On the Computational Complexities of Three Problems Related to a Privacy Measure for Large Networks Under Active Attack

Tanima Chatterjee[a,1], Bhaskar DasGupta[a,1,*], Nasim Mobasheri[a,1], Venkatkumar Srinivasan[a,1], Ismael G. Yero[b,2]

[a] *Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA*

[b] *Departamento de Matemáticas, Escuela Politécnica Superior, Universidad de Cádiz, 11202 Algeciras, Spain*

Abstract

With the arrival of modern internet era, large public networks of various types have come to existence to benefit the society as a whole and several research areas such as sociology, economics and geography in particular. However, the societal and research benefits of these networks have also given rise to potentially significant privacy issues in the sense that malicious entities may violate the privacy of the users of such a network by analyzing the network and deliberately using such privacy violations for deleterious purposes. Such considerations have given rise to a new active research area that deals with the quantification of privacy of users in large networks and the corresponding investigation of computational complexity issues of computing such quantified privacy measures. In this paper, we formalize three natural problems related to such a privacy measure for large networks and provide non-trivial theoretical computational complexity results for solving these problems. Our results show the first two problems can be solved efficiently, whereas the third problem is provably hard to

*Corresponding author

Email addresses: tchatt2@uic.edu (Tanima Chatterjee), bdasgup@uic.edu (Bhaskar DasGupta), nmobas2@uic.edu (Nasim Mobasheri), vsrini7@uic.edu (Venkatkumar Srinivasan), ismael.gonzalez@uca.es (Ismael G. Yero)

[1] Research partially supported by NSF grant IIS-1160995.

[2] This research was done while the author was visiting the University of Illinois at Chicago, USA, supported by "Ministerio de Educación, Cultura y Deporte", Spain, under the "José Castillejo" program for young researchers (reference number: CAS15/00007)

solve within a logarithmic approximation factor. Furthermore, we also provide computational complexity results for the case when the privacy requirement of the network is severely restricted, including an efficient logarithmic approximation.

Keywords: Privacy measure, social networks, active attack, computational complexity

2010 MSC: 68Q25, 68W25, 05C85

---

1. Introduction

Social networks have become an important center of attention in our modern information society by transforming human relationships into a huge interchange of, very often, *sensitive* data. There are many truly beneficial consequences when social network data are released for justified mining and analytical purposes. For example, researchers in sociology, economics and geography, as well as vendors in service-oriented systems and internet advertisers can benefit and improve their performances by a fair study of the social network data. But, such benefits are *not* free of cost as dishonest individuals or organizations may compromise the *privacy* of its users while scrutinizing a public social network and may deliberately use such privacy violations for harmful or other unfair commercial purposes. A common way to handle this kind of unwelcome intrusion on the user's privacy is to somehow *anonymize* the data by removing most potentially identifying attributes. However, even after such anonymization, often it may still be possible to infer many sensitive attributes of a social network that may be linked to its users, such as node degrees, inter-node distances or network connectivity, and therefore *further* privacy-preserving methods need to be investigated and analyzed. These additional privacy-preserving methods of social networks are based on the concept of k-anonymity introduced for micro-data in [16], aiming to ensure that *no* record in a database can be re-identified with a probability higher than $1/k$.

Crucial to modelling a social network anonymization process are the ad-

versary's background knowledge of any object and the structural information about the network that is available. For example, assuming the involved social network as a simple graph in which individuals are represented by nodes and relationships between pairs of individuals are represented by edges, the adversary's background knowledge about a target (a node) could be the node degree [12], the node neighborhood [24], *etc.* In such scenarios, it frequently suffices to develop attacks to re-identify the individuals and their relationships. Such attacks are usually called *passive* (see [14] for more information). Some examples of passive attacks and the corresponding privacy-preserving methods for social networks can be found in references [12, 24, 25].

In contrast, Backstrom *et al.* introduced the concept of the *active* attacks in [1]. Such attacks are mainly based on creating and inserting in a network some nodes (the "attacker nodes") under control by the adversary. These attacker nodes could be newly created accounts with pseudonymous or spoofed identities (commonly called Sybil nodes), or existing legitimate individuals in the network which are in the adversary's proximity. The goal is to establish links with some other nodes in the network (or even links between other nodes) in order to create some sort of "fingerprints" in the network that will be further released. Clearly, once the releasing action has been achieved, the adversary could retrieve the fingerprints already introduced, and use them to re-identify other nodes in the network. Backstrom *et al.* in [1] showed that $O(\sqrt{\log n})$ attacker nodes in a network could in fact seriously compromise the privacy of any arbitrary node. In recent years, several research works have appeared that deal with decreasing the impact of these active attacks (see, for instance, [20]). For other related publications on privacy-preserving methods in social networks, see [15, 21, 24].

There are already many well-known active attack strategies for social networks in order to find all possible vulnerabilities. However, somewhat surprisingly, not many prior research works have addressed the goal of measuring how resistant is a given social network against these kinds of active attacks to the privacy. Very recently a novel privacy measure for social networks was introduced in [18]. The privacy measure proposed there was called the (k, ℓ)-*anonymity*,

where k is a number indicating a privacy threshold and ℓ is the *maximum* number of attacker nodes that can be inserted into the network; ℓ may be estimated through some statistical methods[3]. As claimed by Trujillo-Rasua and Yero in [18], graphs satisfying (k, ℓ)-anonymity can prevent adversaries who control at most ℓ nodes in the network from re-identifying individuals with probability higher than $1/k$. This privacy measure relies on a graph parameter called the k-metric anti-dimension.

Consider a simple connected unweighted graph $G = (V, E)$ and let $\text{dist}_{u,v}$ be the length (number of edges) of a shortest path between two nodes $u, v \in V$. For an ordered set $S = u_1, \ldots, u_t$ of nodes of G and a node $v \in V$, the vector $\mathbf{d}_{v,-S} = (\text{dist}_{v,u_1}, \ldots, \text{dist}_{v,u_t})$ is called the *metric representation* of v with respect to S. Based on the above definition, an ordered set $S \subset V$ of nodes is called a *k-anti-resolving set* for G if k is the largest positive integer such that for every node $v \in V \setminus S$ there exist at least $k-1$ different nodes $v_1, \ldots, v_{k-1} \in V \setminus S$ such that $\mathbf{d}_{v,-S} = \mathbf{d}_{v_1,-S} = \cdots = \mathbf{d}_{v_{k-1},-S}$, *i.e.*, v and $v_1, \ldots, v_{k-1}$ have the same metric representation with respect to S. The *k-metric anti-dimension* of G, denoted by $\text{adim}_k(G)$, is the minimum cardinality of any k-anti-resolving set in G. Note that k-anti-resolving sets may *not* exist in a graph for every k.

The connection between (k, ℓ)-anonymity privacy measure and the k-metric anti-dimension can be understood in the following way. Suppose that an adversary takes control of a set of nodes S of the graph (*i.e.*, S plays the role of attacker nodes), and the background knowledge of such an adversary regarding a target node v is the metric representation of the node v with respect to S. The (k, ℓ)-anonymity privacy measure is a privacy metric that naturally evolves from the adversary's background knowledge. Intuitively, if S (the attacker nodes of an adversary) is a k-anti-resolving set then the adversary cannot uniquely re-identify other nodes in the network (based on the metric representation) from

[3]Note that other different privacy notions with the *same* name also exists, *e.g.*, Feder and Nabar in [6] investigated (k, ℓ)-anonymity where ℓ represented the number of common neighbors of two nodes.

these attacker nodes with a probability higher than $1/k$ (based on uniform sampling of other nodes), and if the k-metric anti-dimension of the graph is ℓ then the adversary must use at least ℓ attacker nodes to get the probability of privacy violation down to $1/k$.

1.1. Organization of the Paper

It is desirable to know how secure a given social network is against active attacks. This necessitates the study of computational complexity issues for computing (k, ℓ)-anonymity. Currently known results only include some heuristic algorithms with no provable guarantee on performances such as in [18], or algorithms for very special cases. In fact, it is not even known if any version of the related computational problems is NP-hard. We formalize three computational problems related to measuring the (k, ℓ)-anonymity of graphs and present non-trivial computational complexity results for these problems. The rest of the paper is organized as follows:

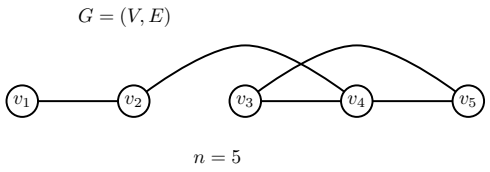
- ▷ In Section 2 we review some basic terminologies and notations and present the three computational problems that we consider in this paper. For the benefit of the reader, we also briefly review some standard algorithmic complexity concepts and results that will be used later.
- ▷ In Section 3, we state the results in this paper mathematically precisely along with some informal remarks. We group our results based on the problem definitions and the expected size of the attacker nodes.
- ▷ Sections 4–6 are devoted to the proofs of the results stated in Section 3.
- ▷ We finally conclude in Section 8 with some possible future research directions.

Historical note on the results of this paper While our paper was still under submission/review, Zhang and Gao in [23] independently and without knowing our results provided an alternate proof to the NP-completeness result reported in Theorem 2(a) using a different reduction.

2. Basic Terminologies, Notations and Problem Definitions

In this section, we first describe the terminologies and notations required to describe our computational problems, and subsequently describe several versions of the problems we consider.

2.1. Basic Terminologies and Notations



dist_{v_i,v_j} values

	v_1	v_2	v_3	v_4	v_5
v_1	0	1	3	2	3
v_2	1	0	2	1	2
v_3	3	2	0	1	1
v_4	2	1	1	0	1
v_5	3	2	1	1	0

Figure 1: An example to illustrate the notations in Section 2.1.

Let $G = (V, E)$ be our *undirected unweighted* input graph over n nodes $v_1, v_2, \ldots, v_n$. We use dist_{v_i,v_j} to denote the distance (number of edges in a shortest path) between nodes v_i and v_j. For illustrating various notations, we use the example in Fig. 1.

- $\mathbf{d}_{v_i} = (\text{dist}_{v_i,v_1}, \text{dist}_{v_i,v_2}, \ldots, \text{dist}_{v_i,v_n})$. For example, $\mathbf{d}_{v_2} = (1, 0, 2, 1, 2)$.
- $\mathsf{diam}(G) = \max_{v_i,v_j \in V} \{ \text{dist}_{v_i,v_j} \}$ is the *diameter* (length of a longest shortest path) of the graph $G = (V, E)$. For example, $\mathsf{diam}(G) = 3$.
- $\mathsf{Nbr}(v_\ell) = \{ v_j \mid \{v_\ell, v_j\} \in E \}$ is the (open) *neighborhood* of node v_ℓ in $G = (V, E)$. For example, $\mathsf{Nbr}(v_2) = \{v_1, v_4\}$.
- For a subset of nodes $V' \subset V$ and any $v_i \in V \setminus V'$, $\mathbf{d}_{v_i,-V'}$ denotes the metric representation of v_i with respect to V', *i.e.*, the vector of $|V'|$ elements obtained from $\mathbf{d}_{v_i}$ by deleting dist_{v_i,v_j} for every $v_j \in V \setminus V'$. For example, $\mathbf{d}_{v_2,-\{v_1,v_3\}} = (1, 2)$.
- $\mathcal{D}_{V'',-V'} = \{ \mathbf{d}_{v_i,-V'} \mid v_i \in V'' \}$ for any $V'' \subseteq V \setminus V'$. For example, if $V'' = \{v_2, v_4\}$ then $\mathcal{D}_{V'',-\{v_1,v_3\}} = \{(1, 2), (2, 1)\}$.

- ▶ $\Pi = \{V_1, V_2, \ldots, V_k\}$ is a partition of $V' \subseteq V$ if and only if $\cup_{t=1}^{k} V_t = V'$ and $V_i \cap V_j = \emptyset$ for $i \neq j$.

  - ▷ Partition $\Pi' = \{V'_1, V'_2, \ldots, V'_\ell\}$ is called a *refinement*[4] of partition Π, denoted by $\Pi' \prec_r \Pi$, provided $\cup_{t=1}^{\ell} V'_t \subset \cup_{t=1}^{k} V_t$ and the following two conditions are satisfied:

    i. All the sets in Π' are pairwise disjoint.

    ii. There exists a total and surjective function $f : \{1, \ldots, \ell\} \mapsto \{1, \ldots, t\}$ such that $\forall j \in \{1, \ldots, \ell\} : V'_j \subseteq V_{f(j)}$.

    For example, if $\Pi = \Big\{ \{v_1, v_2\}, \{v_3, v_4, v_5\} \Big\}$ and $\Pi' = \Big\{ \{v_1, v_2\}, \{v_3\}, \{v_4\} \Big\}$ then $\Pi' \prec_r \Pi$.

- ▶ The equality relation over a set of vectors, all of same length, defines an *equivalence relation*. The following notations are used for such an equivalence relation over the set of vectors $\mathcal{D}_{V \setminus V', -V'}$ for some $\emptyset \subset V' \subset V$.

  - ▷ The set of equivalence classes, which forms a partition of $\mathcal{D}_{V \setminus V', -V'}$, is denoted by $\Pi^{=}_{V \setminus V', -V'}$. For example, $\Pi^{=}_{\{v_1, v_2, v_3\}, -\{v_4, v_5\}} = \Big\{ \big\{(2,3)\big\}, \big\{(1,2)\big\}, \big\{(1,1)\big\} \Big\}$.

  - ▷ Abusing terminologies slightly, two nodes $v_i, v_j \in V \setminus V'$ will be said to belong to the *same* equivalence class if $\mathbf{d}_{v_i, -V'}$ and $\mathbf{d}_{v_j, -V'}$ belong to the same equivalence class in $\Pi^{=}_{V \setminus V', -V'}$, and thus $\Pi^{=}_{V \setminus V', -V'}$ also defines a partition into equivalence classes of $V \setminus V'$. For example, $\Pi^{=}_{\{v_1, v_2, v_3\}, -\{v_4, v_5\}}$ will also denote $\Big\{ \big\{v_1\big\}, \big\{v_2\big\}, \big\{v_3\big\} \Big\}$.

  - ▷ The *measure* of the equivalence relation is defined as $\mu\left(\mathcal{D}_{V \setminus V', -V'}\right) \stackrel{\text{def}}{=} \min_{\mathcal{Y} \in \Pi^{=}_{V \setminus V', -V'}} \Big\{ \mid \mathcal{Y} \mid \Big\}$. Thus, if a set S is a k-anti-resolving set then $\mathcal{D}_{V \setminus S, -S}$ defines a partition into equivalence classes whose measure is *exactly* k. For example, $\mu\left(\mathcal{D}_{\{v_1, v_2, v_3\}, -\{v_4, v_5\}}\right) = 1$ and $\{v_4, v_5\}$ is a 1-anti-resolving set.

[4] Our definition is slightly different from the standard definition of refinement since we allow $\cup_{t=1}^{\ell} V'_t \subset \cup_{t=1}^{k} V_t$.

2.2. Problem Definitions

It is desirable to know how secure a given social network is against active attacks. This necessitates the study of computational complexity issues for computing (k, ℓ)-anonymity. We formalize three computational problems related to measuring the (k, ℓ)-anonymity of graphs. For all the problem versions, let $G = (V, E)$ be the (connected undirected unweighted) input graph representing the social network under study.

Problem 1 (metric anti-dimension or ADIM)). *Given G, find a subset of nodes V' that* maximizes $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right)$.

Notation related to Problem 1 $k_{\text{opt}} = \max\limits_{\emptyset \subset V' \subset V} \left\{ \mu\left(\mathcal{D}_{V\setminus V',-V'}\right) \right\}$.

Problem 1 simply finds a k-anti-resolving set for the largest possible k. Intuitively, it sets an absolute bound on the privacy violation probability of an adversary assuming that the adversary can use *any* number of attacker nodes. In practice, however, the number of attacker nodes employed by the adversary may be limited, which leads us to the second problem formulation stated below.

Problem 2 ($k_{\geq}$-metric anti-dimension or $\text{ADIM}_{\geq k}$). *Given G and a positive integer k, find a subset of nodes V' of* minimum *cardinality such that $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) \geq k$, if such a V' exists.*

Notation and assumption related to Problem 2
$\mathcal{L}_{\text{opt}}^{\geq k} = \left|V_{\text{opt}}^{\geq k}\right| = \min\left\{ |V'| \;\middle|\; \mu\left(\mathcal{D}_{V\setminus V',-V'}\right) \geq k \right\}$ for some $\emptyset \subset V_{\text{opt}}^{\geq k} \subset V$. If $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) \geq k$ for *no* V' then we set $\mathcal{L}_{\text{opt}}^{\geq k} = \infty$ and $V_{\text{opt}}^{\geq k} = \emptyset$.

Problem 2 finds a k-anti-resolving set for a given k while simultaneously minimizing the number of attacker nodes.

The remaining third version of our problem formulation relates to a trade-off between privacy violation probability and the corresponding minimum number of attacker nodes needed to achieve such a violation. To understand this motivation, suppose that G has a k-metric anti-dimension of ℓ, a k'-metric anti-dimension of ℓ', $k' > k$ and $\ell' < \ell$. This provides a trade-off between privacy

and number of attacker nodes, namely we may allow a smaller privacy violation probability $1/k'$ but the network can tolerate *adversarial control* of a *fewer* number ℓ' of nodes or we may allow a larger privacy violation probability $1/k$ but the network can tolerate adversarial control of a larger number ℓ of nodes. Such a trade-off may be crucial for a network administrator in administering privacy of a network or for an individual in its decision to join a network. Clearly, this necessitates solving a problem of the following type.

Problem 3 ($k_{=}$-metric antidimension or $\textsc{Adim}_{=k}$). *Given G and a positive integer k, find a subset of nodes V' of* minimum *cardinality such that $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) = k$, if such a V' exists.*

Notation and assumption related to Problem 3
$\mathcal{L}^{=k}_{\text{opt}} = \left|V^{=k}_{\text{opt}}\right| = \min\left\{ |V'| \;\middle|\; \mu\left(\mathcal{D}_{V\setminus V',-V'}\right) = k\right\}$ for some $\emptyset \subset V^{=k}_{\text{opt}} \subset V$. If $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) = k$ for *no* V' then we set $\mathcal{L}^{=k}_{\text{opt}} = \infty$ and $V^{=k}_{\text{opt}} = \emptyset$

2.3. Standard Algorithmic Complexity Concepts and Results

For the benefit of the reader, we summarize the following concepts and results from the computational complexity theory domain. *We assume that the reader is familiar with standard O, Ω, o and ω notations used in asymptotic analysis of algorithms* (*e.g., see* [4]).

An algorithm $\mathcal{A}$ for a minimization (resp., maximization) problem is said to have an *approximation ratio* of ε (or is simply an *ε-approximation*) [19] provided $\mathcal{A}$ runs in polynomial time in the size of its input and produces a solution with an objective value *no larger than* ε times (resp., *no smaller than* $1/\varepsilon$ times) the value of the optimum. $\text{DTIME}\left(n^{\log\log n}\right)$ refers to the class of problem that can be solved by a deterministic algorithm running in $(n^{\log\log n})$ time when n is the size of the input instance; it is widely believed that $\mathsf{NP} \not\subset \text{DTIME}(n^{\log\log n})$.

The minimum set-cover problem (Sc) is a well-known combinatorial problem that is defined as follows [4, 9]. Our input is an universe $\mathcal{U} = \{a_1, a_2, \ldots, a_n\}$ of n elements, and a collection of m sets $S_1, S_2, \ldots, S_m \subseteq \mathcal{U}$ over this universe with $\cup_{j=1}^{m} S_j = U$. A valid solution of Sc is a subset of indices $\mathcal{I} \subseteq \{1, 2, \ldots, m\}$

such that every element in $\mathcal{U}$ is "covered" by a set whose index is in $\mathcal{I}$, *i.e.*, $\forall\, a_j \in \mathcal{U}\ \exists\, i \in \mathcal{I}\ :\ a_j \in S_i$. The objective of Sc is to *minimize* the number $|\mathcal{I}|$ of selected sets. We use the notation $\mathsf{opt}_{\mathrm{SC}}$ to denote the size (number of sets) in an optimal solution of an instance of Sc. On the inapproximability side, Sc is NP-hard [9] and, assuming $NP \not\subseteq \mathrm{DTIME}\big(n^{\log\log n}\big)$, Sc does not admit a $(1-\varepsilon)\ln n$-approximation for any constant $0 < \varepsilon < 1$ [7]. On the algorithmic side, Sc admits a $(1+\ln n)$-approximation using a simple greedy algorithm [10] that can be easily implemented to run in $O\big(\sum_{i=1}^{m} |S_i|\ \big)$ time [4].

Finally, in the context of proving NP-completeness, a "decision version" of an optimization problem by the standard method of using an additional parameter and formulating a decision (*i.e.*, yes/no) question on the value of the objective function with respect to this new parameter (*e.g.*, see [9]). For reader's convenience we explicitly write down these decision versions below.

Problem 4 (decision version of metric anti-dimension or Adim)). *Given G and a positive integer ζ, is there a subset of nodes V' such that $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) \geq \zeta$?*

Problem 5 (decision version of $k_{\geq}$-metric anti-dimension or $\mathrm{Adim}_{\geq k}$). *Given G and two positive integers k and ζ, is there a subset of nodes V' such that $|V'| \leq \zeta$ and $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) \geq k$?*

Problem 6 (decision version of $k_{=}$-metric antidimension or $\mathrm{Adim}_{=k}$). *Given G and two positive integers k and ζ, is there a subset of nodes V' such that $|V'| \leq \zeta$ and $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) = k$?*

It is also standard to state that an optimization problem is NP-complete to mean that the decision versions of the optimization problem is NP-complete, and therefore we will follow the same practice in this paper.

3. Our Results

In this section we provide precise statements of our results, leaving their proofs in Sections 4–6.

3.1. Polynomial Time Solvability of ADIM and $\text{ADIM}_{\geq k}$

Theorem 1.

(a) *Both* ADIM *and* $\text{ADIM}_{\geq k}$ *can be solved in* $O\left(n^4\right)$ *time.*

(b) *Both* ADIM *and* $\text{ADIM}_{\geq k}$ *can also be solved in* $O\left(\frac{n^4 \log n}{k}\right)$ *time "with high probability"* (i.e., *with a probability of at least* $1 - n^{-c}$ *for some constant* $c > 0$).

Remark 1. *The randomized algorithm in Theorem* 1(b) *runs faster that the deterministic algorithm in Theorem* 1(a) *provided* $k = \omega(\log n)$.

3.2. Computational Complexity of $\text{ADIM}_{=k}$

3.2.1. The Case of Arbitrary k

Theorem 2.

(a) $\text{ADIM}_{=k}$ *is* NP*-complete for any integer* k *in the range* $1 \leq k \leq n^{\varepsilon}$ *where* $0 \leq \varepsilon < \frac{1}{2}$ *is any arbitrary constant, even if the diameter of the input graph is* 2.

(b) *Assuming* $\mathsf{NP} \not\subseteq \text{DTIME}\left(n^{\log \log n}\right)$, *there exists a universal constant* $\delta > 0$ *such that* $\text{ADIM}_{=k}$ *does not admit a* $\left(\frac{1}{\delta} \ln n\right)$*-approximation for any integer* k *in the range* $1 \leq k \leq n^{\varepsilon}$ *where* $0 \leq \varepsilon < \frac{1}{2}$ *is any arbitrary constant, even if the diameter of the input graph is* 2.

(c) *If* $k = n - c$ *for some constant* c *then* $\mathcal{L}_{\text{opt}}^{=k} = c$ *if a solution exists and* $\text{ADIM}_{=k}$ *can be solved in polynomial time.*

Remark 2.

(a) *For* $k = 1$, *the inapproximability ratio in Theorem* 2(b) *is asymptotically optimal up to a constant factor because of the* $(1 + \ln(n-1))$*-approximation of* $\text{ADIM}_{=1}$ *in Theorem* 3(a).

(b) *The result in Theorem* 2(b) *provides a much stronger inapproximability result compared to that in Theorem* 2(a) *at the expense of a slightly weaker complexity-theoretic assumption* (i.e., $\mathsf{NP} \not\subseteq \text{DTIME}\left(n^{\log \log n}\right)$ *vs.* $P \neq \mathsf{NP}$).

3.2.2. The Case of $k = 1$

Note that even when $k = 1$ $\textsc{Adim}_{=k}$ is NP-hard and even hard to approximate within a logarithmic factor due to Theorem 2. We show the following algorithmic results for $\textsc{Adim}_{=k}$ when $k = 1$.

Theorem 3.

(a) $\textsc{Adim}_{=1}$ *admits a* $(1 + \ln(n-1))$*-approximation in* $O\left(n^3\right)$ *time.*

(b) *If* G *has at least one node of degree* 1 *then* $\mathcal{L}_{\text{opt}}^{=1} = 1$ *and thus* $\textsc{Adim}_{=1}$ *can be solved in* $O\left(n^3\right)$ *time.*

(c) *If* G *does not contain a cycle of* 4 *edges then* $\mathcal{L}_{\text{opt}}^{=1} \leq 2$ *and thus* $\textsc{Adim}_{=1}$ *can be solved in* $O\left(n^3\right)$ *time.*

4. Proof of Theorem 1

(a) We first consider the claim for $\textsc{Adim}_{\geq k}$. We begin by proving some structural properties of valid solutions for $\textsc{Adim}_{\geq k}$.

Proposition 1. *Consider two subsets of nodes* $\emptyset \subset V_1 \subset V_2 \subset V$*. Let* $v_i, v_j \in V_2$ *be two nodes such that they do not belong to the same equivalence class in* $\Pi^{=}_{V \setminus V_1, -V_1}$*. In this case* v_i *and* v_j *do not belong to the same equivalence class in* $\Pi^{=}_{V \setminus V_2, -V_2}$ *also, and thus* $\Pi^{=}_{V \setminus V_2, -V_2} \prec_r \Pi^{=}_{V \setminus V_1, -V_1}$*.*

Proof. Since v_i and v_j are not in the same equivalence class in $\Pi^{=}_{V \setminus V_1, -V_1}$, we have $\mathbf{d}_{v_i, -V_1} \neq \mathbf{d}_{v_j, -V_1}$ which in turn implies (since $V_1 \subset V_2$) $\mathbf{d}_{v_i, -V_2} \neq \mathbf{d}_{v_j, -V_2}$ which implies v_i and v_j are not in the same equivalence class in $\Pi^{=}_{V \setminus V_2, -V_2}$. ❑

Note that $\Pi^{=}_{V \setminus V_2, -V_2} \prec_r \Pi^{=}_{V \setminus V_1, -V_1}$ in Proposition 1 implies does not necessarily imply that $\mu\left(\mathcal{D}_{V \setminus V_2, -V_2}\right) < \mu\left(\mathcal{D}_{V \setminus V_1, -V_1}\right)$. For example, for the example in Fig. 1 $\Pi^{=}_{\{v_1, v_2\}, -\{v_3, v_4, v_5\}} = \left\{ \{v_1\}, \{v_2\} \right\} \prec_r \Pi^{=}_{\{v_1, v_2, v_3\}, -\{v_4, v_5\}} = \left\{ \{v_1\}, \{v_2\}, \{v_3\} \right\}$ but $\mu\left(\mathcal{D}_{\{v_1, v_2, v_3\}, -\{v_4, v_5\}}\right) = \mu\left(\mathcal{D}_{\{v_1, v_2\}, -\{v_3, v_4, v_5\}}\right) = 1$. The following proposition gives some condition for this to happen.

Proposition 2. *Consider two subsets of nodes $\emptyset \subset V_1 \subset V_2 \subset V$, let $S_1, \ldots, S_\ell \subseteq V \setminus V_1$ be all equivalence classes (subsets of nodes) in $\Pi^{=}_{V\setminus V_1, -V_1}$ such that $|S_1| = |S_2| = \cdots = |S_\ell| = \mu\left(\mathcal{D}_{V\setminus V_1,-V_1}\right)$, and assume that $\emptyset \subset V_2 \cap S_j \subset S_j$ for some $j \in \{1,\ldots,\ell\}$. In this case $\mu\left(\mathcal{D}_{V\setminus V_2,-V_2}\right) < \mu\left(\mathcal{D}_{V\setminus V_1,-V_1}\right)$.*

Proof. By Proposition 1, $\Pi^{=}_{V\setminus V_2,-V_2} \prec_r \Pi^{=}_{V\setminus V_1,-V_1}$. If there exists a S_j such that $\emptyset \subset V_2 \cap S_j \subset S_j$ then $\Pi^{=}_{V\setminus V_2,-V_2}$ contains an equivalence class $\emptyset \subset S_{j'} \subset S_j$. This implies $\mu\left(\mathcal{D}_{V\setminus V_2,-V_2}\right) \leq |S_{j'}| < |S_j| = \mu\left(\mathcal{D}_{V\setminus V_1,-V_1}\right)$. ❑

Based on the above structural properties, we design Algorithm I for $\textsc{Adim}_{\geq k}$ as shown below.

Algorithm I: $O\left(n^4\right)$ time deterministic algorithm for $\textsc{Adim}_{\geq k}$.

---

1. Compute $\mathbf{d}_i$ for all $i = 1, 2, \ldots, n$ in $O\left(n^3\right)$ time using the Floyd-Warshall algorithm [4, p. 629]

2. $\widehat{\mathcal{L}^{\geq k}_{\text{opt}}} \leftarrow \infty$; $\widehat{V^{\geq k}_{\text{opt}}} \leftarrow \emptyset$

3. **for** each $v_i \in V$ **do** $\quad$ ($*$ we *guess* v_i to belong to $V^{\geq k}_{\text{opt}}$ $*$)

3.1 $V' = \{v_i\}$; done $\leftarrow$ FALSE

3.2 **while** $\big($ $(V \setminus V' \neq \emptyset)$ AND (NOT done) $\big)$ **do**

3.2.1 compute $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right)$

3.2.2 **if** $\Big($ $\big(\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) \geq k\big)$ and $\big(|V'| < \widehat{\mathcal{L}^{\geq k}_{\text{opt}}}\big)$ $\Big)$

3.2.3 **then** $\widehat{\mathcal{L}^{\geq k}_{\text{opt}}} \leftarrow |V'|$; $\widehat{V^{\geq k}_{\text{opt}}} \leftarrow V'$; done $\leftarrow$ TRUE

3.2.4 **else** let $V_1, \ldots, V_\ell$ be *all* equivalence classes in $\Pi^{=}_{V\setminus V',-V'}$ such that $|V_1| = \cdots = |V_\ell| = \mu\left(\mathcal{D}_{V\setminus V',-V'}\right)$

3.2.5 $V' \leftarrow V' \cup \left(\cup_{t=1}^{\ell} V_t\right)$

4. **return** $\widehat{\mathcal{L}^{\geq k}_{\text{opt}}}$ and $\widehat{V^{\geq k}_{\text{opt}}}$ as our solution

---

Lemma 4 (Proof of correctness). *Algorithm I returns an optimal solution*

for $\textsc{Adim}_{\geq k}$.

Proof. Assume that $V_{\text{opt}}^{\geq k} \neq \emptyset$ since otherwise our returned solution is correct. Fix any optimal solution (subset of nodes) $V_{\text{opt}}^{\geq k}$ of measure $\mu\left(\mathcal{D}_{V\setminus V_{\text{opt}}^{\geq k}, -V_{\text{opt}}^{\geq k}}\right) \geq k$ and select any arbitrary node $v_\ell \in V_{\text{opt}}^{\geq k}$. Consider the iteration of the **for** loop in Step 3 when v_i is equal to v_ℓ. We now analyze the run of *this particular iteration*.

Let $\{v_\ell\} = V_1 \subset V_2 \subset \cdots \subset V_\kappa$ be the κ subsets of nodes that were assigned to V' in *successive* iterations of the **while** loop in Step 3.2. We have the following cases to consider.

Case 1: $V_{\text{opt}}^{\geq k} = V_t$ for some $t \in \{1, 2, \ldots, \kappa\}$. Our solution is a set $\widehat{V_{\text{opt}}^{\geq k}}$ such that $\mu\left(\mathcal{D}_{V\setminus \widehat{V_{\text{opt}}^{\geq k}}, -\widehat{V_{\text{opt}}^{\geq k}}}\right) \geq k$ and $\widehat{\mathcal{L}_{\text{opt}}^{\geq k}} \leq \mathcal{L}_{\text{opt}}^{\geq k}$.

Case 2: $V_{\text{opt}}^{\geq k} \neq V_t$ for any $t \in \{1, 2, \ldots, \kappa\}$. Since $V_1 = \{v_\ell\} \subset V_{\text{opt}}^{\geq k}$ and $V_t \neq V_{\text{opt}}^{\geq k}$ for any $t \in \{1, 2, \ldots, \kappa\}$, only one of the following cases is possible:

Case 2.1: there exists $r \in \{1, \ldots, \kappa - 1\}$ such that $V_r \subset V_{\text{opt}}^{\geq k}$ but $V_{r+1} \not\subseteq V_{\text{opt}}^{\geq k}$. Let $V_{r,1}, V_{r,2}, \ldots, V_{r,p} \subseteq V \setminus V_r$ be all the $p > 0$ equivalence classes (subsets of nodes) in $\Pi^{=}_{V\setminus V_r, -V_r}$ such that $|V_{r,1}| = |V_{r,2}| = \cdots = |V_{r,p}| = \mu\left(\mathcal{D}_{V\setminus V_r, -V_r}\right)$. Now we note the following:

- By Step 3.2.5, $V_{r+1} = V_r \cup V_{r,1} \cup V_{r,2} \cup \cdots \cup V_{r,p}$.
- Thus, $V_r \subset V_{\text{opt}}^{\geq k}$ and $V_{r+1} \not\subseteq V_{\text{opt}}^{\geq k}$ implies $V_{r,1} \cup V_{r,2} \cup \cdots \cup V_{r,p} \not\subseteq V_{\text{opt}}^{\geq k}$, and therefore there exists an index $1 \leq s \leq p$ such that $Z = V_{r,s} \setminus V_{\text{opt}}^{\geq k} \neq \emptyset$. Let $Z' = V_{r,s} \setminus Z$ (Z' could be empty). For this case, for some $\emptyset \subset Z'' \subseteq Z$, Z'' is an equivalence class in $\Pi^{=}_{V\setminus(V_r \cup Z'), -(V_r \cup Z')}$ implying

$$\mu\left(\mathcal{D}_{V\setminus(V_r\cup Z'), -(V_r\cup Z')}\right) \leq |Z''| \leq |Z| \tag{1}$$

Since $V_r \cup Z' \subseteq V_{\text{opt}}^{\geq k}$, we have

$$\Pi^{=}_{V\setminus V_{\text{opt}}^{\geq k}, -V_{\text{opt}}^{\geq k}} \prec_r \Pi^{=}_{V\setminus(V_r\cup Z'), -(V_r\cup Z')}$$

(in Proposition 1, set $V_2 = V_{\text{opt}}^{\geq k}$ and $V_1 = V_r \cup Z'$)

$$\Rightarrow\ k \leq \mu\left(\mathcal{D}_{V\setminus V_{\text{opt}}^{\geq k},-V_{\text{opt}}^{\geq k}}\right) \leq \mu\left(\mathcal{D}_{V\setminus(V_r\cup Z'),-(V_r\cup Z')}\right)$$
$$\leq |Z| \leq |V_{r,s}| = \mu\left(\mathcal{D}_{V\setminus V_r,-V_r}\right)$$
by (1)

Thus, $\mu\left(\mathcal{D}_{V\setminus V_r,-V_r}\right) \geq k$ and $|V_r| < \left|V_{\text{opt}}^{\geq k}\right| = \mathcal{L}_{\text{opt}}^{\geq k}$, contradicting the optimality of $\mathcal{L}_{\text{opt}}^{\geq k}$.

Case 2.2: $V_\kappa \subset V_{\text{opt}}^{\geq k}$**.** If done was set to TRUE at the last iteration of the **while** loop, then $\mu\left(\mathcal{D}_{V\setminus V_\kappa,-V_\kappa}\right) \geq k$ and $|V_\kappa| < \left|V_{\text{opt}}^{\geq k}\right| = \mathcal{L}_{\text{opt}}^{\geq k}$, contradicting the optimality of $\mathcal{L}_{\text{opt}}^{\geq k}$. Thus, done must have remained FALSE after the last iteration of the **while** loop, implying $\mu\left(\mathcal{D}_{V\setminus V_\kappa,-V_\kappa}\right) < k$. Let $V_{\kappa,1}, V_{\kappa,2}, \ldots, V_{\kappa,p} \subseteq V \setminus V_\kappa$ be all the $p > 0$ equivalence classes (subsets of nodes) in $\Pi^{=}_{V\setminus V_\kappa,-V_\kappa}$ such that $|V_{\kappa,1}| = |V_{\kappa,2}| = \cdots = |V_{\kappa,p}| = \mu\left(\mathcal{D}_{V\setminus V_\kappa,-V_\kappa}\right)$. Since $V_\kappa \subset V_{\text{opt}}$, we have

$$\Pi^{=}_{V\setminus V_{\text{opt}},-V_{\text{opt}}} \prec_r \Pi^{=}_{V\setminus V_\kappa,-V_\kappa}$$
(in Proposition 1, set $V_2 = V_{\text{opt}}$ and $V_1 = V_\kappa$)
$$\Rightarrow\ k \leq \mu\left(\mathcal{D}_{V\setminus V_{\text{opt}},-V_{\text{opt}}}\right) \leq \mu\left(\mathcal{D}_{V\setminus V_\kappa,-V_\kappa}\right) \leq |Z| \leq |V_{\kappa,p}| = \mu\left(\mathcal{D}_{V\setminus V_\kappa,-V_\kappa}\right)$$
by (1)

Thus, $\mu\left(\mathcal{D}_{V\setminus V_\kappa,-V_\kappa}\right) \geq k$ contradicting our assumption of $\mu\left(\mathcal{D}_{V\setminus V_\kappa,-V_\kappa}\right) < k$.

❑

Lemma 5 (Proof of time complexity). *Algorithm I runs in* $O\left(n^4\right)$ *time.*

Proof. There are n choices for the **for** loop in Step 3. For each such choice, we analyze the execution of the **while** loop in Step 3.2. The running time in each iteration of the **while** loop is dominated by the time taken to compute $\Pi^{=}_{V\setminus\left(V'\cup\left(\cup_{t=1}^{\ell}V_t\right)\right),-V'\cup\left(\cup_{t=1}^{\ell}V_t\right)}$ from $\Pi^{=}_{V\setminus V',-V'}$. Suppose that $\cup_{t=1}^{\ell}V_t = \left\{v_{i_1}, v_{i_2}, \ldots, v_{i_p}\right\}$. By Proposition 1,

$$\Pi^{=}_{V\setminus\left(V'\cup\left\{v_{i_1},v_{i_2},\ldots,v_{i_p-1},v_{i_p}\right\}\right),-V'\cup\left\{v_{i_1},v_{i_2},\ldots,v_{i_p-1},v_{i_p}\right\}}$$
$$\prec_r \Pi^{=}_{V\setminus\left(V'\cup\left\{v_{i_1},v_{i_2},\ldots,v_{i_p-1}\right\}\right),-V'\cup\left\{v_{i_1},v_{i_2},\ldots,v_{i_p-1}\right\}} \prec_r \cdots$$

$$\prec_r \Pi^{=}_{V\setminus(V'\cup\{v_{i_1},v_{i_2}\}),-V'\cup\{v_{i_1},v_{i_2}\}} \prec_r \Pi^{=}_{V\setminus(V'\cup\{v_{i_1}\}),-V'\cup\{v_{i_1}\}} \prec_r \Pi^{=}_{V\setminus V',-V'}$$

Thus, it follows that the *total* time to execute *all* iterations of the **while** loop *for a specific choice* of v_i in Step 3 is of the order of n times the time taken to solve a problem of the following kind:

> *for a subset of nodes* $\emptyset \subset V_1 \subset V$, *given* $\Pi^{=}_{V\setminus V_1,-V_1}$ *and a node* $v_j \in V \setminus V_1$, *compute* $\Pi^{=}_{V\setminus(V_1\cup\{v_j\}),-(V_1\cup\{v_j\})}$.

Since $\Pi^{=}_{V\setminus(V_1\cup\{v_j\}),-(V_1\cup\{v_j\})}$ is a refinement of $\Pi^{=}_{V\setminus V_1,-V_1}$ by Proposition 1, we can use the following simple strategy. For every set $S \in \Pi^{=}_{V\setminus V',-V'}$, we split $S \setminus \{v_j\} = \{v_{i_1}, v_{i_2}, \ldots, v_{i_s}\}$ into two or more parts, if needed, by doing a bucket-sort (with n bins) in $O(n\,|S|)$ time on the sequence of values $\mathrm{dist}_{v_{i_1},v_j}, \ldots, \mathrm{dist}_{v_{i_s},v_j}$. The total time taken for all sets in $\Pi^{=}_{V\setminus V',-V'}$ is thus $\sum_{S\in\Pi^{=}_{V\setminus V',-V'}} O\left(n\,|S|\right) = O\left(n^2\right)$. ❑

This completes the proof for ADIM$_{\geq k}$. Now we consider the claim for ADIM. Note that ADIM can be solved in $O\left(n^5\right)$ time by solving ADIM$_{\geq k}$ for $k = n-1, n-2, \ldots, 1$ in this order and selecting the largest k as k_{opt} for which $\mathcal{L}^{\geq k}_{\mathrm{opt}} < \infty$. However, we can modify the steps of Algorithm I directly to solve ADIM in $O\left(n^4\right)$ time, as shown in Algorithm II.

Algorithm II: $O\left(n^4\right)$ time deterministic algorithm for ADIM

(changes from Algorithm-I are shown enclosed in ▭)

---

1. Compute $\mathbf{d}_i$ for all $i = 1, 2, \ldots, n$ in $O\left(n^3\right)$ time using the Floyd-Warshall algorithm [4, p. 629]

2. $\widehat{V^{\geq k}_{\mathrm{opt}}} \leftarrow \emptyset$; $\boxed{\widehat{k_{\mathrm{opt}}} \leftarrow 0}$

3. **for** each $v_i \in V$ **do** (∗ we *guess* v_i to belong to $V^{\geq k}_{\mathrm{opt}}$ ∗)

3.1 $V' = \{v_i\}$

3.2 **while** $\boxed{\left(V \setminus V' \neq \emptyset\right)}$ **do**

3.2.1 compute $\mu\left(\mathcal{D}_{V\setminus V',-V'}\right)$

3.2.2 **if** $\boxed{\left(\mu\left(\mathcal{D}_{V\setminus V',-V'}\right) > \widehat{k_{\text{opt}}}\right)}$

3.2.3 **then** $\boxed{\widehat{k_{\text{opt}}} \leftarrow \mu\left(\mathcal{D}_{V\setminus V',-V'}\right)}$; $\widehat{V_{\text{opt}}^{\geq k}} \leftarrow V'$

3.2.4 **else** let $V_1, \ldots, V_\ell$ be all equivalence classes in $\Pi^{=}_{V\setminus V',-V'}$ such that $|V_1| = \cdots = |V_\ell| = \mu\left(\mathcal{D}_{V\setminus V',-V'}\right)$

3.2.5 $V' \leftarrow V' \cup \left(\cup_{t=1}^{\ell} V_t\right)$

4. **return** $\boxed{\widehat{k_{\text{opt}}}}$ and $\widehat{V_{\text{opt}}^{\geq k}}$ as our solution

340 The proof of correctness is very similar (and, in fact simpler due to elimination of some cases) to that of ADIM$_{\geq k}$.

(b) Our solution is the obvious randomization of Algorithm II (for ADIM$_{\geq k}$) or Algorithm-II (for ADIM) as shown below.

Algorithm III (resp. Algorithm-IV): $O\left(\frac{n^4 \log n}{k}\right)$ time randomized algorithm for ADIM$_{\geq k}$ (resp. ADIM)

1. Compute $\mathbf{d}_i$ for all $i = 1, 2, \ldots, n$ in $O\left(n^3\right)$ time using the Floyd-Warshall algorithm [4, p. 629]

2. $\widehat{\mathcal{L}_{\text{opt}}^{\geq k}} \leftarrow \infty$; $\widehat{V_{\text{opt}}^{\geq k}} \leftarrow \emptyset$ (for ADIM$_{\geq k}$)

or

$\widehat{V_{\text{opt}}^{\geq k}} \leftarrow \emptyset$; $\widehat{k_{\text{opt}}} \leftarrow 0$ (for ADIM)

3. **repeat** $\left\lceil \frac{2n \ln n}{k} \right\rceil$ times

3.1 select a node v_i uniformly at random from the n nodes

3.2 execute Step 3.1 and Step 3.2 (and its sub-steps) of Algorithm I (for ADIM$_{\geq k}$)

or

execute Step 3.1 and Step 3.2 (and its sub-steps) of Algorithm II (for ADIM)

4. **return** the best of all solutions found in Step 3

The success probability p is given by

$$\begin{aligned} p &= \Pr\left[\ v_i \in V_{\text{opt}}^{\geq k} \text{ in } \textit{at least one} \text{ of the } \left\lceil \tfrac{2n \ln n}{k} \right\rceil \text{ iterations } \right] \\ &= 1 - \Pr\left[v_i \notin V_{\text{opt}}^{\geq k} \text{ in } \textit{each} \text{ of the } \left\lceil \tfrac{2n \ln n}{k} \right\rceil \text{ iterations} \right] \\ &\geq 1 - \left(1 - \frac{k}{n}\right)^{\left\lceil \frac{2n \ln n}{k} \right\rceil} > 1 - \frac{1}{\mathbf{e}^{2 \ln n}} = 1 - \frac{1}{n^2} \end{aligned}$$

5. Proof of Theorem 2

The standard NP-complete *minimum dominating set* (MDS) problem for a graph is defined as follows [9]. Our input is a connected undirected unweighted graph $G = (V, E)$. A subset of nodes $V' \subset V$ is called a *dominating set* if and only if every node in $V \setminus V'$ is adjacent to some node in V'. The objective of MDS is to find a dominating set of nodes of *minimum* cardinality. Let $\nu(G)$ denote the cardinality of a minimum dominating set for a graph G. It is well-known that the MDS and SC problems have precisely the same approximability via approximation-preserving reductions in both directions and, in particular, there exists a standard reduction from SC to MDS as follows. Given an instance $\mathcal{U} = \{a_1, a_2, \ldots, a_n\}$ and $S_1, S_2, \ldots, S_m \subseteq \mathcal{U}$ of SC, we create the following instance $G_1 = (V_1, E_1)$ of MDS. V_1 has an *element node* v_{a_i} for every element $a_i \in \mathcal{U}$ and a *set node* v_{S_j} for every set S_j with $j \in \{1, 2, \ldots, m\}$. There are two types of edges in E_1. Every set node v_{S_j} has an edge to every other set node v_{S_ℓ} and the collection of these edges is called the set of *clique edges*. Moreover, a set node v_{S_j} is connected to an element node v_{a_i} if and only if $a_i \in S_j$ and the collection of these edges is called the set of *membership edges*. A standard straightforward argument shows that $\mathcal{I} \subset \{1, 2, \ldots, m\}$ is a solution of SC if and only if the collection of set nodes $\{\, v_{S_i} \,|\, i \in \mathcal{I} \,\}$ is a solution of MDS on G_1 and thus $\mathsf{opt}_{\text{SC}} = \nu\,(G_1)$.

(a) ADIM$_{=k}$ belongs to NP for any k since given any solution V' it is straightforward to verify if $|V'| \leq \zeta$ and $\mu\left(\mathcal{D}_{V \setminus V', -V'}\right) = k$. Thus we need to show that it is also NP-hard.

For the purpose of our NP-hardness reduction, it would be more convenient to work with a restricted version of Sc known as the *exact cover by 3-sets* (X3C) problem. Here we have exactly n elements and exactly n sets where n is a multiple of three, every set contains exactly 3 elements and every element occurs in exactly 3 sets. Note that we need at least $\frac{n}{3}$ sets to cover all the n elements. Letting $\mathsf{opt}_{\text{X3C}}$ to denote the number of sets in an optimal solution of X3C, it is well-known that problem of deciding whether $\mathsf{opt}_{\text{X3C}} = \frac{n}{3}$ is in fact NP-complete (*e.g.*, see [9, p. 221]).

Let $n_1 = \frac{-6k+\sqrt{36k^2+24(n-k)}}{4}$ be the real-valued solution of the quadratic equation $n_1\left(2k + \frac{2n_1}{3}\right) + k = n$. Note that since $k \leq n^{\varepsilon}$ for some constant $\varepsilon < \frac{1}{2}$, we have $n_1 = \Theta\left(\sqrt{n}\right)$, *i.e.*, n and n_1 are "polynomially related".

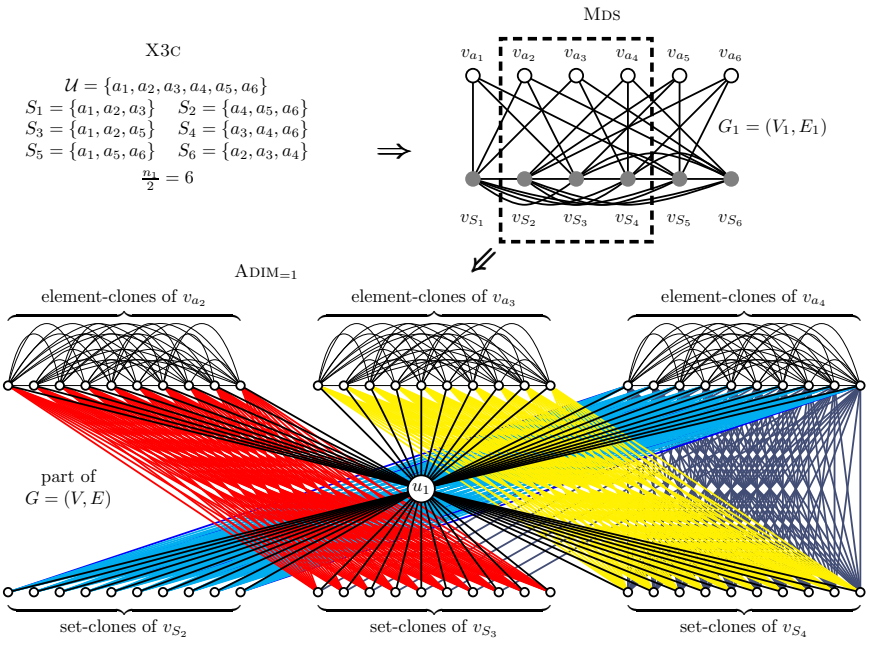


Figure 2: Illustration of the NP-hardness reduction in Theorem 2(a). Only a part of the graph G is shown for visual clarity (for example, non-member edges are not shown).

We assume without loss of generality that n_1 is an even integer, and start with an instance of X3C of $\frac{n_1}{2}$ elements and transform it to an instance graph

$G_1 = (V_1, E_1)$ having n_1 nodes of MDS via the reduction outlined before. Since $\frac{n_1}{2}$ is polynomially related to n, such an instance of X3C is NP-complete with respect to n being the input size. We reduce G_1 to an instance $G = (V, E)$ of $\text{ADIM}_{=k}$ in polynomial time as follows (see Fig. 2 for an illustration):

- We "clone" each element node $v_{a_j} \in V_1$ to get $2k + \frac{2n_1}{3}$ copies, *i.e.*, every node v_{a_j} is replaced by $2k + \frac{2n_1}{3}$ new nodes $v_{a_j,1}, v_{a_j,2}, \ldots, v_{a_j,2k+\frac{2n_1}{3}}$,. We refer to these nodes as *clones* of the element node v_{a_j} (or, sometimes simply as *element-clone nodes*). There are precisely $n_1\left(k + \frac{n_1}{3}\right)$ such nodes.

- We "clone" each set node $v_{S_j} \in V_1$ to get $2k + \frac{2n_1}{3}$ copies, *i.e.*, every node v_{S_j} is replaced by $2k + \frac{2n_1}{3}$ new nodes $v_{S_j,1}, v_{S_j,2}, \ldots, v_{S_j,2k+\frac{2n_1}{3}}$,. We refer to these nodes as *clones* of the set node v_{S_j} (or, sometimes simply as *set-clone nodes*). There are precisely $n_1\left(k + \frac{n_1}{3}\right)$ such nodes.

- We add k new nodes $u_1, u_2, \ldots, u_k$. We refer to these nodes as *clique nodes*.

- We add an edge between every pair of clique nodes u_i and u_j. We refer to these edges as *clique edges*. There are precisely $\binom{k}{2}$ such edges.

- We add an edge between every clique node and every non-clique node, *i.e.*, we add every edge in the set

$$\left\{ \left\{u_i, v_{a_j,\ell}\right\} \mid 1 \le i \le k,\, 1 \le j \le \frac{n_1}{2},\, 1 \le \ell \le 2k + \frac{2n_1}{3} \right\} \bigcup \left\{ \left\{u_i, v_{S_j,\ell}\right\} \mid 1 \le i \le k,\, 1 \le j \le \frac{n_1}{2}, 1 \le \ell \le 2k + \frac{2n_1}{3} \right\}$$

  We refer to these edges as the *partition-fixing* edges. There are precisely $kn_1\left(k + \frac{n_1}{3}\right)$ such edges.

- We add an edge between every pair of distinct element-clone nodes $v_{a_j,\ell}$ and $v_{a_{j'},\ell'}$. We refer to these as the *element-clone edges*. There are precisely $\binom{2k+(2n_1)/3}{2}$ such edges.

- For every element a_i and every set S_j such that $a_i \notin S_j$, we add the following $\left(2k + \frac{2n_1}{3}\right)^2$ edges:

$$\left\{v_{S_j,\ell}, v_{a_i,p}\right\} \quad \text{for} \quad 1 \leq \ell, p \leq 2k + \frac{2n_1}{3}$$

  We refer to these edges as the *non-member* edges corresponding to the element node a_i and the set node S_j. There are precisely $\frac{3n_1}{2}\left(2k + \frac{2n_1}{3}\right)^2$ such edges.

Note that G has precisely $n_1\left(2k + \frac{2n_1}{3}\right) + k = n$ nodes and thus our reduction is polynomial time in n. Since any clique node is adjacent to every other node in G, it follows that $\mathsf{diam}(G) = 2$. We now show the validity of our reduction by showing that

$$(\star)\ \nu\left(G_1\right) = \frac{n_1}{3} \ \text{ if and only if } \ \mathcal{L}_{\text{opt}}^{=k} \leq \frac{n_1}{3}$$

Proof of $\nu\left(G_1\right) = \frac{n_1}{3} \Rightarrow \mathcal{L}_{\text{opt}}^{=k} \leq \frac{n_1}{3}$

Consider an optimal solution $V_1' \subset \left\{v_{S_1}, v_{S_2}, \ldots, v_{S_{n_1}}\right\}$ of MDS on G_1 with $\nu\left(G_1\right) = |V_1'| = \frac{n_1}{3}$. We now construct a solution $V' \subset V$ of ADIM$_{=k}$ on G by setting $V' = \left\{v_{S_j,1} \,|\, v_{S_j} \in V_1'\right\}$. Note that $|V'| = |V_1'| = \frac{n_1}{3}$. We claim that V' is a valid solution of ADIM$_{=k}$ by showing that

(*a*) $\{u_1, u_2, \ldots, u_k\} \in \Pi^{=}_{V\setminus V', -V'}$ and

(*b*) any other equivalence class in $\Pi^{=}_{V\setminus V', -V'}$ has at least k nodes.

To prove (a), consider a clique node u_i and any other non-clique node. Then, the following cases apply:

- Suppose that the non-clique node is a element-clone node $v_{a_j,\ell} \in V\setminus V'$ for some j and ℓ. Since V_1' is a solution of MDS on G_1, there exists a set node $v_{S_p} \in V_1'$ such that $\left\{v_{S_p}, v_{a_j}\right\} \in E_1$ and consequently $\left\{v_{S_p,1}, v_{a_j,\ell}\right\} \notin E$. This implies that there exists a node $v_{S_p,1} \in V'$ such that $1 = \text{dist}_{u_i, v_{a_j,\ell}} \neq \text{dist}_{v_{S_p,1}, v_{a_j,\ell}}$, and therefore $v_{a_j,\ell}$ *cannot* be in the same equivalence class with u_i.

- Suppose that the non-clique node is a set-clone node $v_{S_j,p} \in V \setminus V'$. Pick any set-clone node $v_{S_\ell,1} \in V'$. Then, $1 = \text{dist}_{u_i,v_{S_j,p}} \neq \text{dist}_{v_{S_j,p},v_{S_\ell,1}}$, and therefore $v_{S_j,p}$ *cannot* be in the same equivalence class with u_i.

To prove (b), note the following:

- Since $\mathsf{diam}(G) = 2$, $\text{dist}_{v_{S_i,p},v_{S_j,q}} = 2$ for any two *distinct* set-clone nodes $v_{S_i,p}$ and $v_{S_j,q}$, and thus all the set nodes in $V \setminus V'$ belong together in the *same* equivalence class in $\Pi^{=}_{V\setminus V',-V'}$. There are at least $n_1\left(k + \frac{n_1}{3}\right) - \frac{n_1}{3} > k$ such nodes in $V \setminus V'$. Thus, any equivalence class that contains these set-clone nodes cannot have less than k nodes.

- Consider now an equivalence class in $\Pi^{=}_{V\setminus V',-V'}$ that contains a copy $v_{a_i,j}$ of the element node v_{a_i} for some i and j. Consider another copy $v_{a_i,\ell}$ of the element node v_{a_i} for some $\ell \neq j$. For any set node $v_{S_p,1} \in V'$, if $a_i \notin S_p$ then $\text{dist}_{v_{S_p,1},v_{a_i,j}} = \text{dist}_{v_{S_p,1},v_{a_i,\ell}} = 1$, whereas if $a_i \in S_p$ then, since $\mathsf{diam}(G) = 2$, it follows that $\text{dist}_{v_{S_p,1},v_{a_i,j}} = \text{dist}_{v_{S_p,1},v_{a_i,\ell}} = 2$. Thus, any equivalence class that contains at least one clone of an element node must contain all the $2k + \frac{2n_1}{3} > k$ clones of that element node and thus such an equivalence class cannot have a number of nodes that is less than k.

Proof of $\mathcal{L}^{=k}_{\text{opt}} \leq \frac{n_1}{3} \Rightarrow \nu(G_1) = \frac{n_1}{3}$

Since we know that $\nu(G_1)$ is always at least $\frac{n_1}{3}$, it suffices to show that $\mathcal{L}^{=k}_{\text{opt}} \leq \frac{n_1}{3} \Rightarrow \nu(G_1) \leq \frac{n_1}{3}$. Consider an optimal solution $V^{=k}_{\text{opt}} \subset V$ with $\mathcal{L}^{=k}_{\text{opt}} = \left|V^{=k}_{\text{opt}}\right| = \frac{n_1}{3}$. Since $V^{=k}_{\text{opt}}$ is a solution of $\textsc{Adim}_{=k}$ on G, there exists a subset of nodes, say $\widehat{V} \subset V \setminus V^{=k}_{\text{opt}}$, such that $|\widehat{V}| = k$ and $\widehat{V} \in \Pi^{=}_{V\setminus V^{=k}_{\text{opt}},-V^{=k}_{\text{opt}}}$.

Proposition 3. *$\widehat{V}$ does not contain any set-clone or element-clone nodes and thus $\widehat{V} = \{u_1, u_2, \ldots, u_k\}$.*

Proof. Suppose that $\widehat{V}$ contains at least one element-clone node $v_{a_i,j}$ for some i and j. But, $V \setminus V^{=k}_{\text{opt}}$ contains at least $2k + \frac{2n_1}{3} - \frac{n_1}{3} - 1 > k$ other clones of

the element node a_i and all these clones must belong together with $v_{a_i,j}$ in the *same* equivalence class. This implies $|\widehat{V}| \geq 2k + \frac{2n_1}{3} - \frac{n_1}{3} > k$, a contradiction.

Similarly, suppose that $\widehat{V}$ contains at least one set-clone node $v_{S_i,j}$ for some i and j. But, $V \setminus V^{=k}_{\text{opt}}$ contains at least $2k + \frac{2n_1}{3} - \frac{n_1}{3} - 1 > k$ other clones of the set node S_i and all these clones must belong together with $v_{S_i,j}$ in the *same* equivalence class. This implies $|\widehat{V}| \geq 2k + \frac{2n_1}{3} - \frac{n_1}{3} > k$, a contradiction. ❑

Proposition 4. *$V^{=k}_{\text{opt}}$ does not contain two or more clones of the same set node.*

Proof. Suppose that $V^{=k}_{\text{opt}}$ contains two set-clone nodes $v_{S_j,p}$ and $v_{S_j,q}$ of the same set node v_{S_j}. But, $V \setminus V^{=k}_{\text{opt}}$ contains at least $2k + \frac{2n_1}{3} - \frac{n_1}{3} - 1 > k$ other clones of the element node a_i and all these clones must belong together in the *same* equivalence class S. If we remove $v_{S_j,p}$ from $V^{=k}_{\text{opt}}$ then $v_{S_j,p}$ gets added to this equivalence class. Thus, such a removal produced another valid solution but with one node less than $\mathcal{L}_{\text{opt}}$, contradicting the optimality of $\mathcal{L}^{=k}_{\text{opt}}$. ❑

Proposition 5. *$V^{=k}_{\text{opt}}$ does not contain any element-clone node.*

Proof. Suppose that $V^{=k}_{\text{opt}}$ contains at least one element-clone node and thus at most $\frac{n_1}{3} - 1$ set-clone nodes. Note that $V \setminus V^{=k}_{\text{opt}}$ contains at least $2k + \frac{2n_1}{3} - \frac{n_1}{3}$ clones of every element node a_i. Consider an element-clone node $v_{a_i,p} \in V \setminus V^{=k}_{\text{opt}}$ and a clique node u_j. Since $\widehat{V} = \{u_1, u_2, \ldots, u_k\} \in \Pi^{=}_{V \setminus V^{=k}_{\text{opt}}, -V^{=k}_{\text{opt}}}$, there must be a node in $V^{=k}_{\text{opt}}$ such that the distance of this node to u_j is different from the distance to $v_{a_i,p}$. Such a node in $V^{=k}_{\text{opt}}$ cannot be an element-clone node, say $v_{a_\ell,q}$ since $\text{dist}_{v_{a_i,p},v_{a_\ell,q}} = \text{dist}_{u_j,v_{a_\ell,q}} = 1$. Since there is an edge between every set-clone node and every clique node, such a node must be a set-clone node, say $v_{S_r,s}$ for some r and s, such that $\text{dist}_{v_{a_i,p},v_{S_r,s}} = 2$, *i.e.*, $a_i \in S_r$. Since every set in X3C contains exactly 3 elements and $3 \times \left(\frac{n_1}{3} - 1\right) < n_1$, there must exist an element-clone node $v_{a_i,p}$ such that the distance of $v_{a_i,p}$ to any node in $V^{=k}_{\text{opt}}$ is exactly the same as the distance of u_j to that node in $V^{=k}_{\text{opt}}$. This implies $v_{a_i,p} \in \widehat{V}$, contradicting Proposition 3. ❑

By Proposition 4 and Proposition 5, $V_{\text{opt}}^{=k}$ contains exactly one clone of a subset of set nodes. Without loss of generality, assume that $V_{\text{opt}}^{=k} = \left\{ v_{S_j,1} \,|\, j \in J,\, J \subset \left\{1, 2, \ldots, \frac{n_1}{2}\right\} \right\}$ and let $V_1' = \left\{ v_{S_j} \,|\, v_{S_j,1} \in V_{\text{opt}}^{=k} \right\}$. Note that $|V_1'| = \left|V_{\text{opt}}^{=k}\right|$. We are now ready to finish our proof by showing V_1' is indeed a valid solution of MDS on G_1. Suppose not, and let v_{a_i} be an element-node that is not adjacent to any node in V_1'. For this case,

$$\forall\, v_{S_j} \in V_1' \,:\, \left\{v_{a_i}, v_{S_j}\right\} \notin E_1 \;\Rightarrow\; \forall\, v_{S_j,1} \in V_{\text{opt}}^{=k} \,:\, \left\{v_{a_i,1}, v_{S_j,1}\right\} \in E$$
$$\Rightarrow\; \forall\, v_{S_j,1} \in V_{\text{opt}}^{=k} \,:\, \text{dist}_{v_{a_i,1}, v_{S_j,1}} = 1 \;\Rightarrow\; v_{a_i,1} \in \widehat{V}$$

which contradicts Proposition 3.

(b) The proof is similar to that of (a) but this time we start with a general version of SC as opposed to the restricted X3C version, and show that the reduction is approximation-preserving in an appropriate sense. In the sequel, we use the standard notation poly(n) to denote a polynomial n^c of n (for some constant $c > 0$). We recall the following details of the inapproximability reduction of Feige in [7]. Given an instance formula ϕ of the standard Boolean satisfiability problem (SAT), Feige reduces ϕ to an instance $\mathcal{U}, S_1, S_2, \ldots, S_m$ of SC (with $m = \text{poly}(n)$) in $O(n^{\log \log n})$ time such that the following properties are satisfied for any constant $0 < \varepsilon < 1$:

- For some $Q > 0$, either $\mathsf{opt}_{\text{SC}} = \frac{n}{Q}$ or $\mathsf{opt}_{\text{SC}} > \left(\frac{n}{Q}\right)(1-\varepsilon)\ln n$.
- The reduction satisfies the following completeness and soundness properties:

  (completeness) If ϕ is satisfiable then $\mathsf{opt}_{\text{SC}} = \frac{n}{Q}$.

  (soundness) If ϕ is not satisfiable then $\mathsf{opt}_{\text{SC}} > \left(\frac{n}{Q}\right)(1-\varepsilon)\ln n$.

Since $m = \text{poly}(n)$, by adding duplicate copies of a set, if necessary, we can ensure that $m = n^c - n$ for some constant $c \geq 1$. Our reduction from SC to MDS to $\text{ADIM}_{=k}$ is same as in (a) except that some details are different, which we show here.

- We start with an instance of Sc as given by Feige in [7] with n_1 elements and $m = (n_1)^c - n_1$ sets, where $n_1 = \left(\frac{-k+\sqrt{k^2+2(n-k)}}{2}\right)^{1/c}$ is a real-valued solution of the equation $(n_1)^{2c} + k(n_1)^c - \frac{n-k}{2} = 0$. Note that since $k \leq n^\varepsilon$ for some constant $\varepsilon < \frac{1}{2}$, we have $n_1 = \Theta\left(n^{1/(2c)}\right)$, *i.e.*, n and n_1 are polynomially related.

- We make $2(n_1)^c + 2k$ copies of each element node and each set node as opposed to $2k + \frac{2n_1}{3}$ copies that we made in the proof of (a). Note that G has again precisely $(n_1)^c\,(2k + 2(n_1)^c\,) + k = n$ nodes.

- Let $\delta > 0$ be the constant given by $\delta = \frac{\ln n}{(1-\varepsilon)\ln n_1}$. Our claim $(\star)$ in the proof of (a) is now modified to

$$(\star)\qquad \begin{array}{ll} \text{(completeness)} & \text{if } \nu\,(G_1) = \frac{n_1}{Q} \text{ then } \mathcal{L}^{=k}_{\text{opt}} \leq \frac{n_1}{Q} \\ \text{(soundness)} & \text{if } \nu\,(G_1) > \left(\frac{n_1}{Q}\right)(1-\varepsilon)\ln n_1 \\ & \qquad \text{then } \mathcal{L}^{=k}_{\text{opt}} > \left(\frac{n_1}{Q}\right)(1-\varepsilon)\ln n_1 = \left(\frac{n_1}{Q}\right)\frac{1}{\delta}\ln n \end{array}$$

- Our proof of the *completeness* claim follows the "Proof of $\nu\,(G_1) = \frac{n_1}{3} \Rightarrow \mathcal{L}^{=k}_{\text{opt}} \leq \frac{n_1}{3}$" in the proof of (a) with the obvious replacement of $\frac{n_1}{3}$ by $\frac{n_1}{Q}$.

- Note that our soundness claim is equivalent to its contra-positive

$$\text{if}\quad \mathcal{L}^{=k}_{\text{opt}} \leq \left(\frac{n_1}{Q}\right)(1-\varepsilon)\ln n_1 \quad \text{then}\quad \nu\,(G_1) \leq \left(\frac{n_1}{Q}\right)(1-\varepsilon)\ln n_1$$

  and the proof of this contra-positive follows the "Proof of $\mathcal{L}^{=k}_{\text{opt}} \leq \frac{n_1}{3} \Rightarrow \nu\,(G_1) = \frac{n_1}{3}$" in the proof of (a). In the proof, the quantity $2k + \frac{2n_1}{3}$ corresponding to the number of copies for each set and element node needs to be replaced by $2(n_1)^c + 2k$; note that $(2(n_1)^c + 2k) - n_1 \gg k$.

(c) Since $k = n - c$ for some constant c, $\Pi^{=}_{V\setminus V^{=k}_{\text{opt}}, -V^{=k}_{\text{opt}}}$ contains a single equivalence class $V' \subset V$ such that $|V'| = k$. Thus, we can employ the straightforward exhaustive method of selecting every possible subset V' of k nodes to be in $\Pi^{=}_{V\setminus V', -V'}$ and checking if the chosen subset of nodes provide a valid solution. There are $\binom{n}{k} < n^c$ such possible subsets and therefore the asymptotic running time is $O\left(n^c + n^3\right)$ which is polynomial in n. Note that for this case $\mathcal{L}^{=k}_{\text{opt}} = c$ if a solution exists.

6. Proof of Theorem 3

(a) Note that $\mathcal{L}_{\text{opt}}^{=1} \leq n-1$ and thus $V_{\text{opt}}^{=1} \neq \emptyset$. Our algorithm, shown as Algorithm V, uses the greedy logarithmic approximation of Johnson [10] for Sc that selects, at each successive step, a set that contains the maximum number of elements that are still not covered.

Algorithm V: $O\left(n^3\right)$-time $(1+\ln(n-1)\,)$-approximation algorithm for $\textsc{Adim}_{=1}$.

---

1. Compute $\mathbf{d}_i$ for all $i = 1, 2, \ldots, n$ in $O\left(n^3\right)$ time using Floyd-Warshall algorithm.

2. $\widehat{\mathcal{L}_{\text{opt}}^{=1}} \leftarrow \infty$; $\widehat{V_{\text{opt}}^{=1}} \leftarrow \emptyset$

3. **for** each node $v_i \in V$ **do** ($*$ we *guess* the set $\{v_i\}$ to belong to $\Pi^{=}_{V \setminus V_{\text{opt}}^{=1}, -V_{\text{opt}}^{=1}}$ $*$)

3.1 create the following instance of Sc containing $n-1$ elements and $n-1$ sets:

$$\mathcal{U} = \left\{\, a_{v_j} \,|\, v_j \in V \setminus \{v_i\} \,\right\},$$

$$S_{v_j} = \left\{a_{v_j}\right\} \cup \left\{\, a_{v_\ell} \,|\, \text{dist}_{v_i,v_j} \neq \text{dist}_{v_\ell,v_j} \right\} \text{ for } j \in \{1,2,\ldots,n\} \setminus \{i\}$$

3.2 **if** $\cup_{j \in \{1,2,\ldots,n\} \setminus \{i\}} S_{v_j} = \mathcal{U}$ **then**

3.2.1 run the greedy approximation algorithm [10] for this instance of Sc giving a solution $\mathcal{I} \subseteq \{1,2,\ldots,n\} \setminus \{i\}$

3.2.2 $V' = \{\, v_j \,|\, j \in \mathcal{I} \,\}$

3.2.3 **if** $\left(\, |V'| < \widehat{\mathcal{L}_{\text{opt}}^{=1}} \,\right)$ **then** $\widehat{\mathcal{L}_{\text{opt}}^{=1}} \leftarrow |V'|$; $\widehat{V_{\text{opt}}^{=1}} \leftarrow V'$

4. **return** $\widehat{\mathcal{L}_{\text{opt}}^{=1}}$ and $\widehat{V_{\text{opt}}^{=1}}$ as our solution

---

Lemma 6 (Proof of correctness). *Algorithm V returns a valid solution for* $\textsc{Adim}_{=1}$.

Proof. Suppose that our algorithm returns an invalid solution in the iteration of the **for** loop in Step 3 when v_i is equal to v_ℓ for some $v_\ell \in V$. We claim that this cannot be the case since $\{v_\ell\} \in \Pi^{=}_{V \setminus V', -V'}$. Indeed, since $\mathcal{I}$ is a valid

solution of the Sc instance, for every $j \notin \{\ell\} \cup \mathcal{I}$, the following holds:

$$\exists\, t \in \mathcal{I} \,:\, a_{v_j} \in S_{v_t} \;\Rightarrow\; \exists\, v_t \in V' \,:\, \mathrm{dist}_{v_\ell, v_t} \neq \mathrm{dist}_{v_j, v_t}$$

and thus v_ℓ cannot be together with any other node in any equivalence class in $\Pi^{=}_{V \setminus V', -V'}$. ❑

Lemma 7 (Proof of approximation bound). *Algorithm V solves* ADIM$_{=1}$ *with an approximation ratio of* $1 + \ln(n-1)$.

Proof. Fix any optimal solution $V^{=1}_{\mathrm{opt}}$. Since $\mu\left(\mathcal{D}_{V \setminus V^{=1}_{\mathrm{opt}}, -V^{=1}_{\mathrm{opt}}}\right) = 1$, $\{v_\ell\} \in \Pi^{=}_{V \setminus V^{=1}_{\mathrm{opt}}, -V^{=1}_{\mathrm{opt}}}$ for some $v_\ell \in V$. Consider the iteration of the **for** loop in Step 3 when v_i is equal to v_ℓ. We now analyze the run of *this particular iteration*, and claim that the set-cover instance created during this iteration satisfies $\mathsf{opt}_{\mathrm{Sc}} \leq \left|V^{=1}_{\mathrm{opt}}\right| = \mathcal{L}^{=1}_{\mathrm{opt}}$. To see this, construct the following solution of the set-cover instance from V_{opt} containing exactly $\mathcal{L}_{\mathrm{opt}}$ sets:

$$v_i \in V^{=1}_{\mathrm{opt}} \;\equiv\; i \in \mathcal{I}$$

To see that this is indeed a valid solution of the set-cover instance, consider any $a_{v_j} \in \mathcal{U} = \{a_{v_1}, a_{v_2}, \ldots, a_{v_n}\} \setminus \{a_{v_\ell}\}$. Then, the following cases apply showing that a_{v_j} belongs to some set selected in our solution of Sc:

- if $j \in \mathcal{I}$ then $a_{v_j} \in S_{v_j}$ and S_{v_j} is a selected set in the solution.
- if $j \notin \mathcal{I}$ then $v_j \in V \setminus V_{\mathrm{opt}} \;\Rightarrow\; \exists\, v_t \in V_{\mathrm{opt}} \,:\, \mathrm{dist}_{v_\ell, v_t} \neq \mathrm{dist}_{v_j, v_t} \;\Rightarrow\; \exists\, t \in \mathcal{I} \,:\, a_{v_j} \in S_{v_t}$.

Using the approximation bound of the algorithm of [10] it now follows that the quality of our solution $\widehat{\mathcal{L}^{=1}_{\mathrm{opt}}}$ satisfies

$$\widehat{\mathcal{L}^{=1}_{\mathrm{opt}}} = \left|\widehat{V^{=1}_{\mathrm{opt}}}\right| = |\mathcal{I}| < (1 + \ln(n-1)\,)\mathsf{opt}_{\mathrm{Sc}} \leq (1 + \ln(n-1)\,)\mathcal{L}^{=1}_{\mathrm{opt}}$$

❑

Lemma 8 (Proof of time complexity). *Algorithm V runs in* $O\left(n^3\right)$ *time.*

Proof. There are a total of n instances of set cover that we need to build in Step 3.1 and solve by the greedy heuristic in Step 3.2.1. Building the set-cover instance can be done in $O\left(n^2\right)$ time by comparing dist_{v_i,v_j} for all appropriate pairs of nodes v_i and v_j. Since the set-cover instance in Step 3.1 has $n-1$ sets each having no more than $n-1$ elements, each implementation of the greedy heuristic in Step 3.2.1 takes $O\left(n^2\right)$ time. ❑

(b) Let v_i be the node of degree 1. Let v_ℓ be the unique node adjacent to v_i (*i.e.*, $\{v_i, v_\ell\} \in E$). Consider the following solution of $\textsc{Adim}_{=1}$: $V' = \{v_i\}$. We claim that is a valid solution of $\textsc{Adim}_{=1}$ by showing that $\{v_\ell\} \in \Pi^{=}_{V\setminus V',-V'}$. Consider any node $v_j \in V \setminus \{v_i, v_\ell\}$, Then, $1 = \text{dist}_{v_\ell,v_i} \neq \text{dist}_{v_j,v_i}$.

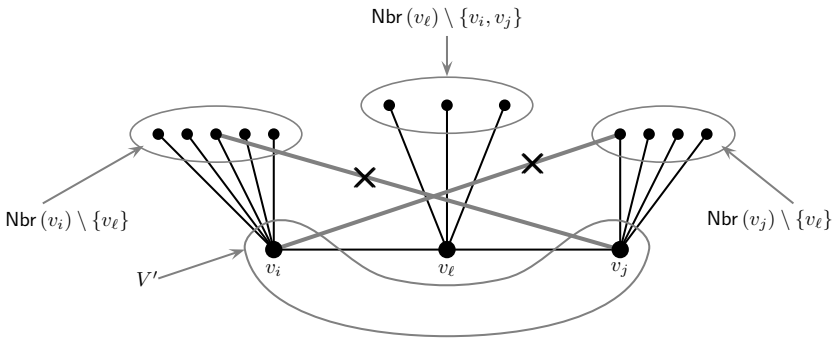


Figure 3: Illustration of the proof of Theorem 3(c). Edges marked by × cannot exist. No node in $\mathsf{Nbr}\,(v_\ell) \setminus \{v_i, v_j\}$ can have an edge to *both* v_i and v_j.

(c) Since G does not contain a 4-cycle, $\mathsf{diam}(G) \geq 2$. Thus, there exists two nodes $v_i, v_j \in V$ such that $\text{dist}_{v_i,v_j} = 2$. Let v_ℓ be a node at a distance of 1 from both v_i and v_j on a shortest path between v_i and v_j (see Fig. 3). Consider the following solution of $\textsc{Adim}_{=1}$: $V' = \{v_i, v_j\}$. Note that $v_\ell \in V \setminus V'$. We claim that this is a valid solution of $\textsc{Adim}_{=1}$ by showing that $\{v_\ell\} \in \Pi^{=}_{V\setminus V',-V'}$ (*i.e.*, no node $v_p \in V \setminus \{v_i, v_j, v_\ell\}$ can belong together with v_ℓ in the same equivalence class of $\Pi^{=}_{V\setminus V',-V'}$) in the following manner:

- If $v_p \in \mathsf{Nbr}(v_i) \setminus \{v_\ell\}$ then $\text{dist}_{v_\ell,v_j} = 1$ but $\text{dist}_{v_p,v_j} \neq 1$ since G has no 4-cycle (see the edges marked $\times$ in Fig. 3).

- If $v_p \in \mathsf{Nbr}(v_j) \setminus \{v_\ell\}$ then $\text{dist}_{v_\ell,v_i} = 1$ but $\text{dist}_{v_p,v_i} \neq 1$ since G has no 4-cycle (see the edges marked $\times$ in Fig. 3).

- If $v_p \in \mathsf{Nbr}(v_\ell) \setminus \{v_i, v_j\}$ then v_p cannot be adjacent to *both* v_i and v_j since G does not contain a 4-cycle. This implies that $\text{dist}_{v_\ell,v_i} = \text{dist}_{v_\ell,v_j} = 1$ but at least one of dist_{v_p,v_i} and dist_{v_p,v_j} is not equal to 1.

- If v_p is any node not covered by the above cases, we have $\text{dist}_{v_p,v_i} > 1$ but $\text{dist}_{v_\ell,v_i} = 1$.

7. Related Works: Other Privacy Concepts and Measures

There is a rich literature on theoretical investigations of privacy measures and privacy preserving computational models in several other application areas such as multi-party communications, distributed computing and game-theoretic settings (*e.g.*, see [2, 11, 22, 8, 3]). However, none of these settings apply directly to our application scenario of active attack model for social networks. The differential privacy model, introduced by Dwork [5] in the context of privacy preservation in statistical databases against malicious database queries, works by computing the correct answer to a query and adding a noise drawn from a specific distribution, and is quite different from the anonymization approach studied in this paper.

8. Concluding Remarks

Prior to our work, known results for the three problems considered in this paper only included some heuristic algorithms with no provable guarantee on performances such as in [18], or algorithms for very special cases. In fact, it was not even known if any version of these computational problems is NP-hard. Our work provides the first non-trivial computational complexity results for effective solution of these problems. Theorem 1 shows that both ADIM and $\text{ADIM}_{\geq k}$

are *provably* computationally easier problems than $\textsc{Adim}_{=k}$. In contrast, Theorem 2(a)–(b) and Theorem 3 show that $\textsc{Adim}_{=k}$ is in general computationally hard but admits approximations or exact solution for specific choices of k or graph topology. We believe that our results will stimulate further research on quantifying and computing privacy measures for networks. In particular, our results raise the following interesting research questions:

- We have only provided a logarithmic approximation algorithm for $\textsc{Adim}_{=1}$. Is it possible to design a non-trivial approximation algorithm for $\textsc{Adim}_{=k}$ for $k > 1$?

- We have provided a logarithmic inapproximability result for $\textsc{Adim}_{=k}$ for every k *roughly* up to $\sqrt{n}$. Can this approximability result be further improved when k is not a constant ?

References

[1] L. Backstrom, C. Dwork and J. Kleinberg. *Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography*, Proc. 16th International Conference on World Wide Web, 181-190, New York, NY, USA, 2007.

[2] R. Bar-Yehuda, B. Chor, E. Kushilevitz and A. Orlitsky. *Privacy, additional information, and communication*, IEEE Transactions on Information Theory, 39, 55-65, 1993.

[3] M. Comi, B. DasGupta, M. Schapira and V. Srinivasan. *On Communication Protocols that Compute Almost Privately*, Theoretical Computer Science, 457, 45-58, 2012.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*, 2nd edition, The MIT Press, 2001.

[5] C. Dwork. *Differential Privacy*, Proc. 33rd International Colloquium on Automata, Languages and Programming, 1-12, 2006.

[6] T. Feder, S. U. Nabar and E. Terzi. *Anonymizing graphs*, CoRR, abs/0810.5578, 2008.

[7] U. Feige. *A threshold for approximating set cover*, Journal of the ACM, 45, 634-652, 1998.

[8] J. Feigenbaum, A. Jaggard and M. Schapira. *Approximate Privacy: Foundations and Quantification*, Proc. ACM Conference on Electronic Commerce, 167-178, 2010.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., 1979.

[10] D. S. Johnson. *Approximation Algorithms for Combinatorial Problems*, Journal of Computer and System Sciences, 9, 256-278, 1974.

[11] E. Kushilevitz. *Privacy and communication complexity*, SIAM Journal on Discrete Mathematics, 5(2), 273-284, 1992.

[12] K. Liu and E. Terzi. *Towards identity anonymization on graphs*, Proc. 2008 ACM SIGMOD International Conference on Management of Data, 93-106, New York, NY, USA, 2008.

[13] S. Mauw, R. Trujillo-Rasua and B. Xuan. *Counteracting active attacks in social network graphs*, 30$^{\text{th}}$ Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy, Trento, Italy, 2016.

[14] A. Narayanan and V. Shmatikov. *De-anonymizing social networks*, 30$^{\text{th}}$ IEEE Symposium on Security and Privacy, 173-187, 2009.

[15] M. Netter, S. Herbst and G. Pernul. *Analyzing privacy in social networks– an interdisciplinary approach*, IEEE 3$^{\text{rd}}$ International Conference on Privacy, Security, Risk and Trust and IEEE 3$^{\text{rd}}$ International Conference on Social Computing, 1327-1334, 2011.

[16] P. Samarati and L. Sweeney. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*, Technical report, 1998.

[17] R. Trujillo-Rasua and I. G. Yero. *Characterizing* 1*-metric antidimensional trees and unicyclic graphs*, The Computer Journal, 59(8), 1264-1273, 2016.

[18] R. Trujillo-Rasua and I. G. Yero. *k-Metric antidimension: A privacy measure for social graphs*, Information Sciences, 328, 403-417, 2016.

[19] V. Vazirani. *Approximation Algorithms*, Springer-Verlag, 2001.

[20] B. Viswanath, M. Mondal, K. P. Gummadi, A. Mislove and A. Post. *Canal: Scaling social network-based sybil tolerance schemes*, Proc. 7^{th} ACM European Conference on Computer Systems, 309-322, New York, NY, USA, 2012.

[21] X. Wu, X. Ying, K. Liu and L. Chen. *A survey of privacy-preservation of graphs and social networks*, in C. C. Aggarwal and H. Wang (eds.), Managing and Mining Graph Data, Vol. 40 of Advances in Database Systems, 421-453. Springer, 2010.

[22] A. C. Yao. *Some complexity questions related to distributive computing (preliminary report)*, Proc. 11^{th} ACM Symposium on Theory of Computing, 209-213, 1979.

[23] C. Zhang and Y. Gao. *On the Complexity of k-Metric Antidimension Problem and the Size of k-Antiresolving Sets in Random Graphs*, Y. Cao and J. Chen (Eds.), COCOON 2017, LNCS 10392, 555-567, Springer, 2017.

[24] B. Zhou, J. Pei and W. S. Luk. *A brief survey on anonymization techniques for privacy preserving publishing of social network data*, SIGKDD Explorations Newsletter, 10(2), 12-22, 2008.

[25] L. Zou, L. Chen and M. T. Özsu. *K-automorphism: A general framework for privacy preserving network publication*, Proc. VLDB Endowment, 2(1), 946-957, 2009.