UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARCELO GARLET MILLANI

Algoritmos e Complexidade para Anonimização de Grau em Grafos Direcionados

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Trabalho realizado na Technische Universität Berlin dentro do acordo de dupla diplomação UFRGS - TU Berlin.

Orientador: Prof. Dr. Marcus Ritt

Porto Alegre
2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco
Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb
Coordenador do Curso de Ciência de Computação: Prof. Raul Fernando Weber
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

RESUMO

A relevância de redes sociais para mineração de dados cresceu consideravelmente nos últimos anos. No entanto, já foi mostrado que a preservação da anonimidade dos usuários envolvidos não é uma tarefa fácil.

Neste trabalho, consideramos o problema de k-anonimidade em grafos direcionados (dígrafos), onde um dígrafo é considerado k-anônimo quando para cada um de seus vértices existe pelo menos $k-1$ outros vértices com o mesmo grau. Esse modelo assume que um agressor tentando desanonimizar a rede conhece os graus do vértices do dígrafo original. Nós analisamos a complexidade do problema, provando sua NP-dificuldade e sua inaproximabilidade em tempo polinomial e FPT. Finalmente, investigamos o caso especial onde o grau é limitado e propomos uma solução eficiente para ele.

Primeiramente, definimos formalmente os problemas a serem tratados. Consideramos apenas o caso em que alteramos o conjunto de arcos, definindo dessa forma dois problemas de decisão. Quando apenas inserimos novos arcos, estamos considerando DIGRAPH DEGREE ANONYMITY A-ADD, definido a seguir.

DIGRAPH DEGREE ANONYMITY A-ADD

Entrada: Um dígrafo $D = (V, A)$ e dois inteiros positivos k e s.

Pergunta: Existe um conjunto de arcos A' sobre V com $|A'| \leq s$ de tal forma que $D' = (V, A \cup A')$ é k-anônimo, ou seja, para cada vértice $v \in V$ existem pelo menos $k-1$ outros vértices em D' com os mesmos graus de entrada e saída?

O problema onde apenas remove-se arcos é chamado de DIGRAPH DEGREE ANONYMITY A-DEL, definido a seguir.

DIGRAPH DEGREE ANONYMITY A-DEL

Entrada: Um dígrafo $D = (V, A)$ e dois inteiros positivos k e s.

Pergunta: Existe um conjunto de arcos A' sobre V com $|A'| \leq s$ de tal forma que $D' = (V, A - A')$ é k-anônimo, ou seja, para cada vértice $v \in V$ existem pelo menos $k-1$ outros vértices em D' com os mesmos graus de entrada e saída?

Podemos provar que ambos os problemas são equivalentes, ou seja, adicionar um arco a um dígrafo D produz o mesmo dígrafo que remover o mesmo arco de $\overline{D}$ e fazer com complemento do resultado. Formalmente, provamos que, para qualquer conjunto de arcos S, vale $D \cup S = \overline{\overline{D} - S}$. Essa equivalência é utilizado em algumas das provas presentes neste trabalho.

Uma vez definidos esses problemas, provamos que ambos são NP-completos. É fácil ver que ambos estão em NP, mas a prova de que são NP-difíceis requer mais argumentações. Para

tal, mostramos uma redução de INDEPENDENT SET em grafos 3-regulares para DIGRAPH DEGREE ANONYMITY A-ADD. A escolha do problema para a redução é intuitiva primeiramente por ter sido usado para o caso não-direcionado. Adicionalmente, inserir arcos entre vértices de um conjunto independente é sempre possível, o que simplifica a prova.

A redução é construída de tal forma que só é possível anonimizar o dígrafo da instância reduzida quando há um conjunto independente suficientemente grande na instância original. Garantimos isso através da inserção de um dígrafo estrela, o qual possui um grau muito superior ao dos outros nodos (lembre que o grafo original é 3-regular, ou seja, possui apenas nodos de grau três). Com isso, são necessárias tantas arestas quanto um dígrafo completo possui para a anonimização. Logo, a instância de INDEPENDENT SET possui uma solução se e somente se a instância equivalente de DIGRAPH DEGREE ANONYMITY A-ADD também possui. Como corolário, concluímos com essa redução que DIGRAPH DEGREE ANONYMITY A-DEL também é NP-difícil.

Em sequência, mostramos que o problema permanece NP-difícil mesmo se o grafo correspondente ao dígrafo D de entrada já é k-anônimo. Esse resultado mostra que, mesmo primeiro aplicando heurísticas existentes para grafos antes de se considerar o problema em dígrafos, não é possível "concluir" em tempo polinomial o processo de anonimização no dígrafo. Para a prova, realizamos novamente uma redução INDEPENDENT SET em grafos 3-regulares para DIGRAPH DEGREE ANONYMITY A-ADD, mas agora adicionamos diversos dígrafos-estrela. As novas estrelas garantem que o grafo correspondente será k-anônimo mas, por diferirem no sentido de seus arcos, o dígrafo não é k-anônimo. O restante da prova é muito similar a anterior.

Depois de provarmos que os problemas de decisão em questão são NP-completos, decidimos analisar a possibilidade de se utilizar algoritmos de aproximação. Ao considerarmos aproximações, trabalhamos com problemas de otimização em vez de decisão. Formalmente, consideraremos o problema a seguir.

DIGRAPH MAXIMUM DEGREE ANONYMITY BY ARC DELETION (DIGRAPH MAX-ANONYM A-DEL)

Entrada: Um dígrafo $D = (V, A)$ e um inteiro $s > 0$.

Tarefa: Encontre um conjunto de arcos S com $|S| \leq s$ de tal forma $D - S$ é k-anônimo e k é maximizado.

Se tentarmos maximizar a anonimidade do dígrafo, não existe um algoritmo de aproximação que garanta algum resultado útil e que rode em tempo polinomial ou FPT. Mais especificamente, provamos que não existe um algoritmo de aproximação de fator $|V|^{1-\epsilon}$, com

$0 < \epsilon \leq 1$.

Provamos inaproximabilidade em tempo polinomial através de uma *gap-reduction* de EXACT 3-COVER, definido a seguir.

EXACT 3-COVER

Entrada: Um universo $A = \{a_1, a_2, \ldots, a_{3h}\}$, uma coleção $\mathcal{B} = \{B_1, B_2, \ldots, B_\beta\}$ de conjuntos de três elementos sobre A, e $h \in \mathbb{N}$.

Pergunta: Existe um conjunto de índices $J \subseteq \{1, 2, \ldots, \beta\}$ com $|J| = h$ tal que $\bigcup_{j \in J} B_j = A$?

No processo de redução, a correlação entre a solução dos dois problemas é garantida ao requerermos que qualquer arco removido tenha um vértice correspondente a um elemento de A em uma das pontas e outro correspondente a um elemento de $\mathcal{B}$. Os graus dos vértices na instância reduzidas são escolhidos de tal forma que, se for possível aumentar os graus de entrada e saída de todos os vértices associados a A em um, então a anonimidade é melhorada consideravelmente. Caso contrário, é possível fazer apenas pequenas melhoras. Essa discrepância permite decidir EXACT 3-COVER baseando-se no valor obtido pela aproximação, a qual garante uma certa qualidade do resultado. Assim sendo, se pudermos aproximar DIGRAPH MAX-ANONYM A-DEL em tempo polinomial, então também podemos resolver EXACT 3-COVER em tempo polinomial, o que não é esperado.

Para concluir os resultados de inaproximabilidade, provamos que mesmo fixando o número de arestas que podem ser removidas para uma constante, o problema da aproximação permanece difícil. Ou seja, provamos que DIGRAPH MAX-ANONYM A-DEL não pode ser aproximado em tempo FPT com respeito ao parâmetro s (número de arestas que podem ser removidas).

Como EXACT 3-COVER é FPT com respeito ao tamanho da solução, a redução anterior não garante inaproximabilidade em tempo FPT. Dessa forma, precisamos escolher um problema que seja W[1]-difícil com respeita a algum parâmetro para montarmos a redução. Similarmente à prova para grafos não-direcionados, mostramos uma redução de CLIQUE, que é W[1]-difícil com respeito ao tamanho do clique, para DIGRAPH MAX-ANONYM A-DEL. Como utilizamos INDEPENDENT SET para provar que DIGRAPH DEGREE ANONYMITY A-ADD é NP-difícil, o uso de seu problema complementar é natural ao tratarmos DIGRAPH DEGREE ANONYMITY A-DEL. Adicionalmente, o uso de CLIQUE para uma *gap-reduction* se torna intuitivo pois a ausência de um clique grande o suficiente impede o decremento suficiente do grau de alguns vértices sem a remoção de diversas arestas. Como um único vértice é capaz de deixar um dígrafo apenas 1-anônimo, a redução pode ser construída de tal forma que a presença de um clique no grafo original permite uma forte melhora na anonimidade, enquanto que sua ausência a impede.

A ideia por trás da prova é adicionar um vértice u com grau muito elevado e ter diversos outros vértice com o mesmo, porém muito menor, grau. O dígrafo reduzido é, portanto, construído de tal forma que é necessário encontrar um clique para se reduzir o grau de u para um outro grau existente sem criar novos graus. Como provamos inaproximabilidade em tempo FPT com respeito ao número de arestas removíveis, é importante que o esse valor dependa exclusivamente do tamanho h do clique, pois CLIQUE é W[1]-difícil com respeito a h. Assim, a presença de um clique garante uma anonimização máxima bastante alta, o que significa que o algoritmo de aproximação também deverá melhorar consideravelmente a anonimização. A ausência de um clique impede grandes melhoras, fazendo com que a aproximação também não melhore muito. Essa diferença permite a resolução do problema de decisão com base na resposta do algoritmo de aproximação, implicando que uma aproximação em tempo FPT para DIGRAPH MAX-ANONYM A-DEL nos permite decidir CLIQUE em tempo FPT, o que não é esperado.

Após provarmos a dificuldade computacional do problema, mostramos uma solução eficiente para o problema no caso especial onde o grau máximo Δ é limitado. Começamos mostrando um algoritmo de tempo polinomial para $\Delta = 1$. Nesse caso, existem três tipos de componentes conexos: um vértice isolado, um caminho e um ciclo. Como o número de possibilidades é restrito, adicionar arestas a um certo vértice para ele obter o grau desejado é fácil. Assim, para solucionar o problema utilizamos programação linear inteira. Embora o problema de programação inteira seja NP-difícil, temos uma instância de tamanho constante, dependendo apenas da magnitude dos valores. Dessa forma, podemos solucioná-lo em tempo polinomial no tamanho da entrada. Se existir alguma solução viável para o programa, então é possível resolver a instância correspondente de DIGRAPH DEGREE ANONYMITY A-ADD. Dessa forma, para $\Delta = 1$ é possível decidir DIGRAPH DEGREE ANONYMITY A-ADD em tempo polinomial.

Nós então estendemos os princípios utilizados no caso em que $\Delta = 1$ para $\Delta = 2$. O número de casos diferentes a serem tratados aumenta consideravelmente em relação ao caso anterior, então optamos por uma estratégia mais genérica. Primeiro, obtemos o número mínimo de arcos que devem ser adicionados a vértices de grau d, para todo d possível. Note que para $\Delta = 2$ existem nove valores diferentes para d. Esses números são obtidos, novamente, através de programação inteira. Depois, adicionamos as arestas necessárias aos respectivos vértices, seguindo uma ordem que garante a existência de vértices do grau certo. Esse processo, no

entanto, pode adicionar arestas que conflitam com outras e não correspondem, portanto, a uma solução válida do problema. Para obtermos uma solução válida, eliminamos sistematicamente os conflitos. Durante esse processo, pode ser necessário testar todas as possíveis combinações de arestas. No entanto, garantimos que isso é apenas necessário quando o número de arestas é menor do que uma constante, garantindo que o passo seja realizado em tempo polinomial. No final, ou obtemos uma quantidade mínima de arcos a serem adicionados, ou concluímos que a instância não possui solução. Como Δ é constante, o tempo de execução do algoritmo é polinomial.

Concluímos o trabalho com a conjectura de que é possível utilizar o programa inteiro genérico descrito neste trabalho para resolver Digraph Degree Anonymity A-Add em tempo FPT com respeito ao grau máximo do dígrafo.

Palavras-chave: Grafos. complexidade.

Technische Universität Berlin
Fakultät Elektrotechnik und Informatik
Institut für Softwaretechnik und Theoretische Informatik
Fachgebiet Algorithmik und Komplexitätstheorie

Algorithms and Complexity for Degree Anonymization in Directed Graphs

Author
Marcelo Garlet Millani
Matriculation No: 359602

Supervisors
Prof. Dr. Rolf Niedermeier
André Nichterlein
Robert Bredereck

A thesis submitted in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

March 2015

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den

....
Unterschrift

Zusammenfassung

Die Bedeutung von sozialen Netzwerken für Data Mining ist stark gestiegen, aber deren Anonymisierung ist keine einfache Aufgabe. Diese Arbeit beschäftigt sich mit dem k-Anonymisierungproblem in gerichteten Graphen, wobei ein Digraph k-Anonym ist wenn es für jeder Knoten mindestens $k-1$ anderen Knoten mit gleichen Grad gibt. Dieses Modell nimmt an, dass ein Angreifer, der ein Netzwerk deanonymizieren will, den Grad der Knoten im originalen Digraph weiß. Wir untersuchen die Komplexität des Problems, und beweisen NP-schwere und FPT- sowie Polynomialzeit Inapproximierbarkeit. Weiterhin analysieren wir den Spezialfall mit geringem Maximalgrad und stellen effiziente Algorithmen dafür vor.

Abstract

Social networks had an ever increasing relevance for data mining, yet preserving the anonymity of users was already shown to be no simple task. In this work we consider the k-anonymity problem in directed graphs, where a digraph is k-anonymous if for every vertex there are at least $k-1$ other vertices with the same degree. The assumption of this model is that an attacker attempting to deanonymize the network knows the degrees of the vertices in the original digraph. We analyze the complexity of the problem, proving NP-hardness, polynomial-time inapproximability and FPT-time inapproximability. Finally, we investigate the special case where the degree is bounded and provide an efficient solution for maximum degree one.

Contents

1 Introduction

Obtaining statistical data about a country's population is often quite expensive and complicated: one can make a census, which requires a lot of personnel and time in order to be processed, or analyze a sample of the population, generally omitting potential outliers that might be of interest. In this context, the data contained in social networks can be of great importance: not only can it be quickly obtained when compared to the previous methods, it also contains information that would be otherwise difficult to obtain, like interpersonal relationships or minor interests.

This increased amount of information per person, together with the ease of obtaining such data without the "owner" knowing it, comes at a price however: can it be assumed that the users agree to give their own information to third-parties and, in a sense, make it public? Clearly, one may add some clauses to the terms of agreement stating that the user has to agree with such practices, but, given the increased public awareness and worry about data privacy, people may become reluctant to submit to those terms. It is therefore necessary to find adequate methods to simultaneously provide important information from a database and protect users' privacy.

Social networks can be typically represented as graphs: every user is a vertex and relationships are represented as edges or arcs, depending on their meaning. While undirected graphs model networks where relations are symmetric (e.g., friendship), directed graphs represent asymmetric relations, like a subscription to a blog or one person following the posts of another. One anonymization method that comes naturally for both cases is to simply remove potential unique identifiers from the vertices. This procedure, however, is not always effective [2], as one may infer the identity of a vertex based on previous knowledge of its degree, for example, or through other structural similarities. The unique structural properties of the neighborhood of some node have to be, therefore, concealed by modifying the graph so that they are no longer unique. One simple property to consider is the degree of a vertex. If for every vertex there are at least $k-1$ other vertices with the same degree, then the attacker can only guess the original identity of a vertex with a certainty of $1/k$ and the graph is said to be k-anonymous.

We consider in this work the problem of adding at most s arcs to an input digraph such that the resulting digraph is k-anonymous, where k is also given as input. Since both problems are very similar, we also analyze the variant where arcs are removed instead of added. We prove NP-hardness, polynomial-time as well as FPT-time inapproximability. We also provide an efficient algorithm for cases where the maximum degree is small. Our results are listed in Table 1.1.

Digraph Degree Anonymization	
decision problem	NP-hard
polynomial-time approximation	no $\|V\|^{1-\epsilon}$-approximation
FPT-time approximation	no $\|V\|^{1-\epsilon}$-approximation
$\Delta \leq 2$	polynomial-time solvable

Table 1.1: Summary of our results.

Related Work The concept of k-anonymity was introduced by Samarati and Sweeney [17] for databases and has received increasing interest since then. In particular, the k-anonymity problem has already been researched for tabular data by Bredereck et al. [5, 6, 7].

With respect to undirected graphs, it was shown by Backstrom et al. [2] that simply de-identifying the nodes is not enough to preserve the privacy in a graph. Liu and Terzi [14] formally defined the k-anonymity problem in undirected graphs, considering situations where an attacker that is trying to deanonymize the vertices only knows the degree of each vertex in the original graph. They address the issue of solving the problem without changing the vertex set, but do not analyze whether it is NP-hard or not. The computational complexity of the problem was researched by Hartung et al. [11], and it was proven to be NP-hard. The proof consists of a reduction from Independent Set. They also state that the problem is not only NP-hard, but also W[1]-hard with respect to the maximum number of edge additions.

There are NP-completeness results for the edge deletion variant as well as for vertex deletion due to Bredereck et al. [4]. There are also both polynomial and FPT-time inapproximability results due to Bazgan and Nichterlein [3] for these two problems. The latter result consists of a reduction from the clique problem. On the positive side, they provide an FPT-time algorithm for the combined parameter maximum degree and solution size.

Liu and Terzi [14] also proposed a heuristic for k-anonymity by edge addition, and it was proven by Hartung et al. [12] that this heuristic is optimal when the solution is big enough.

A survey about different topics related to undirected graph anonymization was written by Wu et al. [20]. With respect to directed graphs, López and Sebé [15] consider anonymization for a network of blogs and users where an attacker knows the sorting of the blogs according to their PageRank relevance. To the best of our knowledge, degree anonymity in directed graphs has not been considered thus far.

2 Preliminaries and Definitions

We use standard graph notation. For a more detailed description, there are plenty of books about graph theory, like Modern Graph Theory from Bollobás or Graph Theory by Diestel.

As for computation complexity, more information can be obtained in books like Computational Complexity: A Modern Approach by Arora and Barak, or Computability and Complexity Theory by Homer and Selman.

2.1 Graph Notation

An undirected graph G is a pair (V, E), where V is the set of vertices in G and E the set of edges. Each edge is a set with exactly two vertices from V. The degree of a vertex $v \in V$ is the number of edges that contain v, formally defined as

$$\deg(v) = |\{\{u, v\} : \{u, v\} \in E\}|\,.$$

The maximum degree in a graph is called Δ and the minimum degree, δ.

A directed graph D, or digraph, is a pair (V, A), with V being the set of vertices in D and A the set of arcs. Each arc is an ordered pair of distinct vertices, the first component indicating the origin of the arc and the second, the destination. The indegree of a vertex v of a digraph $D = (V, A)$ denotes the number of arcs with v as the second component and is written as $\deg^-(v)$. The outdegree of v is the number of arcs with v as the first component and is written as $\deg^+(v)$. They are formally defined as:

$$\deg^-(v) = |\{(u, v) : (u, v) \in A\}|\,,$$
$$\deg^+(v) = |\{(v, u) : (v, u) \in A\}|\,.$$

We say that u is a successor of v (and v a predecessor of u) when $(v, u) \in A$. Analogously to graphs, the maximum indegree in a digraph is given by Δ^-, and this notation is extended with the same logic for δ^-, Δ^+ and δ^+. Additionally, we define $\Delta = \max\{\Delta^-, \Delta^+\}$ and $\delta = \min\{\delta^-, \delta^+\}$.

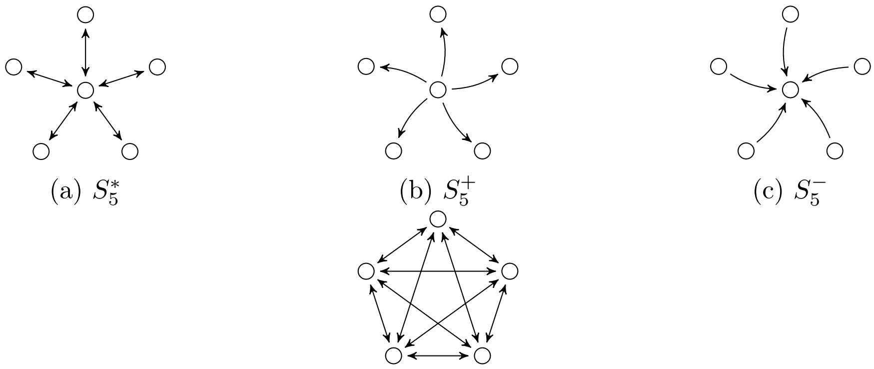
For simplicity reasons, instead of writing "v has indegree 3 and outdegree 4", we write "v has degree $\binom{3}{4}$".

The set of vertices of a graph G or digraph D is denoted by $V(G)$ or $V(D)$. In digraphs, the set of arcs is given by $A(D)$, while the set of edges in graphs is $E(G)$. Additionally, the underlying undirected graph of a digraph is written as $G(D)$, and defined as

$$G(D) = (V(D), \{\{v, u\} : (v, u) \in A(D)\}).$$

(a) S_5 (b) K_5

Figure 2.1: Examples of common graphs.



(a) S_5^* (b) S_5^+ (c) S_5^- (d) K_5^D

Figure 2.2: Examples of common digraphs. To prevent illustrations from getting cluttered with arcs, we use one line where both endpoints have an arrow to denote two arcs $(v, u), (u, v)$.

The equivalent digraph of a graph is called $D(G)$ and defined as

$$D(G) = (V(G), \{(v, u), (u, v) : \{v, u\} \in E(G)\}).$$

Note that there are digraphs D where $D \neq D(G(D))$, but for every graph G it holds that $G = G(D(G))$.

An undirected star S_n is an undirected graph with one central node and n leaves. The central node is adjacent to every leaf, thus having degree n, and each leaf has degree 1. An undirected complete graph K_n is an undirected graph with n nodes where every node is adjacent to every other node. Examples of such graphs can be seen in Figure 2.1.

Digraphs have three variants for S_n: S_n^* is just $D(S_n)$; S_n^+ has all arcs going out of the center, which then has degree $\binom{0}{n}$ (hence $+$); S_n^- has all arcs coming into the center, which has degree $\binom{n}{0}$. A complete digraph K_n^D is simply defined as $D(K_n)$. Examples of such digraphs can be seen in Figure 2.2.

Both graphs and digraphs can be used in set operations. Let G, G' be two graphs, D, D' two digraphs, E' a set of edges between vertices of G, A' a set of arcs between vertices of D, V' a set of vertices, and $\oplus$ any set operator. Then:

$$\begin{aligned}
G \oplus E' &= (V(G), E(G) \oplus E'),\\
G \oplus V' &= (V(G) \oplus V', E(G)),\\
G \oplus G' &= (V(G) \oplus V(G'), E(G) \oplus E(G')),\\
D \oplus A' &= (V(D), A(D) \oplus A'),\\
D \oplus V' &= (V(D) \oplus V', A(A)),\\
D \oplus D' &= (V(D) \oplus V(D'), A(D) \oplus A(D')).
\end{aligned}$$

The complement of a graph $G = (V, E)$, denoted as $\overline{G} = (V, \overline{E})$, is a graph that contains every edge that is not in G and does not contain any edge that is in G. The complemented edge set is thus defined as

$$\overline{E} = \{\{u, v\} : u \in V \wedge v \in V\} - E.$$

The same notation is valid for digraphs, and the complemented arc set is defined as

$$\overline{A} = \{(u, v), (v, u) : u \in V \wedge v \in V\} - A.$$

2.2 Problem Classes

In a decision problem P_d, it is asked whether a certain input word is contained in a certain language. Since any discrete value can be encoded in a word and a language can be any set of words, we can understand a decision problem as asking whether some instance I contains a certain Boolean property (i.e., it either has the property or not). If it does, then we say that $I \in P_d$. As an example, consider the problem of finding an element e in a list L. Effectively, we want to ask if the instance (e, L) has the property $e \in L$. The answer can be either **true** or **false**.

After defining decision problems, a natural question arises: is every decision problem decidable? Or, in other terms, is it always possible to construct an algorithm that guarantees a correct answer in finite time? In 1936, Alan Turing proved that the halting problem cannot be solved by a Turing machine [19]. Note that, according to the widely believed Church's conjecture, a Turing machine is the most powerful computational model physically possible. In practice, this means that, ignoring resource limitations, anything that the fastest supercomputer can compute, so can a common cellphone and a Turing machine.

For those problems that are indeed decidable, we may further ask: how many operations are necessary to solve a certain instance? We could measure the amount of time it takes to run an algorithm, but that depends on the hardware in which it is executed. The abstract operations executed however depend only on the algorithm. Additionally, we are interested in atomic operations that do not require further computations based on the size of the operands. For example, checking if $a \in A$ could be one operation in the algorithm, but it actually requires multiple operations depending on the data structure in which A is stored. So if A were a linked list, computing $a \in A$ would take at most $c \cdot |A|$ operations, for some constant c (for comparing values, proceeding to the next element, etc.). It is irrelevant for the analysis whether a comparison takes 1, 2 or 5 cycles on some hardware. Despite not being measured in seconds or in any time unit, the amount of operations required to solve an instance is called time complexity. Observe that physical time grows with the amount of operations, so the name is adequate. Furthermore, when analyzing how big an instance can be so that it may be solved by a certain algorithm within a practically useful time frame, the presence of constant factors may be ignored. For example, consider the values in Table 2.1. Even on a machine that executes 10^{10} operations per second, it is impractical to solve an input of size 1000 with an algorithm that has a time complexity of 2^n, and it would remain impractical even if the complexity were $2^n \cdot 10^{-6}$. Therefore, it is convenient to apply the big O notation. We say that $f(x) \in O(g(x))$ if there is a constant $c \in \mathbb{Q}$ such that for any $x > x_0 \in \mathbb{Q}$ it holds that $f(x) < cg(x)$. As such, we can observe that $5 \cdot 2^n \in O(2^n)$ and $2^n + n \in O(2^n)$ and treat both functions as simply $O(2^n)$.

Even if a problem is decidable, the time complexity of the best known algorithm can be so high that it would take more time to solved some instances in a modern machine than to reach the heat death of the universe. To identify which problems are known to be tractable, problems in computer science are divided in many classes. With respect to time complexity, the most important classes are P and NP.

A decision problem P_d is said to be in P if and only if there is an algorithm for a deterministic Turing machine that solves any instance of such problem in polynomial time. Similarly, P_d is in NP if and only if it can be solved by a non-deterministic Turing machine in polynomial time.

Let A be a polynomial-time algorithm that converts any instance I of any NP problem to some instance I_R of a problem H. Intuitively, if A exists and solving I_R always allows us to solve I correctly, then H is said to be NP-hard (also when H is not a decision problem or even decidable) and A is called a reduction. It is not hard to see that giving a polynomial-time reduction from a NP-hard problem to some other problem H' is sufficient to prove that H' is in NP-hard. A diagram of how P and NP correlate with NP-hardness and NP-completeness under the assumption that P $\neq$ NPcan be seen in Figure 2.3. A polynomial-time reduction guarantees that, if it is possible to solve any NP-hard problem in polynomial time by a deterministic Turing machine then it is also possible to solve any NP problem in polynomial time or, in other words, P = NP. Whether P equals NP or not is one of the biggest open questions in computer science, but it is strongly believed that P $\neq$ NP[10]. Note that, since this is an open question, there is no known polynomial-time algorithm for any NP-hard problem. This means that the known algorithms for these problems have time complexities like 2^n or $n!$, for example. To better visualize how much worse non-polynomial functions can be, refer to Table 2.1.

While classes like P and NP are only defined for decision problems, there are many cases where simply classifying a given instance as either accepted or rejected is not enough.

n	n^2	n^{10}	2^n	$n!$
2	4	1024	4	2
10	10^2	10^{10}	$> 10^3$	$> 10^6$
1000	10^6	10^{30}	$> 10^{301}$	$> 10^{2567}$

Table 2.1: Growth of different functions. Only the first three columns represent polynomial functions.

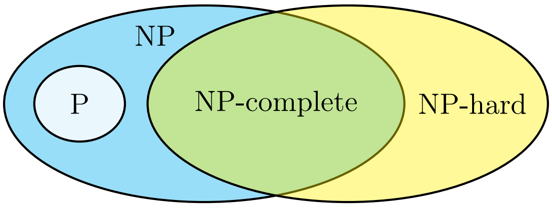


Figure 2.3: Diagram of problem classes, assuming that P $\neq$ NP.

Sometimes we need to maximize or minimize some aspect of an instance, but we do not know before hand any upper or lower bound for the result. Trying to describe such problems as decision problems would be impractical. Therefore, optimization problems have some notational differences from decision problems. In an optimization problem P_o, the objective is to produce an output S such that some specific aspect of the input instance I is either maximized or minimized. We also define $\text{Opt}(I) \in \mathbb{Q}$ as the magnitude of the desired aspect in the optimal solution for the instance I. In maximization problems, this value is called "score" and in minimization problems it is called "cost".

While for simplicity reasons we generally consider the complexity based on the total size of the input as a function of a single variable, the complexity of some algorithms depend on multiple parameters. For example, the NP-hard problem Vertex Cover can be solved in $1.2738^k n^{O(1)}$ time [8], where k is the size of the vertex cover and n is the number of vertices. Therefore, if k is small enough, Vertex Cover can be solved efficiently even when n is huge. We say that Vertex Cover is fixed-parameter tractable with respect to parameter k. The class of problems that allow this type of solution is called FPT [9]. Formally, we say that a problem with instances of the form (k, n) is in FPT with respect to parameter k if there is an algorithm solving it with running time $f(k) \cdot n^{O(1)}$, for some function f. Similar to the P versus NP problem, there are classes W[1], W[2] ... W[t], with

$$FPT \subseteq W[1] \subseteq W[2] \cdots \subseteq W[t].$$

It is strongly believed that this containment is strict, that is, that FPT≠W[1]. While the exact definition of W[t] is beyond the scope of this work, it suffices to know that there is no known FPT-time algorithm for any W[1]-hard problem, and the existence of such algorithm would imply that FPT=W[1], which is unlikely.

Figure 2.4: Example of an initially 1-anonymous digraph and the respective graph. The dashed arc and edge indicate an optimal solution of size 1 for the degree anonymity problem.

2.3 Problem Definition

We extend the definition of Degree Anonymity due to Liu and Terzi [14] to digraphs as follows:

Digraph Degree Anonymity A-Add
Input: A digraph $D = (V, A)$ and two positive integers k and s.
Question: Is there an arc set A' over V with $|A'| \leq s$ such that $D' = (V, A \cup A')$ is k-anonymous, that is, for every vertex $v \in V$ there are at least $k-1$ other vertices in D' with the same in- and outdegree?

Figure 2.4 presents an example instance for Digraph Degree Anonymity A-Add. The input digraph D is 1-anonymous since there is only 1 vertex with degree $\binom{0}{1}$. If we add the dashed arc to it, it becomes 7-anonymous. When we add the dashed edge to G, however, it becomes only 3-anonymous.

It can be sometimes convenient to consider arc deletion instead of addition. Some of the proofs provided later become much easier to understand in one approach than in the other. The problem is defined as:

Digraph Degree Anonymity A-Del
Input: A digraph $D = (V, A)$ and two positive integers k and s.
Question: Is there an arc set A' over V with $|A'| \leq s$ such that $D' = (V, A - A')$ is k-anonymous, that is, for every vertex $v \in V$ there are at least $k-1$ other vertices in D' with the same in- and outdegree?

We show here that both problems are equivalent, that is, adding an arc to a digraph D produces the same digraph as removing that arc from $\overline{D}$ and taking the complement of the result.

Observation 2.1. *For every digraph $D = (V, A)$ and arc-set $S \subseteq V \times V$ it holds that $D \cup S = \overline{\overline{D} - S}$.*

Proof. Let D be a digraph and S some arc-set between the vertices of D. It holds that:

$$\begin{aligned}\overline{\overline{D} - S} &= (V, U - ((U - A) - S) \\ &= (V, U \cap \overline{(U \cap \overline{A} \cap \overline{S})}) \\ &= (V, U \cap (\overline{U} \cup A \cup S)) \\ &= (V, \varnothing \cup A \cup S) \\ &= (V, A \cup S) = D \cup S\end{aligned}$$

□

Using this observation, we can easily reduce DIGRAPH DEGREE ANONYMITY A-ADD and DIGRAPH DEGREE ANONYMITY A-DEL to each other.

Observation 2.2. *For every digraph D and all integers k, s, it holds that (D, k, s) is a yes-instance of* DIGRAPH DEGREE ANONYMITY A-ADD *if and only if $(\overline{D}, k, s)$ is a yes-instance of* DIGRAPH DEGREE ANONYMITY A-DEL.

Proof. We show that D is k-anonymous if and only if $\overline{D}$ is also k-anonymous.

Let $\binom{v_i}{v_o}$ be the degree of some node $v \in V$ in D. Since $\overline{D}$ only contains the arcs that are not in D, the degree of v in $\overline{D}$ is $\binom{|V|-v_i}{|V|-v_o}$. This means that if two vertices $v_1, v_2 \in V$ have the same degree in D, then they also have the same degree in $\overline{D}$. Therefore, the degree anonymity of $\overline{D}$ is equal to that of D.

By Observation 2.1 we conclude that $D \cup S$ is k-anonymous if and only if $\overline{D} - S$ is also k-anonymous, for any arc-set $S \subseteq V \times V$. □

3 Computational Hardness

A trivial brute-force approach to solve Digraph Degree Anonymity A-Add is to test every subset of the set of all possible arcs in a digraph $D = (V, A)$. Since D can have at most $2 \cdot |V|^2$ arcs, this algorithm has to test $2^{|A|} \in O(4^{|V|^2})$ different arc-sets. A natural question that arises from such a complexity is: is there a cleverer approach that solves this problem in polynomial time? As we show in Theorem 3.2, this problem is NP-hard, which means that it is not solvable in polynomial time unless P = NP.

Since there is no hope to find a polynomial-time algorithm to solve Digraph Degree Anonymity A-Add, we will show that it is in NP. Being in NP is helpful because it means that there exists some reduction from this problem to any NP-hard problem. This reductions could potentially be used to take advantage of available tools to well-known problem solvers, like GLPK [16] for integer linear programming, CPLEX [13] for many optimization problems and CryptoMiniSat 4 [18] for Boolean Satisfiability (SAT).

To show that this problem is in NP, it suffices to provide a polynomial-time verification algorithm that, given an instance I of Digraph Degree Anonymity A-Add and a certificate S, returns **true** whenever S is a solution for I and **false** otherwise. One can easily construct a non-deterministic polynomial time solution by generating all possible certificates (in this case, all possible arc sets over V) non-deterministically and running the verification algorithm with each one.

Theorem 3.1. Digraph Degree Anonymity A-Add *is in NP.*

Proof. We provide a polynomial-time verification algorithm for Digraph Degree Anonymity A-Add.

The idea behind this verification algorithm directly follows the problem definition. Given a possible solution $S \subseteq V(D) \times V(D)$ as a certificate, we first need to check if $|S| \leq s$. If this does not hold, then S is clearly not a solution. Otherwise, we need to check if $D \cup S$ is k-anonymous. This can be done by counting the frequency of every degree occurring in $D \cup S$. The certificate S is a solution only if the resulting digraph is k-anonymous.

Checking the size of a set can clearly be done in linear time. Adding S to D takes polynomial time and, depending on the data structures chosen, can be done with $O(|V|^3)$ operations or even faster. To calculate the anonymity of a digraph one has to obtain the degree-frequency, which potentially requires going through the entire structure exactly once, and thus takes $O(|V| \cdot |A|) = O(|V|^3)$ time. The algorithm runs therefore in polynomial time.

□

3.1 NP-hardness

The k-anonymity problem is NP-hard for undirected graphs, so it no surprise that Digraph Degree Anonymity A-Add is also NP-hard. In fact, one can use a very similar reduction as Hartung et al. [11] used, needing only to consider both incoming and outgoing arcs instead of just edges.

To prove that Digraph Degree Anonymity A-Add is NP-hard, we provide a polynomial-time reduction from the NP-complete Independent Set problem, as defined below:

Independent Set
Input: An undirected graph $G = (V, E)$ and a positive integer h.
Question: Is there an independent set $V' \subseteq V$ of size $|V'| = h$, that is, a vertex subset of pairwise nonadjacent vertices?

For the reduction to work, the input graph has to be 3-regular and have at least $2h + 1$ vertices. Therefore, we first need to prove the following lemma, which is rather simple and was omitted from the reference article:

Lemma 3.1. Independent Set *is NP-hard for 3-regular graphs where* $|V| \geq 2h + 1$.

Proof. We give a reduction from Independent Set on 3-regular graphs, which is NP-hard [10]. Let (G, h) be an instance of Independent Set, with $G = (V, E)$ a 3-regular graph. We consider three cases:

Case 1. $|V| = 2h$

We then output a graph G' which consists of a copy of G plus a complete graph K_4, and set $h' = h + 1$, creating the instance (G', h'). Note that G' remains 3-regular, and has $|V| + 4 = 2h' + 2$ vertices. The solution for (G', h') contains at most one vertex from the K_4 added, thus needing at least h vertices from G in order to find an independent set of size $h+1$. A solution for (G, h) is built by removing the vertex taken from the K_4 when solving (G', h').

Case 2. $|V| < 2h$

Then (G, h) is a no-instance. Assume that there is an independent set I, with $|I| = h$. Then $|V - I| < h$ and $V - I$ can "absorb" less than $3h$ edges. However, since G is 3-regular, there are exactly $3h$ edges "leaving" I, which means that some vertex in $V - I$ will have a degree greater than 3. In this case, G is not 3-regular, which contradicts its definition.

The output instance in this case is $(G', h' = 3)$, where G' is composed by two disconnected K_4. Clearly, G' has $8 \geq 2h + 1$ vertices, and is always a no-instance, since it is impossible to choose two vertices from the same K_4, as they will be adjacent by definition of complete graph. The instance (G', h') is, therefore, always a no-instance for Independent Set on 3-regular graphs with $|V| \geq 2h + 1$.

Case 3. $|V| \geq 2h + 1$

Then (G, h) is already an instance of Independent Set on 3-regular graphs with $|V| \geq 2h + 1$.

$\deg^-$ \ $\deg^+$	1	3	$h+2$
1	$k+1$		
3		n	
$h+2$			1

Table 3.1: Number of vertices in D' with the respective in- and outdegrees. For example, there are $k+1$ vertices with degree $\binom{1}{1}$. All zeroes have been omitted.

In all cases there is a polynomial-time reduction from INDEPENDENT SET on 3-regular graphs to INDEPENDENT SET on 3-regular graphs with $|V| \geq 2h+1$, thus proving that the latter is NP-hard. □

Providing a reduction from INDEPENDENT SET on 3-regular graphs to DIGRAPH DEGREE ANONYMITY A-ADD is intuitive since a trivial way to make a digraph k-anonymous is to turn it into a complete digraph, provided that s is big enough. Having all degrees on the input graph to be same simplifies the argumentation, otherwise we would need to consider separate cases for different degrees or add extra steps.

The idea behind the reduction is to add a vertex with a high enough degree so that h vertices must have their degrees largely increased. The reduction is built in such a way that little freedom exists as to which arcs can be added: either all arcs between any two vertices from a subset of $V(D)$ have to be added or there is no solution. It is then only possible to k-anonymize the reduced digraph if there is an independent set in the original graph.

Theorem 3.2. DIGRAPH DEGREE ANONYMITY A-ADD *is NP-complete.*

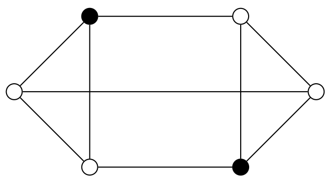
Proof. We give a reduction from INDEPENDENT SET on 3-regular graphs with $|V| \geq 2h+1$, which is NP-hard (Lemma 3.1).

Given an instance (G, h) for INDEPENDENT SET on 3-regular graphs with $|V| \geq 2h+1$, we construct an equivalent instance $(D' = (V', A'), k, s)$ for DIGRAPH DEGREE ANONYMITY A-ADD as follows (recall that $S^*_{h+2} = D(S_{h+2})$ is a star digraph with $h+2$ leaves):

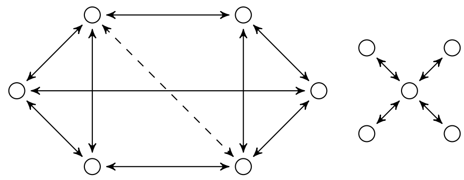
$$D' := D(G) \cup S^*_{h+2},$$
$$s := 2 \cdot \binom{h}{2},$$
$$k := h+1.$$

An example of the reduction can be seen in Figure 3.1, and Table 3.1 shows the distribution of in- and outdegrees of D'.

To prove the correctness of the reduction we show that (G, h) is a yes-instance of INDEPENDENT SET if and only if (D', k, s) is a yes-instance of DIGRAPH DEGREE ANONYMITY A-ADD.



Input graph G. The black nodes represent an independent set of size 2.



Reduced digraph from G with $h = 2$, $k = 3$ and $s = 2$. The dashed edge is one solution for the reduced instance.

Figure 3.1: Example of the reduction from the proof of Theorem 3.2.

Claim 1. If (G, h) is a yes-instance of Independent Set on 3-regular graphs with $|V| \geq 2h + 1$, then (D', k, s) is a yes-instance of Digraph Degree Anonymity A-Add.

Proof. Let $I \in V$ be an Independent Set for G, with $|I| = h$, and $A_I = \{(u, v), (v, u) : u \in I \wedge v \in I\}$ the set with all possible arcs between the vertices of I. Clearly $|A_I| = 2 \cdot \binom{h}{2} = s$. If we add A_I to D', then all vertices $I \in V'$ will have their in- and outdegree increased by $h - 1$ and raised to $h + 2$. Since the center c of the star has degree $\binom{h+2}{h+2}$, after this insertion there are $h + 1 = k$ vertices with degree $\binom{h+2}{h+2}$, all $k + 1$ leaves of the star will remain with degree $\binom{1}{1}$, and the remaining $|V| - |I| \geq k$ vertices will also keep their original degree $\binom{3}{3}$. Therefore, $D' \cup A_I$ is k-anonymous. □

Claim 2. If (G', k, s) is a yes-instance of Digraph Degree Anonymity A-Add, then (G, h) is a yes-instance of Independent Set on 3-regular graphs.

Proof. Let S be a set of arcs such that $D' \cup S$ is k-anonymous. Note that there are $k + 1$ vertices in D' with degree 1 (the leaves of the star), $|V| \geq 2h + 1 = 2k - 1$ of degree 3 and one of degree $h + 2$ (the center of the star). Since we cannot decrease the degree of a vertex, we need to add arcs to $k - 1 = h$ vertices such that their degree is equal to $\binom{h+2}{h+2}$. Because $s = 2 \cdot \binom{h}{2} = h^2 - h$, this is only possible if we add all arcs of a complete digraph with h vertices, which would use all available arcs and, as shown in Claim 3 below, it is not possible to use less than s arcs to increase by $h - 1$ the in- and outdegrees of h vertices. As in a K_h^D every vertex has $h - 1$ neighbors, it is only possible to increase the in- or outdegree of a vertex by $h - 1$, which means all vertices in the complete digraph must already have degree $\binom{3}{3}$, thus eliminating the possibility of adding arcs to the any vertex of the star. □

Claim 3. A complete digraph is required in order to increase the in- and outdegrees of h vertices by $h - 1$ using at most $h^2 - h$ arcs.

Proof. Ideally, each arc will increase the indegree of a vertex by 1 and the outdegree of another also by 1. Since we need to increase $h(h - 1) = h^2 - h$ in- and outdegrees, this cannot be achieved with less than $h^2 - h$ arcs. Moreover, because all arcs connect two vertices that must have their in- and outdegrees increased, this arc set is contained within the complete digraph with these vertices. A complete digraph has twice as many arcs as a complete graph has edges, which means that a K_h^D has $2\binom{h}{2} = h(h-1) = h^2 - h$ arcs. This implies that the solution has all arcs in the complete digraph and no other arcs. □

Observe that the set of vertices $I = \{v : (u, v) \in S\} \in V$ is an independent set of G, since (I, S) forms a complete digraph, $|I| = h$ and $S \cap A = \emptyset$, which means that there are no edges in $E(G)$ between any pair of vertices in I.

From Claims 1 and 2 we conclude that the previously presented reduction is correct and Digraph Degree Anonymity A-Add is NP-hard. Furthermore, from Theorem 3.1 we conclude that Digraph Degree Anonymity A-Add is also NP-complete. □

As Digraph Degree Anonymity A-Add and Digraph Degree Anonymity A-Del are very similar problems, it can be easily shown that the latter is also NP-hard. Since we will need this fact later, we prove it now:

Corollary 3.1. Digraph Degree Anonymity A-Del *is NP-hard.*

Proof. A polynomial-time reduction from Digraph Degree Anonymity A-Add to Digraph Degree Anonymity A-Del was already shown in Observation 2.1. □

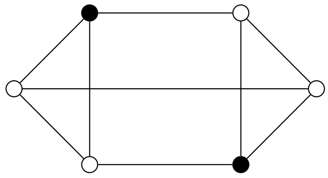
One alternative to deal with Digraph Degree Anonymity A-Add would be to consider $G(D)$ instead of D and use available heuristics, like the one Liu and Terzi [14] proposed. Their heuristic rely on solving the sequence anonymization problem, but the algorithm used is not easily translated to directed graphs. One could apply such a heuristic to $G(D)$ instead, and then "finish" the anonymization process in D. The latter step, however, cannot be computed in polynomial time, unless P = NP, since the problem remains NP-hard, as shown below.

The idea here is very similar to the one provided earlier. We only need to add more star digraphs that have a different degree from the previously added star, but end up with the same degree when converted to a graph.

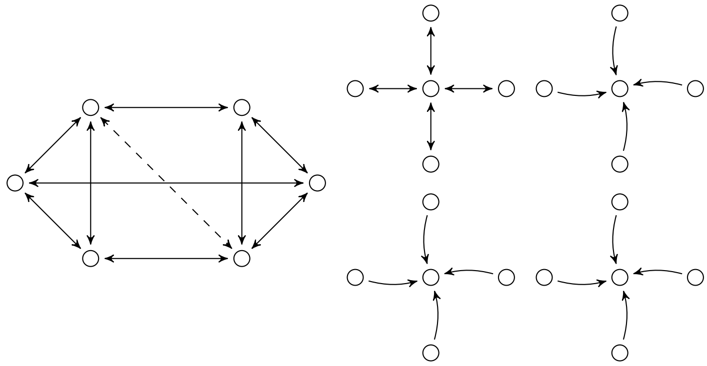
Theorem 3.3. Digraph Degree Anonymity A-Add *remains NP-hard even if the underlying undirected graph of D is already k-anonymous.*

Proof. We give a reduction from Independent Set on 3-regular graphs where $|V| \geq 2h + 1$, similarly to the one present in Theorem 3.2.

Let (G, h) be an instance of Independent Set. We construct an equivalent instance $(D' = (V', A'), k, s)$ following the same steps as before. Additionally, we add $k = h + 1$ stars S^-_{h+2}. Recall that a star S^-_{h+2} has $h+2$ leaves, each with one arc going to the center, and in $S^*_{h+2} = D(S_{h+2})$ all arcs occur in both directions. Altogether, the reduction is as



Input graph G. The nodes colored in black represent an independent set of size 2.



Digraph reduced from G, where $h = 2$, $k = 3$ and $s = 2$. The dashed arcs are one solution for the reduced instance.

Figure 3.2: Example of the reduction presented in Theorem 3.3.

follows (refer to Figure 3.2 for an example):

$$D' = D(G) \cup S^*_{h+2} \cup \bigcup_{i=1}^{k} S^-_{h+2},$$
$$s = 2 \cdot \binom{h}{2},$$
$$k = h + 1.$$

Let $G' = (V', E')$ be an undirected graph with $E' = \{\{u, v\} : (u, v) \in A'\}$. It is clear that G' has $|V| \geq 2h + 1 = 2k - 1 \geq k$ vertices with degree 3, $h + 2 = k + 1$ vertices with degree $h + 2$ (centers of the stars), and $(h + 1)(h + 2) = k(k + 1) \geq k$ vertices with degree 1 (leaves of the stars). It is thus obvious that G' is k-anonymous if $h \geq 0$, which is always the case. Therefore, we can conclude that this reduction always produces a digraph that is already k-anonymous when converted to an undirected graph. The digraph D', however, is not k-anonymous as there is only one degree-$\binom{h+2}{h+2}$ vertex. An overview of the amount of vertices in D' with their in- and outdegrees can be seen in Table 3.2.

In order to k-anonymize D', one has to increase the in- and outdegree of at least k vertices to $h + 2$. Raising the outdegree of the centers of k stars S^-_{h+2} to $h + 2$ would

deg⁻ \ deg⁺	0	1	3	$h+2$
0		$k+1$		
1	$k(k+1)$			
3			n	
$h+2$	k			1

Table 3.2: Number of vertices in D' with the respective in- and outdegrees. All zeroes have been omitted.

require $(h+2)h = h^2 + 2h$ arcs, which is more than s:

$$h^2 + 2h \geq h^2 \geq h^2 - h = s.$$

It is, therefore, impossible to solve the generated Digraph Degree Anonymity A-Add instance by increasing the outdegrees of the centers of the S^-_{h+2}. By an analogous argument, it is also not possible to raise the in- and outdegrees of the leaves of the new stars since their indegrees are all 0. By the argument above, it would require more than s arcs to raise their indegrees from 0 to $h+2$. Using less than k stars is also not possible as there would be less than k vertices with outdegree 0. Since removing arcs is forbidden in this problem, it is impossible to increase the number of vertices with outdegree 0, meaning that the graph would not be k-anonymous.

Since the newly added vertices cannot be part of the solution, using the same argumentation as in Claim 3 from Theorem 3.2, we prove that it is necessary to add all arcs from a complete digraph K_h^D between the vertices in V in order to k-anonymize D'. □

3.2 Polynomial-Time Inapproximability

A natural strategy to deal with NP-hard problems is to accept approximations instead of exact solutions. In the case of Digraph Degree Anonymity A-Add, we could be interested only in increasing the anonymity of the input digraph by adding a limited number of arcs, but the target anonymity would not be precisely given. We would then attempt to maximize the anonymity, but since the problem is NP-hard, finding the optimal solution would be too expensive. We could therefore seek an approximation algorithm that guarantees a solution that is at least Opt/x-anonymous, with Opt being the anonymity of an optimal solution and x some approximation factor, and runs in polynomial time.

Since an approximation for a Boolean value has almost no meaning, it makes more sense to consider an optimization problem instead. We consider here the case where only the maximum size of the arc-set is given. For the arc-deletion case, the definition of the optimization variant is as follows:

DIGRAPH MAXIMUM DEGREE ANONYMITY BY ARC DELETION (DIGRAPH MAX-ANONYM A-DEL)

Input: A digraph $D = (V, A)$ and an integer $s > 0$.

Task: Find an arc-set S with $|S| \leq s$ such that $D - S$ is k-anonymous and k is maximized.

We show in Theorem 3.4 below that a polynomial-time approximation algorithm for DIGRAPH MAX-ANONYM A-DEL with factor $|V|^{1-\epsilon}$ (for $0 < \epsilon \leq 1$) does not exist unless $P = NP$.

To simplify the proof, we will consider arc deletion instead of addition. As shown in Observation 2.2, we can treat a DIGRAPH DEGREE ANONYMITY A-ADD problem as a DIGRAPH DEGREE ANONYMITY A-DEL by using the complementary digraph. Therefore, the inapproximability results obtained for DIGRAPH MAX-ANONYM A-DEL are also valid for the arc-addition variant.

For the proof of Theorem 3.4 we will use the NP-hard EXACT 3-COVER problem, as defined below:

EXACT 3-COVER [10]

Input: A universe $A = \{a_1, a_2, \ldots, a_{3h}\}$, a collection $\mathcal{B} = \{B_1, B_2, \ldots, B_\beta\}$ of 3-element sets over A, and $h \in \mathbb{N}$.

Question: Is there an index set $J \subseteq 1, 2, ..., \beta$ with $|J| = h$, such that $\bigcup\limits_{j \in J} B_j = A$?

To prove an approximation hardness result, it is necessary to provide a reduction that produces a *gap* in the optimum value [1]. The reduction thus allows us to solve the decision problem by checking whether the optimal solution is greater or smaller than a certain known value. To this end we provide a certain interval T such that, for any instance I_O of the optimization problem, it holds that $\text{Opt}(I_O) \notin T$. If there is a polynomial-time factor-q approximation for the optimization problem, with q being the ratio between the highest and lowest values in the interval, then the decision problem can be solved in polynomial time. Hence, using a gap-reduction from an NP-hard problem we can show that a factor-q approximation algorithm that runs in polynomial time exists only if $P = NP$.

A gap-reduction for k-anonymity in undirected graphs was given by Bazgan and Nichterlein [3]. The original paper does not contain all proofs, but a work containing the complete version is to be published. For DIGRAPH DEGREE ANONYMITY A-ADD the reduction is similar, but, in one of the steps, instead of constructing a perfect matching it is possible to simply add a directed cycle. This simplifies the argumentation a little, since a perfect matching is only possible for an even number of vertices. Of course some additional arguments are required in order to consider cases like removing some arc (u, v) without also removing (v, u).

In the reduction from EXACT 3-COVER to DIGRAPH DEGREE ANONYMITY A-DEL the connection between the solution of both problems is ensured by requiring that every removed arc has a vertex corresponding to an element from A as one of its endpoints, and another corresponding to some element in $\mathcal{B}$. The degrees of the vertices in the

$\deg^-$ \ $\deg^+$	2	3	4	7
2	$10h + 11x$			
3		$3h$		
4			x	
7				$\lvert\mathcal{B}\rvert + x$

Table 3.3: Minimum number of vertices in D with the respective in- and outdegrees. All zeroes have been omitted.

reduced instance are chosen such that if it is possible to decrease the in- and outdegree of every vertex associated with A by 1, then the anonymity is considerably improved. If it is not possible, then those vertices will only allow small improvements to the anonymity to be made. By adequately choosing the amount of vertices with each degree, one can construct a gap for the inapproximability result.

Theorem 3.4. *For every* $0 < \epsilon \leq 1$, Digraph Max-Anonym A-Del *is not* $|V|^{1-\epsilon}$ *approximable in polynomial time, even on graphs with maximum in- and outdegree seven, unless* P = NP.

Proof. Let $0 < \epsilon \leq 1$ be a constant. We provide a gap-reduction from Exact 3-Cover, which is NP-complete even when no element occurs in more than three subsets [10]. In this case, it holds that $h \leq \beta \leq 3h$, since, when $h > \beta$, it is not possible to cover all $3h$ elements from A even if we use all $B \in \mathcal{B}$, and if $\beta > 3h$, some $a \in A$ has to appear in more than three $B \in \mathcal{B}$.

Let $(A, \mathcal{B}, h)$ be an instance of Exact 3-Cover where no $a \in A$ occurs in more than three $B \in \mathcal{B}$. We construct an instance (D, s) of Digraph Max-Anonym A-Del as follows (an illustration of the result is provided in Figure 3.3):

1. add to D a vertex v_a for every $a \in A$, and a vertex v_B for every $B \in \mathcal{B}$;
2. add the arcs (v_a, v_B) and (v_B, v_a) for every $a \in A$ and $B \in \mathcal{B}$ where $a \in B$;
3. add four vertices for every v_B, each being its successor and predecessor and having, therefore, in- and outdegree 1;
4. add $3 - \deg^+(v_a)$ vertices for every v_a, each being its successor and predecessor;
5. add $x := \lceil (6h16^{1-\epsilon})^{\frac{1}{\epsilon}} \rceil$ stars S_7^* and x stars S_4^*;
6. create a directed cycle by adding one incoming and one outgoing arc to every vertex with degree $\binom{1}{1}$, increasing their in- and outdegrees by 1;
7. set $s := 6h$.

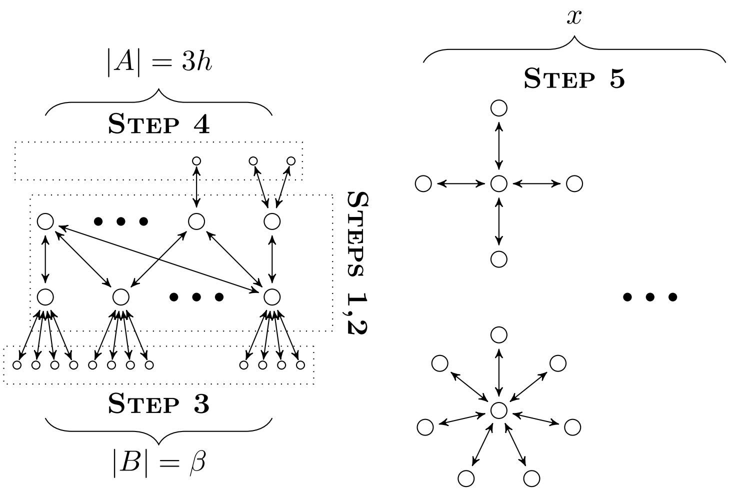


Figure 3.3: Illustration of the reduction procedure from the proof of Theorem 3.4. The cycle produced in Step 6 was omitted.

After that procedure, the number of vertices in D will lie between $14h + 12x$ and $18h + 13x$, being distributed as follows: there are $|A| = 3h$ degree-$\binom{3}{3}$ vertices in D, x vertices with degree $\binom{4}{4}$ and $|B| + x$ vertices with degree $\binom{7}{7}$. The amount of degree-$\binom{2}{2}$ vertices depends on $|\mathcal{B}|$. When $|\mathcal{B}| = 3h$, that amount is maximum and equal to $12h + 11x$. In that case, each $B \in \mathcal{B}$ contains three $a \in A$ and thus every v_a will have degree $\binom{3}{3}$ after Step 2. When $|\mathcal{B}| = h$, the amount of degree-$\binom{2}{2}$ vertices is minimum and equal to $10h + 11x$, since every v_a will have degree $\binom{1}{1}$ after Step 2 and will require 2 additional vertices. This means that the number of degree $\binom{2}{2}$ vertices is between $10h + 11x$ and $12h + 11x$. The degree frequency of D can be observed in Table 3.3. Because $x \geq 6h$, the number of vertices in D is not greater than $16x$. This value will be used later for the analysis of the gap.

We now show that if $(A, \mathcal{B}, h)$ is a yes-instance of Exact 3-Cover, then $\text{Opt}(D, s) \geq x$, and if $(A, \mathcal{B}, h)$ is a no-instance, then $\text{Opt}(D, s) \leq 6h$, where (D, s) is the reduced instance of Digraph Max-Anonym A-Del.

Claim 1. If $(A, \mathcal{B}, h)$ is a yes-instance of Exact 3-Cover, then $\text{Opt}(D, s) \geq x$, where (D, s) is the reduced instance of Digraph Max-Anonym A-Del.

Proof. Let $C \subseteq \mathcal{B}$ be an exact cover for $(A, \mathcal{B}, h)$ of size h, and $S = \{(v_B, v_a), (v_a, v_B) : a \in A \wedge B \in C\}$. Clearly $|S| = 6h$. If we remove S from D, then we reduce the degrees of h vertices v_B from $\binom{7}{7}$ to $\binom{4}{4}$, and all v_a will have their degrees reduced from $\binom{3}{3}$ to $\binom{2}{2}$.

The digraph $D - S$ has no degree-$\binom{3}{3}$ vertices, at least $13h + 11x$ degree-$\binom{2}{2}$ vertices, $x + h$ degree-$\binom{4}{4}$ vertices and x degree-$\binom{7}{7}$ vertices. $D - S$ is therefore x-anonymous, which implies that $\text{Opt}(D, s) \geq x$. □

Claim 2. If $(A, \mathcal{B}, h)$ is a no-instance, then $\text{Opt}(D, s) \leq 6h$, where (D, s) is the reduced instance of Digraph Max-Anonym A-Del.

Proof. We shall prove the contrapositive statement, which is: if $\text{Opt}(D, s) > 6h$, then $(A, \mathcal{B}, h)$ is a yes-instance.

Let S be a set of arcs from D, with $|S| \leq s = 6h$, such that $D - S$ is $(6h+1)$-anonymous.

Since there are no vertices with in- or outdegree less than 2, there can be no arc in S with a degree-$\binom{2}{2}$ vertex as one of its endpoints. This can be shown as follows: assume that there is an arc $(u, v) \in S$ where v is a vertex with degree $\binom{2}{2}$. If v has a degree $\binom{1}{2}$ in $D - S$, then S must have at least $6h$ other arcs with a degree-$\binom{2}{2}$ vertex as one of its endpoints, which means that S has at least $6h + 1 > s$ arcs, which is a contradiction. If v has a smaller in- or outdegree than $\binom{1}{2}$ in $D - S$, then clearly at least $6h + 1$ arcs will still be necessary so that $D - S$ is $(6h + 1)$-anonymous. An analogous argument works for u, which has degree at most $\binom{2}{1}$.

Since there are no degree-$\binom{2}{2}$ vertices as endpoints of the arcs in S, none of the centers of the stars can be an endpoint of one of the arcs of S as the other endpoint would be a vertex of degree $\binom{2}{2}$. Therefore, we have: $S \subseteq \{(v_a, v_B), (v_B, v_a) : a \in A \wedge B \in \mathcal{B}\}$. As D is only $3h$-anonymous because of the $3h$ vertices v_a, it is necessary to increase the number of degree-$\binom{3}{3}$ vertices to $6h + 1$ or to reduce the degree of every degree-$\binom{3}{3}$ vertex to $\binom{2}{2}$. Because every v_B has four degree-$\binom{2}{2}$ adjacent vertices, it is not possible to reduce its degree below $\binom{4}{4}$. Therefore, the only remaining option is to reduce the degree of every v_a to $\binom{2}{2}$, which means it is necessary to remove one incoming and one outgoing arc for every v_a. This will, of course, reduce the in- or outdegree of some v_B. As this v_B cannot have an in- or outdegree in $D - S$ less than four, and one of them will be less than seven, the only viable degree for an affected v_B is $\binom{4}{4}$.

Having any other degree would yield a digraph that is at most $3h$-anonymous, as $|\mathcal{B}| \leq 3h$.

To decrease the degree of every v_a to $\binom{2}{2}$ one has to remove at least $6h$ arcs, meaning that $|S| = 6h = s$. Furthermore, to change the degree of a vertex v_B to $\binom{4}{4}$ it is necessary to remove all of its arcs $(v_B, v_a), (v_a, v_B) \in A(D)$, as every v_B has degree $\binom{7}{7}$ and is adjacent to four degree $\binom{2}{2}$ vertices whose arcs cannot be removed. This implies that, if we remove some arc (v_a, v_B) from D, then we also need to remove the arc (v_B, v_a) and every other arc between this v_B and any v_a. We can express this as

$$\forall (v_a, v_B) \in S((v_B, v_a) \in S \wedge \forall (v_B, v_{a_i}) \in A(D) a_i \in A \Rightarrow (v_B, v_{a_i}) \in S).$$

Hence, one has to choose some vertices v_B which will have all their 6 arcs $(v_B, v_a), (v_a, v_B)$ removed. Since $3h$ vertices v_a must have one incoming and one outgoing arc removed, exactly h vertices v_B have to be chosen, thus removing $6h = |S|$ arcs from D. Note that there is only one v_B for each v_a such that $(v_a, v_B), (v_B, v_a) \in S$. Formally,

$$\forall a \in A \exists! B \in \mathcal{B} (v_a, v_B) \in S.$$

Let $P = \{v, u : (v, u) \in S\}$ be the set of endpoints of S. Since an arc is removed from every v_a, it holds that

$$\bigcup_{B \in \mathcal{B}, v_B \in P} B = A.$$

moreover, because B is unique, we have

$$\forall v_{B_1}, v_{B_2} \in P B_1 \neq B_2 \wedge B_1, B_2 \in B \Rightarrow B_1 \cap B_2 = \varnothing.$$

Therefore, the set $\{B : (v_B, v_a) \in S\}$ is a solution for the instance $(A, \mathcal{B}, h)$ of EXACT 3-COVER. □

From Claims 2 and 1 we build the following gap:

$$\frac{x}{6h} = \frac{x^{\epsilon} x^{1-\epsilon}}{6h} = \frac{6h \cdot 16^{1-\epsilon} \cdot x^{1-\epsilon}}{6h} = (16x)^{1-\epsilon} \geq |V(D)|^{1-\epsilon}$$

Assume that there is an algorithm capable of approximating DIGRAPH MAX-ANONYM A-DEL within a factor $|V|^{1-\epsilon}$ of the optimal solution. Let Approx(D, s) be the achieved anonymity of such algorithm. If Approx(D, s) is smaller than $x/|V|^{1-\epsilon}$, then $(A, \mathcal{B}, h)$ is a no-instance, as Opt(D, s) will have to be smaller than x and, as shown previously, is at most $6h$. If this algorithm finds a solution better than $x/|V|^{1-\epsilon}$, then Opt(D, s) is at least x and $(A, \mathcal{B}, h)$ is a yes-instance, as shown in Claim 2.

Finally, we conclude that, if there is an factor $|V|^{1-\epsilon}$ approximation algorithm for DIGRAPH MAX-ANONYM A-DEL that runs in polynomial time, then EXACT 3-COVER is decidable in polynomial time, and P = NP. □

Note that a trivial algorithm that simply answers $S = \varnothing$ gives a factor-$|V(D)|$ approximation for DIGRAPH MAX-ANONYM A-DEL, as no digraph D can be more than $(|V(D)|)$-anonymous or less than 1-anonymous. While this does not show that the bound of $|V|^{1-\epsilon}$ is tight, it gives very little room for possible approximation algorithms.

3.3 FPT Inapproximability

Since there is no hope to find a polynomial-time approximation for DIGRAPH DEGREE ANONYMITY A-ADD, one may attempt to fix some parameter of the input instance in order to achieve an approximation in FPT time. Here we analyze the complexity of an approximation for a fixed s and prove in Theorem 3.5 below that, with respect to parameter s, there is no practically useful FPT-time approximation algorithm for DIGRAPH MAX-ANONYM A-DEL, unless FPT=W[1].

For the proof, we provide a gap-reduction from the CLIQUE problem, defined below:

CLIQUE [10]
Input: An undirected graph $G = (V, E)$ and a number $h \in \mathbb{N}$.
Question: Is there a subset $V' \subseteq V$ of at least h pairwise adjacent vertices?

Since EXACT 3-COVER is FPT with respect to the parameter solution size h, the reduction provided in Theorem 3.4 does not show FPT inapproximability for DIGRAPH

Max-Anonym A-Del with respect to parameter s, which depended exclusively on h in the given reduction. We need to use, therefore, a problem that is W[1]-hard with respect to some parameter k and provide a reduction where s in the reduced Digraph Max-Anonym A-Del instance depends only on k.

The proof provided by Bazgan and Nichterlein [3] for undirected graphs also consists of a reduction from Clique. Since Independent Set was used to prove that Digraph Degree Anonymity A-Add is NP-hard, it is intuitive to consider its complementary problem for Digraph Degree Anonymity A-Del. Moreover, using Clique in a gap-reduction for Digraph Max-Anonym A-Del is helpful since the absence of a large enough clique prevents sufficiently decreasing the degrees of some vertices without remove many arcs. As the presence of a single outlier is enough to make a graph 1-anonymous, a reduction can be constructed in such a way that if there is a clique in the original graph, then the anonymity can be largely increased. Otherwise, new outliers will have to be created, and the achieved anonymity will be small.

The proof for undirected graphs translates mostly one-to-one to digraphs, needing only some extra arguments since both in- and outdegrees have to be considered. The idea is to add a vertex with a very large degree and having many vertices with the same, but much smaller, degree. The reduced digraph is then built in such a way that one has to find a clique to be able to decrease the degree of the outlier to some other existing degree without creating more outliers. Additionally, since we are proving FPT-inapproximability with respect to parameter s, it is important that s depends only on h as Clique is W[1]-hard with respect to h. And because every complete digraph K_h^D has $2 \cdot \binom{h}{2}$ arcs, it is very natural to set s to that value in this reduction.

Theorem 3.5. *There is no fixed-parameter $|V|^{1-\epsilon}$-approximation algorithm for* Digraph Max-Anonym A-Del *with respect to parameter s, for any $0 < \epsilon \leq 1$, unless FPT=W[1].*

Proof. Let $0 < \epsilon \leq 1$ be a constant. We provide an FPT gap-reduction with gap $|V|^{1-\epsilon}$ from the W[1]-hard Clique problem parameterized by the solution size h [9].

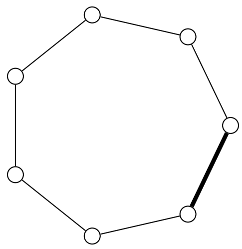
Let $I = (G, h)$ be an instance of Clique. If $\Delta_G + 2h + 1 > |V|$, then we add enough degree $\binom{0}{0}$ vertices to V such that $\Delta_G + 2h + 1 \leq |V|$. This clearly does not affect our solution as the added vertices are not adjacent to any other vertex and, therefore, cannot be part of a clique.

We now construct an instance $I' = (D', s)$ of Digraph Max-Anonym A-Del as follows:

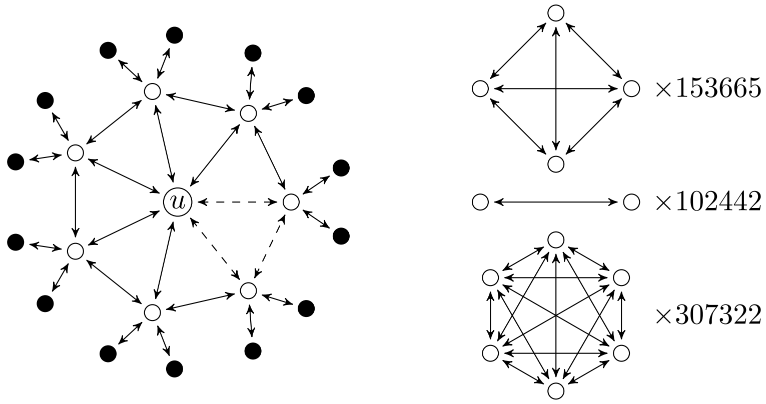
1. start with $D' := D(G)$;

2. add a vertex u to D' and the arcs $\{(u, v), (v, u) : v \in V\}$;

3. for every $v \in V$, add enough degree-$\binom{1}{1}$ vertices w together with the arcs $(v, w), (w, v)$ to D' so that the degree of v becomes $\binom{|V|-h}{|V|-h}$;



Input graph G. The bold edge represents a clique of size 2.



Reduced digraph D' for $\epsilon = 0.75$. In this case, $x = 614656$. The nodes added in Step 3 are colored in black. The dashed arcs represent a solution for the degree anonymization problem.

Figure 3.4: Example for the reduction presented in the proof of Theorem 3.5. The input instance is $(G, 2)$ and the reduced, $(D', 6)$.

4. add enough complete digraphs K_2^D, $K_{|V|-2h+1}^D$ and $K_{|V|-h+1}^D$ to D' so that the amount of vertices in D' with degree $\binom{d}{d}$, for $d \in \{1, |V|-2h, |V|-h\}$, lies between $x+h$ and $x+h+|V|$, with

$$x := \lceil (4 \cdot |V|)^{\frac{3}{\epsilon}} \rceil \geq 64 \cdot |V|^3 ;$$

5. set $s := 2(h + \binom{h}{2}) = h^2 + h$.

Note that Step 3 is always possible since $\Delta_G + 2h + 1 \leq |V|$. This inequality implies that at least $|V| \cdot (h+1)$ and at most $|V| \cdot (|V| - h)$ vertices w will be added.

Before Step 4 is executed, there are at most $|V|^2 - h \cdot |V|$ degree-$\binom{1}{1}$ vertices in D' (only those produced at Step 3). Since this amount is clearly smaller than x, we always need to add some K_2^D to D'. The amount of vertices with degree $\binom{|V|-2h}{|V|-2h}$ is 0, so we have to add $\lceil x + h/|V| - 2h + 1 \rceil$ complete digraphs $K_{|V|-2h+1}^D$ to D'. Finally, only the vertices of V have degree $\binom{|V|-h}{|V|-h}$, meaning that $\lceil x + h - |V|/|V| - h + 1 \rceil$ complete digraphs $K_{|V|-h+1}^D$ will be added to D'. The resulting degree distribution after all steps of the reduction can be seen in Table 3.4. The only vertex with degree $\binom{|V|}{|V|}$ is u.

deg⁻ \ deg⁺	1	$\lvert V\rvert-2h$	$\lvert V\rvert-h$	$\lvert V\rvert$
1	$\geq x+h$			
$\lvert V\rvert-2h$		$\geq x+h$		
$\lvert V\rvert-h$			$\geq x+h$	
$\lvert V\rvert$				1

Table 3.4: Degree frequency in D'. All zeroes have been omitted.

We now show that if I is a yes-instance, then $\text{Opt}(I') \geq x$, and if I is a no-instance, then $\text{Opt}(I') < 2s$.

Claim 1. If I is a yes-instance, then $\text{Opt}(I') \geq x$.

Proof. Let C be the solution for I, that is, a clique of size h in G. If we remove all arcs in D' between the vertices of C, then their degrees will be decreased by $h-1$ and equal to $\binom{|V|-2h+1}{|V|-2h+1}$. This amounts to $2\binom{h}{2}$ removed arcs. Removing all $2h$ arcs between those vertices and u will further decrease their degrees to $\binom{|V|-2h}{|V|-2h}$ and will also decrease the degree of u to $\binom{|V|-h}{|V|-h}$.

Let S be the set of removed arcs. It follows from above that $|S| = 2\binom{h}{2} + 2h = s$, which means that S is a viable solution. Furthermore, there will be no degree-$\binom{|V|}{|V|}$ vertex in $D' - S$. The number of degree-$\binom{|V|-h}{|V|-h}$ vertices will be decreased by $h-1$, which means it will be greater than x as there were at least $x+h$ vertices with this degree before. The amount of degree-$\binom{|V|-2h}{|V|-2h}$ vertices will increase by h and will continue to be greater than x. The number of degree-$\binom{1}{1}$ vertices will remain unchanged. Therefore, $D' - S$ is x-anonymous, which means that $\text{Opt}(I') \geq x$. □

Claim 2. If I is a no-instance, then $\text{Opt}(I') < 2s$.

Proof. We shall prove the contrapositive statement, which is: if $\text{Opt}(I') \geq 2s$, then I is a yes-instance.

Let S be a set of arcs from D' such that $D' - S$ is at least $2s$-anonymous. Note that $D' - S$ only has vertices with degree $\binom{d}{d}$, for $d \in \{1, |V|-2h, |V|-h\}$, since u is the only vertex with degree $\binom{|V|}{|V|}$, which means that it must have a lower degree in $D' - S$. If it does not, the resulting digraph would not be $2s$-anonymous. Changing the degrees of $2s$ vertices to the same value would obviously require removing at least $2s$ arcs, which is not possible.

Reducing the degree of u to $\binom{|V|-h}{|V|-h}$ requires the removal of at least h incoming and h outgoing arcs from u (it will be shown later that further reducing its degree is not possible). This means that at least h vertices from V will have their in- or outdegrees decreased by 1. Let C be the set of such vertices. It follows from the argument above that the degrees of the vertices of C in $D' - S$ have to be either $\binom{1}{1}$ or $\binom{|V|-2h}{|V|-2h}$. For the latter, one has to decrease their in- and outdegrees by at least $h-1$ without removing more than $s-2h = 2 \cdot \binom{h}{2}$ arcs. As shown in Claim 2, this is only possible if C is a complete digraph with h vertices and we remove all of the arcs between any two vertices in C, which will require $2 \cdot \binom{h}{2}$ removals. This will reduce the degrees of the vertices of C to $\binom{|V-2h+1|}{|V-2h+1|}$. If we additionally remove all arcs between any vertex of C and u, all vertices of C will have degree $\binom{|V-2h|}{|V-2h|}$ and u will have degree $\binom{|V|-h}{|V|-h}$. Since this consumes all available arcs, further reducing the degree of u or the degrees of the vertices in C is impossible, as it would require more than s arcs to be removed.

With respect to the degree distribution, h vertices had their degrees reduced from $\binom{|V|-h}{|V|-h}$ to $\binom{|V|-2h}{|V|-2h}$ and one vertex had its degree reduced from $\binom{|V|}{|V|}$ to $\binom{|V|-h}{|V|-h}$. This means that $D' - S$ is at least x-anonymous. Note that $2s < x$:

$$2s = 4h + 4\binom{h}{2} = 2h + 2h^2 \leq 2\,|V|^2 + 2\,|V| \leq 4\,|V|^3 < x$$

Since C is a complete digraph, we conclude that if $\mathrm{Opt}(I') \geq 2s$, then I is a yes-instance. □

From Claims 1 and 2 we conclude that if I is a yes-instance, then $\mathrm{Opt}(I') \geq x$, and if I is a no-instance, then $\mathrm{Opt}(I') < 2s$.

Note that, by construction, the number of vertices in $V(D') = V'$ lies between $3(x+h)+1$ and $3(x+h+|V|)+1$. It is clear from the lower bound that $3x < |V'|$. We now show that $4x \geq |V'|$. It follows that:

$$\begin{aligned} \left|V'\right| &\leq 3(x+h+|V|)+1 \\ &= 3x + 3(h+|V|)+1 \end{aligned} \tag{3.1}$$

Since $x \geq 64 \cdot |V|^3$ and $h \leq |V|$:

$$\begin{aligned} x &\geq 64 \cdot |V|^3 > 64(|V|) > 32(h+|V|) \\ &> 3(h+|V|)+1 \end{aligned} \tag{3.2}$$

for $|V| \geq 1$. From (3.1) and (3.2):

$$\left|V'\right| \leq 3x + 3(h+|V|)+1 \leq 4x$$

With this, we build the following gap:

$$\begin{aligned}\frac{x}{2s} &\geq \frac{|V'|}{8\cdot 2(h+\frac{h(h-1)}{2})} = \left|V'\right|^{1-\epsilon}\frac{|V'|^{\epsilon}}{8(h^2+h)} > \left|V'\right|^{1-\epsilon}\frac{|V'|^{\epsilon}}{16h^2}\\ &\geq \left|V'\right|^{1-\epsilon}\frac{|V'|^{\epsilon}}{16\left|V\right|^2} > \left|V'\right|^{1-\epsilon}\frac{x^{\epsilon}}{16\left|V\right|^2}\\ &\geq \left|V'\right|^{1-\epsilon}\frac{(4\left|V\right|)^{\frac{3}{\epsilon}\cdot\epsilon}}{16\left|V\right|^2} = \left|V'\right|^{1-\epsilon}4\left|V\right| > \left|V'\right|^{1-\epsilon} > \left|V\right|^{1-\epsilon}\end{aligned}$$

We now show that, if there is a factor-$|V(D)|^{1-\epsilon}$ approximation for Digraph Max-Anonym A-Del which runs in FPT time with respect to parameter s, then we can decide Clique in FPT time with respect to parameter h or, in other words, FPT=W[1].

Assume that a factor-$|V(D)|^{1-\epsilon}$ approximation algorithm for Digraph Max-Anonym A-Del exists, for some constant $0 < \epsilon \leq 1$, which runs in FPT time with respect to parameter s. Let (G, h) be an instance of Clique, (D, s) the reduced instance for Digraph Max-Anonym A-Del built with the procedure shown previously and Approx(D, s) the anonymity achieved by the algorithm for the instance (D, s). If Approx$(D, s) \geq x/|V(D)|$ then Opt$(D, s) \geq x$, which means that (G, h) is a yes-instance. If Approx(D, s) is smaller than $x/|V(D)|$ then it is also smaller than $2s$, as shown before, and Opt$(D, s) < 2s$. In this case, (G, h) is a no-instance. Since s depends exclusively on h, it would be possible to decide Clique in FPT time with respect to parameter h, which contradicts the believed assumption that FPT≠W[1]. □

4 Special Case: Bounded Degree

Recall that the maximum in- and outdegree of a digraph are given by Δ^- and Δ^+, respectively. Similarly, δ^- and δ^+ denote the minimum in- and outdegree. Additionally, we define here $\Delta = \max\{\Delta^-, \Delta^+\}$ as the overall maximum in- or outdegree and $\delta = \max\{\delta^-, \delta^+\}$ as the overall minimum in- or outdegree.

We now attempt to ease the problem by restricting the input and solution space. By bounding the degree of a digraph it is possible not only to decrease the maximum number of arcs that can be added, but also how many new arcs a single vertex is allowed to receive. That is, if the input digraph D had a maximum in- or outdegree at most Δ, then $D \cup S$ also has in- or outdegree at most Δ. We then consider the following variation of Digraph Degree Anonymity A-Add:

Digraph Bounded-Degree Anonymity A-Add
Input: A digraph $D = (V, A)$ with maximum in- and outdegree Δ_D and two positive integers k and s.
Question: Is there an arc set A' over V with $|A'| \leq s$ such that $D' = (V, A \cup A')$ is k-anonymous, that is, for every vertex $v \in V$ there are at least $k-1$ other vertices in D' with the same in- and outdegree, and the maximum in- and outdegree of D' is not greater than Δ_D?

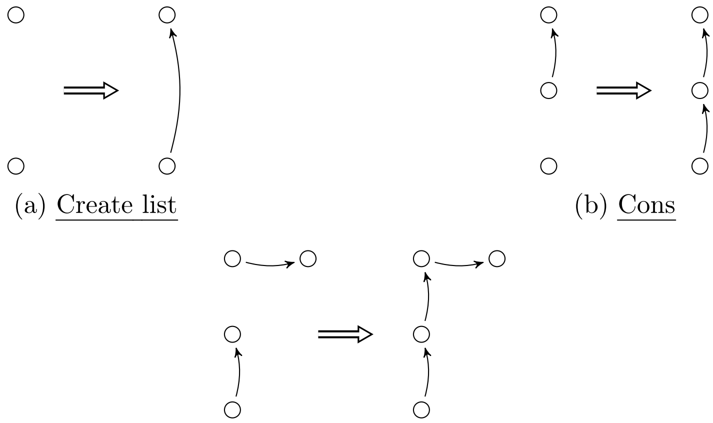
The bounded degree provides an upper bound for s, since adding enough arcs such that the degree of every vertex is $\binom{\Delta}{\Delta}$ requires at most $\Delta \cdot |V(D)|$ arcs, which means that any instance of Digraph Bounded-Degree Anonymity A-Add with $s > \Delta \cdot |V(D)|$ can be reduced to the instance $(D, \Delta \cdot |V(D)|)$.

Theorem 4.1. Digraph Bounded-Degree Anonymity A-Add *can be solved in polynomial time if* $\Delta \leq 1$.

Proof. Let (D, s, k) be an instance of Digraph Bounded-Degree Anonymity A-Add. Since the in- and outdegree of any vertex is at most one, there are three types of connected components: a single vertex with degree $\binom{0}{0}$; a directed cycle with degree-$\binom{1}{1}$ vertices; a directed path with one degree-$\binom{1}{0}$ vertex, one degree-$\binom{0}{1}$ vertex and potentially many degree-$\binom{1}{1}$ vertices. Because a cycle contains a path, we will use the term *list* for the connected component composed by a single path in order to avoid confusions.

Let n_0 be the number of degree-$\binom{0}{0}$ vertices in D, n_l the number of lists and n_1, the number of degree-$\binom{1}{1}$ vertices. Note that the number of vertices with degree $\binom{1}{0}$ or $\binom{0}{1}$ is the same and equal to n_l.

Observe that, when adding an arc to the digraph, there are three possible cases:



(a) Create list (b) Cons (c) Concatenate

Figure 4.1: All possible operations on a digraph with bounded degree $\binom{1}{1}$ that do not destroy this property. Cycles can be created by concatenating a list with itself.

	n_0	n_l	n_1
create list	-2	$+1$	
cons	-1		$+1$
concatenate		-1	$+2$

Table 4.1: Effect of each operation over the vertex frequency.

Case 1. Both endpoints have degree $\binom{0}{0}$.

The arc then creates a new list. This decreases n_0 by 2 and increases n_l by 1.

Case 2. One endpoint has degree $\binom{0}{0}$ and the other, either $\binom{1}{0}$ or $\binom{0}{1}$.

This increases the length of some list by 1, decreases n_0 by 1 an increases n_1 by 1. We use a jargon from functional programming languages and refer to this as cons.

Case 3. One endpoint has degree $\binom{1}{0}$ and the other, $\binom{0}{1}$.

Either two lists are connected or one cycle is created. In both cases, n_l is decreased by 1 and n_1 is increased by 2. This operation is called concatenate.

Note that every operation is always possible, that is, it is guaranteed that, if there are two vertices with the given degrees, then the arc that would be added does not exist. Figure 4.1 illustrates each operation.

We show an algorithm which adds the minimum amount of arcs to D in order to make it k-anonymous.

To decide Digraph Bounded-Degree Anonymity A-Add when $\Delta \leq 1$, we only need to solve the following integer linear programs (ILPs). The logical $\vee$s indicate that one and only one of the inequalities need to be satisfied. To represent this, we can construct $3^2 = 8$ ILPs, each having a different combination of inequalities. In the ILP below, c_l is the amount of create list operations, c_o is the amount of cons and c_c, the amount of concatenates.

$$\begin{aligned}
&\text{minimize } c_l + c_o + c_c \\
&\quad \text{s.t.} \\
&\qquad n_1 + 2c_l + c_c = 0 \;\vee\; n_1 + 2c_l + c_c \geq k \\
&\qquad n_l + c_l - c_o = 0 \;\vee\; n_l + c_l - c_o \geq k \\
&\qquad n_0 - 2c_o - c_c = 0 \;\vee\; n_0 - 2c_o - c_c \geq k \\
&\qquad c_l + c_o + c_c \leq s \\
&\qquad c_l, c_o, c \geq 0 \in \mathbb{N}
\end{aligned}$$

By minimizing $c_l + c_o + c_c$ we also minimize the solution size $|S|$, since each operation corresponds to the addition of one arc. And since it is always possible to execute each operation (i.e., there are no conflicts with existing arcs), it is clear that any solution for the ILP provides a solution for the anonymization problem. An arc-set S such that $D \cup S$ is k-anonymous can then be obtained by executing c_l create list operations, then c_o cons and finally, c_c concatenate. This order is important since it might be necessary to make a cons or concatenate even though $n_l = 0$.

Calculating n_l, n_0 and n_1 can be done in $O(|V(D)|)$ time. The ILPs can be solved in polynomial time since they have a constant amount of variables and inequalities, depending only on the magnitude of k and $V(D)$. Obtaining the solution S takes $O(|V(D)|^2$ time, but this step is not necessary for the decision problem. Therefore, Digraph Bounded-Degree Anonymity A-Add can be decided in polynomial time if the in- and outdegrees are not greater than 1. □

While for $\Delta \leq 1$ there are only three different cases to be considered when adding one arc, for $\Delta \leq 2$ this number increases considerably. We will then use a more generic approach by first obtaining the minimum amount of arcs that must be added to vertices of degree d, for every possible d. To this end, we build an integer linear program which minimizes the number of added arcs.

Let i_d be the total number of incoming arcs added to degree-d vertices, o_d the total number of outgoing arcs added to degree-d vertices, and n_d the initial amount of degree-d vertices in the input digraph D. Additionally, let $M = \{0, 1, \ldots \Delta\}$ and $M^2 = \{\binom{a}{b} : a, b \in M\}$ be the set of all possible degrees.

We want to minimize the sum over all i_d while guaranteeing that, for every degree d, there are either no degree-d vertices or at least k of them. Furthermore, whenever one vertex receives an incoming arc, another must receive an outgoing arc. This means that the sum of incoming arcs has to be the same as the sum of outgoing arcs added. By further observing that the number of degree-$\binom{p}{q}$ decreases whenever we add an incoming or outgoing arc to a degree-$\binom{p}{q}$ vertex, and increases in other two obvious cases, we build the following ILP:

$$\begin{aligned}
&\text{minimize} \sum_{d \in M^2} i_d \\
&\quad \text{s.t.} \\
&\qquad n_d + i_{\binom{p-1}{q}} + o_{\binom{p}{q-1}} - i_d - o_d \geq k \\
&\qquad \vee\ n_d + i_{\binom{p-1}{q}} + o_{\binom{p}{q-1}} - i_d - o_d = 0, && \forall d = \binom{p}{q} \in M^2 \\
&\qquad \sum_{d \in M^2} (i_d - o_d) = 0 \\
&\qquad i_d, o_d \in \mathbb{N} && \forall d \in M^2
\end{aligned}$$

This ILP has $(\Delta + 1)^2$ variables and $2(\Delta + 1)^2 + 1$ inequalities. In total, $2^{(\Delta+1)^2}$ ILPs have to be solved.

Note that it is not possible to increase the amount of nodes with degree $\binom{\delta}{\delta}$. This means that we only need $(\Delta - \delta + 1)^2$ variables and $2(\Delta - \delta + 1)^2 + 1$ inequalities, since no arcs can be added to vertices with in- and outdegree less than δ and their respective variables can be ignored. This could potentially be used to, for example, anonymize a subdigraph where $\Delta - \delta$ is small, but Δ is large.

To solve an instance of Digraph Bounded-Degree Anonymity A-Add for an input digraph D, we first build this ILP and solve it. We then construct a demand arc-collection S, which represents how much the in- and outdegree of each vertex has to increase. This arc-collection is constructed in three phases. First, we construct a set H with all vertices and their original degrees. This set is obtained with

$$H := \{(v, d) : v \in V(D) \wedge d = \begin{pmatrix} \deg^-(v) \\ \deg^+(v) \end{pmatrix}\}$$

Then, we build a collection I with all vertices that will have their indegree increased, and a collection O with vertices that will have their outdegrees increased. A vertex that occurs once in I will have its indegree increased by one, one that occurs twice will have its indegree increased by two, and so on. To build these collections, we need to iterate through all i_d and o_d, choosing degree-d vertices and updating their in- and outdegrees along the way. To guarantee that the needed vertices exist, we need to iterate in the right order. To this end, we will sort the degrees so that $\binom{p}{q}$ precedes $\binom{r}{s}$ (formally written as $\binom{p}{q} \prec \binom{r}{s}$) if and only if $p + q < r + s$. An illustration of how this ordering looks like is presented in Figure 4.2. It is clear that some degree-d vertex may not exist if we do not first obtain all degree-d' vertices, with $d' \prec d$. For example, if we try to add an incoming arc to a degree-$\binom{0}{1}$ vertex before adding all outgoing arcs to degree-$\binom{0}{0}$ vertices, it might happen that no degree-$\binom{0}{1}$ vertex is available at that point. If, however, we iterate in the previously specified order, it is always possible to obtain a demand arc-collection when a solution to the respective ILP exists.

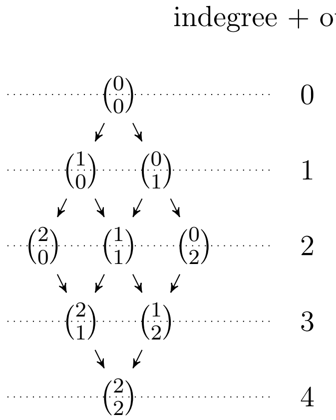


Figure 4.2: Illustration of how the degree of some vertex may evolve as we obtain a demand arc-collection from a solution for the general ILP described in chapter 4. An arrow from some element a to another element b indicates that a degree-a vertex may become a degree-b vertex when an incoming or outgoing arc is added to it. This example is for the case where $\Delta = 2$.

The collections I and O are obtained through the following steps:

1. iterate through every degree d, following the order given by $\prec$;
2. add i_d different vertices v with $(v, d) \in H$ to I;
3. for every added v, replace $(v, d = \binom{p}{q})$ in H with $(v, \binom{p+1}{q})$;
4. add o_d different vertices v with $(v, d) \in H$ to O;
5. for every added v, replace $(v, d = \binom{p}{q})$ in H with $(v, \binom{p}{q+1})$.

Finally, we obtain the demand arc-collection. This is done by arbitrarily choosing and removing one vertex v from I and another vertex u from O, and then adding the arc (v, u) to the demand arc-collection S, until both I and O are empty. However, a solution to the ILP does not guarantee a solution to the respective DIGRAPH BOUNDED-DEGREE ANONYMITY A-ADD instance, since the ILP does not consider, for example, whether an arc between two vertices already exist. To address this issue, we will first define the concept of conflicting arc. A conflicting arc is an arc that cannot be added to a digraph D because it violates some structural restriction. The demand arc-collection may contain three types of conflicting arcs: loops (i.e., arcs of the form (v, v)), preexisting arcs and arcs being added more than once (i.e., they are not present in the original digraph but appear multiple times in S). Examples of conflicting arcs are illustrated in Figure 4.3, together with their respective solutions. Solving these conflicts is not always easy, but when enough arcs are added, it is possible to swap some of the endpoints from the arcs so that all conflicts are solved. For $\Delta \leq 2$, it is always possible to solve the conflicts

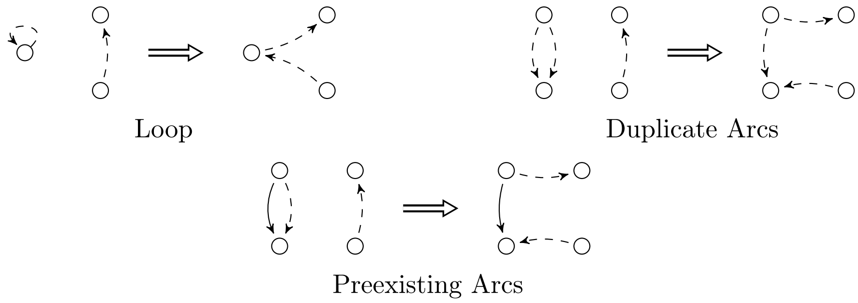


Figure 4.3: Examples of how each conflicting arc can be solved. Arcs from the demand arc-collection S are dashed.

when the demand arc-collection contains at least seven arcs. This is shown in the Lemma below:

Lemma 4.1. *Given a digraph D with maximum in- and outdegree $\Delta \leq 2$, and a demand arc-collection S constructed with the procedure described in chapter 4, with $|S| \geq 7$, it is always possible to find some arc-set S' such that $|S'| = |S|$ and S' contains no conflicting arcs.*

Proof. We describe an algorithm that produces S'. We iteratively reduce one conflict to another until there is only one type of conflict. We then show how to effectively solve this type of conflict, thus obtaining a valid arc-set S'.

First, we remove all loops. Let $(a, a) \in S$ be some loop in S. To remove it without changing the degree of any vertex, we need some other arc $(b, c) \in S$, with $b \neq a \neq c$. Since $\Delta = 2$, there can be at most one other arc in S with a as the first component, and another with a as the second component. This implies that, when $|S| \geq 4$, the arc (b, c) exists in S, for some vertices $b, c \in V(D)$ with $b \neq a \neq c$. We then remove (a, a) and (b, c) from S and add $(a, c), (b, a)$ to it. This does not change the degrees of the vertices nor does it add new loops.

Then, we remove all arcs that appear twice in S. Let (a, b) be some arc that appears twice in S. We need some other arc $(c, d) \in S$ with $a \neq c \neq b$ and $a \neq d \neq b$. At this point, we know that the arcs (a, a) and (b, b) are not present in S. Furthermore, because $\Delta \leq 2$, there are no additional arcs with a as the first component or b as the second component. There might be, however, two more arcs with a as the second component, and two with b as the first component. This means that, if $|S| \geq 7$, there will always be some arc $(c, d) \in S$ with $a \neq c \neq b$ and $a \neq d \neq b$. We then remove (c, d) and one occurrence of (a, b) from S, and add the arcs $(a, d), (c, b)$ to it. This does not add new duplicates, since there can be no other arcs in S with a as the first component or b as the second component.

Finally, we solve all conflicts caused by arcs that are present both in $A(D)$ and in S. Let $(a, b) \in S$ be such an arc. We need some arc $(c, d) \in S$ with $a \neq d$ and $b \neq c$. Since there is already an arc (a, b) in $A(D)$, we know that there are no other arcs in S or $A(D)$ with a as the first component or b as the second. There can be, however, at most two arcs in S with a as the second component, and two with b as the first component. Therefore, if $|S| \geq 6$, it is always possible to find some $(c, d) \in S$ with $a \neq d$ and $b \neq c$. The conflict is then solved by removing $(a, b), (c, d)$ from S and adding $(a, d), (c, b)$. For the reasons given earlier, this will not generate duplicate arcs in S, loops or preexisting arcs.

After these three phases, S contains no conflicting arcs and can now be directly converted into an arc-set S' such that $A(D) \cap S = \varnothing$.

□

Using the Lemma above, we can construct a polynomial-time algorithm for Digraph Bounded-Degree Anonymity A-Add when $\Delta = 2$:

Theorem 4.2. Digraph Bounded-Degree Anonymity A-Add *is polynomial-time solvable when* $\Delta \leq 2$.

Proof. We provide a polynomial-time algorithm for Digraph Bounded-Degree Anonymity A-Add with $\Delta \leq 2$. Let (D, k, s) be an instance of Digraph Bounded-Degree Anonymity A-Add.

First, we construct and solve the general ILP described in chapter 4. This ILP has a constant amount of variables and inequalities, and is solvable in polynomial time. If it has no solution, then (D, k, s) is a no-instance.

Then, we obtain the demand arc-collection S as previously described. If S contains no conflicting arcs, we are done. If S contains conflicting arcs and $|S| \geq 7$, then we solve the conflicts using the algorithm described in Lemma 4.1. If there are conflicts but $|S| < 7$, we try all possible arc-sets of size $|S|$, accepting the first one that does not have conflicts. As the number of different arc-sets is given by the binomial $\binom{a}{|S|}$, with $a = \binom{|V(D)|}{2}$ being the number of possible arcs, testing all of them takes polynomial-time as well. If this is still not possible, we reconstruct the ILP, adding the restriction

$$\sum_{d \in M^2} i_d \geq |S| + 1.$$

and repeat the process from the beginning. This is done at most 6 times, since all conflicts can be solved when $|S| \geq 7$ and the ILP has a solution.

We conclude that Digraph Bounded-Degree Anonymity A-Add is solvable in polynomial-time when $\Delta \leq 2$. □

If a generalization of Lemma 4.1 for any Δ exists, then Digraph Bounded-Degree Anonymity A-Add can be solved in FPT-time with respect to parameter Δ. Since we do not answer this question here, we will leave the following conjecture for future works:

Conjecture 4.1. Digraph Bounded-Degree Anonymity A-Add *can be solved in FPT time with respect to parameter* Δ.

5 Conclusion

In this work, we analyzed the time complexity of Digraph Degree Anonymity A-Add, proving NP-completeness. We also studied the optimization variant Digraph Max-Anonym A-Del, and we showed that, unless P = NP, there is no practically useful polynomial-time approximation for it. With respect to the solution size, we also provided FPT-time inapproximability results. These three results are very similar to the already existing ones for undirected graphs.

On the positive side, we provided a polynomial-time algorithm for the special case where the maximum in- and outdegrees are not greater than 2. We also described an integer linear program for any maximum degree Δ, and postulate that its solution can be used to solve Digraph Bounded-Degree Anonymity A-Add in FPT time with respect to parameter maximum degree with a scheme similar to the one we provided for $\Delta = 1$ and $\Delta = 2$.

The complexity of the optimization variant of Digraph Degree Anonymity A-Add where the solution size is minimized remains unknown, and its study is an important step in the analysis of the feasibility of solving the k-anonymity problem in digraphs. Since inapproximability results exist for this variant in undirected graphs, we suppose that the problem is also inapproximable in digraphs. Moreover, there are some digraph classes that were not investigated in this study where Digraph Degree Anonymity A-Add might become easier, like trees. While adding any arc to a directed tree would destroy its tree property, one can still attempt to anonymize a forest instead. The roots would probably need to be ignored, since they are often easily distinguishable from the remaining vertices. Directed acyclic graphs might be more interesting than trees, since it is possible to add arcs to them without introducing cycles.

Finally, even though the integer linear program provided in chapter 4 provides an FPT-time solution with respect to the maximum amplitude (i.e. $\Delta - \delta$) for the sequence anonymization problem, it would be interesting to investigate whether the heuristic available for undirected graphs due to Liu and Terzi [14] can be used efficiently for digraphs or is NP-hard. Finding practically useful heuristics for Digraph Degree Anonymity A-Add is also a topic that was not considered in this work, but is nonetheless of great importance.

Bibliography

[1] Sanjeev Arora and Carsten Lund. Hardness of approximations. PWS Publishing Company, 1996. 21

[2] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. Commun. ACM, 54(12):133–141, December 2011. 5, 6

[3] Cristina Bazgan and André Nichterlein. Parameterized inapproximability of degree anonymization. In Marek Cygan and Pinar Heggernes, editors, Parameterized and Exact Computation, Lecture Notes in Computer Science, pages 75–84. Springer International Publishing, 2014. URL http://dx.doi.org/10.1007/978-3-319-13524-3_7. 6, 21, 26

[4] Robert Bredereck, Sepp Hartung, André Nichterlein, and GerhardJ. Woeginger. The complexity of finding a large subgraph under anonymity constraints. In Leizhen Cai, Siu-Wing Cheng, and Tak-Wah Lam, editors, Algorithms and Computation, volume 8283 of Lecture Notes in Computer Science, pages 152–162. Springer Berlin Heidelberg, 2013. 6

[5] Robert Bredereck, Thomas Köhler, André Nichterlein, Rolf Niedermeier, and Geevarghese Philip. Using patterns to form homogeneous teams. Algorithmica, 2013. URL http://fpt.akt.tu-berlin.de/publications/homteam.pdf. Online Available. 6

[6] Robert Bredereck, André Nichterlein, and Rolf Niedermeier. Pattern-guided k-anonymity. Algorithms, 6(4):678–701, 2013. URL http://www.mdpi.com/1999-4893/6/4/678. 6

[7] Robert Bredereck, André Nichterlein, Rolf Niedermeier, and Geevarghese Philip. The effect of homogeneity on the computational complexity of combinatorial data anonymization. Data Mining and Knowledge Discovery, 28(1):65–91, 2014. URL http://dx.doi.org/10.1007/s10618-012-0293-7. 6

[8] Jianer Chen, IyadA. Kanj, and Ge Xia. Improved parameterized upper bounds for vertex cover. In Rastislav Královič and Paweł Urzyczyn, editors, Mathematical Foundations of Computer Science 2006, volume 4162 of Lecture Notes in Computer Science, pages 238–249. Springer Berlin Heidelberg, 2006. 11

[9] Rodney G. Downey and Michael R. Fellows. Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, 2013. 11, 26

[10] Michael R. Garey and David S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA, 1990. 10, 15, 21, 22, 25

[11] Sepp Hartung, André Nichterlein, Rolf Niedermeier, and Ondřej Suchý. A refined complexity analysis of degree anonymization in graphs. In FedorV. Fomin, Rūsiņš Freivalds, Marta Kwiatkowska, and David Peleg, editors, Automata, Languages, and Programming, volume 7966 of Lecture Notes in Computer Science, pages 594–606. Springer Berlin Heidelberg, 2013. 6, 15

[12] Sepp Hartung, Clemens Hoffmann, and André Nichterlein. Improved upper and lower bound heuristics for degree anonymization in social networks. In Proceedings of the 13th Symposium on Experimental Algorithms (SEA' 14), volume 8504 of LNCS, pages 376–387. Springer, 2014. 6

[13] IBM. IBM ILOG CPLEX Optimization Studio, 2015. URL http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/. 14

[14] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08, pages 93–106, New York, NY, USA, 2008. ACM. 6, 12, 18, 38

[15] Nacho López and Francesc Sebé. Privacy preserving release of blogosphere data in the presence of search engines. Information Processing & Management, 49(4): 833–851, 2013. 6

[16] Andrew Makhorin. GLPK (GNU Linear Programming Kit), 2012. URL https://www.gnu.org/software/glpk/. 14

[17] Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information. In PODS, volume 98, 1998. 6

[18] Mate Soos. CryptoMiniSat 4, 2014. URL http://www.msoos.org/cryptominisat4/. 14

[19] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, s2-42(1):230–265, 1937. 9

[20] Xintao Wu, Xiaowei Ying, Kun Liu, and Lei Chen. A survey of privacy-preservation of graphs and social networks. In Managing and mining graph data, pages 421–453. Springer, 2010. 6