Graph Editing to a Given Degree Sequence[☆,☆☆]

Petr A. Golovach[a], George B. Mertzios[b,*]

[a]*Department of Informatics, University of Bergen, N-5020 Bergen, Norway*
[b]*School of Engineering and Computing Sciences, Durham University, UK*

Abstract

We investigate the parameterized complexity of the graph editing problem called Editing to a Graph with a Given Degree Sequence where the aim is to obtain a graph with a given degree sequence σ by at most k vertex deletions, edge deletions and edge additions. We show that the problem is W[1]-hard when parameterized by k for any combination of the allowed editing operations. From the positive side, we show that the problem can be solved in time $2^{O(k(\Delta^*+k)^2)}n^2 \log n$ for n-vertex graphs, where $\Delta^* = \max \sigma$, i.e., the problem is FPT when parameterized by $k+\Delta^*$. We also show that Editing to a Graph with a Given Degree Sequence has a polynomial kernel when parameterized by $k+\Delta^*$ if only edge additions are allowed, and there is no polynomial kernel unless NP $\subseteq$ co-NP /poly for all other combinations of the allowed editing operations.

Keywords: Parameterized complexity, graph editing, degree sequence

1. Introduction

The aim of graph editing (or graph modification) problems is to modify a given graph by applying a bounded number of permitted operations in order to satisfy a certain property. Typically, vertex deletions, edge deletions and edge additions are the considered as the permitted editing operations, but in some cases other operations like edge contractions and vertex additions are also permitted.

We are interested in graph editing problems where the aim is to obtain a graph satisfying some given degree constraints. These problems usually turn

[☆]A preliminary version of this paper appeared as an extended abstract in the proceedings of CSR 2016.

[☆☆]The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement n. 267959, the Government of the Russian Federation grant 14.Z50.31.0030 and by the EPSRC Grant EP/K022660/1.

[*]Corresponding author

Email addresses: petr.golovach@uib.no (Petr A. Golovach), george.mertzios@durham.ac.uk (George B. Mertzios)

out to be NP-hard (with rare exceptions). Hence, we are interested in the parameterized complexity of such problems. Before we state our results we briefly discuss the known related (parameterized) complexity results.

Related work. The investigation of the parameterized complexity of the editing problems with degree constraints was initiated by Moser and Thilikos in [24] and Mathieson and Szeider [23]. In particular, Mathieson and Szeider [23] considered the Degree Constraint Editing problem that asks for a given graph G, nonnegative integers d and k, and a function $\delta\colon V(G) \to 2^{\{0,\ldots,d\}}$, whether G can be modified into a graph G' such that the degree $d_{G'}(v) \in \delta(v)$ for each $v \in V(G')$, by using at most k editing operations. They classified the complexity of the problem depending on the set of allowed editing operations. In particular, they proved that if only edge deletions and additions are permitted, then the problem can be solved in polynomial time for the case where the set of feasible degrees $|\delta(v)| = 1$ for $v \in V(G)$. Without this restriction on the size of the sets of feasible degrees, the problem is NP-hard even on subcubic planar graphs whenever only edge deletions are allowed [10] and whenever only edge additions are allowed [16]. If vertex deletions can be used, then the problem becomes NP-complete and W[1]-hard with parameter k, even if the sets of feasible degrees have size one [23]. Mathieson and Szeider [23] showed that Degree Constraint Editing is FPT when parameterized by $d + k$. They also proved that the problem has a polynomial kernel in the case where only vertex and edge deletions are allowed and the sets of feasible degrees have size one. Further kernelization results were obtained by Froese, Nichterlein and Niedermeier [16]. In particular, they proved that the problem with the parameter d admits a polynomial kernel if only edge additions are permitted. They also complemented these results by showing that there is no polynomial kernel unless NP $\subseteq$ co-NP /poly if only vertex and edge deletions are allowed. Golovach proved in [19] that, unless NP $\subseteq$ co-NP /poly, the problem does not admit a polynomial kernel when parameterized by $d + k$ if vertex deletion and edge addition are in the list of operations, even if the sets of feasible degrees have size one. The case where the input graph is planar was considered by Dabrowski et al. in [14]. Golovach [18] introduced a variant of Degree Constraint Editing in which, besides the degree restrictions, it is required that the graph obtained by editing should be connected. This variant for planar input graphs was also considered in [14].

Froese, Nichterlein and Niedermeier [16] also considered the Π-Degree Sequence Completion problem which, given a graph G, a nonnegative integer k, and a property Π of graph degree sequences, asks whether it is possible to obtain a graph G' from G by adding at most k edges such that the degree sequence of G' satisfies Π. They stated some sufficient conditions for Π such that the problem is FPT and, in some cases, admits a polynomial kernel when parameterized by k and the maximum degree of G if these conditions are fulfilled. There are numerous results (see, e.g., [4, 9, 12, 13]) about the graph editing problem where the aim is to obtain a (connected) graph whose vertices satisfy some parity restrictions on their degree. In particular, if the obtained graph is required to be a connected graph with vertices of even degree, we obtain the classical Editing to Eulerian Graph problem (see. [4, 13]).

Another variant of graph editing with degree restrictions is the Degree Anonymization problem introduced by Liu and Terzi [22] motivated by some privacy and social networks applications. A graph G is *h-anonymous* for a positive integer h if for any $v \in V(G)$, there are at least $h-1$ other vertices of the same degree. Degree Anonymization asks, given a graph G, a nonnegative h, and a positive integer k, whether it is possible to obtain an h-anonymous graph by at most k editing operations. The investigation of the parameterized complexity of Degree Anonymization was initiated by Hartung et al. [20] and Bredereck et al. [6] (see also [5, 21]). In particular, Hartung et al. [20] considered the case where only edge additions are allowed. They proved that the problem is W[1]-hard when parameterized by k, but it becomes FPT and has a polynomial kernel when parameterized by the maximum degree Δ of the input graph. Bredereck et al. [6] considered vertex deletions. They proved that the problem is W[1]-hard when parameterized by $h + k$, but it is FPT when parameterized by $\Delta + h$ or by $\Delta + k$. Also the problem was investigated for the cases when vertex additions [5] and edge contractions [21] are the editing operations.

Our results. Recall that the *degree sequence* of a graph is the nonincreasing sequence of its vertex degrees. We introduce the graph editing problem where the aim is to obtain a graph with a given *degree sequence* by using the operations *vertex deletion*, *edge deletion*, and *edge addition*, denoted by *vd*, *ed*, and *ea*, respectively. Formally, the problem is stated as follows. Let $S \subseteq \{vd, ed, ea\}$.

Editing to a Graph with a Given Degree Sequence	
Instance:	A graph G, a nonincreasing sequence of nonnegative integers σ and a nonnegative integer k.
Question:	Is it possible to obtain a graph G' with the degree sequence σ from G by at most k operations from S?

Notice that we can assume that the length of σ is at most $|V(G)|$ and it is exactly $|V(G)|$ if $vd \notin S$ as, otherwise, we have a trivial no-answer. Also if $vd \in S$, then the number of vertex deletions is implicitly defined by the length of σ and is $|V(G)| - |\sigma|$.

It is worth highlighting here the difference between this problem and the Editing to a Graph of Given Degrees problem studied in [16, 19, 23]. In Editing to a Graph of Given Degrees, a function $\delta : V(G) \to \{1, \ldots, d\}$ is given along with the input and, in the target graph G', every vertex v is required to have the *specific* degree $\delta(v)$. In contrast, in the Editing to a Graph with a Given Degree Sequence, only a degree sequence is given with the input and the requirement is that the target graph G' has this degree sequence, without specifying which specific vertex has which specific degree. To some extend, this problem can be seen as a generalization of the Degree Anonymization problem [5, 6, 20, 21], as one can specify (as a special case) the target degree sequence in such a way that every degree appears at least h times in it.

In practical applications with respect to privacy and social networks, we might want to appropriately "smoothen" the degree sequence of a given graph in such a way that it becomes difficult to distinguish between two vertices with (initially) similar degrees. In such a setting, it does not seem very natural to specify in advance a *specific* desired degree to every *specific* vertex of the target graph. Furthermore, for anonymization purposes in the case of a social network, where the degree distribution often follows a so-called power law distribution [2], it seems more natural to identify a smaller number of vertices having all the same "high" degree, and a greater number of vertices having all the same "small" degree, in contrast to the more modest h-anonymization requirement where *every* different degree must be shared among at least h identified vertices in the target graph.

In Section 2, we observe that for any nonempty $S \subseteq \{vd, ed, ea\}$, Editing to a Graph with a Given Degree Sequence is NP-complete and W[1]-hard when parameterized by k. Therefore, we consider a stronger parameterization by $k + \Delta^*$, where $\Delta^* = \max \sigma$. In Section 3, we show that Editing to a Graph with a Given Degree Sequence is FPT when parameterized by $k + \Delta^*$. In fact, we obtain this result for the more general variant of the problem, where we ask whether we can obtain a graph G' with the degree sequence σ from the input graph G by at most k_{vd} vertex deletions, k_{ed} edge deletions and k_{ea} edge additions. We show that the problem can be solved in time $2^{O(k(\Delta^*+k)^2)} n^2 \log n$ for n-vertex graphs, where $k = k_{vd} + k_{ed} + k_{ea}$. The algorithm uses the random separation techniques introduced by Cai, Chan and Chan [8] (see also [1]). First, we construct a true biased Monte Carlo algorithm, that is, a randomized algorithm whose running time is deterministic and that always returns a correct answer when it returns a yes-answer but can return a false negative answer with a certain (small) probability. Then we explain how it can be derandomized. In Section 4, we show that Editing to a Graph with a Given Degree Sequence has a polynomial kernel when parameterized by $k + \Delta^*$ if $S = \{ea\}$, but for all other nonempty $S \subseteq \{vd, ed, ea\}$, there is no polynomial kernel unless NP $\subseteq$ co-NP /poly. Finally in Section 5 we conclude the paper and discuss future research directions.

2. Basic definitions and preliminaries

Graphs. We consider only finite undirected graphs without loops or multiple edges. The vertex set of a graph G is denoted by $V(G)$ and the edge set is denoted by $E(G)$.

For a set of vertices $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G induced by U, and by $G - U$ we denote the graph obtained from G by the removal of all the vertices of U, i.e., the subgraph of G induced by $V(G) \setminus U$. If $U = \{u\}$, we write $G - u$ instead of $G - \{u\}$. Respectively, for a set of edges $L \subseteq E(G)$, $G[L]$ is a subgraph of G induced by L, i.e, the vertex set of $G[L]$ is the set of vertices of G incident to the edges of L, and L is the set of edges of $G[L]$. For a nonempty set U, $\binom{U}{2}$ is the set of unordered pairs of elements of U. For a set of edges L, by $G - L$ we denote the graph obtained from G by the removal of all the edges of L. Respectively, for $L \subseteq \binom{V(G)}{2} \setminus E(G)$, $G + L$ is the graph obtained from G by the addition of the edges that are elements of L. If $L = \{a\}$, then for simplicity, we write $G - a$ or $G + a$.

For a vertex v, we denote by $N_G(v)$ its *(open) neighborhood*, that is, the set of vertices which are adjacent to v, and for a set $U \subseteq V(G)$, $N_G(U) = (\bigcup_{v\in U} N_G(v)) \setminus U$. The *closed neighborhood* $N_G[v] = N_G(v) \cup \{v\}$, and for a positive integer r, $N_G^r[v]$ is the set of vertices at distance at most r from v. For a set $U \subseteq V(G)$ and a positive integer r, $N_G^r[U] = \bigcup_{v\in U} N_G^r[v]$. The *degree* of a vertex v is denoted by $d_G(v) = |N_G(v)|$. The *maximum degree* $\Delta(G) = \max\{d_G(v) \mid v \in V(G)\}$.

For a graph G, we denote by $\sigma(G)$ its degree sequence. Notice that $\sigma(G)$ can be represented by the vector $\delta(G) = (\delta_0, \ldots, \delta_{\Delta(G)})$, where $\delta_i = |\{v \in V(G) \mid d_G(v) = i\}|$ for $i \in \{0, \ldots, \Delta(G)\}$. We call $\delta(G)$ the *degree vector* of G. For a sequence $\sigma = (\sigma_1, \ldots, \sigma_n)$, we define $\delta(\sigma) = (\delta_0, \ldots, \delta_r)$, where $r = \max \sigma$ and $\delta_i = |\{\sigma_j \mid \sigma_j = i\}|$ for $i \in \{0, \ldots, r\}$. Clearly, $\delta(G) = \delta(\sigma(G))$, and the degree vector can be easily constructed from the degree sequence and vice versa. Slightly abusing notation, we write for two vectors of nonnegative integers, that $(\delta_0, \ldots, \delta_r) = (\delta'_0, \ldots, \delta'_{r'})$ for $r \leq r'$ if $\delta_i = \delta'_i$ for $i \in \{0, \ldots, r\}$ and $\delta'_i = 0$ for $i \in \{r + 1, \ldots, r'\}$.

Parameterized Complexity. Parameterized complexity is a two dimensional framework for studying the computational complexity of a problem. One dimension is the input size n and another one is a parameter k. It is said that a problem is *fixed parameter tractable* (or FPT), if it can be solved in time $f(k) \cdot n^{O(1)}$ for some function f. A *kernelization* for a parameterized problem is a polynomial algorithm that maps each instance (x, k) with the input x and the parameter k to an instance (x', k') such that i) (x, k) is a YES-instance if and only if (x', k') is a YES-instance of the problem, and ii) $|x'| + k'$ is bounded by $f(k)$ for a computable function f. The output (x', k') is called a *kernel*. The function f is said to be a *size* of a kernel. Respectively, a kernel is *polynomial* if f is polynomial. A decidable parameterized problem is FPT if and only if it has a kernel, but it is widely believed that not all FPT problems have polynomial kernels. In particular, Bodlaender et al. [3] introduced techniques that allow to show that a parameterized problem has no polynomial kernel unless NP $\subseteq$ co-NP /poly. We refer to the recent books of Cygan et al. [11] and Downey and Fellows [15] for detailed introductions to parameterized complexity.

Solutions of Editing to a Graph with a Given Degree Sequence. Let (G, σ, k) be an instance of Editing to a Graph of Given Degree Sequence. Let $U \subset V(G)$, $D \subseteq E(G - U)$ and $A \subseteq \binom{V(G)\setminus U}{2} \setminus E(G)$. We say that (U, D, A) is a *solution* for (G, σ, k), if $|U| + |D| + |A| \leq k$, and the graph $G' = G - U - D + A$ has the degree sequence σ. We also say that G' is obtained by editing with respect to (U, D, A). If *vd*, *ed* or *ea* is not in S, then it is assumed that $U = \emptyset$, $D = \emptyset$ or $A = \emptyset$ respectively. If $S = \{ea\}$, then instead of $(\emptyset, \emptyset, A)$ we simply write A.

We conclude this section by showing that Editing to a Graph with a Given Degree Sequence is hard when parameterized by k.

Theorem 1. *For any nonempty $S \subseteq \{vd, ed, ea\}$,* Editing to a Graph with a Given Degree Sequence *is* NP*-complete and* W[1]*-hard when parameterized by k.*

Proof. Suppose that $ed \in S$. We reduce the Clique problem which asks, given a graph G and a positive integer k, whether G has a clique of size k. This problem is known to be NP-complete [17] and W[1]-hard when parameterized by k [7] even if the input graph restricted to be regular. Let (G, k) be an instance of Clique, where G is an n-vertex d-regular graph, $d \geq k-1$. Consider the sequence $\sigma = (\sigma_1, \ldots, \sigma_n)$, where

$$\sigma_i = \begin{cases} d & \text{if } 1 \leq i \leq n-k, \\ d-(k-1) & \text{if } n-k+1 \leq i \leq n. \end{cases}$$

Let $k' = k(k-1)/2$. We claim that (G, k) is a yes-instance of Clique if and only if (G, σ, k') is a yes-instance of Editing to a Graph with a Given Degree Sequence. If K is a clique of size k in G, then the graph G' obtained from G by the deletion of the $k' = k(k-1)/2$ edges of $D = E(G[K])$ has the degree sequence σ. Assume that (U, D, A) is a solution of (G, σ, k). Clearly, $U = \emptyset$ even if $vd \in S$, because σ contains n elements. Since $\sum_{i=1}^{n} \sigma_i = dn - k(k-1)$, we have that $A = \emptyset$. The degree sequence of $G - D$ contains exactly k elements $d-(k-1)$, that is, k vertices of $G-D$ have degree $d-(k-1)$. Denote by K the set of these vertices. Since $k' = k(k-1)/2$ and the deletion of an edge decreases the degree of its end-vertices by one, all deleted edges should have their end-vertices in K and each vertex of K is incident to $k-1$ deleted edges. It means that K is a clique of G.

Suppose that $ea \in S$. We reduce Independent Set problem which asks, given a graph G and a positive integer k, whether G has an independent set of size k. The reduction is essentially dual to the reduction in the previous case. Again, Independent Set is NP-complete [17] and W[1]-hard when parameterized by k [7] even if the input graph restricted to be regular. Let (G, k) be an instance of Independent Set, where G is an n-vertex d-regular graph and $k \leq n$. Consider the sequence $\sigma = (\sigma_1, \ldots, \sigma_n)$, where

$$\sigma_i = \begin{cases} d+(k-1) & \text{if } 1 \leq i \leq k, \\ d & \text{if } k+1 \leq i \leq n. \end{cases}$$

Let $k' = k(k-1)/2$. We show that (G, k) is a yes-instance of Independent if and only if (G, σ, k') is a yes-instance of Editing to a Graph with a Given Degree Sequence. If I is an independent set of size k in G, then we can add $k' = k(k-1)/2$ edges joining the vertices of I. Then the degree sequence of the obtained graph G' is σ. Suppose that (U, D, A) is a solution of (G, σ, k). Clearly, $U = \emptyset$ even if $vd \in S$, because σ contains n elements. Since $\sum_{i=1}^{n} \sigma_i = dn + k(k-1)$, we have that $D = \emptyset$. The degree sequence of $G + A$ contains exactly k elements $d+(k-1)$, that is, k vertices of $G+A$ have degree $d+(k-1)$. Denote by I the set of these vertices. Since $k' = k(k-1)/2$ and the addition of an edge increases the degree of its end-vertices by one, all added edges should have their end-vertices in I and each vertex of I is incident to $k-1$ added edges. It means that I is an independent set of G.

Finally, assume that $S = \{vd\}$. We again reduce the Clique problem for regular graphs. Let (G, k) be an instance of Clique, where G is an n-vertex d-regular graph with m edges. We assume without loss of generality that $d - (k-1) \geq 3$. The graph G' is constructed from G by subdividing each edge of G, i.e., for each $xy \in E(G)$, we construct a new vertex u_{xy} and replace xy by xu_{xy} and yu_{xy}. We say that u_{xy} is a *subdivision vertex* of G'. Let $k' = k(k-1)/2$. Consider the sequence $\sigma = (\sigma_1, \dots, \sigma_p)$, where $p = n + m - k'$ and

$$\sigma_i = \begin{cases} d & \text{if } 1 \leq i \leq n-k, \\ d-(k-1) & \text{if } n-k+1 \leq i \leq n, \\ 2 & \text{if } n+1 \leq i \leq p. \end{cases}$$

We prove that that (G, k) is a yes-instance of Clique if and only if (G', σ, k') is a yes-instance of Editing to a Graph with a Given Degree Sequence. Let K be a clique of size k in G. We define $U = \{u_{xy} \mid x, y \in K,\ xy \in E(G)\}$, that is, U contains vertices obtained by the subdivision of the edges joining the vertices of K. Since K is a clique of size k, $|U| = k(k-1)/2 = k'$. It is straightforward to verify that $G' - U$ has the degree sequence σ. Assume now that (U, D, A) is a solution of (G', σ, k'). Since $S = \{vd\}$, $D = A = \emptyset$. Notice that σ contains $m - k'$ elements 2 and G' has m vertices of degree 2 by the condition $d - (k-1) \geq 3$ and the definition of σ. Moreover, all vertices of degree 2 are subdivision vertices. It implies that U contains only subdivision vertices. The deletion of a subdivision vertex decreases the degrees of its two neighbors by one. Let K be the set of neighbors of the vertices of U. Clearly, $K \subseteq V(G)$. Observe that K has exactly k vertices of degree $d-(k-1)$ in $G'-U$ by the definition of σ. Therefore, for each $x \in K$, U contains $k-1$ subdivision vertices that are adjacent to x in G'. Since $|U| \leq k'$, we conclude that K is a clique of G. $\square$

3. FPT-algorithm for Editing to a Graph with a Given Degree Sequence

In this section we show that Editing to a Graph with a Given Degree Sequence is FPT when parameterized by $k + \Delta^*$, where $\Delta^* = \max \sigma$. In fact, we obtain this result for the more general variant of the problem:

Extended Editing to a Graph with a Given Degree Sequence	
Instance:	A graph G, a nonincreasing sequence of nonnegative integers σ and nonnegative integers k_{vd}, k_{ed}, k_{ea}.
Question:	Is it possible to obtain a graph G' with $\sigma(G') = \sigma$ from G by at most k_{vd} vertex deletions, k_{ed} edge deletions and k_{ea} edge additions?

Notice that we can solve Editing to a Graph with a Given Degree Sequence using an algorithm for Extended Editing to a Graph with a Given Degree Sequence by trying all at most $(k+1)^2$ possible values of k_{vd}, k_{ed} and k_{ea} with $k_{vd}+k_{ed}+k_{ea}=k$.

Theorem 2. Extended Editing to a Graph with a Given Degree Sequence *can be solved in time* $2^{O(k(\Delta^*+k)^2)}n^2\log n$ *for n-vertex graphs, where* $\Delta^*=\max\sigma$ *and* $k=k_{vd}+k_{ed}+k_{ea}$.

Proof. First, we construct a randomized true biased Monte Carlo FPT-algorithm for Extended Editing to a Graph with a Given Degree Sequence parameterized by $k+\Delta^*$ based on the random separation techniques introduced by Cai, Chan and Chan [8] (see also [1]). Then we explain how this algorithm can be derandomized.

Let $(G,\sigma,k_{vd},k_{ed},k_{ea})$ be an instance of Extended Editing to a Graph with a Given Degree Sequence, $n=|V(G)|$.

In the first stage of the algorithm we preprocess the instance to get rid of vertices of high degree or solve the problem if we have a trivial no-instance by the following reduction rule.

Vertex deletion rule. If G has a vertex v with $d_G(v)>\Delta^*+k_{vd}+k_{ed}$, then delete v and set $k_{vd}=k_{vd}-1$. If $k_{vd}<0$, then stop and return a NO-answer.

To show that the rule is *safe*, i.e., by the application of the rule we either correctly solve the problem or obtain an equivalent instance, assume that $(G,\sigma,k_{vd},k_{ed},k_{ea})$ is a yes-instance of Extended Editing to a Graph with a Given Degree Sequence. Let (U,D,A) be a solution. We show that if $d_G(v)>\Delta^*+k_{vd}+k_{ed}$, then $v\in U$. To obtain a contradiction, assume that $d_G(v)>\Delta^*+k_{vd}+k_{ed}$ but $v\notin U$. Then $d_{G'}(v)\leq\Delta^*$, where $G'=G-U-D+A$. It remains to observe that to decrease the degree of v by at least $k_{vd}+k_{ed}+1$, we need at least $k_{vd}+k_{ed}+1$ vertex or edge deletion operations; a contradiction. We conclude that if $(G,\sigma,k_{vd},k_{ed},k_{ea})$ is a yes-instance, then the instance obtained by the application of the rule is also a yes-instance. It is straightforward to see that if $(G',\sigma,k'_{vd},k_{ed},k_{ea})$ is a yes-instance of Extended Editing to a Graph with a Given Degree Sequence obtained by the deletion of a vertex v and (U,D,A) is a solution, then $(U\cup\{v\},D,A)$ is a solution for the original instance. Hence, the rule is safe.

We exhaustively apply the rule until we either stop and return a NO-answer or obtain an instance of the problem such that the degree of any vertex v is at most Δ^*+k. To simplify notations, we assume that $(G,\sigma,k_{vd},k_{ed},k_{ea})$ is such an instance.

In the next stage of the algorithm we apply the random separation technique. We color the vertices of G independently and uniformly at random with three colors. In other words, we partition $V(G)$ into three sets R_v, Y_v and B_v (some sets could be empty), and say that the vertices of R_v are *red*, the vertices of Y_v are *yellow* and the vertices of B_v are *blue*. Then the edges of G are colored independently and uniformly at random either *red* or *blue*. We denote by R_e the set of red and by B_e the set of blue edges respectively.

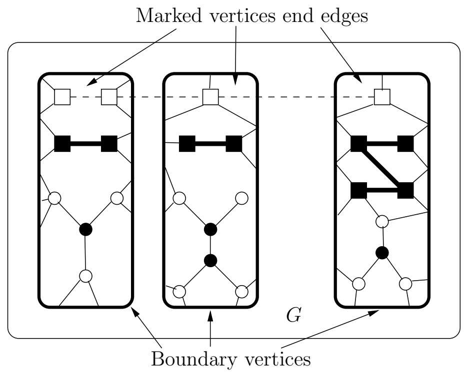

Figure 1: The structure of a solution. The vertices of U are shown by black bullets, the vertices of X are shown by black squares, the vertices of Y are white squares, the edges of D are shown by thick lines and the edges of A are shown by dashed lines.

Before we proceed with the formal description of our algorithm, we briefly discuss main ideas behind it. Assume that $(G, \sigma, k_{vd}, k_{ed}, k_{ea})$ is a yes-instance, and let (U, D, A) be a solution. Let also X and Y be the sets of vertices incident to the edges of D and A respectively. See Fig. 1. We say that the vertices of U, the neighbors of the vertices of U and the vertices of the sets X and Y are *marked* vertices (of a solution). Notice that the marked vertices are exactly the vertices whose degrees are modified by the editing operations with respect to the solution including the deleted vertices. Respectively, the degrees of unmarked vertices remain the same. We say that the unmarked vertices adjacent to the marked vertices are *boundary*. We also call edges incident to the marked vertices *marked*. Notice that $|X| \leq 2|D| \leq 2k_{ed}$ and $|Y| \leq 2|A| \leq 2k_{ea}$ and recall that $|U| \leq k_{vd}$. Taking into account that $\Delta(G) \leq \Delta^* + k$, we obtain that for the components of G induced by marked vertices and edges, we have that they have size $O(k(\Delta^* + k))$ and are separated from each other and the remaining part of the graph by $O(k(\Delta^* + k)^2)$ boundary vertices. Because we color vertices and edges independently and uniformly at random, with sufficiently high probability $(1/2)^{O(\Delta^*+k)}$ that depends only on the parameters, the marked vertices and edges together with boundary vertices are colored *correctly* with respect to the solution, that is, the vertices of U are red, the vertices of Y are yellow, the edges of D are red and the remaining vertices and edges are blue. The correctness here means that the random coloring allows to distinguish elements of the solution. In our algorithm, we are trying to find (U, D, A) (or another solution) using these structural properties and the assumption that a solution is colored correctly. To do it, we analyze the sets of vertices R_v, Y_v, B_v and the sets of edges R_e, B_e. We consider the vertices of R_v, their neighbors, the vertices of Y_v and the vertices incident to the edges of R_e to be *candidates* to be marked vertices of a solution. Respectively, the edges incident to these vertices are *candidates* to be marked edges of a solution, and the neighbors of these candidate vertices are *candidates* to be the boundary vertices. We consider the components of G induced by the candidates to be marked vertices. By the assumption about the correctness of the coloring, we have that either all red or all yellow elements of each component are participating in a solution (that is, all red vertices and edges are deleted and at least one edge incident to each yellow vertex is added) or is excluded from a solution. In particular, it means that we can ignore components that are too big. We order the remaining components and use a dynamic programming algorithm to find a solution. In this algorithm, we consider components and decide whether we include its elements in a solution or not. We also keep track of added edges using the observation that the components are separated by the candidates to be the boundary vertices. It ensures that yellow vertices of distinct components are not adjacent.

Now we formalize these ideas. We are looking for a solution (U, D, A) of $(G, \sigma, k_{vd}, k_{ed}, k_{ea})$ such that the vertices of U are colored red, the vertices incident to the edges of A are yellow and the edges of D are red. Moreover, if X and Y are the sets of vertices incident to the edges of D and A respectively, then the vertices of $(N_G^2[U] \cup N_G[X \cup Y]) \setminus (U \cup Y)$ and the edges of $E(G) \setminus D$ incident to the vertices of $N_G[U] \cup X \cup Y$ should be blue. We say that a solution (U, D, A) of $(G, \sigma, k_{vd}, k_{ed}, k_{ea})$ is a *colorful* solution if there are $R_v^* \subseteq R_v$, $Y_v^* \subseteq Y_v$ and $R_e^* \subseteq R_e$ such that the following holds.

i) $|R_v^*| \le k_{vd}$, $|R_e^*| \le k_{ed}$ and $|Y_v^*| \le 2k_{ea}$.

ii) $U = R_v^*$, $D = R_e^*$, and for any $uv \in A$, $u, v \in Y_v^*$ and $|A| \le k_{ea}$.

iii) If $u, v \in R_v \cup Y_v$ and $uv \in E(G)$, then either $u, v \in R_v^* \cup Y_v^*$ or $u, v \notin R_v^* \cup Y_v^*$.

iv) If $u \in R_v \cup Y_v$ and $uv \in R_e$, then either $u \in R_v^* \cup Y_v^*, uv \in R_e^*$ or $u \notin R_v^* \cup Y_v^*, uv \notin R_e^*$.

v) If $uv, vw \in R_e$, then either $uv, vw \in R_e^*$ or $uv, vw \notin R_e^*$.

vi) If distinct $u, v \in R_v$ and $N_G(u) \cap N_G(v) \neq \emptyset$, then either $u, v \in R_v^*$ or $u, v \notin R_v^*$.

vii) If $u \in R_v$ and $vw \in R_e$ for $v \in N_G(u)$, then either $u \in R_v^*, vw \in R_e^*$ or $u \notin R_v^*, vw \notin R_e^*$.

We also say that (R_v^*, Y_v^*, R_e^*) is the *base* of (U, D, A).

Our aim is to find a colorful solution if it exists. We do it by a dynamic programming algorithm based on the following properties of colorful solutions.

Let

$$L = R_e \cup \{e \in E(G) \mid e \text{ is incident to a vertex of } R_v\} \cup \{uv \in E(G) \mid u, v \in Y_v\},$$

and $H = G[L]$. Denote by $H_1, \ldots, H_s$ the components of H. Let $R_v^i = V(H_i) \cap R_e$, $Y_v^i = V(H_i) \cap Y_v$ and $R_e^i = E(H_i) \cap R_e$ for $i \in \{1, \ldots, s\}$.

Claim A. *If (U, D, A) is a colorful solution and (R_v^*, Y_v^*, R_e^*) is its base, then if H_i has a vertex of $R_v^* \cup Y_v^*$ or an edge of R_e^*, then $R_v^i \subseteq R_v^*$, $Y_v^i \subseteq Y_v^*$ and $R_e^i \subseteq R_r^*$ for $i \in \{1, \ldots, s\}$.*

Proof of Claim A. Suppose that H_i has $u \in R_v^* \cup Y_v^*$ or $e \in R_e^*$.

If $v \in R_v^i \cup Y_v^i$, then H_i has a path $P = x_0 \dots x_\ell$ such that $u = x_0$ or $e = x_0x_1$, and $x_\ell = v$. By induction on ℓ, we show that $v \in R_v^*$ or $v \in Y_v^*$ respectively. If $\ell = 1$, then the statement follows from iii) and iv) of the definition of a colorful solution. Suppose that $\ell > 1$. We consider three cases.

Case 1. $x_1 \in R_v \cup Y_v$. By iii) and iv), $x_1 \in R_v^* \cup Y_v^*$ and, because the (x_1, x_ℓ)-subpath of P has length $\ell - 1$, we conclude that $v \in R_v^*$ or $v \in Y_v^*$ by induction.

Assume from now that $x_1 \notin R_v \cup Y_v$.

Case 2. $x_0x_1 \in R_e$. Clearly, if for the first edge e of P, $e \in R_e^*$, then $x_0x_1 = e \in R_e^*$. Suppose that for the first vertex $u = x_0$ of P, $u \in R_v^* \cup Y_v^*$. Then by iv), $x_0x_1 \in R_e^*$. If $x_1x_2 \in R_e$, then $x_1x_2 \in R_e^*$ by v). Since $x_1x_2 \in R_e^*$ and the (x_1, x_ℓ)-subpath of P has length $\ell - 1$, we have that $v \in R_v^*$ or $v \in Y_v^*$ by induction. Suppose that $x_1x_2 \notin R_e$. Then because $x_1x_2 \in L$, $x_2 \in R_v$ and by vii), $x_2 \in R_v^*$. If $\ell = 2$, then $x_\ell \in R_v^*$. Otherwise, as the (x_2, x_ℓ)-subpath of P has length $\ell - 2$, we have that $v \in R_v^*$ or $v \in Y_v^*$ by induction.

Case 2. $x_0x_1 \notin R_e$. Then $u = x_0 \in R_v^* \cup Y_v^*$. Because $x_0x_1 \in L$, $x_0 \in R_v^*$. If $x_1x_2 \in R_e$, then $x_1x_2 \in R_e^*$ by vii). Since $x_1x_2 \in R_e^*$ and the (x_1, x_ℓ)-subpath of P has length $\ell - 1$, we have that $v \in R_v^*$ or $v \in Y_v^*$ by induction. Suppose that $x_1x_2 \notin R_e$. Then because $x_1x_2 \in L$, $x_2 \in R_v$ and by vi), $x_2 \in R_v^*$. If $\ell = 2$, then $x_\ell \in R_v^*$. Otherwise, as the (x_2, x_ℓ)-subpath of P has length $\ell - 2$, we have that $v \in R_v^*$ or $v \in Y_v^*$ by induction.

Suppose that $e' \in R_e^i$. Then H_i has a path $P = x_0 \dots x_\ell$ such that $u = x_0$ or $e = x_0x_1$, and $x_{\ell-1}x_\ell = e'$. Using the same inductive arguments as before, we obtain that $e' \in R_e^*$. $\square$

By Claim A, we have that if there is a colorful solution (U, D, A), then for its base (R_v^*, Y_v^*, R_e^*), $R_v^* = \bigcup_{i \in I} R_v^i$, $Y_v^* = \bigcup_{i \in I} Y_v^i$ and $R_e^* = \bigcup_{i \in I} R_e^i$ for some set of indices $I \subseteq \{1, \dots, s\}$.

The next property is a straightforward corollary of the definition of H.

Claim B. *For distinct $i, j \in \{1, \dots, s\}$, if $u \in V(H_i)$ and $v \in V(H_j)$ are adjacent in G, then either $u, v \in B_v$ or ($u \in Y_v^i$ and $v \in B_v$) or ($u \in B_v$ and $v \in Y_v^j$).*

We construct a dynamic programming algorithm that consecutively for $i = 0, \dots, s$, constructs the table T_i that contains the records of values of the function γ:

$$\gamma(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I),$$

where

i) $t_{vd} \le k_{vd}$, $t_{ed} \le k_{ed}$ and $t_{ea} \le k_{ea}$,

ii) $X = \{d_1, \dots, d_h\}$ is a collection (multiset) of integers, where $h \in \{1, \dots, 2t_{ea}\}$ and $d_i \in \{0, \dots, \Delta^*\}$ for $i \in \{1, \dots, h\}$,

iii) $\delta = (\delta_0, \ldots, \delta_r)$, where $r = \max\{\Delta^*, \Delta(G)\}$ and δ_i is a nonnegative integer for $i \in \{0, \ldots, r\}$,

such that (U, D, A) is a *partial solution* with the base (R_v^*, Y_v^*, R_e^*) defined by $I \subseteq \{1, \ldots, i\}$ with the following properties.

iv) $R_v^* = \bigcup_{i \in I} R_v^i$, $Y_v^* = \bigcup_{i \in I} Y_v^i$ and $R_e^* = \bigcup_{i \in I} R_e^i$, and $t_{vd} = |R_v^*|$ and $t_{ed} = |R_e^*|$.

v) $U = R_v^*$, $D = R_e^*$, $|A| = t_{ea}$ and for any $uv \in A$, $u, v \in Y_v^*$.

vi) The multiset $\{d_{G'}(y) \mid y \in Y_v^*\} = X$, where $G' = G - U - D + A$.

vii) $\delta(G') = \delta$.

In other words, t_{vd} ,t_{ed} and t_{ea} are the numbers of deleted vertices, deleted edges and added edges respectively, X is the multiset of degrees of yellow vertices in the base of a partial solution, and δ is the degree vector of the graph obtained from G by the editing with respect to a partial solution. Notice that the values of γ are defined only for some $t_{vd}, t_{ed}, t_{ea}, X, \delta$ that satisfy i)–iii), as a partial solution with the properties iv)–vii) not necessarily exists, and we only keep records corresponding to the arguments $t_{vd}, t_{ed}, t_{ea}, X, \delta$ for which γ is defined.

Now we explain how we construct the tables for $i \in \{0, \ldots, s\}$.

Construction of T_0. The table T_0 contains the unique record $(0, 0, 0, \emptyset, \delta) = (\emptyset, \emptyset, \emptyset, \emptyset)$, where $\delta = \delta(G)$ (notice that the length of δ can be bigger than the length of $\delta(G)$).

Construction of T_i for $i \geq 1$. We assume that T_{i-1} is already constructed. Initially we set $T_i = T_{i-1}$. Then for each record $\gamma(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ in T_{i-1}, we construct new records $\gamma(t'_{vd}, t'_{ed}, t'_{ea}, X', \delta') = (U', D', A')$ and put them in T_i unless T_i already contains the value $\gamma(t'_{vd}, t'_{ed}, t'_{ea}, X', \delta')$. In the last case we keep the old value.

Let $(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ in T_{i-1}.

- If $t_{vd} + |R_v^i| > k_{vd}$ or $t_{ed} + |R_e^i| > k_{ed}$ or $t_{ea} + 2|Y_v^i| > k_{ea}$, then stop considering the record. Otherwise, let $t'_{vd} = t_{vd} + |R_v^i|$ and $t'_{ed} = t_{ed} + |R_e^i|$.
- Let $F = G - U - D + A - R_v^i - R_e^i$.
- Let $\bigcup_{j \in I} Y_v^j = \{x_1, \ldots, x_h\}$, $d_F(x_f) = d_f$ for $f \in \{1, \ldots, h\}$. Let $Y_v^i = \{y_1, \ldots, y_\ell\}$. Consider every $E_1 \subseteq \binom{Y_v^i}{2} \setminus E(F[Y_v^i])$ and $E_2 \subseteq \{x_f y_i \mid 1 \leq f \leq h, 1 \leq j \leq \ell\}$ such that $|E_1| + |E_2| \leq k_{ea} - t_{ea}$, and set $\alpha_f = |\{x_f y_j \mid x_f y_j \in E_2, 1 \leq j \leq \ell\}|$ for $f \in \{1, \ldots, h\}$ and set $\beta_j = |\{e \mid e \in E_1, e \text{ is incident to } y_j\}| + |\{x_f y_j \mid x_f y_j \in E_2, 1 \leq f \leq h\}|$ for $j \in \{1, \ldots, \ell\}$.
  - If $d_f + \alpha_f > \Delta^*$ for some $f \in \{1, \ldots, h\}$ or $d_F(y_j) + \beta_j > \Delta^*$ for some $j \in \{1, \ldots, \ell\}$, then stop considering the pair (E_1, E_2).

- Set $t'_{ea} = t_{ea} + |E_1| + |E_2|$, $X' = \{d_1 + \alpha_1, \ldots, d_h + \alpha_h, d_F(y_1) + \beta_1, \ldots, d_F(y_\ell) + \beta_\ell\}$.
- Let $F' = F + E_1 + E_2$. Construct $\delta' = (\delta'_0, \ldots, \delta'_r) = \delta(F')$.
- Set $U' = U \cup R^i_v$, $D' = D \cup R^i_e$, $A' = A \cup E_1 \cup E_2$, $I' = I \cup \{i\}$, set $\gamma(t'_{vd}, t'_{ed}, t'_{ea}, X', \delta') = (U', D', A', I')$ and put the record in T_i.

We consecutively construct $T_1, \ldots, T_s$. The algorithm returns a YES-answer if T_s contains a record $(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ for $\delta = \delta(\sigma)$ and (U, D, A) is a colorful solution in this case. Otherwise, the algorithm returns a NO-answer.

The correctness of the algorithm follows from the next claim.

Claim C. *For each $i \in \{1, \ldots, s\}$, the table T_i contains a record $\gamma(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$, if and only if there are $t_{vd}, t_{ed}, t_{ea}, X, \delta$ satisfying i)-iii) such that there is a partial solution (U^*, D^*, A^*) and $I^* \subseteq \{1, \ldots, i\}$ that satisfy iv)-vii). In particular $t_{vd}, t_{ed}, t_{ea}, X, \delta$, (U, D, A) and I satisfy i)–vii) if $\gamma(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ is in T_i.*

Proof of Claim C. We prove the claim by induction on i. It is straightforward to see that it holds for $i = 0$. Assume that $i > 0$ and the claim is fulfilled for T_{i-1}.

Suppose that a record $\gamma(t'_{vd}, t'_{ed}, t'_{ea}, X', \delta') = (U', D', A', I')$ was added in T_i. Then ether $\gamma(t'_{vd}, t'_{ed}, t'_{ea}, X', \delta') = (U', D', A', I')$ was in T_{i-1} or it was constructed for some record $(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ from T_{i-1}. In the first case, $t'_{vd}, t'_{ed}, t'_{ea}, X', Q'$, (U', D', A') and $I' \subseteq \{1, \ldots, i\}$ satisfy i)-vii) by induction. Assume that $\gamma(t'_{vd}, t'_{ed}, t'_{ea}, X', \delta') = (U', D', A', I')$ was constructed for some record $(t_{vd}, t_{ed}, t_{ea}, X, Q) = (U, D, A, I)$ from T_{i-1}. Notice that $i \in I'$ in this case. Let $I = I' \setminus \{i\}$. Consider $\bigcup_{j \in I} Y^j_v = \{x_1, \ldots, x_h\}$ and $Y^i_v = \{y_1, \ldots, y_\ell\}$. By Claim B, x_f and y_j are not adjacent for $f \in \{1, \ldots, h\}$ and $j \in \{1, \ldots, \ell\}$. Then it immediately follows from the description of the algorithm that $t'_{vd}, t'_{ed}, t'_{ea}, X', \delta'$, (U', D', A') and I' satisfy i)–vii).

Suppose that there are $t_{vd}, t_{ed}, t_{ea}, X, \delta$ satisfying i)-iii) such that there is a partial solution (U^*, D^*, A^*) and $I^* \subseteq \{1, \ldots, i\}$ that satisfy iv)-vii). Suppose that $i \notin I^*$. Then T_{i-1} contains a record $\gamma(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ by induction and, therefore, this record is in T_i. Assume from now that $i \in I^*$. Let $I' = I^* \setminus \{i\}$. Consider $R'_v = \bigcup_{j \in I'} R^j_v$ and $Y'_v = \bigcup_{j \in I'} Y^j_v$. Let $E_1 = \{uv \in A \mid u, v \in T^i_v\}$ and $E_2 = \{uv \in A \mid u \in Y'_v, v \in Y^i_v\}$. Define $U' = U \setminus R^i_v$, $D' = D \setminus R^i_e$ and $A' = A \setminus (E_1 \cup E_2)$. Let $t'_{vd} = |U'|$, $t_{ed} = |D'|$ and $t_{ea} = |A'|$. Consider the multiset of integers $X' = \{d_F(v) \mid v \in Y'_v\}$ and the sequence $\delta' = (\delta'_1, \ldots, \delta'_r) = \delta(F)$ for $F = G - U' - D' + A'$. We obtain that $t'_{vd}, t'_{ed}, t'_{ea}, X', \delta'$, (U', D', A') and $I' \subseteq \{1, \ldots, i-1\}$ satisfy i)-vii). By induction, T_{i-1} contains a record $\gamma(t'_{vd}, t'_{ed}, t'_{ea}, X', \delta') = (U'', D'', A'', I'')$. Let $Y'_v = \{x_1, \ldots, x_h\}$, $\bigcup_{j \in I''} Y^j_v = \{x'_1, \ldots, x'_h\}$ and assume that $d_F(x_f) = d_{F'}(x'_f)$ for $f \in \{1, \ldots, h\}$, where $F' = G - U'' - D'' + A''$. Consider E'_2 obtained from E_2 by the replacement of every edge $x_f v$ by $x'_f v$ for $f \in \{1, \ldots, h\}$ and $v \in Y^i_v$. It remains to observe that when we consider $\gamma(t'_{vd}, t'_{ed}, t'_{ea}, X', \delta') = (U'', D'', A'', I'')$ and the pair (E_1, E'_2), we obtain $\gamma(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ for $U = U'' \cup R_v^i$, $D = D'' \cup R_e^i$, $A = A'' \cup E_1 \cup E_2'$ and $I = I'' \cup \{i\}$. $\square$

Now we evaluate the running time of the dynamic programming algorithm.

First, we upper bound the size of each table. Suppose that $\gamma(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ is included in a table T_i. By the definition and Claim C, $\delta = \delta(G')$ for $G' = G - U - D + A$. Let $\delta = \{\delta_0, \ldots, \delta_r\}$ and $\delta(G) = (\delta_0', \ldots, \delta_r')$. Let $i \in \{0, \ldots, r\}$. Denote $W_i = \{v \in V(G) \mid d_G(v) = i\}$. Recall that $\delta(G) \leq \Delta^* + k$. If $\delta_i' > \delta_i$, then at least $\delta_i' - \delta_i$ vertices of W_i should be either deleted or get modified degrees by the editing with respect to (U, D, A). Since at most k_{vd} vertices of W_i can be deleted and we can modify degrees of at most $(k+\Delta^*)k_{vd} + 2(k_{ed}+k_{ea})$ vertices, $\delta_i' - \delta_i \leq (k+\Delta^*+1)k_{vd} + 2(k_{ed}+k_{ea})$. Similarly, if $\delta_i > \delta_i'$, then at least $\delta_i - \delta_i'$ vertices of $V(G) \setminus W_i$ should get modified degrees. Since we can modify degrees of at most $(k+\Delta^*)k_{vd} + 2(k_{ed}+k_{ea})$ vertices, $\delta_i - \delta_i' \leq (k + \Delta^*)k_{vd} + 2(k_{ed} + k_{ea})$. We conclude that for each $i \in \{0, \ldots, r\}$,

$$\delta_i' - (k + \Delta^* + 1)k_{vd} + 2(k_{ed} + k_{ea}) \leq \delta_i \leq \delta_i' + (k + \Delta^*)k_{vd} + 2(k_{ed} + k_{ea})$$

and, therefore, there are at most $(2(k + \Delta^*)k_{vd} + 4(k_{ed} + k_{ea}) + 1)^r$ distinct vectors δ. Since $r = \max\{\Delta^*, \Delta(G)\} \leq \Delta^* + k$, we have $2^{O((\Delta^*+k)\log(\Delta^*+k))}$ distinct vectors δ. The number of distinct multisets X is at most $(\Delta^* + 1)^{2k}$ and there are at most $3(k + 1)$ possibilities for t_{vd}, t_{ed}, t_{ea}. We conclude that each T_i has $2^{O((\Delta^*+k)\log(\Delta^*+k))}$ records.

To construct a new record $\gamma(t_{vd}', t_{ed}', t_{ea}', X', \delta') = (U', D', A', I')$ from $\gamma(t_{vd}, t_{ed}, t_{ea}, X, \delta) = (U, D, A, I)$ we consider all possible choices of E_1 and E_2. Since these edges have their end-vertices in a set of size at most $2k_{ea}$ and $|E_1| + |E_2| \leq k_{ea}$, there are $2^{O(k \log k)}$ possibilities to choose E_1 and E_2. The other computations in the construction of $\gamma(t_{vd}', t_{ed}', t_{ea}', X', \delta') = (U', D', A', I')$ can be done in linear time. We have that T_i can be constructed from T_{i-1} in time $2^{O((\Delta^*+k)\log(\Delta^*+k))} \cdot n$ for $i \in \{1, \ldots, s\}$. Since $s \leq n$, the total time is $2^{O((\Delta^*+k)\log(\Delta^*+k))} \cdot n^2$.

We proved that a colorful solution can be found in time $2^{O((\Delta^*+k)\log(\Delta^*+k))} \cdot n^2$ if it exists. Clearly, any colorful solution is a solution for $(G, \sigma, k_{vd}, k_{ed}, k_{ea})$ and we can return it, but nonexistence of a colorful solution does not imply that there is no solution. Hence, to find a solution, we run the randomized algorithm N times, i.e., we consider N random colorings and try to find a colorful solution for them. If we find a solution after some run, we return it and stop. If we do not obtain a solution after N runs, we return a NO-answer. The next claim shows that it is sufficient to run the algorithm $N = 6^{2k(\Delta^*+k)^2}$ times.

Claim D. *There is a positive p that does not depend on the instance such that if the randomized algorithm has not found a solution for $(G, \sigma, k_{vd}, k_{ed}, k_{ea})$ after $N = 6^{2k(\Delta^*+k)^2}$ executions, then the probability that $(G, \sigma, k_{vd}, k_{ed}, k_{ea})$ is a no-instance is at least p.*

Proof of Claim D. Suppose that $(G, \sigma, k_{vd}, k_{ed}, k_{ea})$ has a solution (U, D, A). Let X be the set of end-vertices of the edges of D and Y is the set of end-vertices of A. Let $W = N_G^2[U] \cup N_G[X \cup Y]$ and denote by L the set of edges incident to the vertices of $N_G[U] \cup X \cup Y$. The algorithm colors the vertices of G independently and uniformly at random with three colors and the edges are colored by two colors. Notice that if the vertices of W and the edges of L are colored correctly with respect to the solution, i.e., the vertices of U are red, the vertices of Y are yellow, all the other vertices are blue, the edges of D are red and all the other edges are blue, then (U, D, A) is a colorful solution. Hence, the algorithm can find a solution in this case.

We find a lower bound for the probability that the vertices of W and the edges of L are colored correctly with respect to the solution. Recall that $\Delta(G) \leq \Delta^* + k$. Hence, $|W| \leq k_{vd}(\Delta^* + k)^2 + 2(k_{ed} + k_{ea})(\Delta^* + k) \leq 2k(\Delta^* + k)^2$ and $|L| \leq k_{vd}(\Delta^* + k)^2 + 2(k_{ed} + k_{ea})(\Delta^* + k) \leq 2k(\Delta^* + k)^2$. As the vertices are colored with three colors and the edges by two, we obtain that the probability that the vertices of W and the edges of L are colored correctly with respect to the solution is at least $3^{-2k(\Delta^*+k)^2} \cdot 2^{-2k(\Delta^*+k)^2} = 6^{-2k(\Delta^*+k)^2}$.

The probability that the vertices of W and the edges of L are not colored correctly with respect to the solution is at most $1 - 6^{-2k(\Delta^*+k)^2}$, and the probability that these vertices are not colored correctly with respect to the solution for neither of $N = 6^{2k(\Delta^*+k)^2}$ random colorings is at most $(1 - 1/N)^N$, and the claim follows. $\square$

Claim D implies that the running time of the randomized algorithm is $2^{O(k(\Delta^*+k)^2)} \cdot n^2$.

The algorithm can be derandomized by standard techniques (see [1, 8]) because random colorings can be replaced by the colorings induced by *universal sets*. Let m and r be positive integers, $r \leq m$. An (m, r)-*universal set* is a collection of binary vectors of length m such that for each index subset of size r, each of the 2^r possible combinations of values appears in some vector of the set. It is known that an (m, r)-universal set can be constructed in FPT-time with the parameter r. The best construction is due to Naor, Schulman and Srinivasan [25]. They obtained an (m, r)-universal set of size $2^r \cdot r^{O(\log r)} \log m$, and proved that the elements of the sets can be listed in time that is linear in the size of the set.

In our case we have $m = |V(G)| + |E(G)| \leq ((\Delta^* + k)/2 + 1)n$ and $r = 4k(\Delta^* + k)^2$, as we have to obtain the correct coloring of W and L corresponding to a solution (U, D, A). Observe that colorings induced by a universal set are binary and we use three colors. To fix it, we assume that the coloring of the vertices and edges is done in two stages. First, we color the elements of G with two colors: red and green, and then recolor the green elements by yellow or blue. By using an (m, r)-universal set of size $2^r \cdot r^{O(\log r)} \log m$, we get $4^r \cdot r^{O(\log r)} \log m$ colorings with three colors. We conclude that the running time of the derandomized algorithm is $2^{O(k(\Delta^*+k)^2)} \cdot n^2 \log n$. $\square$

4. Kernelization for Editing to a Graph with a Given Degree Sequence

In this section we show that Editing to a Graph with a Given Degree Sequence has a polynomial kernel when parameterized by $k+\Delta^*$ if $S=\{ea\}$, but for all other nonempty $S\subseteq\{vd, ed, ea\}$, there is no polynomial kernel unless NP $\subseteq$ co-NP /poly.

Theorem 3. *If $S=\{ea\}$, then* Editing to a Graph with a Given Degree Sequence *parameterized by $k+\Delta^*$ has a kernel with $O(k{\Delta^*}^2)$ vertices, where $\Delta^*=\max\sigma$.*

Proof. Let (G,σ,k) be an instance of Editing to a Graph with a Given Degree Sequence and $\Delta^*=\max\sigma$. If $\Delta(G)>\Delta^*$, (G,σ,k) is a no-instance, because by edge additions it is possible only to increase degrees. Hence, we immediately stop and return a NO-answer in this case. Assume from now that $\Delta(G)\leq\Delta^*$.

For $i\in\{0,\ldots,\Delta^*\}$, denote $W_i=\{v\in V(G)\mid d_G(v)=i\}$ and $\delta_i=|W_i|$. Our kernelization algorithm is based in the following observation. If some W_i is sufficiently large, then we can choose a subset $W_i'\subseteq W_i$ whose size is bounded by a polynomial of k and Δ^* and assume that we never add edges incident to the vertices $W_i\setminus W_i'$, that is, the vertices of $W_i\setminus W_i'$ are irrelevant. Formally, let $s_i=\min\{\delta_i, 2k(\Delta^*+1)\}$ and let $W_i'\subseteq W_i$ be an arbitrary set of size s_i for $i\in\{0,\ldots,\Delta^*\}$. We consider $W=\bigcup_{i=0}^{\Delta^*}W_i'$ and prove the following claim.

Claim A. *If (G,σ,k) is a yes-instance of* Editing to a Graph with a Given Degree Sequence*, then there is $A\subseteq\binom{V(G)}{2}\setminus E(G)$ such that $\sigma(G+A)=\sigma$, $|A|\leq k$ and for any $uv\in A$, $u,v\in W$.*

Proof of Claim A. Suppose that $A\subseteq\binom{V(G)}{2}\setminus E(G)$ is a solution for (G,σ,k), i.e., $\sigma(G+A)=\sigma$ and $|A|\leq k$, such that the total number of end-vertices of the edges of A in $V(G)\setminus W$ is minimum. Suppose that there is $i\in\{0,\ldots,\Delta^*\}$ such that at least one edge of A has at least one end-vertex in $W_i\setminus W_i'$. Clearly, $s_i=2k(\Delta^*+1)$. Denote by $\{x_1,\ldots,x_p\}$ the set of end-vertices of the edges of A in W_i and let $\{y_1,\ldots,y_q\}$ be the set of end-vertices of the edges of A in $V(G)\setminus W_i$. Since $p+q\leq 2k$, $\Delta(G)\leq\Delta^*$ and $s_i=2k(\Delta^*+1)$, there is a set of vertices $\{x_1',\ldots,x_p'\}\subseteq W_i'$ such that the vertices of this set are pairwise nonadjacent and are not adjacent to the vertices of $\{y_1,\ldots,y_q\}$. We construct $A'\subseteq\binom{V(G)}{2}\setminus E(G)$ by replacing every edge x_iy_j by $x_i'y_j$ for $i\in\{1,\ldots,p\}$ and $j\in\{1,\ldots,q\}$, and every edge x_ix_j is replaced by $x_i'x_j'$ for $i,j\in\{1,\ldots,p\}$. It is straightforward to verify that A' is a solution for (G,σ,k), but A' has less end-vertices outside W contradicting the choice of A. Hence, no edge of A has an end-vertex in $V(G)\setminus W$. $\square$

If $\delta_i\leq 2k(\Delta^*+1)$ for $i\in\{0,\ldots,\Delta^*\}$, then we return the original instance (G,σ,k) and stop, as $|V(G)|\leq 2k(\Delta^*+1)^2$. From now we assume that there is $i\in\{0,\ldots,\Delta^*\}$ such that $\delta_i>2k(\Delta^*+1)$. Using Claim A, we construct the graph G' from G by the deletion of the set of irrelevant vertices $V(G) \setminus W$. Notice that the deletion of this set could decrease the degrees of the remaining vertices. To avoid this situation, we add auxiliary vertices $v_1, \ldots, v_h$ and join them by edges with each $u \in W$ in such a way that $d_G(u) = d_{G'}(u)$. Formally, we do the following.

- Delete all the vertices of $V(G) \setminus W$.
- Construct $h = \Delta^* + 2$ new vertices $v_1, \ldots, v_h$ and join them by edges pairwise to obtain a clique.
- For any $u \in W$ such that $r = |N_G(u) \cap (V(G) \setminus W)| \geq 1$, construct edges $uv_1, \ldots, uv_r$.

Notice that $d_{G'}(v_1) \geq \ldots \geq d_{G'}(v_h) \geq \Delta^* + 1$ and $d_{G'}(u) = d_G(u)$ for $u \in W$. Observe also that $|V(G')| \leq 2k(\Delta^* + 1)^2$. Now we consider the sequence σ and construct the sequence σ' as follows.

- The first h elements of σ' are $d_{G'}(v_1), \ldots, d_{G'}(v_h)$.
- Consider the elements of σ in their order and for each integer $i \in \{0, \ldots, \Delta^*\}$ that occurs j_i times in σ, add $j_i - (\delta_i - s_i)$ copies of i in σ'.

We claim that (G, σ, k) is a yes-instance of Editing to a Graph with a Given Degree Sequence if and only if (G', σ', k) is a yes-instance of the problem.

Suppose that (G, σ, k) is a yes-instance of Editing to a Graph with a Given Degree Sequence. By Claim A, it has a solution $A \subseteq \binom{V(G)}{2} \setminus E(G)$ such that for any $uv \in A$, $u, v \in W$. It is straightforward to verify that $\sigma(G' + A) = \sigma'$, i.e., A is a solution for (G', σ', k). Assume that $A \subseteq \binom{V(G')}{2} \setminus E(G)$ is a solution for (G', σ', k). Recall that $d_{G'}(v_1), \ldots, d_{G'}(v_h)$ are the first elements of σ', $d_{G'}(v_1) \geq \ldots \geq d_{G'}(v_h) \geq \Delta^* + 1$ and $d_{G'}(u) = d_G(u) \leq \Delta^*$ for $u \in W$. It follows that for any $uv \in A$, $u, v \notin \{v_1, \ldots, v_h\}$. Otherwise, if A contains some edge $v_i u$, the degree of v_i gets increased and we cannot obtain a graph with the degree sequence σ'. We conclude that for any $uv \in A$, $u, v \in W$. Then it is straightforward to check that $\sigma(G + A) = \sigma$, i.e., A is a solution for (G, σ, k).

It is easy to verify that (G', σ', k) can be constructed in polynomial time. Since $|V(G')| = O(k{\Delta^*}^2)$, we obtain a required kernel. □

We complement Theorem 3 by showing that it is unlikely that Editing to a Graph with a Given Degree Sequence parameterized by $k + \Delta^*$ has a polynomial kernel for $S \neq \{ea\}$. The proof is based on the cross-composition technique introduced by Bodlaender, Jansen and Kratsch [3].

Theorem 4. *If S is nonempty and $S \subseteq \{vd, ed, ea\}$ but $S \neq \{ea\}$, then* Editing to a Graph with a Given Degree Sequence *has no polynomial kernel unless* NP $\subseteq$ co-NP /poly *when the problem is parameterized by $k + \Delta^*$ for $\Delta^* = \max \sigma$.*

Proof. We refer to the book of Cygan et al. [11] for the detailed introduction to the cross-composition technique. Here we only briefly remind main definitions and statements that are needed for the proof.

Recall that, formally, a parameterized problem $\mathcal{P} \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet.

Let Σ be a finite alphabet. An equivalence relation $\mathcal{R}$ on the set of strings Σ^* is called a *polynomial equivalence relation* if the following two conditions hold:

i) there is an algorithm that given two strings $x, y \in \Sigma^*$ decides whether x and y belong to the same equivalence class in time polynomial in $|x| + |y|$,

ii) for any finite set $S \subseteq \Sigma^*$, the equivalence relation $\mathcal{R}$ partitions the elements of S into a number of classes that is polynomially bounded in the size of the largest element of S.

Let $L \subseteq \Sigma^*$ be a language, let $\mathcal{R}$ be a polynomial equivalence relation on Σ^*, and let $\mathcal{P} \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. An *OR-cross-composition of* L *into* $\mathcal{P}$ (with respect to $\mathcal{R}$) is an algorithm that, given t instances $x_1, x_2, \ldots, x_t \in \Sigma^*$ of L belonging to the same equivalence class of $\mathcal{R}$, takes time polynomial in $\sum_{i=1}^{t} |x_i|$ and outputs an instance $(y, k) \in \Sigma^* \times \mathbb{N}$ such that:

i) the parameter value k is polynomially bounded in $\max\{|x_1|, \ldots, |x_t|\} + \log t$,

ii) the instance (y, k) is a yes-instance for $\mathcal{P}$ if and only if at least one instance x_i is a yes-instance for L for $i \in \{1, \ldots, t\}$.

It is said that L *OR-cross-composes into* $\mathcal{P}$ if a cross-composition algorithm exists for a suitable relation $\mathcal{R}$.

In particular, Bodlaender, Jansen and Kratsch [3] proved that if an NP-hard language L OR-cross-composes into the parameterized problem $\mathcal{P}$, then $\mathcal{P}$ does not admit a polynomial kernelization unless NP $\subseteq$ co-NP /poly.

We prove that the Clique problem which asks, given a graph G and a positive integer k, whether G has a clique of size k, OR-cross-composes into Editing to a Graph with a Given Degree Sequence if $S \neq \{ea\}$. Recall that Clique is NP-complete [17] for regular graphs. Notice also that the constructions used here are very similar to the reduction used in the proof of Theorem 1.

Suppose that $ed \in S$. We assume that two instances (G, k) and (G', k') of Clique are equivalent if $|V(G)| = |V(G')|$, $k = k'$ and G, G' are d-regular for some nonnegative integer d. Let $(G_1, k), \ldots, (G_t, k)$ be equivalent instances of Clique, where $G_1, \ldots, G_t$ are d-regular, $n = |V(G_1)| = \ldots = |V(G_t)|$ and $d \geq k - 1$. We construct the graph G by taking the disjoint union of copies of $G_1, \ldots, G_t$. Consider the sequence $\sigma = (\sigma_1, \ldots, \sigma_{nt})$, where

$$\sigma_i = \begin{cases} d & \text{if } 1 \leq i \leq nt - k, \\ d - (k-1) & \text{if } nt - k + 1 \leq i \leq nt. \end{cases}$$

Let $k' = k(k-1)/2$. We claim that (G_i, k) is a yes-instance of Clique for some $i \in \{1, \ldots, t\}$ if and only if (G, σ, k') is a yes-instance of Editing to a Graph with a Given Degree Sequence. If K is a clique of size k in G_i, then the graph G' obtained from G by the deletion of the $k' = k(k-1)/2$ edges of $D = E(G[K])$ has the degree sequence σ. Assume that (U, D, A) is a solution of (G, σ, k). Clearly, $U = \emptyset$ even if $vd \in R$, because σ contains nt elements. Since $\sum_{i=1}^{nt} \sigma_i = dn - k(k-1)$, we have that $A = \emptyset$. It remains to notice that since in $G - D$ exactly k vertices have degree $d - (k-1)$, $G[D]$ is a compete graph with k vertices, i.e., G contains a clique of size k. Clearly, any clique K of size k is a clique of some G_i for $i \in \{1, \ldots, t\}$.

Assume that $vd \in S$. Now we assume that two instances (G, k) and (G', k') of Clique are equivalent if $|V(G)| = |V(G')|$, $|E(G)| = |E(G')|$, $k = k'$ and G, G' are d-regular for some nonnegative integer d. Let $(G_1, k), \ldots, (G_t, k)$ be equivalent instances of Clique, where $G_1, \ldots, G_t$ are d-regular, $n = |V(G_1)| = \ldots = |V(G_t)|$, $m = |E(G_1)| = \ldots = |E(G_t)|$ and $d - (k-1) \geq 3$. We construct the graph G as follows.

- Take the disjoint union of copies of $G_1, \ldots, G_t$.
- For each edge $uv \in E(G_i)$ for $i \in \{1, \ldots, t\}$, subdivide it, i.e., construct a new vertex w and replace uv by uw and wv. We call the new vertices *subdivision* vertices.

Let $k' = k(k-1)/2$. Consider the sequence $\sigma = (\sigma_1, \ldots, \sigma_p)$, where $p = (n+m)t - k'$ and

$$\sigma_i = \begin{cases} d & \text{if } 1 \leq i \leq nt - k, \\ d - (k-1) & \text{if } nt - k + 1 \leq i \leq nt, \\ 2 & \text{if } nt + 1 \leq i \leq p. \end{cases}$$

We claim that (G_i, k) is a yes-instance of Clique for some $i \in \{1, \ldots, t\}$ if and only if (G, σ, k') is a yes-instance of Editing to a Graph with a Given Degree Sequence. If K is a clique of size k in G_i, then the graph G' obtained from G by the deletion of the $k' = k(k-1)/2$ subdivision vertices corresponding to the edges $G[K]$ has the degree sequence σ. Assume that (U, D, A) is a solution of (G, σ, k). Because σ has p elements and $|V(G)| - p = t(n+m) - p = k'$, U contains k' vertices and $D = A = \emptyset$. By the construction of G and σ, U contains only vertices of degree 2. As $d - (k-1) \geq 3$, we have that U contains k' subdivision vertices. It remains to notice that because in $G - U$ k vertices have degree $d - (k-1)$, the subdivision vertices of U correspond to the edges of a compete graph with k vertices, i.e., G contains a clique of size k. Clearly, any clique K of size k is a clique of some G_i for $i \in \{1, \ldots, t\}$. $\square$

5. Conclusions

In this paper we investigated the parameterized complexity of the problem, given a graph G and a degree sequence σ, to construct a graph G' from G by at most k vertex deletions, edge deletions, and edge additions, such that σ is the degree sequence of G'. We proved that, for any combination of these permitted edit operations, the problem is W[1]-hard parameterized by k. On the positive side we proved that the problem is FPT when parameterized by $k + \Delta^*$, where $\Delta^* = \max \sigma$. Furthermore we proved that, when parameterized by $k + \Delta^*$, the problem admits a polynomial kernel if only edge additions are allowed, while there is no polynomial kernel for all other combinations of permitted edit operations unless NP $\subseteq$ co-NP /poly.

It can be noted that we can obtain similar result using the maximum degree of the input graph $\Delta(G)$ instead of Δ^*, that is, for the for the parameterization by $k + \Delta(G)$. Notice that if $\Delta(G) < \Delta^* - k$, then we have a trivial no-instance of the problem, because the degree of a vertex can be increased only by edge additions and we cannot add more than k edges. Therefore, if EDITING TO A GRAPH WITH A GIVEN DEGREE SEQUENCE is FPT when parameterized by $k + \Delta^*$, then it is FPT when parameterized by $k + \Delta(G)$. In the same way, we obtain that if $S = \{ed\}$, then EDITING TO A GRAPH WITH A GIVEN DEGREE SEQUENCE admits a polynomial kernel when parameterized by $k + \Delta(G)$. Finally, we observe that for the graphs constructed in the proof of Theorem 4, the maximum degree is the same as the maximum value in the corresponding degree sequences. Hence, we have that if S is nonempty and $S \subseteq \{vd, ed, ea\}$ but $S \neq \{ea\}$, then EDITING TO A GRAPH WITH A GIVEN DEGREE SEQUENCE has no polynomial kernel unless NP $\subseteq$ co-NP /poly when the problem is parameterized by $k + \Delta(G)$. It would be interesting to study the complexity of the problem with respect to other parameters as well.

A further interesting research direction is to consider the same problem using a different type of edit operations such as vertex additions and edge contractions, among others. Moreover, specific graph classes could be also investigated in order to reduce the complexity in special cases of the input. A related question that deserves a separate effort is to investigate the problem where the aim is to obtain a graph with the degree sequence that is close, in some sense, to a given degree sequence. In particular, it could be interesting in the context of anonymizing social networks to modify a graph to get its degree sequence close to a power-law degree distribution.

References

[1] N. Alon, R. Yuster, U. Zwick, Color-coding, J. ACM 42 (4) (1995) 844–856.

[2] A.-L. Barabasi, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.

[3] H. L. Bodlaender, B. M. P. Jansen, S. Kratsch, Kernelization lower bounds by cross-composition, SIAM J. Discrete Math. 28 (1) (2014) 277–305. URL `http://dx.doi.org/10.1137/120880240`

[4] F. T. Boesch, C. L. Suffel, R. Tindell, The spanning subgraphs of Eulerian graphs, Journal of Graph Theory 1 (1) (1977) 79–84.

[5] R. Bredereck, V. Froese, S. Hartung, A. Nichterlein, R. Niedermeier, N. Talmon, The complexity of degree anonymization by vertex addition, in: AAIM 2014, vol. 8546 of Lecture Notes in Computer Science, Springer, 2014, pp. 44–55.

[6] R. Bredereck, S. Hartung, A. Nichterlein, G. J. Woeginger, The complexity of finding a large subgraph under anonymity constraints, in: ISAAC 2013, vol. 8283 of Lecture Notes in Computer Science, Springer, 2013, pp. 152–162.

[7] L. Cai, Parameterized complexity of cardinality constrained optimization problems, Comput. J. 51 (1) (2008) 102–121.
URL `http://dx.doi.org/10.1093/comjnl/bxm086`

[8] L. Cai, S. M. Chan, S. O. Chan, Random separation: A new method for solving fixed-cardinality optimization problems, in: IWPEC, vol. 4169 of Lecture Notes in Computer Science, Springer, 2006, pp. 239–250.

[9] L. Cai, B. Yang, Parameterized complexity of even/odd subgraph problems, J. Discrete Algorithms 9 (3) (2011) 231–240.

[10] G. Cornuéjols, General factors of graphs, J. Comb. Theory, Ser. B 45 (2) (1988) 185–198.
URL `http://dx.doi.org/10.1016/0095-8956(88)90068-8`

[11] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.
URL `http://dx.doi.org/10.1007/978-3-319-21275-3`

[12] M. Cygan, D. Marx, M. Pilipczuk, M. Pilipczuk, I. Schlotter, Parameterized complexity of Eulerian deletion problems, Algorithmica 68 (1) (2014) 41–61.

[13] K. K. Dabrowski, P. A. Golovach, P. van 't Hof, D. Paulusma, Editing to Eulerian graphs, in: FSTTCS 2014, vol. 29 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014, pp. 97–108.

[14] K. K. Dabrowski, P. A. Golovach, P. van 't Hof, D. Paulusma, D. M. Thilikos, Editing to a planar graph of given degrees, in: CSR 2015, vol. 9139 of Lecture Notes in Computer Science, Springer, 2015, pp. 143–156.

[15] R. G. Downey, M. R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science, Springer, 2013.
URL `http://dx.doi.org/10.1007/978-1-4471-5559-1`

[16] V. Froese, A. Nichterlein, R. Niedermeier, Win-win kernelization for degree sequence completion problems, in: SWAT 2014, vol. 8503 of Lecture Notes in Computer Science, Springer, 2014, pp. 194–205.

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

[18] P. A. Golovach, Editing to a connected graph of given degrees, in: MFCS 2014, Part II, vol. 8635 of Lecture Notes in Computer Science, Springer, 2014, pp. 324–335.

[19] P. A. Golovach, Editing to a graph of given degrees, Theor. Comput. Sci. 591 (2015) 72–84.
URL `http://dx.doi.org/10.1016/j.tcs.2015.04.034`

[20] S. Hartung, A. Nichterlein, R. Niedermeier, O. Suchý, A refined complexity analysis of degree anonymization in graphs, Inf. Comput. 243 (2015) 249–262.
URL `http://dx.doi.org/10.1016/j.ic.2014.12.017`

[21] S. Hartung, N. Talmon, The complexity of degree anonymization by graph contractions, in: TAMC 2015, vol. 9076 of Lecture Notes in Computer Science, Springer, 2015, pp. 260–271.

[22] K. Liu, E. Terzi, Towards identity anonymization on graphs, in: SIGMOD 2008, ACM, 2008, pp. 93–106.

[23] L. Mathieson, S. Szeider, Editing graphs to satisfy degree constraints: A parameterized approach, J. Comput. Syst. Sci. 78 (1) (2012) 179–191.

[24] H. Moser, D. M. Thilikos, Parameterized complexity of finding regular induced subgraphs, J. Discrete Algorithms 7 (2) (2009) 181–190.

[25] M. Naor, L. Schulman, A. Srinivasan, Splitters and near-optimal derandomization, in: 36th Annual Symposium on Foundations of Computer Science (FOCS 1995), IEEE, 1995, pp. 182–191.