left(\frac{8\bar{w}\log n}{Cs^2(1-\epsilon)^2}\right)^{\frac{1}{3-\beta}}$ and $r_k < (1-\delta)\mu = \epsilon\mu$. Recalling the expression of μ as well as that $r_k = \max\left(M, \frac{(\alpha_k)^{4-\beta}}{\sqrt{n}}\right)$, the last condition can be reformulated in terms of n as[7]: $n > n_{th}^{(1)} = \max\left\{\exp\left[\frac{1}{8}\left(\frac{\bar{w}}{Cs^2}\right)^{2-\beta}\left(\frac{\epsilon}{2\left\lceil\frac{6(4-\beta)}{3-\beta}+1\right\rceil}\right)^{\beta-3}\right], \left(\frac{2\bar{w}}{Cs^2\epsilon}\right)^{\frac{2}{1-2\gamma}}\right\}$.

At last, jointly considering all pairs in $\mathcal{P}_k$, the probability that $\sum_{[l_1,l_2]\in\mathcal{P}_{k-1}} \mathbb{1}_{i,l} \leq r_k$ for some $[i_1,i_2] \in \mathcal{P}_k$, is:

$$\begin{aligned}&\mathbb{P}\left(\exists[i_1,i_2]\in\mathcal{P}_k \middle| \sum_{[l_1,l_2]\in\mathcal{P}_{k-1}} \mathbb{1}_{i,l} \leq r_k\right)\\ &\quad\leq \sum_{[i_1,i_2]\in\mathcal{P}_k} \mathbb{P}\left(\sum_{[l_1,l_2]\in\mathcal{P}_{k-1}} \mathbb{1}_{i,l} \leq r_k\right) < nn^{-2} = n^{-1}\end{aligned} \tag{9}$$

provided that jointly $n > n_{th}^{(1)}$ and $\alpha_k^* < \alpha_k < n^\gamma$, as immediate consequence of probability sub-additivity. ∎

We remark that, as a consequence of (4) and of the above result, we have to impose $\alpha_k > \max\left(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*\right)$.

A stronger statement than the previous one is provided by the following Corollary, which ensures that only a fraction of the good pairs matched in the previous slice are enough to guarantee percolation of good pairs in the current slice.

Corollary 4: Consider the good pairs $[i_1,i_2] \in \mathcal{P}_k$, with vertex weight $w_i \in [\alpha_{k+1},\alpha_k]$. Also, consider the subset $\mathcal{S}_0^k \subseteq \mathcal{P}_{k-1}$, such that $|\mathcal{S}_0^k|$ is a fraction ($\eta_k \geq \eta_0 > 0$) of the good pairs in $\mathcal{P}_{k-1}$. Given a generic good pair $[i_1,i_2] \in \mathcal{P}_k$, for any $\epsilon > 0$, with probability greater than $1-n^{-2}$, the number of its neighbor good pairs in $\mathcal{S}_0^k$ is greater than $r_k = \max(M, \frac{(\alpha_k)^{4-\beta}}{\sqrt{n}})$. This result holds as long as $\left(\frac{8\bar{w}\log n}{\eta Cs^2(1-\epsilon)^2}\right)^{\frac{1}{3-\beta}} = \alpha_k^* < \alpha_k < n^\gamma$ (with $1/4 < \gamma < 1/2$), and $n > n_{th}^{(1)}/\eta$. Also, the above property holds uniformly over the good pairs in $\mathcal{P}_k$ with a probability greater than $1-n^{-1}$, under the same conditions as before on α_k and n.

Proof: The proof follows exactly the same lines as the proof of Theorem 3, by replacing N_{k-1} with ηN_{k-1}. ∎

[7]The second term in the right hand side of the inequality can be easily obtained by upper bounding α_k with n^γ.

Similarly, the theorem below proves that the probability that a bad pair has a number of neighbor good pairs greater than or equal to a given threshold tends to zero.

Theorem 4: Consider the bad pairs $[i_1,j_2]$, with vertex weights $w_i, w_j < \alpha_k$, being $\alpha_k < n^\gamma$ ($\gamma < 1/2$). Uniformly over such pairs $[i_1,j_2]$, the number of their neighbor good pairs $[l_1,l_2] \in \mathcal{P}_{k-1}$ is smaller than $r_k = \max\left(M, \frac{(\alpha_k)^{4-\beta}}{\sqrt{n}}\right)$, for $M = \left\lceil\frac{6(4-\beta)}{3-\beta}+1\right\rceil$, with probability greater than $1-n^{-1}$. The result holds for any

$$n > n_{th}^{(2)} = \max\left\{(2C_0)^{\frac{2(4-\beta)}{3-\beta}}, \left(\frac{10Cs^2}{\bar{w}^2}\right)^{\frac{2(4-\beta)}{3-\beta}}, e^{\frac{\left\lceil\frac{6(4-\beta)}{3-\beta}+1\right\rceil(1+\log(2/\min(1,C_0)))}{\left(\left\lceil\frac{6(4-\beta)}{3-\beta}+1\right\rceil-\frac{1}{10}\right)\frac{3-\beta}{2(4-\beta)}-3}}\right\}, \tag{10}$$

where $C_0 = \frac{\bar{w}}{4Cs^2\left\lceil\frac{6(4-\beta)}{3-\beta}+1\right\rceil^{1/(4-\beta)}}$.

Proof: The proof can be found in in the Supplemental Material. ∎

Corollary 4 and Theorem 4 provide the basic tools to show that the DDM algorithm can match all good pairs in slices $\mathcal{P}_k$ for $k \geq 2$, with $\alpha_k > \max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*)$. That is, the good pair matching process successfully percolates from one slice to the next till we reach α_k^*, without requiring a pre-existing seed set in $\mathcal{P}_k$.

C. Matching Pairs in $\mathcal{P}_h$ and $\mathcal{P}_q$

We now turn our attention to vertices with weight $\alpha_h \leq \max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*)$. Previous recursion over the slices cannot be further employed because either Corollary 4 does not apply, or (4) does not hold. Thus DDM is forced to operate in a different way. Before going into the details of how the DDM algorithm operates, we introduce Theorems 5, 6 and 7, which enlighten the basic properties of the graph that will be exploited by the DDM algorithm. In particular, Theorems 5 and 7 provide weaker versions of Theorem 4, while Theorem 6 strengthens the statement of Theorem 4.

Theorem 5: Consider the good pairs $[i_1,i_2] \in \mathcal{P}_h$, with vertex weight $w_i \in [\alpha_{h+1},\alpha_h]$. Also, assume that, for some $\eta > 0$, at least a fraction η of neighbor good pairs, $[l_1,l_2] \in \mathcal{P}_{h-1}$, have been previously identified. Then, for any $0 < \epsilon < 1$, at least a fraction $(1-\epsilon)$ of pairs $[i_1,i_2] \in \mathcal{P}_h$ have a number of neighbors among the identified pairs $[l_1,l_2] \in \mathcal{P}_{h-1}$ greater than $r_h = \left\lceil\frac{6(4-\beta)}{3-\beta}+1\right\rceil$ w.h.p., as long as $\alpha_h \to \infty$.

Proof: The proof can be found in the Supplemental Material. ∎

Furthermore, consider slices in the interval $h \in [h_{\min}, h_{\max}]$, where $h_{\min}$ has been chosen so as to guarantee $\alpha_{h_{\min}} \geq (\frac{8\bar{w}\log n}{Cs^2(1-\epsilon)^2})^{\frac{1}{3-\beta}}$, while $h_{\max}$ is such that that $\alpha_{h_{\max}} \to \infty$. Then a sufficiently large n can be found such that uniformly on $h \in [h_{\min}, h_{\max}]$ we have $\mu_h > \max\left(2r_h, -8\log\frac{\epsilon}{2}\right)$ (i.e., $\exp(-\delta^2\mu_h/2) < \epsilon/2$). This is because, by construction, for every n, μ_h is decreasing with h. Thus, if for a given n the expression $\mu_{h_{\max}} > \max\left(2r_h, -8\log\frac{\epsilon}{2}\right)$ holds, the relationship is

automatically satisfied for any $h < h_{\max}$. Now, for $n \geq n_{th}^{(3)}$, by sub-additivity of probability we can bound the probability that the DDM algorithm at some stage fails to identify at least a fraction $1-\epsilon$ of good pairs. Specifically, the bound is given by: $\sum_{h_{\min}}^{h_{\max}} \exp\Big(-\epsilon^2\Big(1-\frac{\epsilon}{2}\Big)N_h/8\Big) = \sum_{h_{\min}}^{h_{\max}} \exp\Big(-\epsilon^2\Big(1-\frac{\epsilon}{2}\Big)N_{h_{\min}}2^{(h-h_{\min})(\beta-1)}/8\Big) = \Theta(\exp(-\epsilon^2(1-\epsilon/2)N_{h_{\min}+1}/8)) \rightarrow 0$. We conclude that, for any $\epsilon > 0$, we can iteratively identify at least a fraction $1-\epsilon$ of good pairs jointly in all slices w.h.p., as long as for each slice h the assumptions of Theorem 5 are satisfied for some $\eta > 0$.

Theorem 6: Consider the bad pairs $[i_1, l_2]$, with vertex weight $w_i < 2\max(2\frac{65}{(\epsilon')^2}\log(n), \alpha_k^*)$ and $w_l < 2\max(2\frac{65}{(\epsilon')^2}\log(n), \alpha_k^*)$. Uniformly over such pairs $[i_1, l_2]$, for any sufficiently large n with a probability greater than $1-n^{-1}$, the number of their neighbor good pairs $[j_1, j_2]$, with weight $w_j < \alpha_k^*$ is smaller than $r_h = \left\lceil \frac{6(4-\beta)}{3-\beta} + 1 \right\rceil$.

Proof: The proof can be found in the Supplemental Material. ∎

Theorem 7: Consider the good pairs $[i_1, i_2] \in \mathcal{P}_q$, with vertex weight $w_i \in [\alpha_{q+1}, \alpha_q]$. A finite fraction $f(\alpha_q)$ $(0 < f(\alpha_q) < 1)$ of such pairs have a number of neighbors among the identified pairs $[l_1, l_2] \in \mathcal{P}_{q-1}$ greater than $r_q = \left\lceil \frac{6(4-\beta)}{3-\beta} + 1 \right\rceil$, with a probability at least $1-n^{-1}$. This result holds provided that at least a fraction $f(\alpha_{q-1}) \geq f(\alpha_q)$ of neighbor good pairs $[l_1, l_2] \in \mathcal{P}_{q-1}$ (i.e., pairs whose vertices have weight $w_j \in [\alpha_q, \alpha_{q-1}]$) have been previously identified. The above property holds for properly selected values of $f(\alpha_q)$, whenever $\alpha_q > (\frac{32\bar{w}}{Cs^2 f(\alpha_q)})^{\frac{1}{3-\beta}}$ and $n > \frac{2\alpha_q^{\beta-1}}{10^4 Cs^2 f(\alpha_q)}$.

Proof: The proof can be found in the Supplemental Material. ∎

D. Matching Pairs in $\mathcal{P}_0$ and $\hat{\mathcal{P}}_0$

Theorem 8 guarantees that every good pair in $\hat{\mathcal{P}}_0$ (and, thus, in $\mathcal{P}_0$) is matched, while no bad pairs are matched. This holds provided that $r_0 = n^{\gamma/2}$ and one of the slices $\mathcal{S}_0^k$ is used as seed set, for k satisfying $\max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*) < \alpha_k < \log^{\xi} n$, with an opportunely defined $\xi < \infty$.

Theorem 8: Consider a generic good pair $[i_1, i_2] \in \hat{\mathcal{P}}_0$ and a slice $\mathcal{P}_k$ satisfying $\max(2\frac{65}{(\epsilon')^2}\log n, \alpha_k^*) < \alpha_k < \log^{\xi} n$, for some $\xi < \infty$. For a sufficiently large n, with probability greater than $1-n^{-1}$, the number of good pairs $[l_1, l_2] \in \mathcal{S}_0^k$ (with $\mathcal{S}_0^k \subseteq \mathcal{P}_k$ and $\frac{|\mathcal{S}_0^k|}{|\mathcal{P}_k|} > \eta > 0$) that are neighbors of $[i_1, i_2]$ is greater than $r_0 = n^{\gamma/2}$. Also, for sufficiently large n, with probability greater than $1-n^{-2}$, the number of neighbor good pairs $[l_1, l_2] \in \mathcal{P}_k$ of a bad pair $[i_1, j_2] \in \hat{\mathcal{P}}_0$ is smaller than r_0. The above properties hold uniformly over all good pairs in $\hat{\mathcal{P}}_0$ w.h.p.

Proof: The proof can be found in the Supplemental Material. ∎

VI. Uniformly Distributed Seeds

So far we assumed that all the initial seeds in $\mathcal{A}_0$ belong to $\mathcal{P}_1$. Now, we show that DDM can properly percolate when seeds are uniformly distributed over the slices. Note that, although the uniform distribution is likely one of the most relevant cases, our results hold for any arbitrary distribution of the seeds over the graph. We start introducing the key parameter that characterizes the ability to start the bootstrap percolation process over $\mathcal{P}_1$ (and then over the whole $\mathcal{P}$).

Definition 3: We denote the set of edges between the seed set $\mathcal{A}_0$ and the rest of pairs $\mathcal{P}(\mathcal{G}_{\mathrm{g}}) \setminus \mathcal{A}_0$, by $\partial\mathcal{A}_0$.

Then we can prove the following result.

Theorem 9: Whenever the seed set $\mathcal{A}_0$ is chosen in such a way that:

$$|\partial\mathcal{A}_0| \gg n^{\gamma + \frac{(1-2\gamma)r_1 + \gamma(\beta-1)-1}{r_1-1}}, \tag{11}$$

our DDM algorithm percolates identifying $\Theta(n)$ good pairs.

Proof: First, by exploiting the monotonicity property of the percolation process, we show that a properly dimensioned set of seeds belonging to slice $\mathcal{P}_k$, $k > 1$, is at least equivalent to a single seed belonging to $\mathcal{P}_1$, i.e., it has an equivalent or stronger impact on the evolution of mark counters of other pairs. Similar arguments can be used to show that a group of seeds in $\mathcal{P}_1$ behaves as a seed in $\mathcal{P}_0$.

More formally, we start by considering, as before, the evolution of the PGM algorithm operating with a seed set $\mathcal{A}_0$ of pairs in $\mathcal{P}_1$. Then we compare it to the evolution of a modified version of the PGM algorithm operating on a seed set $\mathcal{A}_0^*$, which differs from $\mathcal{A}_0$ in that a fraction of seeds in $\mathcal{P}_1$ is replaced with groups of seeds in $\mathcal{P}_k$. The modified version of the PGM algorithm handles every group of seeds belonging to $\mathcal{P}_k$ as a single seed (i.e., all the seeds in the same group are selected by the algorithm at the same time and simultaneously included in $\mathcal{Z}$). Also, while proceeding, the two versions of the algorithm process exactly the same sequence of seeds. We show that, by properly setting the size of the group of seeds in $\mathcal{P}_k$, denoted by S_k, we can guarantee that the process of matching good pairs percolates faster starting from $\mathcal{A}_0^*$ than from $\mathcal{A}_0$.

Consider a generic good pair $[i_1, i_2]$ in $\mathcal{P}_1$. Note that, by construction, the number of edges between $[i_1, i_2]$ and a given good pair $[l_1, l_2] \in \mathcal{A}_0$ is either 0 or 1. The probability that such edge exists in $\mathcal{P}(\mathcal{G}_{\mathrm{g}})$ is upper-bounded by $p_{1,1} = \frac{w_i \alpha_1}{n\bar{w}}$. Instead, the probability that at least an edge exists between $[i_1, i_2]$ in $\mathcal{P}_1$ and the corresponding group of S_k seeds in $\mathcal{P}_k$ is lower-bounded by $p_{1,S_k} = 1 - (1 - \frac{w_i \alpha_{k+1}}{n\bar{w}})^{S_k}$. By setting $S_k > \frac{\alpha_1}{\alpha_{k+1}} + \epsilon$, for any $\epsilon > 0$, it can be easily shown that, for sufficiently large n, $p_{1,S_k} > p_{1,1}$. In other words, the group of S_k seeds belonging to $\mathcal{P}_k$ in $\mathcal{A}_0^*$ distributes to any good pair in $\mathcal{P}_1 \setminus \mathcal{A}_0$ a number of marks that upper bounds those distributed by the corresponding seed in $\mathcal{A}_0$. This immediately implies that $\mathcal{B}_t'(\mathcal{A}_0^*) \setminus \mathcal{A}_0 \supseteq \mathcal{B}_t'(\mathcal{A}_0) \setminus \mathcal{A}_0$ for any t. Therefore, at t_1 defined as in Theorem 2, $\mathcal{B}_{t_1}'(\mathcal{A}_0^*)$ must necessarily include all pairs in $\mathcal{P}_1 \setminus \mathcal{A}_0$. In addition, it is straightforward to show that every pair in $\mathcal{A}_0 \setminus \mathcal{A}_0^*$ has at least r_1 neighbors among good pairs in $\mathcal{P}_1 \setminus \mathcal{A}_0$ and, thus, it is included in $\mathcal{B}_{t_1}'(\mathcal{A}_0^*)$. Next, we have to show that $\mathcal{B}_{t_1}''(\mathcal{A}_0^*) = \emptyset$. This can be done by following the lines of Theorem 2, i.e., uniformly upper-bounding the probability of adding marks at any time t to bad pairs in $\mathcal{P}_1$, and, then, repeating the arguments of Corollary 1.

To conclude the proof, we underline that in the extreme case where all seeds in $|\mathcal{A}_0|$ are replaced by groups of seeds in $\mathcal{P}_k$, we have $|\mathcal{A}_0^*| = S_k|\mathcal{A}_0|$. The equality corresponds to $|\partial\mathcal{A}_0| = \Theta\left(\frac{\alpha_k}{\bar{w}} S_k|\mathcal{A}_0|\right)$, which yields $|\partial\mathcal{A}_0| = \Theta\left(n^\gamma|\mathcal{A}_0|\right)$. Then, iterating the previous argument for all slices containing seeds and using (7), we get the assertion. ∎

From (11), it immediately descends that, for any choice of seeds, we can correctly match $\Theta(n)$ good pairs provided that the size of the seed set is at least of order $n^{\frac{1}{2}+\epsilon}$, for an arbitrarily small ϵ. This is achieved by choosing γ sufficiently close to $1/2$ and r_1 large enough.

VII. Computational Complexity of the DDM Algorithm

We now address the computational complexity of our algorithm assuming that $\mathcal{G}_1$ and $\mathcal{G}_2$ are given and that we have the list of nodes sorted in decreasing order based on their observable degree. This implies that the selection of nodes within a given slice can be accomplished directly exploiting the above list, without any additional sorting cost.

The algorithmic complexity of the PGM algorithm over $\hat{\mathcal{P}}_1$ is determined by considering that at every step t a good pair is chosen and all its neighbor pairs (which are by construction $O\left((N_1\frac{\alpha_1^2}{n})^2\right)$) are sequentially explored, and eventually placed in $\Delta\mathcal{B}(t)$ and then in $\Delta\mathcal{A}(t)$ (see Alg. 1). All previous steps require a number of operations that is linear with the number of explored pairs (we remark that, in order to reduce the complexity of the algorithm, pairs are dynamically generated as they are explored for the first time). Taking into account that the algorithm stops after at most N_1 steps, the complexity is then given by $O\left(N_1\left(N_1\frac{\alpha_1^2}{n}\right)^2\right)$.

Similarly, the complexity of matching pairs in slice $\mathcal{P}_k$ (respectively, $\mathcal{P}_h$ and $\mathcal{P}_q$) is $O\left(N_k(N_k\frac{\alpha_k^2}{n})^2\right)$ (respectively, $O\left(N_h(N_h\frac{\alpha_h^2}{n})^2\right)$ and $O\left(N_q(N_q\frac{\alpha_q^2}{n})^2\right)$). This is because pairs in $\mathcal{S}_0^k$ (respectively, $\mathcal{S}_0^h$, $\mathcal{S}_0^q$) are to be sequentially selected, and, for each of them, all neighbor pairs must be explored.

The total complexity of our algorithm is given by the sum of the above terms, i.e., $O(n\Delta_2)$ where Δ_2 is the second moment of the node degree. Note that Δ_2 is $O(n^{\frac{3-\beta}{2}})$.

VIII. Experimental Validation

Our results hold asymptotically as the number of nodes tends to infinite, thus it is difficult to validate them considering networks of finite size. Nevertheless, in this section, we show that the dramatic impact of power-law degree on the performance of graph matching algorithms is evident even on small-scale systems. Another important goal of this section is to compare the performance achieved by graph-matching algorithms (PGM or DDM) in Chung-Lu graphs, with the performance obtained in real social networks. Recall that the former only capture effects due to the (marginal) degree distribution of the nodes, while the latter possess several other structural features not reproduced by the Chung-Lu model.

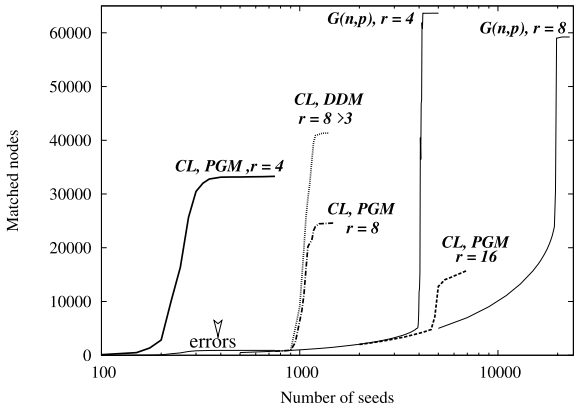

Fig. 3. Number of matched good pairs vs number of seeds, for different graphs and algorithms: Chung-Lu vs $G(n,p)$, PGM vs DDM, fixed $s = 0.7$.

In this section, we first consider synthetic ground-truth (Chung-Lu and $G(n,p)$) graphs containing 63,731 nodes, with average node degree 25.64. The reason for these specific values is that, in the following, we will consider an early snapshot of Facebook having exactly these properties. The considered snapshot, first analyzed in [10], contains (undirected) friendship relationships among users, and it is publicly available at [11]. The maximum node degree is 1,098, and the power law exponent, estimated using the maximum-likelihood approach [12], is 2.94.

Our evaluation proceeded as follows. First, we generated a Chung-Lu graph using a sequence of node weights ω_i exactly matched to the degree sequence of the Facebook snapshot. By so doing, we obtained a synthetic graph of the kind analyzed in this paper, and which at least preserves the degree distribution of a real social network. We also generated a $G(n,p)$ graph with the same number of nodes and average node degree, as reference. The above two graphs (either the Chung-Lu or the $G(n,p)$) were taken as the ground-truth[8] $\mathcal{G}_\mathrm{g}$, and we used an edge sampling probability $s = 0.7$ to generate $\mathcal{G}_1$ and $\mathcal{G}_2$. On these two systems, we compared the performance of the original PGM algorithm and the DDM algorithm. Results are shown in Fig. 3, where curves labelled CL refer to the Chung-Lu graph. Note that, for the $G(n,p)$ graph, we do not distinguish PGM from DDM, since the two algorithms obtain here the same performance. Seeds are selected uniformly at random among all nodes, and we averaged the results of 1000 runs in which the set of seeds varies. We observed the following facts: i) for fixed r, the critical number of seeds is largely smaller in CL than in $G(n,p)$ (by more than an order of magnitude) confirming the dramatic impact of power-law degree distribution on graph-matching performance; ii) the standard PGM algorithm performs quite well, when we run it on the CL graph. Specifically, the error ratio of PGM was found to be negligible (smaller than 1e-4) for $r = 8$ and $r = 16$, and only around 2.6% with $r = 4$ (the appearance of such non-negligible errors is explicitly highlighted in Fig. 3 by an arrow); iii) errors are essentially eliminated by PGM with

[8]Ideally, one would like to assess the performance of graph matching algorithms given real graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, but this is hard to do without knowledge of the ground-truth. Therefore, to check the validity of our results, we used a known graph (synthetic or real) as ground-truth.

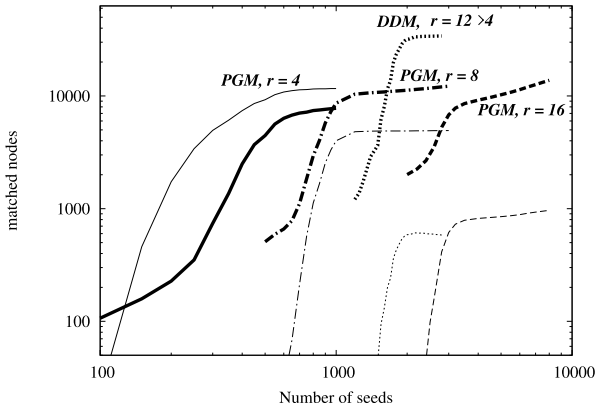

Fig. 4. Number of good (thick lines) and bad (thin lines) matches vs number of seeds, using the Facebook snapshot as ground-truth. Comparison between DDM and PGM, for fixed $s = 0.7$.

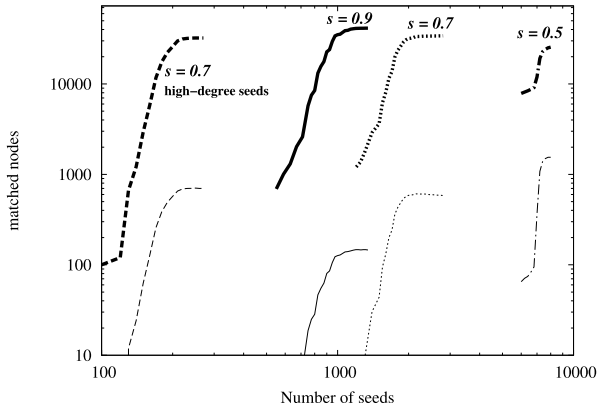

Fig. 5. Number of good (thick lines) and bad (thin lines) matched pairs for DDM ($r = 12 \to 4$), in the Facebook graph, for different values of thinning probability s and seed selection strategy.

sufficiently large r, but at the expense of reducing the final number of correctly matched pair, since nodes having degree smaller than r either in $\mathcal{G}_1$ or in $\mathcal{G}_2$ cannot be matched by PGM; iv) the DDM algorithm, by using different r for different nodes, can easily overcome the above limitation of PGM, as can be seen by the curved labelled[9] DDM ($r = 8 \to 3$).

A rather different picture emerged when we took the real Facebook snapshot as the ground-truth graph. Results are shown in Fig. 4, where we used a log y scale to better show the extent of the errors. In the plot, we adopt the same line style for a given version of algorithm (with specified parameters), and we use thick curves to represent correctly matched pairs, and thin curves to represent the corresponding errors. We show the performance of three versions of PGM (with $r = 4, 8, 16$) and one version of DDM, with $r = 12 \to 4$. We observed the following facts: i) PGM produces a very large number of errors, especially for small r. In particular, for $r = 4$, the number of errors even exceeds the number of correct matches: PGM makes so many errors that it fails to reach all matchable pairs. With $r = 8$, the error ratio is still very high (29%), while for $r = 16$ it is about 6.7%; ii) the position of the phase transition is roughly the same in the Facebook system and in the corresponding CL graph (i.e., around 1,000 for $r = 8$); iii) DDM ($r = 12 \to 4$) performs quite well, correctly matching more than 30K nodes with error ratio around 1.7% (one can use different choices of $r_{\max}$ and $r_{\min}$ to trade-off error ratio, critical number of seeds and total number of matched pairs; we do not show these trade-offs for lack of room).

We believe that the main reason why the Facebook graph is much harder to match than its CL counterpart is related to its large clustering coefficient (14.8%). Indeed, we have shown in a parallel work [13] that clustering can severely degrade the performance of graph matching algorithms based on bootstrap percolation. This suggests that, in the case of real social networks with significant degrees of clustering, more sophisticated algorithms are necessary to bring down the errors to an acceptable rate.

[9]Hereinafter, we will denote by DDM ($r = r_{\max} \to r_{\min}$) a version of DDM which employs $r = r_{\max}$ in the first slice, and progressively reduces the threshold down to $r = r_{\min}$ to match low-degree nodes.

At last, we show in Fig. 5 the impact of the thinning probability s and the seed selection strategy. In particular, we consider the Facebook system and the DDM algorithm ($r = 12 \to 4$). The curve labelled $s = 0.7$ is the same as in Fig. 4. We tried both a larger value of $s = 0.9$ and a smaller value $s = 0.5$. Moreover, in the case of $s = 0.7$, we investigated what happens when seeds are selected uniformly at random among nodes with degree larger than or equal to 100 (labelled by "high-degree seeds"). As expected: i) as s decreases, the critical number of seeds increases, as well as the error ratio (for $s = 0.5$, the error ratio of our algorithm is 5.7%, suggesting that the matching problem becomes harder and harder to solve (if not impossible) for small s (in line with theoretical results in [2]; ii) far fewer seeds are necessary when one can select seeds among high-degree nodes, confirming the fact that what really matters is the cardinality of the set of edges going out of the seed set, rather than the number of seeds alone.

IX. Related Work

In this section we first summarize the contribution of [14] on bootstrap percolation in random graphs. Then we limit ourselves to mentioning papers that have proposed and analysed algorithms to solve the network de-anonymization problem in the case of the random graph model introduced in [2], i.e., when $\mathcal{G}_1$ and $\mathcal{G}_2$ are obtained by independent edge sampling of the unknown ground-truth graph $\mathcal{G}_{\rm g}$. For a more general overview on network de-anonymization, we recommend the bibliography in [6] and [4].

The work in [14] analyses bootstrap percolation in Chung-Lu graphs. Similarly to our work, it considers different slices of the graph, each including nodes with different degree. It then leverages the results by Janson *et al.* [9] and the monotonicity property of bootstrap percolation, in order to show that percolation successfully occurs. Unlike [14], however, network de-anonymization has to deal with both good and bad pairs, and it turns out that the aforementioned monotonic property holds for both of them. This leads to the need for a different algorithm and a significantly more complex analysis.

For the case in which $\mathcal{G}_{\rm g}$ is an Erdös–Rényi graph, [2] finds sufficient conditions on p and s under which users can in principle be identified even in the absence of seeds.

In particular, graphs whose average vertex degree is of order $\log n$ (or larger) can potentially be de-anonymized when s is $\Theta(1)$. The analysis of [2] is independent of the specific matching algorithm and consists in showing that the solution of the network de-anonymization problem corresponds w.h.p. to the unique minimum of a specific objective function, under the above mentioned conditions. A similar approach has been adopted in [4] to show the conditions on the sampling probability under which graph de-anonymization can be successfully performed, either exactly or with an arbitrarily small percentage of errors, in the case of configuration graphs. The work in [4] also presents a greedy approximation to the optimal algorithm, which, unlike the optimal one, is computationally feasible (it has polynomial complexity). Another attempt to devise a polynomial time algorithm for matching two similar graphs without side information (i.w., without seeds) has been done in [15].

In [6] the PGM algorithm has been proposed as a simple and scalable algorithm for network de-anonymization, when an initial set of seeds is available. By exploiting bootstrap percolation results in [9], the authors carry out an asymptotic analysis of the minimal number of seeds needed to almost perfectly de-anonymize the network in the case in which $\mathcal{G}_{\mathrm{g}}$ is an Erdös–Rényi random graph, showing in particular the existence of a phase transition.

An algorithm to reconcile scale-free networks has been recently proposed in [5]. Differently from us, in [5] authors do not identify any phase transition effect related to bootstrap percolation. Actually, they consider a simple direct identification strategy that requires $\Omega(\frac{n}{\log n})$ seeds and essentially prove that their algorithm is unlikely to match bad pairs. Also, their analysis is complicated by the adoption of the preferential attachment model by Barabási and Albert [16], whereas here we adopt a different model of scale-free networks (the Chung-Lu model) that greatly simplifies the analysis.

Finally, a preliminary version of our work has appeared in [17]. In parallel and independently from us, another network de-anonymization algorithm has been recently proposed and analyzed in [18], adopting exactly the same Chung-Lu model considered in this paper. Even if the algorithm in [18] exhibits significant differences with respect to our DDM algorithm, it achieves essentially the same performance in terms of the asymptotic minimum number of seeds needed to guarantee an almost perfect network de-anonymization.

X. Conclusions

We analytically investigated the de-anonymization problem in social networks represented by scale-free graphs, by exploiting bootstrap percolation results and a novel graph slicing technique. Our main finding is that, to successfully identify most of the nodes, the seed set can be as small as n^{ϵ} (for any $\epsilon > 0$) when seeds are properly selected, and of the order of $n^{\frac{1}{2}+\epsilon}$ when they are uniformly distributed among the nodes. Our asymptotic results have been empirically validated by simulation experiments on synthetic graphs as well as on a Facebook snapshot. Our work can be extended along several directions, including the effect of spurious edges in the network graphs, that of erroneously selected seeds and partially overlapping node sets, as done in [8] for Erdös–Rényi graphs.

References

[1] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proc. 30th IEEE Symp. Secur. Privacy*, May 2009, pp. 173–187.

[2] P. Pedarsani and M. Grossglauser, "On the privacy of anonymized networks," in *Proc. 17th SIGKDD*, 2011, pp. 1235–1243.

[3] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, 2012, pp. 628–637.

[4] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural data de-anonymization: Quantification, practice, and implications," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, 2014, pp. 1040–1053.

[5] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," in *Proc. VLDB*, 2014, pp. 377–388.

[6] L. Yartseva and M. Grossglauser, "On the performance of percolation graph matching," in *Proc. 1st ACM Conf. Online Social Netw.*, 2013, pp. 119–130.

[7] F. Chung and L. Lu, "The average distance in a random graph with given expected degrees," *Internet Math.*, vol. 1, no. 1, pp. 91–113, 2004.

[8] E. Kazemi, S. H. Hassani, and M. Grossglauser, "Growing a graph matching from a handful of seeds," in *Proc. Endowment Int. Conf. Very Large Data Bases (VLDB)*, 2015, pp. 1010–1021.

[9] S. Janson, T. Łuczak, T. Turova, and T. Vallier, "Bootstrap percolation on the random graph $G_{n,p}$," *Ann. Appl. Probab.*, vol. 22, no. 5, pp. 1989–2047, 2012.

[10] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in Facebook," in *Proc. 2nd WONS*, 2009, pp. 37–42.

[11] *Facebook Friendships Network Dataset—KONECT*. (May 2015). [Online]. Available: http://konect.uni-koblenz.de/networks/facebook-wosn-links

[12] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, 2009.

[13] C.-F. Chiasserini, M. Garetto, and E. Leonardi, "Impact of clustering on the performance of network de-anonymization," in *Proc. ACM COSN*, 2015, pp. 83–94.

[14] H. Amini and N. Fountoulakis, "Bootstrap percolation in power-law random graphs," *J. Statist. Phys.*, vol. 155, no. 1, pp. 72–92, 2014.

[15] P. Pedarsani, D. R. Figueiredo, and M. Grossglauser, "A Bayesian method for matching two similar graphs without seeds," in *Proc. IEEE 51st Allerton*, Oct. 2013, pp. 1598–1607.

[16] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[17] C. F. Chiasserini, M. Garetto, and E. Leonardi, "De-anonymizing scale-free social networks by percolation graph matching," in *Proc. INFOCOM*, 2015, pp. 1–9.

[18] K. Bringmann, T. Friedrich, and A. Krohmer, "De-anonymization of heterogeneous random graphs in quasilinear time," in *Proc. 22nd Annu. Eur. Symp. Algorithms (ESA)*, 2014, pp. 197–208.

Carla-Fabiana Chiasserini (M'98–SM'09) received the Ph.D. degree from the Politecnico di Torino in 2000, where she is currently an Associate Professor. She has authored over 260 papers at major venues. She serves as an Associate Editor of several prestigious journals. Her research interests include protocols and performance analysis of wireless networks.

Michele Garetto (M'04) received the Dr.Ing. degree in telecommunication engineering and the Ph.D. degree in electronic and telecommunication engineering from the Politecnico di Torino, Italy, in 2000 and 2004, respectively. In 2002, he was a Visiting Scholar with the Networks Group, University of Massachusetts, Amherst, MA, USA, and in 2004, he held a post-doctoral position with the Electrical and Computer Engineering Department, Rice University, Houston, TX, USA. He is currently an Assistant Professor with the University of Torino, Italy.

Emilio Leonardi (M'99–SM'09) received the Dr.Ing. degree in electronics engineering and the Ph.D. degree in telecommunications engineering from the Politecnico di Torino in 1991 and 1995, respectively. He is currently a Professor with the Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino. His research interests are in the field of performance evaluation of complex networks.