On the Feasible Region of Efficient Algorithms for Attributed Graph Alignment

Ziao Wang*, Ning Zhang*, Weina Wang†, and Lele Wang*
*University of British Columbia, Vancouver, BC V6T1Z4, Canada, {ziaow, ningz, lelewang}@ece.ubc.ca
†Carnegie Mellon University, Pittsburgh, PA 15213, USA, weinaw@cs.cmu.edu

2022 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-2159-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISIT50566.2022.9834398

***Abstract*—Graph alignment aims at finding the vertex correspondence between two correlated graphs, a task that frequently occurs in graph mining applications such as social network analysis. Attributed graph alignment is a variant of graph alignment, in which publicly available side information or attributes are exploited to assist graph alignment. Existing studies on attributed graph alignment focus on either theoretical performance without computational constraints or empirical performance of efficient algorithms. This motivates us to investigate efficient algorithms with theoretical performance guarantee. In this paper, we propose two polynomial-time algorithms that exactly recover the vertex correspondence with high probability. The feasible region of the proposed algorithms is *near optimal* compared to the information-theoretic limits. When specialized to the seeded graph alignment problem, the proposed algorithms *strictly improve* the best known feasible region for exact alignment by polynomial-time algorithms.**

I. Introduction

The graph alignment problem, also referred to as the graph matching or noisy graph isomorphism problem, is the problem of finding the correspondence between the vertices of two correlated graphs. This problem has been given increasing attention for its applications in social network de-anonymization. For instance, datasets of social networks are typically anonymized for privacy protection. However, an attacker may be able to de-anonymize the dataset by aligning its user-user connection graph with that of publicly available data. *Attributed graph alignment* is a variant of graph alignment in which side information, referred to as attributes of vertices, is also publicly available in addition to the user-user connection information. This variant is motivated by the largely available information on social network users in practice such as education background, hobbies, and birthplaces.

In this paper, we focus on the attributed graph alignment problem under the attributed Erdős–Rényi pair model $\mathcal{G}(n, p, s_{\mathrm{u}}; m, q, s_{\mathrm{a}})$ first proposed in [1]. In this model, a base graph G is generated on the vertex set $[n+m]$ where the vertices from the set $[n]$ represent *users* and the rest of the vertices represent *attributes*. Between each pair of users, an edge is generated independently and identically with probability p to represent their connection. For each user-attribute pair, an edge is generated independently and identically with probability q to represent their association. Note that there are no edges between attributes. The graph G is then independently subsampled to two graphs G_1 and G_2, where each user-user edge is subsampled with probability s_{u} and each user-attribute edge is subsampled with probability s_{a}. To model the anonymization procedure, a random permutation Π^* chosen uniformly at random is applied to the *users* in G_2 to generate an anonymized version G_2'. Our goal in this model is to achieve exact alignment, i.e, exactly recovering the permutation Π^* using G_1 and G_2'.

For the attributed graph alignment problem, and the graph alignment problem in general, two often asked questions are the following. First, *for what region of graph statistics is exact alignment feasible with unlimited computational power?* This region is usually referred to as the information-theoretically feasible region or the information-theoretic limits. Second, *for what region of graph statistics is exact alignment feasible with polynomial-time algorithms?* This region is usually referred to as the feasible region of polynomial-time algorithms. Characterizing these two feasible regions and their relationship is of utmost importance to developing a fundamental understanding of the graph alignment problem. For the attributed graph alignment problem, the first question has been partially answered in [1], where the feasible region (achievability results) and infeasible region (converse results) are characterized with a gap in between in some regimes. However, the second question on the feasible region of polynomial-time algorithms has not been studied before, and it is the focus of this paper.

There has been massive study on the graph alignment problem under the Erdős–Rényi pair model without attributes. A line of research focuses on the information-theoretic limits of exact alignment [2, 3, 4, 5]. It is shown that exact alignment is information-theoretically feasible when the intersection graph is dense enough. A sharp threshold of exact alignment has been established, while there still exists some gap between the converse and the achievability results. Another line of research focuses on polynomial-time algorithms for exact alignment [6, 7, 8, 9]. Compared to the information-theoretic limits, the existing polynomial-time algorithms further require high edge correlation between the pair of graphs to achieve exact alignment. The question whether there exist polynomial-time algorithms that achieve the known information-theoretic limits is still left open.

In this work, we consider the attributed graph alignment problem and characterize the feasible regions of two polynomial-time algorithms that we propose. The two algorithms are designed for two different regimes of parameters based on the richness of attribute information: the algorithm AttrRich is designed for the regime where $mqs_{\mathrm{a}}^2 = \Omega(\log n)$, referred to as the attribute-information

rich regime; and the algorithm AttrSparse is designed for the regime where $mqs_{\mathrm{a}}^2 = o(\log n)$, referred to as the attribute-information sparse regime. In both algorithms, we first explore the user-attribute connections to align a set of anchor users, and then utilize the user-user connections to the anchors to align the rest of users. Due to the regime difference, AttrRich is able to generate a much larger set of anchors in the first step than AttrSparse. Therefore, AttrRich and AttrSparse make use of the anchors differently in the second step: AttrRich explores one-hop user-user connections to align the rest of users, while AttrSparse explores multiple-hop user-user connections to align the rest of users.

Our characterizations of the feasible regions of AttrRich and AttrSparse are illustrated in Figure 1 as areas ② and ③, respectively. The information-theoretically feasible and infeasible regions given in [1] are also illustrated in the figure for comparison. We can see that there is a gap between the feasible region achieved by AttrRich and AttrSparse and the known information-theoretically feasible region. It is left open whether this gap is a fundamental limit of polynomial-time algorithms. [1] In addition, we also specialize the attributed graph alignment to the so-called seeded graph alignment problem and show that our proposed algorithms strictly improve the known feasible region of polynomial-time algorithms for the seeded graph alignment problem.

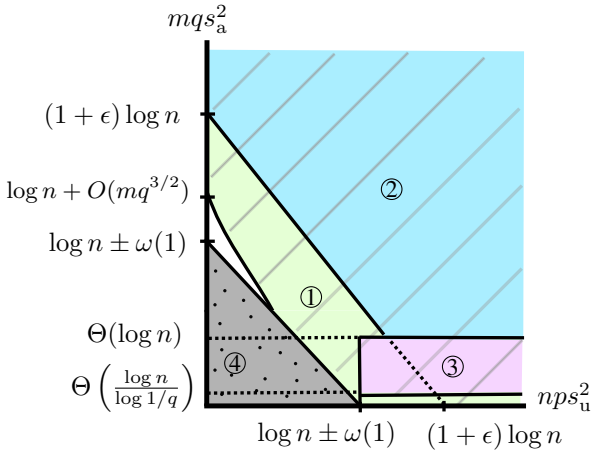

Fig. 1: Comparison between the feasible regions of the proposed algorithms and the information-theoretic limits: the shaded area (①+②+③) represents the information-theoretically feasible region given in [1]; area ② is the feasible region for Algorithm AttrRich and area ③ is the feasible region for Algorithm AttrSparse; area ④ is the information-theoretically infeasible region given in [1].

II. Model

In this section, we describe a random process that generates a pair of correlated graphs, which we refer to as the attributed Erdős–Rényi pair model $\mathcal{G}(n, p, s_{\mathrm{u}}; m, q, s_{\mathrm{a}})$. Under this model, we define the exact alignment problem.

Base graph generation. We first generate a base graph G, whose vertex set $\mathcal{V}(G)$ consists of two disjoint sets, the *user vertex set* $\mathcal{V}^{\mathrm{u}} = \{1, 2, \ldots, n\}$ and the *attribute vertex set* $\mathcal{V}^{\mathrm{a}} = \{n+1, n+2, \ldots, n+m\}$. There are two types of edges in the base graph G, the user-user edges (edges connecting a pair of users) and the user-attribute edges (edges connecting a user vertex and an attribute vertex). The user-user edges are generated independently and identically with probability p, and the user-attribute edges are generated independently and identically with probability q. Throughout this paper, we assume that $p = o(1)$ and $q = o(1)$. We write $i \overset{G}{\sim} j$ if vertices i and j are connected in graph G.

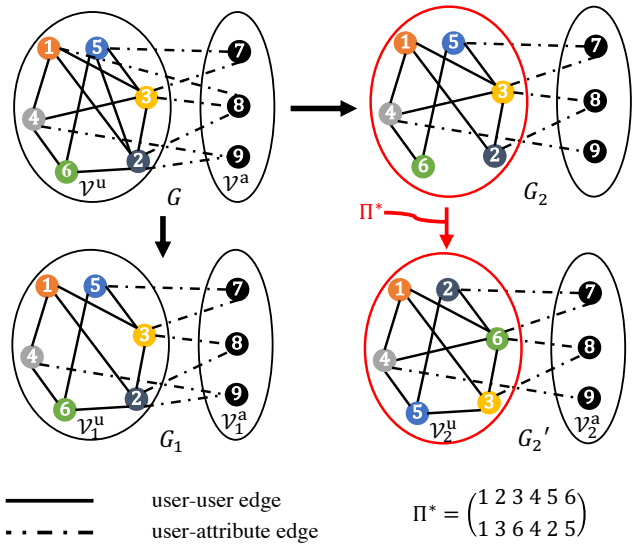

Fig. 2: An illustration of attributed Erdős–Rényi pair model. We first sample a base graph G. Then we get G_1 and G_2 through edge subsampling G. The anonymized graph G_2' is obtained through apply the permutation Π^* on the user vertex set of G_2.

Edge subsampling. From the base graph G, we obtain two correlated graphs G_1 and G_2 by subsampling the edges in G independently. More specifically, we get G_1 and G_2 by independently including each user-user edge in G with probability s_{u} and independently including each user-attribute edge with probability s_{a}. Throughout this paper, we assume that $s_{\mathrm{u}} = \Theta(1)$ and $s_{\mathrm{a}} = \Theta(1)$.

Anonymization. From the G_2 generated as above, we get an anonymized graph G_2' by applying an unknown permutation Π^* on the user vertices of G_2, where Π^* is drawn uniformly at random from the set of all possible permutations on $\mathcal{V}^{\mathrm{u}}$. We use $\mathcal{V}_2^{\mathrm{u}}$ to denote the user vertex set of G_2' and use $\mathcal{V}_1^{\mathrm{u}}$ to denote the user vertex set of G_1. Finally, we remark that this subsampling process is a special case of an earlier described attributed Erdős–Rényi pair model in [1].

Exact alignment. Given an observable pair (G_1, G_2'), our goal is to recover the unknown permutation Π^*, which allows us to recover the original labels of user vertices in the anonymized graph G_2'. We say exact alignment is achieved *with high probability* (w.h.p.) if $\lim_{n\to\infty} \mathsf{P}(\hat{\Pi} \neq \Pi^*) = 0$. It is worth mentioning that $\mathsf{P}(\hat{\Pi} \neq \Pi^*) = \mathsf{P}(\hat{\Pi} \neq \Pi^* | \Pi^* = \pi_{\mathrm{id}})$ due to the symmetry among user vertices. Thus, we later

[1] We comment that efficient algorithms for attributed graph alignment are also studied in the line of work [10, 11, 12], where the focus is the empirical performance rather than the theoretical feasible regions.

assume without loss of generality that the underlying true permutation is the identity permutation.

Relation to the seeded graph alignment problem. Another well-studied graph alignment problem with side information is the seeded graph alignment problem, where we have access to part of the true correspondence between user vertices. To make a comparison between the two models, here we describe the seeded Erdős–Rényi pair model $\mathcal{G}(N, \alpha, p, s)$. We first sample a base graph G from the Erdős–Rényi graph on N vertices with edge probability p. Then two correlated copies G_1 and G_2 are obtained by independently subsampling the edges in the base graph where each edge is preserved with probability s. The anonymized graph G_2' is obtained by applying an unknown permutation Π^* on G_2, where Π^* is drawn uniformly at random. Then, a subset $\mathcal{V}_s \subset \mathcal{V}(G_1)$ of size $\lfloor N\alpha \rfloor$ is chosen uniformly at random and we define the vertex pairs $\mathcal{I}_0 = \{(v_1, \Pi^*(v_1)) : v_1 \in \mathcal{V}_s\}$ as the *seed set*. The graph pair (G_1, G_2') together with the seed set $\mathcal{I}_0$ are given and the goal of the exact alignment is to recover the underlying permutation for the remaining vertices w.h.p.

Comparing the seeded Erdős–Rényi pair model and the attributed Erdős–Rényi pair model, we can see that the seed set and the attribute set both provide side information to assist the alignment of the remaining vertices. Nevertheless, there are two main differences between the two models. First, in the attributed Erdős–Rényi pair model, we allow different edge probabilities and subsampling probabilities for user-user edges and user-attribute edges, whereas in the seeded Erdős–Rényi pair model, the edge probability is identical for all edges and so is the subsampling probability. Second, while there are edges between seeds in seeded Erdős–Rényi pair model, there are no attribute-attribute edges in the attributed Erdős–Rényi pair model. However, it can be shown that the existence of edges between seeds has no influence on the information-theoretic limits for exact alignment in the seeded Erdős–Rényi pair model. This further suggests that the information-theoretic limits on attributed graph alignment recover the information-theoretic limits on seeded graph alignment if we specialize $p = q$ and $s_{\mathrm{u}} = s_{\mathrm{a}}$ in the attributed Erdős–Rényi pair model $\mathcal{G}(n, p, s_{\mathrm{u}}; q, s_{\mathrm{a}})$.

Other notation. Our algorithms rely on exploring the neighborhood similarity of user vertices in G_1 and G_2'. Here we introduce our notation of local neighborhoods. We define $\mathcal{N}_1^{\mathrm{a}}(i) \triangleq \{j \in \mathcal{V}_1^{\mathrm{a}} : i \overset{G_1}{\sim} j\}$ as the set of attribute neighbors of a user vertex i in G_1 and $\mathcal{N}_2^{\mathrm{a}}(i) \triangleq \{j \in \mathcal{V}_2^{\mathrm{a}} : i \overset{G_2'}{\sim} j\}$ as the set of attribute neighbors of a user vertex i in G_2'. For two user vertices i and j in the same graph, let $d(i, j)$ be the length of the shortest path connecting i and j via user-user edges. For a user vertex $i \in \mathcal{V}_1^{\mathrm{u}}$, we define the set of l-hop user neighbors of vertex i as $\mathcal{N}_1^{\mathrm{u}}(i, l) \triangleq \{j \in \mathcal{V}_1^{\mathrm{u}} : d(i, j) \le l\}$ for any positive integer l. By convention, when $l = 1$, we simply write $\mathcal{N}_1^{\mathrm{u}}(i) \equiv \mathcal{N}_1^{\mathrm{u}}(i, 1)$. The quantities $\mathcal{N}_2^{\mathrm{u}}(i, l)$ and $\mathcal{N}_2^{\mathrm{u}}(i)$ are defined similarly for user vertices in G_2'.

Reminder of the Landau notation.

Notation	Definition
$f(n) = \omega(g(n))$	$\lim_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} = \infty$
$f(n) = o(g(n))$	$\lim_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} = 0$
$f(n) = O(g(n))$	$\limsup_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} < \infty$
$f(n) = \Omega(g(n))$	$\liminf_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} > 0$
$f(n) = \Theta(g(n))$	$f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

III. Main results

In this section, we propose two polynomial-time algorithms for the attributed graph alignment problem. Their feasible regions are characterized in the following two theorems.

Theorem 1. *Consider the attributed Erdős–Rényi pair $\mathcal{G}(n, p, s_{\mathrm{u}}; m, q, s_{\mathrm{a}})$ with $p = o(1)$, $q = o(1)$, $s_{\mathrm{u}} = \Theta(1)$, and $s_{\mathrm{a}} = \Theta(1)$. Assume that*

$$mqs_{\mathrm{a}}^2 = \Omega(\log n) \tag{1}$$

and that there exists some constant $\epsilon > 0$ such that

$$mqs_{\mathrm{a}}^2 + nps_{\mathrm{u}}^2 \ge (1 + \epsilon) \log n. \tag{2}$$

Then there exists a polynomial-time algorithm, namely, Algorithm AttrRich *with the parameters chosen in* (8) *and* (9)*, that achieves exact alignment w.h.p.*

Theorem 2. *Consider the attributed Erdős–Rényi pair $\mathcal{G}(n, p, s_{\mathrm{u}}; m, q, s_{\mathrm{a}})$ with $p = o(1)$, $q = o(1)$, $s_{\mathrm{u}} = \Theta(1)$, and $s_{\mathrm{a}} = \Theta(1)$. Assume that*

$$mqs_{\mathrm{a}}^2 = o(\log n), \tag{3}$$

$$nps_{\mathrm{u}}^2 - \log n = \omega(1), \tag{4}$$

and that there exists some constant $\tau > 0$ such that

$$mqs_{\mathrm{a}}^2 \ge \frac{2 \log n}{\tau \log \frac{1}{q}}. \tag{5}$$

Then there exists a polynomial-time algorithm, namely, Algorithm AttrSparse *with the parameters chosen in* (11)*,* (12) *and* (13)*, that achieves exact alignment w.h.p.*

The proofs of Theorems 1 and 2 are omitted due to space limitations; see [13] for the proofs.

A. Algorithm AttrRich

In this subsection, we propose the first algorithm that leads to the feasible region in Theorem 1. This algorithm is designed for the attribute-information rich regime, reflected by the condition $mqs_{\mathrm{a}}^2 = \Omega(\log n)$ in (1), hence named AttrRich.

- **Input:** The graph pair (G_1, G_2') and two thresholds x and y.
- **Step 1: Align through attribute neighbors.** In this step, we only consider the edge connections between users and attributes, and use this information to find the matching for a set of vertices that will be later referred to as anchors. For each pair of users $i \in \mathcal{V}_1^{\mathrm{u}}$ and $j \in \mathcal{V}_2^{\mathrm{u}}$, compute the number of common attribute neighbors

$$C_{ij} \triangleq |\mathcal{N}_1^{\mathrm{a}}(i) \cap \mathcal{N}_2^{\mathrm{a}}(j)|. \tag{6}$$

If $C_{ij} > x$, add (i, j) into $\mathcal{S}_1$. We refer to vertex pairs in the set $\mathcal{S}_1$ as *anchors*. If there exist conflicting pairs in $\mathcal{S}_1$, i.e., two distinct pairs (i_1, j_1) and (i_2, j_2) with $i_1 = i_2$ or $j_1 = j_2$, set $\mathcal{S}_1 = \emptyset$ and declare failure. Otherwise, set $\hat{\Pi}(i) = j$ for all pairs $(i, j) \in \mathcal{S}_1$.

- **Step 2: Align through user neighbors.** In the previous step, we have aligned the anchors using user-attribute edges. In this step, we align the unmatched vertices by their edge connections to the anchors. Let

$$\mathcal{U}_1 \triangleq \{i \in \mathcal{V}_1^{\mathrm{u}} : (i, j) \notin \mathcal{S}_1, \forall j \in \mathcal{V}_2^{\mathrm{u}}\}$$

denote the set of all unmatched vertices in G_1, and let

$$\mathcal{U}_2 \triangleq \{j \in \mathcal{V}_2^{\mathrm{u}} : (i, j) \notin \mathcal{S}_1, \forall i \in \mathcal{V}_1^{\mathrm{u}}\}$$

denote the set of all unmatched vertices in G_2. For each unmatched pair (i, j) with $i \in \mathcal{U}_1$ and $j \in \mathcal{U}_2$, consider the user neighbors of i and the user neighbors of j that are matched as pairs in $\mathcal{S}_1$, and compute the number of such matched pairs

$$W_{ij} \triangleq \sum_{k \in \mathcal{N}_1^{\mathrm{u}}(i), l \in \mathcal{N}_2^{\mathrm{u}}(j)} \mathbb{1}_{\{(k,l) \in \mathcal{S}_1\}}. \tag{7}$$

For each $i \in \mathcal{U}_1$, if $W_{ij} > y|\mathcal{S}_1|$ for a unique $j \in \mathcal{U}_2$, set $\hat{\Pi}(i) = j$. Otherwise, declare failure. If $\hat{\Pi}$ is not a bijection from $\mathcal{V}_1^{\mathrm{u}}$ to $\mathcal{V}_2^{\mathrm{u}}$, declare failure.

- **Output:** The estimated permutation $\hat{\Pi}$.

In this algorithm, there are two threshold parameters x and y. In the analysis of Theorem 1, we choose

$$x = (1 - \delta_x) mqs_{\mathrm{a}}^2, \tag{8}$$

where $1 - \delta_x = \frac{\Delta_x}{\log \frac{1}{q}}$ with constant $\Delta_x \geq \max\{1, \frac{3 \log n}{mqs_{\mathrm{a}}^2}\}$, and

$$y = (1 - \delta_y) ps_{\mathrm{u}}^2, \tag{9}$$

where $1 - \delta_y = \frac{\Delta_y}{\log \frac{1}{p}}$ with constant $\Delta_y \geq 2$.

Remark 1 (Complexity of Algorithm ATTRRICH)**.** The time complexity for computing C_{ij} for all pairs $(i, j) \in \mathcal{V}_1^{\mathrm{u}} \times \mathcal{V}_2^{\mathrm{u}}$ is $O(n^2 m)$ since there are n^2 pairs and for each pair, there are m attributes to consider. The time complexity for computing W_{ij} for all pairs $(i, j) \in \mathcal{U}_1 \times \mathcal{U}_2$ is $O(n^3)$. This is because there are at most n^2 pairs $(i, j) \in \mathcal{U}_1 \times \mathcal{U}_2$, and for each (i, j) pair, computing W_{ij} needs to scan through all pairs $(k, l) \in \mathcal{S}_1$. A necessary condition for the algorithm to execute Step 2 is that there are no conflicting pairs in $\mathcal{S}_1$, which implies that $|\mathcal{S}_1| \leq n$. Therefore, the time complexity of Algorithm ATTRRICH is $O(n^2 m)$ if $m = \omega(n)$, and $O(n^3)$ if $m = O(n)$.

B. Algorithm ATTRSPARSE

In this subsection, we propose the second algorithm that leads to the feasible region in Theorem 2. This algorithm is designed for the attribute-information sparse regime, reflected by the condition $mqs_{\mathrm{a}}^2 = o(\log n)$ in (3), hence named ATTRSPARSE. In Step 2 of this algorithm, we consider two different cases. In the case when the user-user connection is dense, we perform a similar process as in Step 2 of Algorithm ATTRRICH. In the case when the user-user connections is sparse, we call a seeded alignment algorithm proposed in [14], which is restated in Subsection III-C.

- **Input:** The graph pair (G_1, G_2'), three thresholds y, z and η, an integer l, and the model parameters n and p.
- **Step 1: Align through attribute neighbors.** Similar to Step 1 of Algorithm ATTRRICH, for each pair of users $i \in \mathcal{V}_1^{\mathrm{u}}$ and $j \in \mathcal{V}_2^{\mathrm{u}}$, we compute the quantity

$$C_{ij} = |\mathcal{N}_1^{\mathrm{a}}(i) \cap \mathcal{N}_2^{\mathrm{a}}(j)|. \tag{10}$$

Unlike Step 1 of Algorithm ATTRRICH, we create an anchor set using a different threshold z. If $C_{ij} > z$, add (i, j) into $\mathcal{S}_2$. We refer to vertex pairs in the set $\mathcal{S}_2$ as anchors. If there exist conflicting pairs in $\mathcal{S}_2$, i.e., two distinct pairs (i_1, j_1) and (i_2, j_2) with $i_1 = i_2$ or $j_1 = j_2$, set $\mathcal{S}_2 = \emptyset$ and declare failure.

- **Step 2: Align through user-user edges.**
  - If $np > n^{1/7}$, we perform the similar process as in Step 2 of Algorithm ATTRRICH to align the non-anchor vertices. Define

$$\mathcal{U}_3 \triangleq \{i \in \mathcal{V}_1^{\mathrm{u}} : (i, j) \notin \mathcal{S}_2, \forall j \in \mathcal{V}_2^{\mathrm{u}}\}$$

and

$$\mathcal{U}_4 \triangleq \{j \in \mathcal{V}_2^{\mathrm{u}} : (i, j) \notin \mathcal{S}_2, \forall i \in \mathcal{V}_1^{\mathrm{u}}\}.$$

For each unmatched pair $i \in \mathcal{U}_3$ and $j \in \mathcal{U}_4$, compute W_{ij} as defined in (7). For each $i \in \mathcal{U}_3$, if $W_{ij} > y|\mathcal{S}_2|$ for a unique $j \in \mathcal{U}_4$, set $\hat{\pi}(i) = j$. Otherwise, declare failure. If $\hat{\pi}$ is not a bijection from $\mathcal{V}_1^{\mathrm{u}}$ to $\mathcal{V}_2^{\mathrm{u}}$, declare failure.
  - If $np \leq n^{1/7}$, run Algorithm III-C with the induced subgraphs on the user vertices in $\mathcal{V}_1^{\mathrm{u}}$ and $\mathcal{V}_2^{\mathrm{u}}$, the seed set $\mathcal{I}_0 = \mathcal{S}_2$, and parameters l and η.
- **Output:** The estimated permutation $\hat{\Pi}$.

In this algorithm, there are four parameters y, z, l and η that we can choose. In the analysis of Theorem 2, we choose y to be the same value as in (9),

$$z = (1 + \tau) mqs_{\mathrm{a}}^2, \tag{11}$$

(cf. the same τ as in Theorem 2),

$$l = \left\lfloor \frac{(6/7) \log n}{\log(np)} \right\rfloor, \tag{12}$$

and

$$\eta = 4^{2l+2} n^{-2/7}. \tag{13}$$

Remark 2 (Complexity of Algorithm ATTRSPARSE)**.** For the same reason as in Algorithm ATTRRICH, the time complexity of computing C_{ij} for all pairs $(i, j) \in \mathcal{V}_1^{\mathrm{u}} \times \mathcal{V}_2^{\mathrm{u}}$ is $O(n^2 m)$ and that of computing W_{ij} for all pairs $(i, j) \in \mathcal{U}_3 \times \mathcal{U}_4$ is $O(n^3)$. The time complexity of Algorithm III-C is $O(n^{37/7})$ as given in [14], which may be further improved with better data structures. Therefore, if $np > n^{1/7}$, the complexity of Algorithm ATTRSPARSE is $O(n^2 m + n^3)$; otherwise, its complexity is $O(n^2 m + n^{37/7})$.

C. Seeded alignment in the sparse regime [14, Algorithm 3]

Except for the two graphs G_1 and G_2', this algorithm takes a seed set $\mathcal{I}_0$ as input. Recall that the seed set $\mathcal{I}_0$ consists of vertex pairs (i,j) such that $\Pi^*(i)=j$. The algorithm utilizes this seed set to align the remaining vertices.

- **Input:** The graph pair (G_1,G_2'), the seed set $\mathcal{I}_0$, a threshold η, and an integer l.
- **Align high-degree vertices.** Let

$$\mathcal{J}_1 \triangleq \{i\in\mathcal{V}(G_1):(i,j)\neq\mathcal{I}_0,\forall j\in\mathcal{V}(G_2')\},$$

and

$$\mathcal{J}_2 \triangleq \{j\in\mathcal{V}(G_2'):(i,j)\neq\mathcal{I}_0,\forall i\in\mathcal{V}(G_1)\}.$$

For each pair of unseeded vertices $u\in\mathcal{J}_1$ and $v\in\mathcal{J}_2$, and for each pair of their neighbors $i\in\mathcal{N}_1^{\mathrm{u}}(u)\setminus\{u\}$ and $j\in\mathcal{N}_2^{\mathrm{u}}(v)\setminus\{v\}$, compute

$$\lambda_{i,j}^{u,v}=\min_{x\in\mathcal{V}(G_1),y\in\mathcal{V}(G_2')}\Big|\{(k_1,k_2)\in\mathcal{I}_0:\\ k_1\in\mathcal{N}^{\mathrm{u}}_{G_1\setminus\{u,x\}}(i,l),k_2\in\mathcal{N}^{\mathrm{u}}_{G_2'\setminus\{v,y\}}(j,l)\}\Big|,$$

where $\mathcal{N}^{\mathrm{u}}_{G\setminus S}(i_1,l)$ denotes the set of user vertices i_2 such that $d(i_1,i_2)\le l$ in the induced subgraph G with the set of vertices S removed. Let

$$Z_{u,v}=\sum_{i\in\mathcal{N}_1^{\mathrm{u}}(u)\setminus\{u\}}\sum_{j\in\mathcal{N}_2^{\mathrm{u}}(v)\setminus\{v\}}\mathbb{1}_{\{\lambda_{i,j}^{u,v}\ge\eta|\mathcal{I}_0|\}}.$$

If $Z_{u,v}\ge\log n/\log\log n-1$, add (u,v) into set $\mathcal{T}$. Add all the vertex pairs from $\mathcal{I}_0$ to $\mathcal{T}$. If there exist conflicting pairs in $\mathcal{T}$, i.e., two distinct pairs (i_1,j_1) and (i_2,j_2) with $i_1=i_2$ or $j_1=j_2$, set $\mathcal{T}=\emptyset$ and declare failure.
- **Align low-degree vertices.** Let

$$\mathcal{J}_3 \triangleq \{i\in\mathcal{V}(G_1):(i,j)\neq\mathcal{T},\forall j\in\mathcal{V}(G_2')\},$$

and

$$\mathcal{J}_4 \triangleq \{j\in\mathcal{V}(G_2'):(i,j)\neq\mathcal{T},\forall i\in\mathcal{V}(G_1)\}.$$

For all pairs of unmatched vertices $i_1\in\mathcal{J}_3$ and $i_2\in\mathcal{J}_4$, if i_1 is adjacent to a user vertex j_1 in G_1 and i_2 is adjacent to a user vertex j_2 in G_2' such that $(j_1,j_2)\in\mathcal{T}$, then set $\hat{\Pi}(i_1)=i_2$.
- **Finalize and output:** For each vertex pair $(i,j)\in\mathcal{T}$, set $\hat{\Pi}(i)=j$. If $\hat{\Pi}$ is a bijection from $\mathcal{V}(G_1)$ to $\mathcal{V}(G_2')$, output $\hat{\Pi}$, otherwise declare failure.

IV. Discussion

In this section, we briefly summarize the comparison between the feasible region in Theorems 1 and 2 and the existing results. A detailed comparison is referred to [13].

Specialization to the seeded graph alignment. Consider an attributed Erdős–Rényi pair $(G_1,G_2')\sim\mathcal{G}(n,p,s_{\mathrm{u}};m,q,s_{\mathrm{a}})$ with $p=q$ and $s_{\mathrm{a}}=s_{\mathrm{u}}\triangleq s$. Then these m attributes can be viewed as m seeds and (G_1,G_2') can be viewed as a graph pair generated from the seeded Erdős–Rényi pair model $\mathcal{G}(n+m,\frac{m}{m+n},p,s)$ with the edges between these m vertices all removed. Let $N\triangleq m+n$ and $\alpha\triangleq\frac{m}{m+n}$. When specialized to the seeded graph alignment $\mathcal{G}(N,\alpha,p,s)$ with $p=o(1)$, $s=\Theta(1)$ and $N(1-\alpha)=\omega(1)$, the feasible region in Theorems 1 and 2 strictly improves the best known feasible region by polynomial-time algorithms given in [9], [14] and [15], as shown in the blue area in Fig. 3. We note, however, that there is also some region that is feasible by the existing results but not feasible by the proposed algorithms in this paper, as shown in the red area in Fig. 3. Compared to the information-theoretic limits, there is still a small gap between the union of all known feasible regions by polynomial-time algorithms (shown in the red, blue, and green areas in Fig. 3) and an inner bound on the information-theoretic feasible region derived in [1] (shown as the area above the solid green curve in Fig. 3).

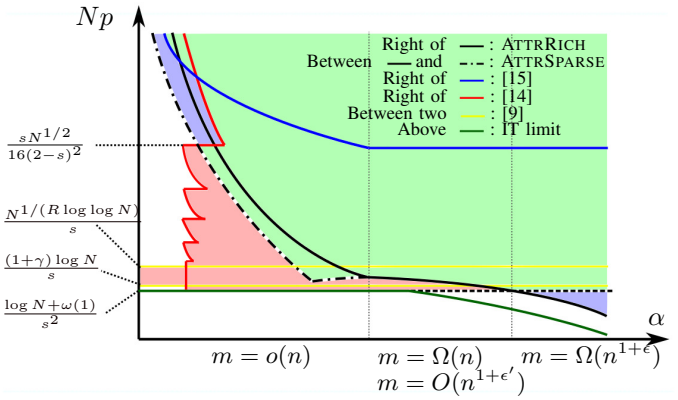

Fig. 3: Comparison between the feasible region of Theorems 1 and 2 and the feasible region of [9], [14] and [15]. On the top-left corner and bottom-right corner, the two blue regions are feasible for our proposed algorithms but not for any existing works. The red region is feasible for existing work [14], but not for our proposed algorithms. The green region is the overlap of our feasible region with the feasible region in the existing works. Constant ϵ' satisfies that $0<\epsilon'<\epsilon$ and $\epsilon-\epsilon'=\Theta(1)$. We note that the algorithm proposed in [9] is designed for the usual Erdős–Rényi pair model *with no seeds*. Therefore, it trivially induces a feasible region for the seeded alignment problem with no constraint on α. We also point out that the feasible region in [9] involves a constraint on s which is not reflected in the plot.

Specialization to the bipartite graph alignment. When $p=0$, the attributed graph alignment problem specializes to the bipartite graph alignment problem, where there is no user-user edge and (G_1,G_2) are bipartite graphs. For this problem, the information-theoretic limit is given by

$$mqs_{\mathrm{a}}^2\ge\log n+\omega(1)$$

and can be achieved in polynomial time using the celebrated Hungarian Algorithm [16]. When specialized to the bipartite graph alignment problem, the proposed Algorithm AttrRich provides an alternative polynomial time algorithm to the Hungarian Algorithm when

$$mqs_{\mathrm{a}}^2\ge(1+\epsilon)\log n,$$

with a slightly lower complexity when $m=o(n)$.

REFERENCES

[1] N. Zhang, W. Wang, and L. Wang, "Attributed graph alignment," in *Proc. IEEE Internat. Symp. Inf. Theory*, 2021, pp. 1829–1834.

[2] P. Pedarsani and M. Grossglauser, "On the privacy of anonymized networks," in *Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)*, 2011, pp. 1235–1243.

[3] D. Cullina and N. Kiyavash, "Improved achievability and converse bounds for Erdős-Rényi graph matching," *ACM SIGMETRICS Perform. Evaluation Rev.*, vol. 44, no. 1, p. 63–72, June 2016.

[4] D. Cullina and N. Kiyavash, "Exact alignment recovery for correlated Erdős-Rényi graphs," 2017. [Online]. Available: https://arxiv.org/abs/1711.06783

[5] Y. Wu, J. Xu, and S. H. Yu, "Settling the sharp reconstruction thresholds of random graph matching," 2021. [Online]. Available: https://arxiv.org/abs/2102.00082

[6] O. Dai, D. Cullina, N. Kiyavash, and M. Grossglauser, "Analysis of a canonical labeling algorithm for the alignment of correlated erdős-rényi graphs," *ACM SIGMETRICS Perform. Evaluation Rev.*, vol. 47, pp. 96–97, 12 2019.

[7] J. Ding, Z. Ma, Y. Wu, and J. Xu, "Efficient random graph matching via degree profiles," 2020. [Online]. Available: https://arxiv.org/abs/1811.07821

[8] Z. Fan, C. Mao, Y. Wu, and J. Xu, "Spectral graph matching and regularized quadratic relaxations: Algorithm and theory," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 July 2020, pp. 2985–2995.

[9] C. Mao, M. Rudelson, and K. Tikhomirov, "Exact matching of random graphs with constant correlation," 2021. [Online]. Available: https://arxiv.org/abs/2110.05000

[10] S. Zhang and H. Tong, "Final: Fast attributed network alignment," in *Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1345–1354.

[11] ——, "Attributed network alignment: Problem definitions and fast solutions," *Proc. IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1680–1692, 2019.

[12] Q. Zhou, L. Li, X. Wu, N. Cao, L. Ying, and H. Tong, *Attent: Active Attributed Network Alignment*. New York, NY, USA: Association for Computing Machinery, 2021, p. 3896–3906.

[13] Z. Wang, N. Zhang, W. Wang, and L. Wang, "On the feasible region of efficient algorithms for attributed graph alignment," 2022. [Online]. Available: https://arxiv.org/abs/2201.10106

[14] E. Mossel and J. Xu, "Seeded graph matching via large neighborhood statistics," *Random Structures & Algorithms*, vol. 57, no. 3, pp. 570–611, 2020.

[15] F. Shirani, S. Garg, and E. Erkip, "Seeded graph matching: Efficient algorithms and theoretical guarantees," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2017, pp. 253–257.

[16] H. W. Kuhn and B. Yaw, "The Hungarian method for the assignment problem," *Naval Res. Logist. Quart*, pp. 83–97, 1955.