The Complexity of Degree Anonymization by Vertex Addition

Robert Bredereck*, Vincent Froese**, Sepp Hartung, André Nichterlein, Rolf Niedermeier, and Nimrod Talmon***

Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany
{robert.bredereck,vincent.froese,sepp.hartung,andre.nichterlein, rolf.niedermeier}@tu-berlin.de,nimrodtalmon77@gmail.com

Abstract. Motivated by applications in privacy-preserving data publishing, we study the problem to make an undirected graph k-anonymous by adding few vertices (together with incident edges). That is, after adding these "dummy vertices", for every vertex degree d in the resulting graph, there shall be at least k vertices with degree d. We explore three variants of vertex addition (justified by real-world considerations) and study their (parameterized) computational complexity. We derive mostly (worst-case) intractability results, even for very restricted cases (including trees or bounded-degree graphs) but also obtain a few encouraging fixed-parameter tractability results.

1 Introduction

This work is concerned with making an undirected graph k-*anonymous*, that is, transforming it (at "low cost") into a graph where every vertex degree occurs either zero or at least k times. This graph modification scenario is motivated by data privacy requests in social networks; it focuses on degree-based attacks on identity disclosure of network nodes. Liu and Terzi [12] (also see Clarkson et al. [5] for an extended version) pioneered degree-based identity anonymization in graphs, which recently developed into a very active research field [1, 2, 3, 4, 10, 14, 18] with theoretical as well as practical work. So far, the most common models have relied on edge modifications (allowing either only edge addition or both edge addition and deletion) [2, 5, 10, 14, 12, 18]. We are aware of one theoretical work [1] that considers vertex deletion as modification operation; there mostly computational hardness results have been achieved. Chester et al. [3] started to investigate vertex addition; here we follow this line of research.

There is good reason why vertex addition may be preferred to other graph modification operations when aiming at k-anonymity. The central point here is the "utility" of the anonymized graph. For instance, in the edge addition scenario inserting a new edge destroys distance properties between vertices and indeed may introduce undesirable and misleading "fake relations". Adding new vertices and connecting them to some of the vertices of the original graph could avoid this problem and gives at least a better chance to preserve essential graph properties such as connectivity,

* Supported by the DFG, project PAWS (NI 369/10).
** Supported by the DFG, project DAMM (NI 369/13).
*** Supported by DFG Research Training Group "Methods for Discrete Structures" (GRK 1408).

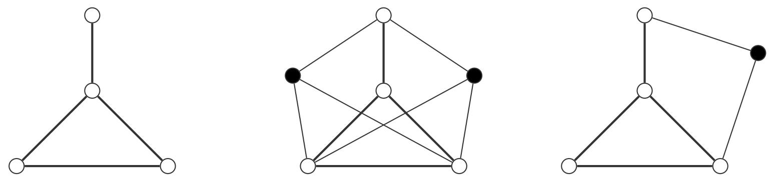

Fig. 1: Example ($k = 2$): The input graph on the left is not yet 2-anonymous. The graph in the middle shows a solution for the vertex cloning variant. The two added vertices (black) are clones of the middle vertex. Note that it is not possible to 2-anonymize the graph by adding only one clone. The graph on the right shows a solution for the general variant where the new vertex can be connected arbitrarily.

shortest paths, or diameter. Chester et al. [3] provide a more thorough discussion of the benefits of vertex addition. The basic decision version of the problem we study is as follows.

Degree Anonymization (va)
Input: A simple undirected graph $G = (V, E)$ and $k, t \in \mathbb{N}$.
Question: Is there a k-anonymous graph $G' = (V \cup V', E \cup E')$ such that $|V'| \leq t$ and $E' \subseteq \{\{u, v\} \subseteq V \cup V' \mid u \in V' \vee v \in V'\}$, where a graph is said to be k-anonymous if and only if every vertex degree in it appears either zero or at least k times ?

It is important to note that Chester et al. [3] studied a slightly different model, with decisive consequences for computational complexity: Their model gets as input a graph $G = (V, E)$, and integers t and k, and also a vertex subset $X \subseteq V$, and the task is to k-anonymize the degree sequence (that is, the vertex degrees sorted in ascending order) of $X \cup V'$ *and* the degree sequence of X. On the contrary, we consider the simpler model where $X = V$, and we require to k-anonymize only the degree sequence of $X \cup V'$ $(= V \cup V')$. Consider the following example highlighting this difference: Let $G = (V, E)$ be an eight-vertex graph containing one star with five leaves plus an edge (that is, $K_{1,5} \cup P_2$). Let $k = 2$ and $X = V$. Since G contains seven vertices of degree 1 and one vertex of degree 5, the solution of Chester et al. [3] will give four as the minimum number of vertices needed to 2-anonymize the degree sequence. In our model, however, this instance can be solved by adding only one new vertex, and connecting it to exactly five old vertices of degree 1 (e.g., transforming the $K_{1,5}$ into a $K_{2,5}$). This happens because the new vertex and the old vertex of degree 5 will form *together* a 2-anonymized "block". However, we believe that our results extend to the model of Chester et al. [3].

Our contributions. Partially answering an open question of Chester et al. [3], we show that Degree Anonymization (va) is weakly NP-hard for a compact encoding of the input. Based on this, we provide several (fixed-parameter) tractability results, exploiting parameterizations by the maximum vertex degree of the input graph, the

Table 1: Overview of our results: Each column represents a different problem variant, where VC (respectively Π, VA) stands for Degree Anonymization (vc) (respectively Π-Preserving Degree Anonymization (va), Degree Anonymization (va)). The first row refers to standard complexity analysis, while the remaining rows show results with respect to several parameters. Here, Δ denotes the maximum degree of the input graph, k is the degree of anonymity, s is the maximum number of added edges, and t is the maximum number of added vertices.

parameter	VC	Π	VA
-	NP-h. [Th. 1]	NP-h. [Th. 3]	weakly NP-h. [Th. 5]
Δ	NP-h., $\Delta = 3$ [Th. 1]	open	open
k	NP-h.[a], $k = 2$ [Th. 2]	NP-h.[a], $k = 2$ [Th. 3]	open
s	open	W[1]-h.[b] [Th. 4]	FPT [Th. 9]
t	W[2]-h. [Th. 2]	W[2]-h. [Th. 3]	XP[c] [Th. 6]
(Δ, k)	open	open	FPT [Th. 8]
(Δ, t)	open	open	FPT [Th. 7]
(t, k)	W[2]-h. [Th. 2]	W[2]-h. [Th. 3]	XP[c] [Th. 6]

[a] Even on trees.
[b] Only for Π = Distances.
[c] Open whether in FPT.

number of added vertices, and the number of (implicitly) added new edges. Moreover, we also study variants of Degree Anonymization (va) where we only allow "cloned" vertices to be added (that is, identical copies of existing vertices with exactly the same neighborhood; this problem variant is denoted by Degree Anonymization (vc)) or where we explicitly demand the preservation of some desirable features such as distance properties (this problem variant is denoted by Π-Preserving Degree Anonymization (va)). For these practically interesting variants we prove computational hardness already for very restricted cases (for instance even on trees). Table 1 surveys most of our results.

Due to the lack of space, most proofs are deferred to a full version.

2 Preliminaries and Problem Definitions

Preliminaries. We consider simple undirected graphs $G = (V, E)$. We denote by $\deg(v)$ the degree of a vertex $v \in V$ and by $\Delta := \max_{v \in V} \deg(v)$ the maximum degree of G. For an integer $0 \leq i \leq \Delta$, we define $B_i := \{v \in V \mid \deg(v) = i\}$, the *block of degree* i. We say that B_i is *empty* (*full*) if $B_i = \emptyset$ ($|B_i| \geq k$). For a full block B_i, we say that it has $z := |B_i - k|$ many *spare* vertices. We call a block B_i *good* if it is empty or full, otherwise we call it *bad* (that is, $0 < |B_i| < k$). The *block sequence* $B(G) := \{(i, |B_i|) \mid B_i \neq \emptyset\}$ of G contains the degrees and sizes of each non-empty block. We call a block sequence *realizable* if it is the block sequence of a graph. For any graph G and for any pair of vertices u and v, we define $\mathrm{dist}_G(u, v)$ to be the length of the shortest path between u and v in G (and $\operatorname{dist}_G(u,v) = \infty$ if there is no path connecting u and v in G). For $n \in \mathbb{N}$, we define $[n] := \{1, 2, \ldots, n\}$.

Problem Definitions. Degree Anonymization (va) allows to add vertices and edges incident to the new vertices. For a given solution of some yes-instance, we denote the actual number of new vertices by t' (obviously, $0 \le t' \le t$) and the total number of newly inserted edges by s.

Π-Preserving Degree Anonymization (va) adds some constraints on the new edges. The idea is to preserve some desirable properties of the input graph. A general definition reads as follows.

Π-Preserving Degree Anonymization (va)
Input: An undirected graph $G = (V, E)$ and $k, t \in \mathbb{N}$.
Question: Is there a k-anonymous graph $G' = (V \cup V', E \cup E')$ such that $|V'| \le t$, $E' \subseteq \{\{u, v\} \subseteq V \cup V' \mid u \in V' \vee v \in V'\}$, and Π is preserved?

We now discuss what "Π is preserved" means for three properties we consider here. First, we say that the *connectedness* remains unchanged if any pair of disconnected vertices in G remains disconnected in G'. As introducing vertices and edges cannot disconnect vertices, this property can be formalized as $\forall\ u, v \in V\colon \operatorname{dist}_G(u,v) = \infty \iff \operatorname{dist}_{G'}(u,v) = \infty$. Second, we say that the *distances* remain unchanged if, for any pair of vertices in G, their distance is the same in G and G', formally, $\forall\ u, v \in V\colon \operatorname{dist}_G(u,v) = \operatorname{dist}_{G'}(u,v)$. Third, we say that the *diameter* remains unchanged if the diameter of G and G' is the same, formally, $\max_{u,v \in V} \operatorname{dist}_G(u,v) = \max_{u,v \in V \cup V'} \operatorname{dist}_{G'}(u,v)$. Note that the diameter property also considers paths between newly added vertices, whereas this is not the case for the first two properties. The reason for this is that the diameter is naturally defined as a single number, whereas the other properties store information for each pair of vertices.

A further constrained variant of Degree Anonymization (va) is to use vertex cloning for modifying the graph. Here, cloning a vertex v means to introduce a new vertex v' and make v' adjacent to all neighbors of v. Formally, we arrive at the following problem:

Degree Anonymization (vc)
Input: An undirected graph $G = (V, E)$ and $k, t \in \mathbb{N}$.
Question: Can G be transformed into a k-anonymous graph by at most t vertex cloning operations?

We remark that there are different cloning variants: Consider two adjacent vertices u and v. If both u and v are cloned, then although the clone u' is adjacent to v and the clone v' is adjacent to u, the clones u' and v' may or may not be adjacent depending on the variant. If the clones are inserted simultaneously at the same time, then u' and v' are not adjacent. If the clones are inserted one after the other, then u' and v' are adjacent (no matter in what order they are inserted). Our results for Degree Anonymization (vc) (Theorems 1 and 2) hold for both variants.

Parameterized Complexity. An instance (I, k) of a parameterized problem consists of the actual instance I and an integer k being the *parameter* [6, 9, 16]. A parameterized problem is called *fixed-parameter tractable* (FPT) if there is an algorithm solving it in $f(k) \cdot |I|^{O(1)}$ time, whereas an algorithm with running time $O(|I|^{f(k)})$ only shows membership in the class XP (clearly, FPT $\subseteq$ XP). One can show that a parameterized problem L is (presumably) not fixed-parameter tractable with a *parameterized reduction* from a W[1]-hard or W[2]-hard problem (such as Clique or Set Cover parameterized by solution size) to L. A parameterized reduction from a parameterized problem L to another parameterized problem L' is a function that, given an instance (I, k), computes in $f(k) \cdot |I|^{O(1)}$ time an instance (I', k') (with $k' \leq g(k)$) such that $(I, k) \in L \Leftrightarrow (I', k') \in L'$.

3 Constrained Degree Anonymization

Cloning seems a natural and well-motivated modification operation for social networks. Unfortunately, we face computational intractability even on very restricted input graphs with maximum degree three. The corresponding reduction is from Independent Set.

Theorem 1. Degree Anonymization (vc) *is NP-hard, even on graphs with maximum degree three.*

Also from the viewpoint of fixed-parameter algorithms, we have no good news with respect to the standard parameter "solution size" t, even on trees. The corresponding reduction is from Set Cover.

Theorem 2. Degree Anonymization (vc) *is NP-hard and W[2]-hard with respect to the number t of clones, even if the degree k of anonymity is two and the graph is a tree.*

We can adjust the reduction from Theorem 2 to also work for Π-Preserving Degree Anonymization (va).

Theorem 3. *For $\Pi \in$ {Distances, Diameter, Connectivity}, Π-*Preserving Degree Anonymization (va) *is NP-hard and also W[2]-hard with respect to the number t of added vertices, even if $k = 2$. For $\Pi \in$ {Distances, Connectivity}, this is also true on trees.*

We strengthen (using a reduction from Clique) parts of Theorem 3 by also showing that the problem remains intractable with respect to the typically larger parameter number s of added edges. For simplicity, we consider s as part of the input.

Theorem 4. *For Π = Distances, Π-*Preserving Degree Anonymization (va) *is W[1]-hard with respect to the number s of new edges.*

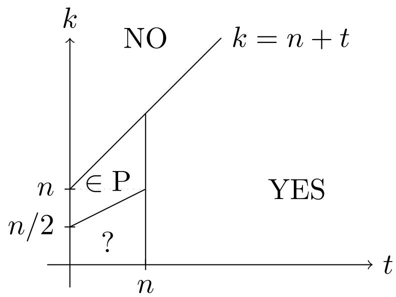

Fig. 2: Visualization of our knowledge about the complexity of Degree Anonymization (va) depending on the values of k and t. The NO-cases follow from Observation 1, the YES-cases are due to Lemma 1, and the polynomial-time solvable cases follow from Lemma 2. For values inside the "?-area", the complexity is open (but for the number version, it is weakly NP-hard, see Theorem 5).

4 Plain Degree Anonymization

In this section, we study the general problem Degree Anonymization (va), without any restrictions on how to connect the new vertices to the input graph. This freedom might raise hope to find solutions more efficiently. Indeed, settling the computational complexity of Degree Anonymization (va) turns out to be tricky in that, on the one hand, we observe that several cases are fairly easy to solve, but we are not aware of any polynomial-time algorithm solving the problem in general. On the other hand, we can only prove weak NP-hardness for a number version of the problem.

In terms of fixed-parameter tractability, however, Degree Anonymization (va) turns out to be more accessible. We obtain some fixed-parameter tractability results regarding, amongst others, certain (combined) parameters (for example, s, (Δ, k), and (Δ, t)), for some of which we proved the cloning and property-preserving problem variants to be W-hard.

Easy Cases. We start analyzing the complexity of Degree Anonymization (va) with respect to the two input values k and t. Figure 2 provides a two-dimensional map indicating those combinations of k and t for which the problem is polynomial-time solvable or even trivial. In the following, we briefly state the corresponding results, starting with the following easy observation:

Observation 1. *Let $I = (G, k, t)$ be an instance of* Degree Anonymization (va) *with G being an n-vertex graph. If $k > n + t$, then I is a no-instance.*

Next, we identify some yes-instances using the fact that—due to a result by Erdős and Kelly [7]—it is always possible to construct a regular graph if we are allowed to add enough, that is, at least n, new vertices.

Lemma 1. *Let $I = (G, k, t)$ be an instance of* Degree Anonymization (va) *with G being an n-vertex graph. If $k \leq n + t$ and $t \geq n$, then I is a yes-instance.*

We finish with some polynomial-time solvable instances which can be solved using f-factors [13, Chapter 10].

Lemma 2. Degree Anonymization (va) *is polynomial-time solvable for $2k > (n + t)$.*

Proof. Let $I = (G = (V, E), k, t)$ be an instance of Degree Anonymization (va) with $2k > (n + t)$. By Observation 1 and Lemma 1, we can assume that $k \leq n + t < 2k$ and $t < n$. Observe that in this case any solution (if existing) transforms G into a regular graph. Hence, the question is whether there is a regular graph H with at most $n + t$ vertices containing G as induced subgraph. We solve this problem by using the polynomial-time solvable f-Factor problem [13, Chapter 10], which is defined as follows:

f-Factor
Input: A graph $G = (V, E)$ and a function $f \colon V \to \mathbb{N}_0$.
Question: Is there an *f-factor*, that is, a subgraph $G' = (V, E')$ of G such that $\deg_{G'}(v) = f(v)$ for all $v \in V$?

Our algorithm is as follows. First, we guess in $O(n^2)$ time the number $t' \leq t$ of vertices that we will add and the degree d of the final regular graph H. Second, we create an f-Factor instance $(G' = (V', E'), f)$ as follows: We define the set $V' := V \cup X$ where X is a set of t' vertices. We start with an edgeless graph G' and then add all edges such that one endpoint is in X, formally, $E' = \{\{u, v\} \mid u \in X, v \in V'\}$. Finally, we set $f(v) := d - \deg_G(v)$ for all $v \in V$ and $f(u) = d$ for all $u \in X$. This completes the f-Factor instance I'. Clearly, I' is a yes-instance if and only if there exists a d-regular graph H with $n + t'$ vertices containing G as an induced subgraph. Hence, our algorithm runs in polynomial time. □

(Weak) NP-Hardness. In Figure 2, we left open the computational complexity of Degree Anonymization (va) for instances with $2k \leq n + t$. We now partially settle this question claiming that an equivalent number version of the problem is weakly NP-hard. To this end, notice that since we are not allowed to add any edges between old vertices, the actual structure of the input graph G becomes negligible and we only need to store the information of how many vertices of which degree it contains (that is, its block sequence $B(G)$):

Observation 2. *Let G and G' be two graphs with identical block sequences, that is, $B(G) = B(G')$. Then, for the* Degree Anonymization (va) *instances $I := (G, k, t)$ and $I' := (G', k, t)$, it holds that I is a yes-instance if and only if I' is a yes-instance.*

Based on Observation 2, we can now define an equivalent number version of Degree Anonymization (va).

Block Sequence Anonymization (va)
Input: A realizable block sequence B and $k, t \in \mathbb{N}$.
Question: Is there a graph G with block sequence B such that (G, k, t) is a yes-instance of Degree Anonymization (va)?

Note that Block Sequence Anonymization (va) is a pure number problem. This helps us to develop a polynomial-time reduction from a weakly NP-hard version of the Subset Sum problem. An NP-hard problem is *weakly* NP-hard if it can be solved in polynomial-time provided that the input is encoded in unary. We conclude with the following theorem:

Theorem 5. Block Sequence Anonymization (va) *is weakly NP-hard.*

Proof (sketch). The reduction is from the weakly NP-hard Change Making problem [15]: Given integers $a_1, \dots, a_n$, m, and b, are there nonnegative integers $x_1, \dots, x_n$ such that $\Sigma_{i \in [n]} x_i \leq m$, and $\Sigma_{i \in [n]} x_i a_i = b$? We can assume, without loss of generality, that $\forall i, j : |a_i - a_j| \geq m^3$. If this property does not hold, then we simply multiply all numbers by m^3, that is, we set $a_i := m^3 \cdot a_i$ and $b := m^3 \cdot b$. It is easy to verify that this new instance is a yes-instance if and only if the original instance is a yes-instance.

We now create an equivalent Block Sequence Anonymization (va) instance (B, k, t), with $t := m$ and $k := t(b + n + 5t + 1)$. The realizable block sequence B is the block sequence of a graph G, which is defined as follows. We introduce several gadgets, that is, subgraphs of G with distinguished vertices of specific degrees which play an important role in the correctness proof. In the following, we only specify the degrees of these *proper* vertices. To realize these gadgets, we add an appropriate number of degree-one neighbors. Our construction ensures that, when k-anonymizing G by adding t vertices, the degree-one vertices will always keep their degree. The construction works as follows.

Add a *b-gadget* consisting of $5t$ *base vertices* of degree $n + t$, b *count vertices* of degree $n + 2t - 1$, and $k - b - 5t$ *b-catch vertices* of degree $n + 2t$. For each $i \in [n]$, add one a_i*-gadget* consisting of one a_i*-vertex* of degree $a_i + n + 4t + 1$ and $k - 1$ a_i*-catch vertices* of degree $a_i + n + 5t + 1$. Finally, add a *dummy gadget* consisting of one *dummy vertex* of degree $n + 4t + 1$ and $k - 1$ *dummy catch vertices* of degree $n + 5t + 1$. This completes the construction. □

Tractability Results. While it remains open whether Degree Anonymization (va) is NP-hard, the weak NP-hardness result for Block Sequence Anonymization (va) (Theorem 5) indicates that also the graph problem may be hard to solve. Hence, a parameterized approach solving Degree Anonymization (va) is reasonable. Notably we provide several (fixed-parameter) tractability results contrasting the hardness results for the constrained problem versions considered in Section 3.

A natural parameter to consider is the solution size t. Unfortunately, we do not know whether Degree Anonymization (va) is fixed-parameter tractable with respect to t; we only know that Degree Anonymization (va) is polynomial-time solvable when t is a constant.

Theorem 6. Degree Anonymization (va) *parameterized by the maximum number t of added vertices is in XP.*

We can, however, "improve" containment in XP with respect to t to fixed-parameter tractability with respect to the combined parameter (t, Δ). Before proving the theorem, we introduce some notation and a helpful lemma.

For a set A of vertices whose addition transforms a graph $G = (V, E)$ into a k-anonymous graph, we call A an *addition set* and we write $G + A$ for the k-anonymous graph. Furthermore, the edges in $G + A$ having at least one endpoint in A (the "added" edges) are denoted by $E(A)$. Hence, $G + A = (V \cup A, E \cup E(A))$.

Clearly, for an addition set A of size t all vertices in $G + A$, except those in A, have degree at most $\Delta + t$ where Δ is the largest degree in G. It may happen that the degree of some (potentially all) vertices from A in $G + A$ is larger than $\Delta + t$. In this case, there are full blocks in $G + A$ of degree larger than $\Delta + t$ consisting only of vertices from A, implying that $t \geq k$. We call blocks corresponding to degrees greater than $\Delta + t$ *large-degree blocks*. Lemma 3 shows that we may assume that there are at most two large-degree blocks which are, in terms of their degree values, not too far away from each other. This will later allow us to guess their degrees.

Lemma 3. *Let (G, k, t) be a yes-instance of* Degree Anonymization (va)*. There is an addition set A of size at most t such that in $G + A$ there are no large-degree blocks, or there is only one large-degree block, or there are only two large-degree blocks whose degrees differ by exactly one.*

Theorem 7. Degree Anonymization (va) *is fixed-parameter tractable with respect to the combined parameter (t, Δ).*

Proof. Our algorithm consists of three phases. First (Phase I), we guess what the solution looks like, specifically guessing the degrees of the good blocks, and the degrees of the new vertices, while respecting the guessed degrees of the good blocks. Then (Phase II), we use a bottom-up lazy method to solve the instance for the old vertices, but with respecting guessed degrees of the new vertices. Finally (Phase III), we use integer linear programming to solve the instance for the new vertices. A detailed description follows.

Phase I: we guess the subgraph induced by the new vertices (in $O(2^{t^2})$ time). We know, from Lemma 3, that the number of possible blocks in the solution is upper-bounded by $\Delta + t + 2 = O(\Delta + t)$. We guess the degrees of the large-degree blocks (in $O(n)$ time). Then, we guess, for each block, whether it is empty or full (in $O(2^{\Delta+t})$). Finally, we guess the degree of each new vertex (in $(\Delta + t)^t$ time). Phase I runs in $n \cdot O\left(2^{t^2} \cdot 2^{\Delta+t} \cdot (\Delta + t)^t\right) = n \cdot f_1(t, \Delta)$ time.

For ease of presentation, we say that we *move* a vertex up, meaning that we connect it to some new vertices, thus changing its degree and moving it to a different block of some desired degree. We can choose which new vertices to use in a round-robin way, but considering their guessed degrees (that is, each new vertex participates in the round-robin until it reaches its guessed degree).

Phase II: we perform the following bottom-up lazy method. We start from the lowest degree block, and work all the way up to the highest degree block. If the current block B_i is guessed to become empty, then we move its vertices up, to the first block above it which is guessed to become full (if there is a gap greater than t to such a block, we halt with a negative answer). Otherwise, if it is guessed to become full, then we distinguish between the following two cases: if the number of old vertices in the block plus the number of new vertices guessed to be in this block is at least k, then we do nothing, because it means that this block is already anonymized with respect to the old vertices, and continue to the next block. Otherwise, B_i has a shortage of some z_i many vertices to become full, so we find the maximum $j < i$ such that the number of old vertices in B_j plus the number of new vertices guessed to be in B_j is greater than k (specifically, equals to $k + z_j$ for some z_j spare vertices in B_j; if the gap $i - j$ is greater than t, then we halt with a negative answer, because B_i cannot be k-anonymized). We move $\min(z_i, z_j)$ spare vertices from B_j to B_i. If, after moving these spare vertices, B_i still needs some more vertices (that is, if $z_i > z_j$), then we repeat this step once more, looking for the maximum $j' < j$ such that the number of old vertices in $B_{j'}$ plus the number of new vertices guessed to be in $B_{j'}$ is greater than k, until we have enough vertices in the current block. If in the end of this phase, all of the blocks are anonymized, we continue to the next phase. The overall cost of Phase II is $O(\Delta + t)^3 = f_2(t, \Delta)$.

Our approach is lazy for two reasons. The first reason is that we use the spare vertices from the *closest* full block below the current one. The second reason is that we move the minimum number of vertices to make the blocks anonymized with respect to the old vertices, that is, we only change the bad blocks to become full, but not overfull.

Phase III: We check if we reached the exact guessed total number s of edges added. If so, then we halt with a positive answer, as this means that the new vertices reached their guessed degrees. If we reached a larger number, then we halt with a negative answer, since Phase II is lazy, it means that we cannot k-anonymize the graph using the guessed number of edges added. If we reached a smaller number, then we still have some hope of reaching s, because of the laziness of Phase II, so we try to move some more vertices, until we reach the guessed total number of edges added, while not destroying the anonymity of the blocks. To this end, denote the number of spare vertices in each full block B_i by z_i. Notice that we can move any number of up to z_i vertices from this block, to any full block above it, and no other moves are possible. Now our problem reduces to the following integer linear program:

Input: n' numbers $\{z'_1, \ldots, z'_{n'}\}$, $n' \times m'$ matrix $A = a_{ij}$, and integer Z.
Task: Maximize $\sum_{i \in [n']} \sum_{j \in [m']} a_{ij} x_{ij}$ such that $\sum_{i \in [n']} \sum_{j \in [m']} a_{ij} x_{ij} \leq Z$ and $\forall j : \sum_{i \in [n']} a_{ij} \leq z_j$.

Specifically, we set n' and m' to be the number of full blocks. For each full block, we set z'_i to be z_i and $a_{i,j}$ to be the gap between the jth full block and the ith full block. Fortunately, the number of variables is upper-bounded by the number of full blocks squared, (therefore, upper-bounded by $O((\Delta + t)^2)$). By a famous result of Lenstra [11], it follows that the running time is exponential only in the number of variables, therefore the cost of this phase is $\text{poly}(n) \cdot f_3(t, \Delta)$.

We now prove the correctness of the algorithm. As the algorithm only performs permitted operations (that is, adds up to t new vertices and connects up to s edges, each incident to at least one new vertex), it follows that if the input is a no-instance, then the algorithm returns a negative answer. Otherwise, if the input is a yes-instance, then at least one set of guesses from Phase I will be correct. Any solution must at least move the vertices that are moved in Phase II, and then the problem reduces to the ILP presented in Phase III. □

The question whether fixed-parameter tractability also holds for the parameter t or Δ alone remains open. Nevertheless, we find that fixed-parameter tractability also holds for the combined parameter (Δ, k).

Theorem 8. Degree Anonymization (va) *is fixed-parameter tractable with respect to the combined parameter* (Δ, k)*.*

Contrasting the W[1]-hardness of Π-Preserving Degree Anonymization (va) parameterized by the number s of new edges (Theorem 4), we conclude with fixed-parameter tractability for Degree Anonymization (va) with respect to s. We again assume that s is given as part of the input.

Theorem 9. Degree Anonymization (va) *is fixed-parameter tractable with respect to the number s of newly inserted edges.*

5 Conclusion

Table 1 in the introductory section overviews most of our results and leaves several specific open questions. Moreover, it is fair to say that our positive algorithmic results are basically of classification nature and require further improvement for practical relevance. Indeed, a more holistic approach in terms of a full-fledged multivariate complexity analysis [8, 17], perhaps also driven by the analysis of real-world network data characteristics, may help to derive practically useful algorithmic results. A deeper investigation of approximation algorithms (cf. [3, 4]) may be beneficial as well. Finally, typical social network properties such as measured by the clustering coefficient or the average path length are studied in experimental work [3], but the complexity of Π-Preserving Degree Anonymization (va) with respect to these properties is unexplored so far.

Bibliography

[1] R. Bredereck, S. Hartung, A. Nichterlein, and G. J. Woeginger. The complexity of finding a large subgraph under anonymity constraints. In *Proc. 24th ISAAC*, volume 8283 of *LNCS*, pages 152–162. Springer, 2013.

[2] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra. An algorithm for k-degree anonymity on large networks. In *Proc. ASONAM'13*, pages 671–675. ACM Press, 2013.

[3] S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, and S. Venkatesh. Why Waldo befriended the dummy? k-anonymization of social networks with pseudo-nodes. *Social Netw. Analys. Mining*, 3(3):381–399, 2013.

[4] S. Chester, B. M. Kapron, G. Srivastava, and S. Venkatesh. Complexity of social network anonymization. *Social Netw. Analys. Mining*, 3(2):151–166, 2013.

[5] K. L. Clarkson, K. Liu, and E. Terzi. Towards identity anonymization in social networks. In *Link Mining: Models, Algorithms, and Applications*, pages 359–385. Springer, 2010.

[6] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

[7] P. Erdős and P. Kelly. The minimal regular graph containing a given graph. *Amer. Math. Monthly*, 70:1074–1075, 1963.

[8] M. R. Fellows, B. M. P. Jansen, and F. A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *Eur. J. Combin.*, 34(3):541–566, 2013.

[9] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[10] S. Hartung, A. Nichterlein, R. Niedermeier, and O. Suchý. A refined complexity analysis of degree anonymization in graphs. In *Proc. 40th ICALP*, volume 7966 of *LNCS*, pages 594–606. Springer, 2013. to appear in *Information and Computation*.

[11] H. W. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8:538–548, 1983.

[12] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *ACM SIGMOD Conference*, SIGMOD '08, pages 93–106. ACM, 2008.

[13] L. Lovász and M. D. Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. North-Holland, 1986.

[14] X. Lu, Y. Song, and S. Bressan. Fast identity anonymization on graphs. In *Proc. 23rd DEXA*, volume 7446 of *LNCS*, pages 281–295. Springer, 2012.

[15] G. S. Lueker. Two NP-complete problems in nonnegative integer programming. Technical report, Computer Science Laboratory, Princeton University, 1975.

[16] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[17] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proc. 27th STACS*, volume 5 of *LIPIcs*, pages 17–32. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010.

[18] B. Zhou and J. Pei. The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowl. Inf. Syst.*, 28(1):47–77, 2011.