Computer Science Review 48 (2023) 100556

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Review article

A survey of parameterized algorithms and the complexity of edge modification ☆

Christophe Crespelle [a], Pål Grønås Drange [b,*], Fedor V. Fomin [b], Petr Golovach [b]

[a] Université Côte d'Azur, France
[b] University of Bergen, Norway

ARTICLE INFO

Article history:
Received 12 January 2022
Received in revised form 9 January 2023
Accepted 10 April 2023
Available online 26 April 2023

Keywords:
Algorithms
Parameterized graph algorithms
Graph modification
Edge modification

ABSTRACT

The survey is a comprehensive overview of the developing area of parameterized algorithms for graph modification problems. It describes state of the art in kernelization, subexponential algorithms, and parameterized complexity of graph modification. The main focus is on edge modification problems, where the task is to change some adjacencies in a graph to satisfy some required properties. To facilitate further research, we list many open problems in the area.

Contents

☆ This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 749022 and is supported by the Research Council of Norway via the project BWCA (grant no. 314528).

* Corresponding author.

E-mail addresses: christophe.crespelle@univ-cotedazur.fr (C. Crespelle), Pal.Drange@uib.no (P.G. Drange), Fedor.Fomin@uib.no (F.V. Fomin), Petr.Golovach@uib.no (P. Golovach).

https://doi.org/10.1016/j.cosrev.2023.100556

1. Introduction

A variety of algorithmic graph problems can be formulated as problems of modifying a graph such that the resulting graph satisfies some desired properties. In particular, in the past 30 years, graph modification problems served as a strong inspiration for developing new approaches in parameterized algorithms and complexity. In this survey we are concerned with a specific type of graph modification problems, namely edge modification problems. Even for this special version of graph modification problem there is a plethora of algorithmic results in the literature. We focus on new developments in the area of parameterized algorithms and complexity for edge modification problems including kernelization, subexponential algorithms, and algorithms for finding various cuts and connectivity augmentations, as well as achieving various vertex-degree constrains. We also provide open problems for further research.

One of the classic results about graph modification problems is the work of Lewis and Yannakakis [1], that provides necessary and sufficient conditions (assuming P ≠ NP) of polynomial time solvability of vertex-removal problems for hereditary properties. However, when it concerns edge-removal problems, no such dichotomy is known. Since the work of Yannakakis [2], a great deal of work was devoted to establish which edge modification problems are in P and which are NP-complete. There already exist surveys on these topics [3–5] that the interested reader can look up.

The edge modification problems discussed in this survey fall mainly in one of the categories depending on the operations we allow; *adding edges*, *deleting edges*, and the combination of both, which we call *editing edges*. Formally, let $\mathcal{G}$ be a graph class. In the $\mathcal{G}$-Edge Completion problem, the task is to decide whether a given graph G can be transformed into a graph in $\mathcal{G}$ by *adding* at most k edges. We use the following notation. For a set F of pairs of $V(G)$, we denote by $G + F$ the graph obtained from G by making all pairs from F adjacent. Then we formally define $\mathcal{G}$-Edge Completion as follows:

$\mathcal{G}$-Edge Completion

Input: Graph G and integer k
Task: Decide whether there exists a set $F \subseteq [V(G)]^2$ of size at most k such that $G + F$ is in $\mathcal{G}$.

For example, when $\mathcal{G}$ is the class of chordal graphs, then this is the Chordal Completion problem, that is the problem of adding at most k edges to make an input graph chordal, i.e., containing no induced cycle of length more than three. If $\mathcal{G}$ is the class of 2-edge connected graphs, then this is the 2-Edge-Connectivity Augmentation problem. One natural question to ask is why it is the case that Chordal Completion is NP-complete [6], whereas 2-Edge-Connectivity Augmentation (for unweighted graphs) is solvable in polynomial time [7].

In the $\mathcal{G}$-Edge Deletion problem, the task is to decide whether a given graph G can be transformed into a graph in $\mathcal{G}$ by *deleting* at most k edges. We use the notation $G - F$, where $F \subseteq E(G)$, to denote the graph with the vertex set $V(G)$ and edge set $E(G) \setminus F$. Then we define $\mathcal{G}$-Edge Deletion as follows:

$\mathcal{G}$-Edge Deletion

Input: Graph G and integer k
Task: Decide whether there exists a set $F \subseteq E(G)$ of size at most k such that $G - F$ is in $\mathcal{G}$.

When $\mathcal{G}$ is the class of acyclic graphs, for example, then $\mathcal{G}$-Edge Deletion is trivially solvable in polynomial time (finding a minimum spanning tree). When $\mathcal{G}$ is the class of bipartite graphs, the problem is known as Odd Cycle Transversal[1] and is NP-complete [2].

Finally, in the $\mathcal{G}$-Edge Editing problem the task is to decide whether a given graph G can be transformed into a graph $G + F_+ - F_-$ in $\mathcal{G}$ using at most $|F_+| + |F_-| = k$ edges. For a set F of pairs of $V(G)$, we denote by $G \triangle F$ the graph with vertex set $V(G)$, and whose edge set is the symmetric difference of $E(G)$ and F. We define $\mathcal{G}$-Edge Editing as follows:

$\mathcal{G}$-Edge Editing

Input: Graph G and integer k
Task: Decide whether there exists a set $F \subseteq [V(G)]^2$ of size at most k such that $G \triangle F$ is in $\mathcal{G}$.

When $\mathcal{G}$ is the graph class of disjoint unions of complete graphs, i.e., cluster graphs, then this is the problem known as Cluster Editing or Correlation Clustering, the problem of deleting and adding at most k edges in a graph G such that every connected component of the obtained graph is a clique. This problem is known to be NP-complete [8]. On the other hand, the Split Editing problem, the problem of editing to a split graph (we postpone the definition of this graph till the next section) is solvable in polynomial time [9].

In this survey we intentionally tried to avoid discussions of vertex-modification problems; a survey including both would likely result in a full text-book. Many parameterized and kernelization algorithms for vertex modification problems including Vertex Cover, Feedback Vertex Set, Odd Cycle Transversal and many others can be found in the books by Cygan et al. [10] and Fomin et al. [11]. We also decided not to discuss the parameterized complexity of contraction problems since the contraction operation decreases the number of vertices in a graph, and is therefore in some sense closer in spirit to vertex removal problems. For further reading on contraction problems we refer to existing surveys [12–15].

Cai's notation. Leizhen Cai in [16] introduced a notation for graph modification problems which is widely used in the literature. Let $\mathcal{G}$ be a class of graphs, then $\mathcal{G} - ke$ (respectively $\mathcal{G} + ke$) is the class of those graphs that can be obtained from a member of $\mathcal{G}$ by deleting at most k edges (respectively adding at most k edges). We also can use $\mathcal{G} \pm ke$ for the class of graphs that can be obtained from a member of $\mathcal{G}$ by changing at most k adjacencies. With Cai's notation, the $\mathcal{G}$-Edge Completion problem is the problem to decide whether graph G is in $\mathcal{G} - ke$, $\mathcal{G}$-Edge Deletion is to decide whether $G \in \mathcal{G} + ke$, and $\mathcal{G}$-Edge Editing is to decide whether $G \in \mathcal{G} \pm ke$. Similarly, Cai's notation are also used for vertex-modification problems $\mathcal{G} - kv$ and $\mathcal{G} + kv$, although the problem $\mathcal{G} - kv$ is exceedingly rare (adding vertices to obtain a property).

Parameterized complexity. In most modification problems, and in many naturally occurring problems, we are interested in finding the *smallest* possible solution—we are looking for a solution of size at most some prescribed number k. In *parameterized complexity*, we are taking this value into account in the analysis of the running time. We are here looking for algorithms that solve problems in time $f(k)n^{\mathcal{O}(1)}$, where f can be any computable function with input k, called the *parameter*, and n is the size

[1] The problem is strictly speaking called Edge Bipartization but is computationally equivalent to Odd Cycle Transversal.

of the input, usually measured in the number of vertices in the input graph. A problem admitting such an algorithm is said to be *fixed-parameter tractable*. This means, informally and vaguely, that for fixed sized solutions, the problem is in some sense still tractable. Parameterized complexity offers a more fine-grained analysis than what the P vs. NP classification does. In addition to solution size as a parameter, there exist many other natural parameters, for example the maximum degree of the input graph, the treewidth of the input graph (or other width parameters), the size of the minimum vertex cover of the input graph, the diameter of the target graph, and many more. We will always specify which parameterization we are addressing.

More formally, a *parameterized problem* is a language $Q \subseteq \Sigma^* \times \mathbb{N}$ where Σ^* is the set of strings over a finite alphabet Σ, that is, an input of Q is a pair (I, k) where $I \subseteq \Sigma^*$ and $k \in \mathbb{N}$. We refer to k as the *parameter* of the problem. A parameterized problem Q is *fixed-parameter tractable* (FPT) if it can be decided whether $(I, k) \in Q$ in $f(k) \cdot |I|^{\mathcal{O}(1)}$ time for some function f that depends on the parameter k only. Respectively, the parameterized complexity class FPT is composed by fixed-parameter tractable problems.

Parameterized complexity theory also provides tools to rule out the existence of FPT algorithms under plausible complexity-theoretic assumptions. For this, a hierarchy of parameterized complexity classes

$$\mathsf{FPT} \subseteq \mathsf{W[1]} \subseteq \mathsf{W[2]} \subseteq \cdots \subseteq \mathsf{XP}$$

was introduced by Downey and Fellows [17], and it was conjectured that the inclusions are proper. The basic way to show that it is unlikely that a parameterized problem admits an FPT algorithm is to show that the problem is W[1]-hard or even para-NP-hard, that is, already NP-hard when the parameter value is a constant[2]. We refer to the many books on the subject [10,19–21] for a proper introduction to parameterized algorithms and complexity.

Kernelization. A *data reduction rule*, or simply, reduction rule, for a parameterized problem Q is a function $\varphi : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ that maps an instance (I, k) of Q to an equivalent instance (I', k') of Q such that φ is computable in time polynomial in $|I|$ and k. We say that two instances of Q, (I, k) and (I', k') are *equivalent* if $(I, k) \in Q$ if and only if $(I', k') \in Q$. We refer to this property of the reduction rule φ, that it translates an instance to an equivalent one, as to the *safeness* of the reduction rule.

Informally, *kernelization* is a preprocessing algorithm that consecutively applies various data reduction rules in order to shrink the instance size as much as possible. A preprocessing algorithm takes as input an instance (I, k) of Q and returns an equivalent instance (I', k') of Q in polynomial time in $|I| + k$. The quality of a preprocessing algorithm $\mathcal{A}$ is measured by the size of the output. More precisely, the *output size* of a preprocessing algorithm $\mathcal{A}$ is a function $\text{size}_{\mathcal{A}} : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ defined as follows:

$$\text{size}_{\mathcal{A}}(k) = \sup\{|I'| + k' \mid (I', k') = \mathcal{A}(I, k),\ I \in \Sigma^*\}.$$

A *kernelization algorithm*, or simply a *kernel*, for a parameterized problem Q is a preprocessing algorithm $\mathcal{A}$ that, given an instance (I, k) of Q, returns an equivalent instance (I', k') of Q in polynomial time in $|I|+k$ such that $\text{size}_{\mathcal{A}}(k) \le g(k)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$. We say that $g(\cdot)$ is the *size* of a kernel. If $g(\cdot)$ is a polynomial function, we say that Q admits a *polynomial kernel*.

It is well-known that every FPT problem admits a kernel (and vice versa), but, up to some reasonable complexity assumptions, there are FPT problems that have no polynomial kernels. In particular, we are using the composition technique introduced by Bodlaender et al. [22] to show that a parameterized problem does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. For further references on kernelization we refer to the recent book on the subject [11].

ETH. The Exponential Time Hypothesis (ETH) is a widely-believed conjecture of Impagliazzo, Paturi, and Zane [23] informally stating that 3-SAThas no algorithm subexponential in the number of variables. It is known that this conjecture implies that $\mathsf{FPT} \neq \mathsf{W[1]}$, hence it can be also used to give conditional evidence that certain problems are not fixed-parameter tractable. More importantly, ETH allows us to prove quantitative results of various forms. In particular, in this survey we mention a number of results ruling out the possibility of solving certain edge modification problems by subexponential parameterized algorithms.

The formal statement of ETH is the following. For $q \geq 3$, let δ_q be the infimum of the set of constants c for which there exists an algorithm solving q-SATin time $\mathcal{O}(2^{cn})$. Then ETH is that $\delta_3 > 0$. We refer to the book of Cygan et al. [10] for more information on ETH and its applications in parameterized algorithms.

Outline of the survey. The remaining part of this survey is organized as follows. Section 2 reviews results about edge modification problems toward hereditary graph classes. Section 3 deals with modification problems related to connectivity, cuts and clustering.[3] In Section 4 we list results where the aim of the modification problem is to make the input graph satisfy some constraints on the degrees of the vertices. Finally, Section 5 reports on variants that do not fit strictly in the scope of the previous sections but are closely related to the questions considered in this survey.

2. Hereditary graph classes

In this section, we review results on edge modifications where the target class of graphs is *hereditary*. A graph class $\mathcal{G}$ is hereditary when for any graph $G \in \mathcal{G}$, every induced subgraph of G also belongs to the class. Equivalently, this means that deleting any vertex of a graph in $\mathcal{G}$ also yields a graph in $\mathcal{G}$. Restricting ourselves to hereditary graph classes is not a sharp limitation. Although not all classes of graphs are hereditary, most classically studied graph classes are. One reason for this is that heredity is a rather natural property to require from a graph class as soon as belonging to the class is meant to be a characteristic of simplicity for a graph. In this case, it is natural to ask that a subpart of a simple object is also simple. To illustrate how ubiquitous hereditary graphs classes are, we can count forests, bipartite, planar, distance-hereditary, chordal and interval, perfect, comparability, permutation, cluster, cographs, trivially perfect, split, threshold, chain graphs, graphs of bounded treewidth, and graphs of bounded degree, to mention just some of them. Delete a vertex in a graph from any of these classes, and the resulting graph remains in that class. The classical surveys about graph classes are the books of Golumbic [24] and Brandstädt, Le, and Spinrad [25].

There are also a few notable examples of classes of graphs that are not hereditary, for instance the class of regular graphs, connected graphs, or more generally the class of graphs with at most a certain number of connected components, as well as graphs with some certain specified connectivity or degree constraints, and sparseness and density requirements. These classes are treated in Sections 3 and 4.

[2] Formally, para-NP is the parameterized equivalent of NP, and is defined similarly to NP, except that the nondeterministic Turing machine may use FPT time. It holds that para-NP = FPT if and only if PTIME = NP [18].

[3] Note that the clustering approaches that consist in modifying the input graph into some hereditary graph class, such as *cluster graphs* for example, are treated in Section 2.1.

Let $\mathcal{H} = \{H_1, H_2, H_3, \ldots\}$ be a (possibly infinite) set of graphs, we say that a graph G is $\mathcal{H}$-free if for every graph $H \in \mathcal{H}$, H is not an induced subgraph of G. The class of graphs $\mathcal{G}_{\mathcal{H}}$ is the class of all $\mathcal{H}$-free graphs. We say that $\mathcal{G}_{\mathcal{H}}$ is *characterized* by $\mathcal{H}$. When $\mathcal{H}$ is a singleton $\{H\}$, we will simply write H-free, and $\mathcal{G}_H$. It is worth to note that all classes that are defined by forbidden induced subgraphs are hereditary, and that conversely, all hereditary classes of graphs can be defined by a (possibly infinite) set of forbidden induced subgraphs: those minimal graphs (for the induced subgraph ordering) that do not belong to the class. Therefore, the edge modification problems considered here can be formulated as modifying the edge set of the input graph in order to get rid of each obstacle (i.e., forbidden induced subgraph), either by adding an edge or deleting an edge. As in the rest of the survey, the parameter we consider is the number k of modifications that are allowed. The complexities of these problems span a very broad range. For example, Split Editing is solvable in polynomial time [9] and Split Completion is NP-complete [5], Planar edge Deletion (or simply Planar Deletion here) is FPT [26] and Wheel-free edge Deletion is W[2]-hard [27], P_4-free Deletion admits a polynomial kernel [28] and P_5-free Deletion does not [29], Chordal Completion admits a subexponential time algorithm [30] while Cograph Completion does not [31,32].

This section is organized as follows. In the first two subsections, we discuss results on FPT algorithms and polynomial kernels for hereditary graph classes that are characterized by a finite number of forbidden induced subgraphs (Section 2.1) and for those characterized by an infinite number of forbidden induced subgraphs (Section 2.2). The reason for this distinction is the existence of a general result [33] that guarantees the existence of an FPT algorithm for any edge modification problem where the target class is characterized by a finite number of forbidden subgraphs. Therefore, for these classes most of the efforts focused on the existence of polynomial kernels. All the results on subexponential parameterized algorithms, both for finitely and non-finitely characterizable classes are listed in Section 2.3. Finally, Section 2.4 lists some results that deal with restricted input graphs or with target classes that are non-hereditary variants of some hereditary classes.

For more on polynomial kernels with respect to the aforementioned graph classes, one may consult the survey on the kernelization complexity by Liu, Wang, and Guo [34] and the master thesis of Cai [29].

2.1. Classes characterized by a finite number of minimal forbidden subgraphs

Several well known graph classes can be characterized by a finite set of forbidden induced subgraphs. This includes cluster, split, threshold, chain, trivially perfect, cographs, triangle-free, claw-free, line graphs, and many more. As mentioned above, there is a general result by Cai [33] that had a strong impact on the study of parameterized complexity of edge modification problems into classes of graphs defined by a finite family of forbidden subgraphs. This algorithmic result can be stated for the generic $\mathcal{G}$ (k_1, k_2, k_3)-Editing problem, which is defined as follows.

$\mathcal{G}$ (k_1, k_2, k_3)-Editing

Input:	$G = (V, E), k_1, k_2, k_3$
Task:	Are there sets $V_- \subseteq V$ of size at most k_1, $E_- \subseteq E$ of size at most k_2, and $E_+ \subseteq [V]^2$ of size at most k_3, such that $G - V_- - E_- + E_+$ is a graph in $\mathcal{G}$?

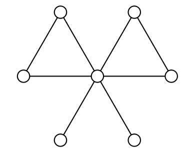

Fig. 1. The graph H_{KW}.

Theorem 2.1 (*Cai's Theorem [33]*). *Let $\mathcal{G}$ be a graph class characterized by a finite set of forbidden induced subgraphs. Then $\mathcal{G}$ (k_1, k_2, k_3)-Editing is solvable in $\mathcal{O}({c_1}^k n^{c_2})$ time, where $k = k_1 + k_2 + k_3$ and c_1 and c_2 depend only on the finite characterization of $\mathcal{G}$.*

In particular, for the problems we are interested in here, this means that completion ($k_1 = 0$ and $k_2 = 0$), deletion ($k_1 = 0$ and $k_3 = 0$) and editing ($k_1 = 0$ and try all couples k_2, k_3 such that $k_2 + k_3 \leq \ell$) are all FPT parameterized by the number of modifications allowed (k_3 in the completion problem, k_2 in the deletion problem, and ℓ in the editing problem). This completely settles the parameterized complexity for many problems (see above for a list of some finitely characterizable graph classes) and has two immediate consequences for the domain:

1. Since the FPT status for modification into finitely characterizable classes is settled, for the hereditary graph classes we are only interested in graphs with infinite characterization (see Section 2.2).
2. For classes defined by a finite set of forbidden subgraphs, from the perspective of parameterized complexity the questions of interest are
   (a) Improving the (exponential) dependence of the running time in Theorem 2.1 on the parameter k. Such improvements can in some cases lead to subexponential parameterized running times, see Section 2.3;
   (b) Exploring the possibility of polynomial kernelization (we focus on these results in this section).

Interestingly, the general result of Cai about the existence of FPT algorithms extends to kernelization for vertex deletion problems. Indeed, in these settings, the task is to hit all the copies of these forbidden subgraphs (so-called *obstacles*) that are originally contained in the graph. Hence, one can construct a simple reduction to the d-Hitting Set problem for a constant d depending on $\mathcal{G}$, and use the classic $\mathcal{O}(k^d)$ kernel for the latter that is based on the sunflower lemma [18,35]. Unfortunately, for edge modification problems, this approach fails utterly: every edge addition and deletion can create new obstacles, and thus it is not sufficient to hit only the original ones. For this reason, kernelization of edge modification problems have received a good deal of attention even for finitely characterizable classes.

From 2007, Guo [53] and Gramm et al. [54] provided kernels for several graph modification problems towards graph classes characterized by a finite set of forbidden induced subgraphs, including cluster, split, threshold, chain, and trivially perfect graphs. Several other positive results followed, which led Fellows et al. to ask whether all $\mathcal{H}$-free modification problems for finite $\mathcal{H}$ admit polynomial, and even linear kernels [55].

This was refuted by Kratsch and Wahlström [56] using the framework of Bodlaender et al. [22], who showed that for a certain graph on seven vertices, namely H_{KW} (depicted on Fig. 1), none of the problems H_{KW}-free Deletion nor H_{KW}-free edge Editing, admit polynomial kernels unless NP $\subseteq$ coNP/poly. (NP $\subseteq$ coNP/poly implies that PH is contained in Σ_3^p. We refer to the textbook on parameterized algorithms [10] for further

Table 1
Kernelization complexity of edge modification problems into hereditary graph classes characterized by a finite number of forbidden induced subgraphs. NOKER means that the problem does not have a polynomial kernel unless NP ⊆ coNP/poly. OPEN means that the complexity is open, while "—" means that the problem is probably open but most likely nobody looked at this question. P means the problem is solvable in polynomial time. A dagger next to the name of the class marks self-complementary classes, for which any result for one of the completion problem or deletion problem automatically gives the same result for the other problem.

Graph class	Polynomial kernel		
	Completion	Deletion	Editing
line	OPEN	OPEN	OPEN
s-plex cluster	—	—	s^2k [36]
chain ($\{K_3, 2K_2, C_5\}$)	as deletion	k^2 [37,38]	k^2 [38]
starforest ($\{K_3, C_4, P_4\}$)	P	$4k$ [39]	as deletion
threshold† ($\{2K_2, C_4, P_4\}$)	k^2 [38]	k^2 [38]	k^2 [38]
split† ($\{2K_2, C_4, C_5\}$)	k [39], $5k^{1.5}$ [40]	k [39], $5k^{1.5}$ [40]	P [9]
clique + IS ($\{P_3, 2K_2\}$)	P	$k/\log k$ [39]	$2k$ [folkl.]
trivially perfect ($\{C_4, P_4\}$)	k^2 [39,40]	k^3 [41]	k^3 [41]
$\{claw, diamond\}$	OPEN	$k^{\mathcal{O}(1)}$ [42]	OPEN
pseudosplit† ($\{2K_2, C_4\}$)	$5k^{1.5}$ [40]	$5k^{1.5}$ [40]	P [9,43]
cluster ($\{P_3\}$)	P	$2k$ [40]	$2k$ [44,45]
$\{K_3\}$	P	$6k$ [46]	as deletion
cograph† ($\{P_4\}$)	k^3 [28]	k^3 [28]	k^3 [28]
$\{paw\}$	k [47]	k^3 [48]	k^6 [48]
$\{diamond\}$	P	k^3 [49,50]	k^8 [50]
$\{claw\}$	OPEN	OPEN	OPEN
$\{K_4\}$	P	k^3 [51]	as deletion
$\{P_\ell\}$, $\ell > 4$	NOKER [52]	NOKER [29]	NOKER [52]
$\{C_\ell\}$, $\ell > 3$	NOKER [52]	NOKER [52]	NOKER [52]

discussions.) This shows that the subtle differences between edge modification and vertex deletion problems have tremendous impact on the kernelization complexity. They conclude by asking whether there is a "simple" graph, like a path or a cycle, for which an edge modification problem does not admit a polynomial kernel under similar assumptions. This question was answered by Guillemot et al. [28] who showed that both for the class of P_ℓ-free graphs (for $\ell \geq 7$) and for the class of C_ℓ-free graphs (for $\ell \geq 4$), the edge deletion problems do not have polynomial kernelization algorithms, unless NP ⊆ coNP/poly. They simultaneously gave a cubic kernel for the Cograph Editing problem, the problem of editing to a graph without induced paths on four vertices, showing that there is a fundamental difference between P_4-free and P_7-free graphs when it comes to modification problems.

This led to further developments on polynomial kernelization for classes characterized by excluding one single graph H. The most prominent result in this direction is the one by Cai and Cai [52] who attempted to obtain a complete dichotomy of the kernelization complexity of edge modification problems for classes of H-free graphs, for every graph H. The project has been very successful—the question is settled for all 3-connected graphs, all paths and cycles, as well as all but a finite number of trees. They show that when H is 3-connected, H-free Deletion and Editing admit no polynomial kernel if and only if H is not complete; and H-free Completion admits no polynomial kernel if and only if H misses at least two edges. More precisely, the results of Cai and Cai are summarized in the following theorem.

Theorem 2.2 ([52]). *Let $\mathcal{G}$ be a hereditary class of graphs characterized by a single forbidden induced subgraph H. Then assuming* NP ⊈ coNP/poly,

- *when H is 3-connected, $\mathcal{G}$-Edge Deletion and $\mathcal{G}$-Edge Editing admit polynomial kernels if and only if H is a complete graph. $\mathcal{G}$-Edge Completion admits a polynomial kernel if and only if H is complete or $K_n - e$, a complete graph minus one edge, and*
- *when H is a fixed path or cycle, $\mathcal{G}$-Edge Deletion, $\mathcal{G}$-Edge Editing, and $\mathcal{G}$-Edge Completion admit polynomial kernels if and only if H has at most 4 edges.*

Moreover, Cai and Cai proved that if $\mathcal{G}$ is characterized by a finite family of forbidden subgraphs $\mathcal{F}$, then $\mathcal{G}$-Edge Deletion admits no polynomial kernel if all graphs in $\mathcal{F}$ are 3-connected and there is a graph $H \in \mathcal{F}$ with fewest edges such that one can add an edge to H to obtain a graph not in $\mathcal{F}$.

As a consequence of Theorem 2.2, the existence of a polynomial kernel for any of H-free Editing, H-free Deletion, or H-free Completion problem is in fact a very rare phenomenon. It essentially happens only for very specific graphs H.

Beside this, one can see in Table 1 that the question of whether certain edge modification problems into H-free graphs admit kernels has been answered for all graphs on three vertices (K_3 and P_3) and for almost all graphs on four vertices. The only case remaining is the claw ($K_{1,3}$), which is unsolved for completion, deletion, and editing. For C_4-free graphs, Guillemot et al. [28] showed that none of the three modification problems admit a kernel. On the positive side, they show the existence of a cubic kernel for each of the three modification problems into the class of P_4-free graphs (cographs). For the class of cographs, there was also some effort put into obtaining the best possible FPT algorithm resulting in 2.56^k complexity for completion and deletion [57] and 4.61^k for editing [58]. The case of diamond-free graphs also drew quite a bit of attention. Fellows et al. [59] designed a k^4 vertex kernel for Diamond-free Deletion, which was improved to k^3 by Sandeep and Sivadasan [49]. Cao et al. [50] also provided a k^3 vertex kernel for the deletion problem, following a different approach, and a k^8 kernel for Diamond-free Editing. Tsur [51] gave a k^{t-1} vertex kernel for the K_t-free deletion problem.

The question about the existence of a polynomial kernel for Claw-free Deletion highlights how little help a finite characteristic provides. Cygan et al. [42] using modulator techniques (obtaining a specific vertex deletion modulator X) similar to that used for showing kernels for modification to trivially perfect graphs, threshold, and chain graphs [32,38] showed that deletion to a subclass of claw-free graphs, Claw-Diamond-free Deletion, admits a polynomial kernel, pinpointing the really hard cases that are left to solve in order to obtain a polynomial kernel for Claw-free Deletion. On the negative side, Cai showed that S_{11}-free Deletion does not have a kernel unless NP $\subseteq$ coNP/poly [29]. Here, S_{11} is the star on 11 vertices, while the claw is the star on 4 vertices. Moreover, since forbidding induced $S_3 = P_3$ is the characterization for cluster graphs, S_3-free Deletion admits a polynomial kernel, and thus there is a gap in our knowledge for the S_t-free Deletion problems with $4 \leq t \leq 10$.

Open problem 2.1 (*[42,52,60]*). *Does* Claw-free Deletion *admit a polynomial kernel?*

By the well-known characterization of line graphs by Beineke [61], a graph is a line graph if and only if it does not contain one of nine graphs as an induced subgraph. One of these graphs is a claw.

Open problem 2.2 (*[60]*). *Does* Line Graph Deletion *admit a polynomial kernel?*

Similar questions are open for Line Graph Completion and Line Graph Editing.

There has also been some attempt to generalize the approach of Cai and Cai [52] to families of hereditary graphs characterized by not only a single obstruction but a finite number of them. This gave the very nice result contained in the work of Aravind, Sandeep, and Sivadasan [62], but which is valid only for restricted input graphs: if the input graphs have bounded degree and if the graphs in $\mathcal{F}$ are connected, then the $\mathcal{F}$-free Deletion problem admits a polynomial kernel.

Among the classes of graphs listed in Table 1, one received a particular attention: cluster graphs (see the survey by Böcker and Baumbach [63] for more on the topic). The reason is that cluster graph modification problems, more precisely deletion and editing, are closely related to the question of community detection, which is central in the domain of complex networks. It is striking to see that despite the simplicity of the structure of cluster graphs (they are disjoint union of cliques), both the editing and deletion problems remain NP-complete. Completion is trivially polynomial: simply turn each connected component into a clique. From a kernelization perspective, Gramm et al. [64] first showed the existence of a k^3 kernel both for Cluster Deletion and Cluster Editing. The editing kernel was improved to linear size, namely $6k$, by Fellows et al. [55] and there were several works putting efforts to further reduce the size of the kernel to $4k$, by Guo [65], and then to $2k$ by Chen and Meng [44] and by Cao and Chen [45] independently. The same efforts were put in trying to obtain the best possible complexity for FPT algorithms solving these modification problems. Gramm et al. [64] first obtained a 2.27^k complexity for editing and 1.77^k for deletion, which was improved by Böcker and Damaschke [66] to 1.76^k and 1.41^k, respectively. The editing version was again improved by Böcker to $\mathcal{O}(1.62^k + m + n)$ [67], where ~ 1.62 is the golden ratio arising from a branching vector $\tau = (2, 1)$.

Van Bevern, Froese, and Komusiewicz [68] looked at parameterized algorithms and kernelization for graph modification problems above packing guarantee. For example, if an input graph G contains ℓ modification-disjoint induced P_3s (no pair of these P_3s share an edge or non-edge), then in order to be transformed into a cluster, graph G requires at least ℓ edits. Then a perhaps more "honest" question is whether $\ell + k$ edits will suffice. For Cluster Editing, Li, Pilipczuk, and Sorge [69] show that the problem is NP-complete for $\ell = 0$.

Open problem 2.3 (*[68]*). *Is* Cograph Editing *(editing to a P_4-free graph) with $\ell + k$ edits, where ℓ is the number of vertex disjoint induced P_4s in the input graph,* FPT *parameterized by k?*

Many variants of the problem of cluster editing have been considered in the literature. They are not listed in Table 1 and we report them below. Guo et al. generalized the Cluster Editing problem to a problem called s-Plex Editing [36]. An s-plex is one way of generalizing the notion of a clique. A set S is an s-plex in a graph G if every vertex $v \in S$ has degree at least $|S| - s$ in $G[S]$. Hence, a clique is a 1-plex. A graph G is then an s-plex cluster if every connected component is an s-plex. They show that the s-plex cluster graphs are characterizable by a finite set of forbidden induced subgraphs and they give an $\mathcal{O}(s^2 k)$ vertex kernel for the problem as well as two FPT algorithms, one running in time $\mathcal{O}\left((2s + \lfloor\sqrt{s}\rfloor)^k \cdot s \cdot (n + m)\right)$ and one running in time $\mathcal{O}\left((2s + \lfloor\sqrt{s}\rfloor)^k + n^4\right)$. It is worth noting that the number of obstructions of these classes depends exponentially on s, but each of the obstructions is of size $\mathcal{O}(s)$.

Fellows et al. [59] studied another relaxed version of the cluster editing problem, where a vertex (s-vertex-overlap) or an edge (s-edge-overlap) is allowed to participate in at most s clusters, where s is part of the input. All the corresponding modification problems are shown to be NP-hard when $s \geq 1$ ($s \geq 2$ in the case of completion), W[1]-hard when parameterized by k and FPT parameterized by (s, k). They also gave an $\mathcal{O}(k^4)$ kernel for 1-Edge-Overlap Deletion (which is exactly Diamond-free Deletion) and an $\mathcal{O}(k^3)$ kernel for 2-Vertex-Overlap Deletion. Other results about different approaches to clustering problems are given in Section 3.

Xia and Zhang [70] studied the problems s-cycle Transversal and $_{\leq}$s-cycle Transversal. In these problems, the task is to find a set of edges $S \subseteq E(G)$ of a given graph G of size at most some given budget k, such that every (not necessarily induced) cycle of length (at most) s in G has an edge in S. For $s = 3$ these problems become Triangle-Free Deletion, which is known to admit a linear kernel [46]. Xia and Zhang show that $_{\leq}$s-cycle Transversal is NP-complete, even on planar graphs of maximum degree seven, for any $s \geq 3$. They give a $6k^2$ vertex kernel for both 4-cycle Transversal and $_{\leq}$4-cycle Transversal, implying that the $\{C_3, C_4\}$-free Deletion problem admits a $6k^2$ kernel. The problems were already known to admit $\mathcal{O}(k^{s-1})$ vertex kernels by a reduction to Hitting Set [35,70].

Due to the structure of the two classes, the modification problems into threshold graphs and chain graphs are closely related. Guo [53] gave a cubic vertex kernel for Threshold Completion and Threshold Deletion (the class is self-complementary) and Bessy and Perez [37] gave a quadratic kernel for Chain Deletion. (The characterizations of all these graph classes in the form of forbidden subgraphs is given in Table 1.) Until recently, it was unknown whether Threshold Editing and Chain Editing were NP-hard or not. This was shown by Drange et al. [38], who obtained quadratic kernels for all three modification problems towards threshold graphs and chain graphs. Furthermore, Chain Deletion was shown to be solvable in $2.57^k n^{\mathcal{O}(1)}$ time by Liu et al. [71]. For split graphs completion and deletion (graphs excluding $\{2K_2, C_4, C_5\}$), Guo [53] initially gave a k^4 kernel which was later improved to k^2 [72].

In the same article, Guo [53] also provided a k^3 kernel for Trivially Perfect Completion (trivially perfect graphs are also known as *quasi-threshold graphs*) and polynomial k^7 kernels have been obtained for the deletion and editing versions of the problem by Drange and Pilipczuk [32]. Later, Dumas, Perez, and Todinca [41] improved the k^7 kernels to k^3 for all three problems, and simultaneously, Bathie et al. [39] and Cao and Ke [40] showed that Trivially Perfect Completion admits a quadratic kernel (Cao and Ke give an explicit $3k^2$ vertex kernel).

Nastos and Gao [57] designed a $2.45^k n^{\mathcal{O}(1)}$ time[4] FPT algorithm for Trivially Perfect Deletion which was later improved to $2.42^k n^{\mathcal{O}(1)}$ by Liu et al. [71].

2.2. Classes characterized by an infinite number of minimal forbidden subgraphs

Although many studied graph classes are finitely characterizable, there are important examples that are outside this regime, such as chordal graphs (defined as graphs with no induced cycle of length at least four) or interval graphs (chordal graphs without asteroidal triples) for example. Therefore, Cai's theorem does not directly cover modification problems into chordal or interval graphs. However, consider the problem Chordal Completion, which constituted a seminal case study for parameterized complexity of edge modification problems. Given an input instance (G, k) of Chordal Completion, we may observe that if G has an induced cycle of length more than $k+3$, then (G, k) is a no-instance [33]. Therefore, even though chordal graphs do not have a finite characterization, the set of obstacles can be bounded by a function of k: (G, k) is a yes-instance of Chordal Completion if and only if (G, k) is a yes-instance of $\mathcal{H}$-free Completion for $\mathcal{H} = \{C_4, C_5, \dots, C_{k+4}\}$ and the $\mathcal{H}$-free graph output is chordal (see Table 5).

Thanks to this fundamental property, Kaplan, Shamir, and Tarjan [74] showed as early as 1994 that Chordal Completion (usually called Minimum Fill-In) can be solved in $16^k \cdot n^{\mathcal{O}(1)}$ time and admits a polynomial kernel with $\mathcal{O}(k^3)$ vertices. In 1996, Cai improved their result on Chordal Completion by giving an FPT algorithm for the problem running in time $\mathcal{O}(4^k \cdot (n+m))$ [33] and in 2000, the analysis of the kernelization algorithm of [74] was improved by Natanzon, Shamir, and Sharan [75] to show that it actually produces a kernel of size $\mathcal{O}(k^2)$. For deletion and editing, no polynomial kernel is known.

Open problem 2.4. *Do* Chordal Deletion *and* Chordal Editing *admit polynomial kernels?*

The related problem of deleting at most k vertices to obtain a chordal graph admits a polynomial kernel [76,77]. A general version of Chordal Editing was shown to be FPT by Cao and Marx [78]. In fact, they showed that $\mathcal{G}$ (k_1, k_2, k_3)-Editing (see Section 2.1 for the definition of the problem), with $\mathcal{G}$ being the class of chordal graphs, is FPT parameterized by $k = k_1 + k_2 + k_3$. That is, vertex deletion, edge deletion,[5] edge completion as well as edge editing to chordal graphs are all FPT as a result. Then, the result of Cao and Marx can be seen as an extension of Cai's theorem to graph classes without finite characterizations.

It could be interesting to see if there are natural ways of extending Cai's theorem to include this result. Answering that question, one needs to take into account that Wheel-free Completion is W[2]-hard, so any such characterization should exclude this class.

Open problem 2.5. *Are there natural extensions of Cai's theorem to include also chordal graphs?*

Kaplan et al. [74] also provided FPT-like algorithms for completion into subclasses of chordal graphs, namely Strongly Chordal Completion and Proper Interval Completion, in $\mathcal{O}(64^k n^{\mathcal{O}(1)})$ time and $\mathcal{O}(16^k n^{\mathcal{O}(1)})$ respectively, and they asked for a similar result for Interval Completion. This was not solved until almost ten years later, when Villanger, Heggernes, Paul, and Telle [80] showed that Interval Completion was indeed fixed-parameter tractable. The complexity of the best FPT algorithm available for the problem was later lowered to $\mathcal{O}(6^k n^{\mathcal{O}(1)})$ time by Cao [81,82]. Villanger, Heggernes, Paul, and Telle [80] asked specifically for a polynomial kernel for Interval Completion. This question was raised again in the work of Bliznets, Fomin, Pilipczuk, and Pilipczuk [83] and became one notoriously hard problem in the domain, but the existence of kernels both for the subclass of proper interval graphs and for the superclass of chordal graphs makes the question particularly appealing.

Open problem 2.6 (*[37,83]*)**.** *Does* Interval Completion *admit a polynomial kernel?*

Open problem 2.7. *Does* Interval Deletion *admit a polynomial kernel?*

Note that the problem of deleting at most k vertices to obtain an interval graph admits a polynomial kernel [84]. Cao [85] gave an algorithm of running time $k^{\mathcal{O}(k)}\mathcal{O}(n+m)$ for Interval Deletion. The existence of single-exponential algorithm for this problem is open.

Open problem 2.8. *Could* Interval Deletion *be solved in time* $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$*?*

Open problem 2.9 (*[85]*)**.** *Is* Interval Editing *fixed-parameterized tractable?*

For the subclass of proper interval graphs, the FPT running time of Kaplan, Shamir, and Tarjan [74] was improved to $\mathcal{O}(4^k n^{\mathcal{O}(1)})$ by Liu et al. [86]. Bessy and Perez [37] gave a polynomial kernel for Proper Interval Completion with $\mathcal{O}(k^3)$ vertices and Cao recently showed that Proper Interval Deletion is FPT, namely solvable in $\mathcal{O}(2^{\mathcal{O}(k \log k)}(n+m))$ time [82].

Open problem 2.10 (*[37]*)**.** *Do* Proper Interval Deletion *and* Proper Interval Editing *admit polynomial kernels?*

We observe that the question above is actually open for most of the subclasses of chordal graphs shown in Table 3 (except 3-leaf powers). Finding a kernel for one of these classes or proving that there is none is a question of high interest.

Another subclass of chordal graphs that received quite a bit of attention in the parameterized framework is the class of p-leaf power graphs. Motivated by the problem of reconstructing evolutionary history, Nishimura, Ragde, and Thilikos [87] defined p-leaf powers as follows. Let T be a tree and L_T be the leaves of T. The p-leaf power of T is the graph $G = (L_T, E)$ where $uv \in E$ if and only if $\text{dist}_T(u, v) \leq p$. It follows that the 1-leaf power graphs are the independent sets and the 2-leaf power graphs are the cluster graphs, i.e. the P_3-free graphs. The editing, deletion, and completion problems towards p-leaf power graphs are NP-hard for every $p \geq 3$.

All three modification problems into the class of 3-*leaf-power* graphs, which are also chordal bull-dart-gem-free[6] graphs, were shown to be FPT by Dom et al. [88] and Bessy, Paul, and Perez [89] later showed that these three problems also admit linear time cubic vertex kernels. The 4-leaf power modification problems were all shown to be fixed-parameter tractable in two articles by Dom, Guo, Hüffner, and Niedermeier [90,91]. For 5-leaf power graphs, there is a linear time recognition algorithm, which leaves

4 It was Nastos and Gao who renewed the interest in the Trivially Perfect Editing problem by discovering that trivially perfect graphs can serve as a measure for hierarchical clusters in social networks [73].

5 The existence of an FPT algorithm for Chordal Deletion had been already established by Marx [79].

6 bull is K_3 where two of the vertices have pendants, gem is $P_4 \cdot K_1$, dart is the $K_4 - e$ with a pendant attached to a degree-three vertex.

the obvious open question below. The question is actually open for all $p \geq 5$, but there is currently no polynomial recognition algorithm known for $p \geq 6$.

Open problem 2.11 (*[91]*). *Is* 5-Leaf Power Editing *(also known as* Closest 5-Leaf Power*)* FPT*?*

Dumas, Perez, and Todinca study modification to the graph class of *strictly chordal graphs* (also known as block duplicate graphs), a class of chordal graphs which is sandwiched between 3-leaf power graphs and 4-leaf power graphs. The class can be defined as the chordal graphs that are dart-gem-free [92]. They show that all three modification problems are NP-complete and give an k^3 kernel for Strictly Chordal Completion and an k^4 kernel for the two remaining problems, Strictly Chordal Editing and Strictly Chordal Deletion.

The class of chordal distance-hereditary graphs is the class of chordal graphs that are also distance-hereditary, i.e., for every induced subgraph G' of G, $\text{dist}_G(u, v) = \text{dist}_{G'}(u, v)$. This class is also referred to as ptolemaic[7] graphs [25], which are also the chordal gem-free graphs. Crespelle, Gras, and Perez [93] initiate the study of Ptolemaic Completion and show that it is NP-complete, and that it admits an k^4 kernel.

One important graph class for which there is no result in the parameterized framework is the class of perfect graphs. It might therefore seem reasonable to start working with modification towards some of its subclasses as a first step in gaining insight into modification towards perfect graphs themselves. One interesting subclass is the class of distance-hereditary graphs. A connected graph G is distance-hereditary when for every two vertices v and u in G, and every connected induced subgraph G' of G, containing v and u, $\text{dist}_G(v, u) = \text{dist}_{G'}(v, u)$. The class is clearly hereditary is also characterized by being house-, hole- (induced cycle of length at least five), domino-, and gem-free graphs, or so-called HHDG-free graphs [25]. For distance-hereditary graphs, the existence of FPT algorithms for edge modification problems follows from a result by Courcelle, Makowsky, and Rotics on *Monadic Second Order Logic* (MSO) [94] since any graph class $\mathcal{G}$ with bounded rank-width and for which membership is definable in the variant of MSO without edge set quantifiers is in FPT. Nevertheless, it would be interesting to improve the complexity of FPT algorithms resulting from the general theorem mentioned above and the question of the existence of polynomial kernels for the three modification problems into distance-hereditary graphs is still open. Kim and Kwon recently showed that the vertex deletion variant admits a polynomial kernel [95].

Another problem (or class of problems) that admits fixed-parameter tractable algorithms as a result of general tools is the problem of Planar Deletion. Here the task is to delete at most k edges to obtain a planar graph. Since the class of graphs Planar $+ ke$ is minor-closed and thus by the fundamental result of Robertson and Seymour [96] is characterized by a finite set of forbidden minors, the minor testing algorithm by Robertson and Seymour from the graph minors project [97] implies that the problem is non-uniformly FPT. Planar Deletion was shown by Kawarabayashi and Reed to admit a linear time FPT algorithm [26]. Using the algorithm by Adler, Grohe, and Kreutzer [98] combined with the minor testing algorithm by Robertson and Seymour, we obtain uniform FPT algorithms for $\mathcal{H}$-minor free Deletion. But the existence of polynomial kernels for these problems is open.

Open problem 2.12. *Does* Planar Deletion *admit a polynomial kernel?*

[7] The *ptolemaic inequality* is defined as $\text{dist}(u, v) \cdot \text{dist}(w, x) \leq \text{dist}(u, w) \cdot \text{dist}(v, x) + \text{dist}(u, x) \cdot \text{dist}(v, w)$.

Open problem 2.13. *Does* $\mathcal{H}$-minor free Deletion *admit a polynomial kernel?*

For the related problem of deleting at most k vertices to obtain an $\mathcal{F}$-minor free graph, a non-uniform polynomial kernel is known when family $\mathcal{F}$ contains at least one planar graph [99].

In 2004, Odd Cycle Transversal (which is Bipartite Vertex Deletion) and its edge version, called Edge Bipartization (which is Bipartite Deletion), were shown to be solvable in time $3^k n^{\mathcal{O}(1)}$ by Reed, Smith, and Vetta [100], inventing the now well-known technique *iterative compression*. Iterative compression has proven to be a very successful technique. One challenge is to get it to work naturally with edge modification problems. The technique has been extremely helpful for many vertex deletion problems, but few edge modification problems. In the case of Edge Bipartization, one reason for the success of iterative compression is the close relation between the edge version and the vertex version of the problem; there is a parameter-preserving reduction from Odd Cycle Transversal to Edge Bipartization [101]. Edge Bipartization was, however, later shown to be solvable in time $2^k \cdot n^{\mathcal{O}(1)}$ [102], and then in time $1.977^k \cdot n^{\mathcal{O}(1)}$ by Pilipczuk, Pilipczuk, and Wrochna [103].

Kratsch and Wahlström [104] proved that there exists a randomized compression such that Edge Bipartization as well as the vertex version, Odd Cycle Transversal, admits a $k^{4.5}$ co-RP kernel. Here, co-RP allows false positives in the sense that if an instance is a no-instance, then the compressed instance is a no-instance with probability at least 1/2, while any yes-instance will be compressed to a yes-instance. Here, we may boost the success probability by running the algorithm polynomially in k many times (not polynomial in n as that would defeat the purpose of a kernelization procedure), and the output instance will then be the "and" over all the compressed instances. Nevertheless, the question is still open in deterministic settings.

Open problem 2.14 (*[43]*). *Does* Edge Bipartization *admit a deterministic polynomial kernel?*

Finally, let us mention the class of *linear forests*, which are the graphs whose connected components are paths. Though the class is pretty simple, it does not admit a finite number of forbidden subgraphs. Feng, Zhou, and Li showed that Linear Forest Deletion admits a polynomial kernel with $9k$ vertices [105]. They also provided an $\mathcal{O}(2.29^k n^{\mathcal{O}(1)})$ time randomized FPT algorithm for solving the problem.

There are many important hereditary classes for which the parameterized complexity of the edge modification problems is still unknown. Among them are comparability, co-comparability and permutation graphs, which are subclasses of perfect graphs, as well as circular-arc and circle graphs. Obtaining positive or negative results for any of these classes would be of high interest. In particular, the following questions were already asked in the literature.

Open problem 2.15 (*[43]*). *Are any of* Comparability Completion, Comparability Deletion, *and* Permutation Completion *in* FPT*?*

Open problem 2.16 (*[106]*). *Is* Proper Circular Arc Deletion *in* FPT*?*

Open problem 2.17 (*[80,107]*). *Is* Perfect Deletion *in* FPT*?*

2.3. Subexponential time algorithms

As usual in algorithms, a natural, but difficult, question to ask is about lower bounds. The case of parameterized complexity is not different. Once a problem has been shown to admit a fixed-parameter tractable algorithm, a natural next question is whether it is possible to improve upon that algorithm. This is especially interesting when the algorithms have running times that are of the order $2^{\Omega(k^2)} \cdot n^{\mathcal{O}(1)}$, or even $2^{\Omega(2^k)} \cdot n^{\mathcal{O}(1)}$.

As mentioned above, the modification problems for finite forbidden induced subgraphs already have nice running times like $6^k \cdot n^{\mathcal{O}(1)}$ or even $3^k \cdot n^{\mathcal{O}(1)}$ and $2^k \cdot n^{\mathcal{O}(1)}$. Is it possible to obtain faster algorithms? Can we improve from $3^k n^{\mathcal{O}(1)}$ to, say, $2^k n^{\mathcal{O}(1)}$ or $1.5^k n^{\mathcal{O}(1)}$? Are there reasons to suspect that we cannot get better than $2^k \cdot n^{\mathcal{O}(1)}$ algorithms? These questions were at the core of what is known as the *optimality programme* [108].

Simultaneously with the optimality programme and the development of polynomial kernel theory, some problems were shown to be solvable in *subexponential parameterized time*, i.e., in time $2^{o(k)} n^{\mathcal{O}(1)}$, or $(1+\epsilon)^k n^{\mathcal{O}(1)}$ for every $\epsilon > 0$, and there was a strong interest in identifying parameterized problems that admit such subexponential parameterized algorithms. The complexity class of problems admitting such an algorithm is called SUBEPT and was defined by Flum and Grohe in their seminal work on parameterized complexity [18]. They noticed that most natural problems do in fact *not* live in this complexity class: the classical NP-hardness reductions paired with the *Exponential Time Hypothesis* (ETH) of Impagliazzo et al. [23] is enough to show that no $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ algorithm exists.

As the first known subexponential parameterized algorithms were for problems with restricted input graphs, such as planar, or more generally H-minor free graphs [109], Chen posed the following question [110]: are there examples of natural problems on graphs, that do not have such a topologically constrained input, and also admit subexponential parameterized algorithms?

Such a problem was first found by Alon, Lokshtanov, and Saurabh [111] who designed a new algorithmic technique called *chromatic coding* and used it to solve Feedback Arc Set on Tournaments (Fast) on tournament graphs in time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$. A tournament graph is a directed graph obtained from a complete undirected graph by choosing an orientation for each edge, i.e., for every pair of vertices u and v, exactly one of uv and vu is an arc. Then Fast is the problem of identifying at most k arcs in the given tournament whose deletion transforms the tournament into an acyclic graph. An important observation for Fast is that instead of deleting an arc, which would make the graph no longer a tournament, we can reverse the arc. Hence, we want to identify at most k arcs such that reversing these arcs yields an acyclic tournament, i.e., a total ordering of the vertices. Fast is also known to admit a quadratic vertex kernel [112] which was improved to a linear kernel by Bessy et al. [113].

Fomin and Villanger [30] gave an algorithm for Chordal Completion (Minimum Fill-In) using ideas from the techniques developed for minimal triangulations and treewidth computations. Numerous $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ algorithms were known [33,74,114] for Chordal Completion, but Fomin and Villanger proved that this problem is solvable in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{k}\log k)} + k^2 nm)$. The additive polynomial factor was due to first preprocessing the graph, thereby obtaining a kernelized instance of polynomial size. The main tools in this algorithm were that of *minimal triangulations and potential maximal cliques*, a framework developed by Bouchitté and Todinca [115,116], further developed by Fomin, Kratsch, Todinca, and Villanger [117].

Following the results of Fomin and Villanger, several new subexponential parameterized time completion results followed. Based on the chromatic coding technique of Alon et al. [111], Ghosh et al. [72] gave an algorithm with the same running time,[8] $2^{\mathcal{O}(\sqrt{k}\log k)} + n^{\mathcal{O}(1)}$, for Split Completion, thus also giving an algorithm for the equivalent problem of *deleting* to a split graph. A natural question arose again on the complexity of completing to $\mathcal{H}$-free graphs: Could this be subexponential time for all $\mathcal{H}$? for finite $\mathcal{H}$? The result by Lokshtanov [27] again immediately gives a negative result here, as his result implies that for $\mathcal{H}$ being the complement of the wheels, $\mathcal{H}$-free edge Completion, that is co-wheel-free Completion, is W[2]-hard. So for general $\mathcal{H}$, the answer is indeed clearly negative. Therefore, a next question was to look for simple $\mathcal{H}$.

And while the classes of chordal and split graphs are rather "simple", they certainly are much more complex than the simple cluster graphs. Therefore, the problems Cluster Editing and Cluster Deletion were natural candidates for subexponential time algorithms. From Cai's theorem, we immediately obtain $2^k n^{\mathcal{O}(1)}$ and $3^k n^{\mathcal{O}(1)}$ algorithms for Cluster Deletion and Cluster Editing, respectively. This question was first answered in the negative by Komusiewicz and Uhlmann studying this problem on bounded degree graphs [118], and then independently by Fomin et al. [123]. Again somewhat surprisingly, we cannot expect algorithms running in time $2^{o(k)} n^{\mathcal{O}(1)}$ solving Cluster Editing. Komusiewicz and Uhlmann gave an elegant reduction proving that both parameterized and exact subexponential time algorithms are not achievable, unless the exponential time hypothesis fails [118]. They show that under the exponential time hypothesis, neither Cluster Editing nor Cluster Deletion can be solved in time $2^{o(k)} n^{\mathcal{O}(1)}$, in time $2^{o(n)}$, or in time $2^{o(m)}$, even on graphs of maximum degree six.

Following the subexponential algorithm for Chordal Completion and Split Completion, it was shown that Trivially Perfect Completion, as well as Chain Completion, Threshold Completion, and Pseudosplit Completion all were solvable in subexponential parameterized time [31]. They simultaneously give negative results, showing that neither completing to a cograph (and thus also deleting, since the class is self-complementary), deleting to trivially perfect graphs, nor completing to C_4-free or $2K_2$-free graphs are in SUBEPT under ETH. Later, Trivially Perfect Editing [32] and Starforest Deletion[9] [119] were also added to the list of problems that are not in SUBEPT under ETH.

Then followed two results by Bliznets et al. [83,124], that Interval Completion and Proper Interval Completion both are solvable in subexponential time, $2^{\mathcal{O}(\sqrt{k}\log k)} n^{\mathcal{O}(1)}$ and $2^{\mathcal{O}(k^{2/3}\log k)} + n^{\mathcal{O}(1)}$, respectively.

Open problem 2.18 (*[124,125]*)**.** *Does* Proper Interval Completion *admit an algorithm of running time* $2^{\mathcal{O}(\sqrt{k}\log k)} n^{\mathcal{O}(1)}$*?*

Drange et al. gave algorithms for Threshold Editing and Chain Editing running in time $2^{\mathcal{O}(\sqrt{k}\log k)} + n^{\mathcal{O}(1)}$, thereby adding these problems to the line of subexponential parameterized time solvable problems [38]. These two graph classes, threshold and chain graphs, are the only classes known for which all three edge modification problems are NP-complete and solvable in subexponential parameterized time. Drange et al. [31] showed that also for Trivially Perfect Completion, or $\{C_4, P_4\}$-free Completion, as well as for Pseudosplit Completion and Threshold Completion, we have subexponential time algorithms.

Later a problem known as Clique Editing, or Sparse Split Editing was introduced as a model for core/periphery structures [126], and for noise reduction [121]. This problem consists of editing a graph to a disjoint union of a clique and an independent set, or, $\{2K_2, P_3\}$-free edge Editing. The problem was shown to be NP-hard independently by Damaschke and Mogren [121] and Kovác, Selecéniová, and Steinová [127] and is

8 The best complexity known for the problem does not need the $\log k$ factor in the exponent, see Exercise 5.17 in [10].

9 Starforest are the graphs where each connected component is a star, they are also the triangle-free trivially perfect graphs.

Table 2
Subexponential parameterized complexity of edge modification problems into hereditary graph classes characterized by a finite number of forbidden induced subgraphs. We provide the best known exponential bound in terms of the parameter k and omit the polynomial part. All problems are FPT by [33]. NOSUB means that there is no parameterized subexponential algorithm (of course up to some complexity assumption). OPEN means that the complexity is open, while "—" means that the problem is probably open but most likely nobody looked at this question. P means the problem is solvable in polynomial time. A dagger next to the name of the class marks self-complementary classes, for which any result for one of the completion problem or deletion problem automatically gives the same result for the other problem.

Graph class	Subexponential time algorithms		
	Completion	Deletion	Editing
line	OPEN	OPEN	OPEN
s-plex cluster	—	—	$(2s+\sqrt{s})^k$ [36], NOSUB [118]
chain ($\{K_3, 2K_2, C_5\}$)	as deletion	SUBEPT $2^{\sqrt{k}\log k}$ [31]	SUBEPT $2^{\sqrt{k}\log k}$ [38]
starforest ($\{K_3, C_4, P_4\}$)	P	NOSUB [119]	as deletion
threshold† ($\{2K_2, C_4, P_4\}$)	SUBEPT $2^{\sqrt{k}\log k}$ [31], NO $2^{k^{1/4}}$ [120]	SUBEPT $2^{\sqrt{k}\log k}$ [31], NO $2^{k^{1/4}}$ [120]	SUBEPT $2^{\sqrt{k}\log k}$ [38]
split† ($\{2K_2, C_4, C_5\}$)	SUBEPT $2^{\mathcal{O}(\sqrt{k})}$ [10]	SUBEPT $2^{\mathcal{O}(\sqrt{k})}$ [10]	P [9]
clique + IS ($\{P_3, 2K_2\}$)	P	SUBEPT $1.6355^{\sqrt{k\ln k}}$ [121]	SUBEPT $2^{\sqrt{k\ln k}}$ [121]
trivially perfect ($\{C_4, P_4\}$)	SUBEPT $2^{\sqrt{k}\log k}$ [31], NO $2^{k^{1/4}}$ [120]	2.42^k [71], NOSUB [31]	NOSUB [32]
$\{claw, diamond\}$	OPEN	NOSUB [42]	—
pseudosplit† ($\{2K_2, C_4\}$)	SUBEPT $2^{\mathcal{O}(\sqrt{k})}$ [10,31]	SUBEPT $2^{\mathcal{O}(\sqrt{k})}$ [10,31]	P [9,43]
cluster ($\{P_3\}$)	P	1.41^k [66], NOSUB [118]	1.76^k [66], NOSUB [118]
$\{K_3\}$	P	NOSUB [122]	as deletion
cograph† ($\{P_4\}$)	2.56^k [57], NOSUB [31,32]	2.56^k [57], NOSUB [31,32]	4.61^k [58], NOSUB [32]
$\{paw\}$	NOSUB [122]	NOSUB [122]	NOSUB [122]
$\{diamond\}$	P	NOSUB [49,122]	NOSUB [122]
$\{claw\}$	NOSUB [122]	NOSUB [122]	NOSUB [122]
$\{K_4\}$	P	NOSUB [122]	as deletion
$\{P_\ell\}$, $\ell > 4$	NOSUB [122]	NOSUB [122]	NOSUB [122]
$\{C_\ell\}$, $\ell > 3$	NOSUB [122]	NOSUB [122]	NOSUB [122]

solvable in subexponential time. Indeed, a polynomial kernel is quite trivial after a twin reduction rule, and then the result follows from guessing a vertex in the clique and a (small) number of other vertices with which its adjacency relationships have to be changed. Damaschke and Mogren showed that several similar problems are solvable in subexponential parameterized time and they showed that Clique Deletion is solvable in time $\mathcal{O}(1.6355^{\sqrt{k\ln k}}n^{\mathcal{O}(1)})$ [121].

Using the H-Bag Editing problem from Damaschke and Mogren [121], Meesum, Misra, and Saurabh [128] showed that the r-Rank Reduction Editing problem is solvable in time $2^{\mathcal{O}(\sqrt{k}\log k)}\cdot n^{\mathcal{O}(1)}$. In this problem, we are asked to edit the input graph G to a graph G' by modifying at most k edges so that rank($A_{G'}$), the rank of the adjacency matrix of G' is at most r. A similar result is shown by Meesum and Saurabh [129] for the directed case.

There are still some graph classes for which the question of whether the edge modification problems admit a subexponential algorithm is not entirely settled. In particular, the case of triangle free graphs and 3-leaf powers is appealing since there exist some polynomial kernels for these problems.

Open problem 2.19. *Do the following problems admit subexponential parameterized algorithms:* Triangle-free Deletion, Linear Forest Deletion, Planar Deletion, *and* 3-Leaf Powers Completion?

As for the lower computational bounds, many problems are known not to be solvable in time $2^{o(k)}n^{\mathcal{O}(1)}$, that is they do not admit subexponential parameterized algorithms, under some complexity hypothesis such as P $\neq$ NP or ETH or NP $\not\subseteq$ coNP/poly, see Tables 2 and 4. For many problems the question of obtaining lower bounds on the subexponential complexity is open.

Fomin and Villanger [30] noted that, unless ETH fails, Chordal Completion cannot be solved in time $2^{o(k^{1/6})}n^{\mathcal{O}(1)}$. Later, Bliznets et al. [120] showed that this can be tightened quite a bit: unless ETH fails, there is a positive natural number $c > 1$ such that Chordal Completion cannot be solved in time $2^{\mathcal{O}(k^{1/4}/\log^c k)}n^{\mathcal{O}(1)}$, and the same lower bound holds for Interval Completion, Proper Interval Completion, Trivially Perfect Completion, Threshold Completion (and so Threshold Deletion since the class is self-complementary). This, however, still leaves a gap for almost all the problems between $k^{1/2}$ and $k^{1/4}$ in the exponent. Is the correct running times for these problems closer to $2^{\mathcal{O}(k^{1/4}/\log^c k)}n^{\mathcal{O}(1)}$, to $2^{\mathcal{O}(k^{1/2})}+n^{\mathcal{O}(1)}$ or to $2^{\mathcal{O}(\sqrt{k}\log k)}+n^{\mathcal{O}(1)}$? For chordal graphs, we know that the exponent $1/2$ of k is optimal as it was shown by Cao and Sandeep [130] (again up to ETH). Therefore, the only open question is on the optimality of the $2^{\mathcal{O}(k^{1/2}\log k)}$. For Proper Interval Completion the gap on the exponent of k is larger than for the other problems cited above since we only know an algorithm running in time $k^{\mathcal{O}(k^{2/3})}+n^{\mathcal{O}(1)}$ [124].

Table 3
Kernelization complexity of edge modification problems into hereditary graph classes whose number of minimal forbidden induced subgraph is infinite. OPEN means that the complexity is open. P means that the problem is solvable in polynomial time.

Graph class	Polynomial kernel		
	Completion	Deletion	Editing
linear forest	P	$9k$ [105]	as deletion
distance-hereditary	OPEN	OPEN	OPEN
planar	P	OPEN	as deletion
H-minor free	P	OPEN	as deletion
bipartite	P	k^3 [104][a]	as deletion
3-leaf power	k^3 [89]	k^3 [89]	k^3 [89]
4-leaf power	OPEN	OPEN	OPEN
proper interval	k^3 [37]	OPEN	OPEN
interval	OPEN	OPEN	OPEN
strongly chordal	OPEN	OPEN	OPEN
chordal	k^2 [75]	OPEN	OPEN

[a]The kernel for deleting to bipartite graphs, Edge Bipartization, is a co-RP kernel, and the size is the number of bits in the representation up to a polylogarithmic factor.

Table 4
Subexponential parameterized complexity of edge modification problems into hereditary graph classes whose number of minimal forbidden induced subgraphs is infinite. NOSUB means that there is no parameterized subexponential algorithm (of course up to some complexity assumption). OPEN means that the complexity is open, while "—" means that the problem is probably open but most likely has not been studied. P means that the problem is solvable in polynomial time.

Graph class	Subexponential parameterized complexity		
	Completion	Deletion	Editing
linear forest	P	NOSUB[a]	as deletion
distance-hereditary	—	NOSUB[a]	NOSUB[a]
planar	P	OPEN	as deletion
H-minor free	P	OPEN	as deletion
bipartite	P	NOSUB[a]	as deletion
3-leaf power	OPEN	NOSUB[a]	NOSUB[a]
4-leaf power	—	—	—
proper interval	SUBEPT $2^{\mathcal{O}(k^{2/3})\log k}$ [124] NO $2^{k^{1/4}}$ [120]	OPEN	OPEN
interval	SUBEPT, $2^{\sqrt{k}\log k}$ [83] NO $2^{k^{1/4}}$ [120]	OPEN	OPEN
strongly chordal	OPEN	OPEN	OPEN
chordal	SUBEPT $2^{\sqrt{k}\log k}$ [30] NO $2^{\sqrt{k}}$ [130]	OPEN	OPEN

[a]For Linear Forest Deletion the subexponential lower bound follows from reduction from Hamiltonicity. For Edge Bipartization the result is folklore. For 3-Leaf Deletion and Editing, the lower bound follows from the lower bounds for clustering. For Distance-Hereditary Deletion and Editing, the lower bounds follow from the lower bounds on Cograph Deletion and Editing.

Table 5
Parameterized complexity of edge modification problems into hereditary graph classes whose number of minimal forbidden induced subgraph is infinite. OPEN means that the complexity is open. P means that the problem is solvable in polynomial time.

Graph class	Parameterized complexity (best known)		
	Completion	Deletion	Editing
linear forest	P	2.29^k [105][a]	as deletion
distance-hereditary	FPT [94]	FPT [94]	FPT [94]
planar	P	FPT [26,97]	as deletion
H-minor free	P	FPT [97]	as deletion
bipartite	P	1.977^k [103]	as deletion
3-leaf power	FPT [88]	FPT [88]	FPT [88]
4-leaf power	FPT [90,91]	FPT [90,91]	FPT [90,91]
proper interval	$2^{\mathcal{O}(k^{2/3})\log k}$ [124]	FPT [131]	FPT [131]
interval	$2^{\sqrt{k}\log k}$ [83]	$2^{\mathcal{O}(k)\log k}$ [82]	OPEN
strongly chordal	64^k [74]	OPEN	OPEN
chordal	$2^{\sqrt{k}\log k}$ [30]	$2^{\mathcal{O}(k\log k)}$ [78]	$2^{\mathcal{O}(k\log k)}$ [78]

[a]For Linear Forest the FPT algorithm is randomized.

There were also some attempts to obtain general results about the (non-)existence of subexponential parameterized algorithms for edge modification problems into H-free graph classes [122]. These are results of impossibility: when H has at least two edges (resp. non-edges), H-free edge deletion (resp. completion) is NP-complete and not in SUBEPT; when H has at least three vertices, H-free edge editing is NP-complete and not in SUBEPT.

Theorem 2.3 *([122]). Let $\mathcal{G}$ be a hereditary class of graphs characterized by graph H. Then unless ETH fails,*

- *If H has fewer than two edges, then $\mathcal{G}$-Edge Deletion is solvable in polynomial time. Otherwise, the problem cannot be solved in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ unless ETH fails.*
- *If H has fewer than two non-edges, then $\mathcal{G}$-Edge Completion is solvable in polynomial time. Otherwise, the problem cannot be solved in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ unless ETH fails.*
- *If H has fewer than three vertices, then $\mathcal{G}$-Edge Editing is solvable in polynomial time. Otherwise, the problem cannot be solved in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ unless ETH fails.*

Even more recently, such kind of results where extended to the question of the existence of approximation algorithms [132]: when H is 3-connected and has at least two non-edges, then there does not exist any poly(OPT)-approximation algorithm running in parameterized subexponential time (in OPT), unless ETH fails, for H-free edge deletion as well as H-free edge completion. Moreover, the same holds for H being a cycle on at least 4 vertices or a path on at least 5 vertices. With previous results, this solves all cases of paths and cycles except the cograph edge deletion problem, for which [132] suggests the existence of a parameterized subexponential approximation algorithm, because of the existence of a kernel to the problem [28].

Among the most interesting open questions in the topic of subexponential parameterized algorithms is to explain why some problems admit such algorithms, which we saw is an exceptional case. In particular, one can ask whether the existence of a polynomial kernel is a prerequisite for an $\mathcal{H}$-free Modification problem to admit a subexponential time algorithm. Actually, there are examples of problems that admit subexponential time algorithms and that do not have polynomial kernels under the assumption of NP $\not\subseteq$ coNP/poly, but these problems are not of the $\mathcal{H}$-free Modification type. Indeed, it is easy to come up with problems that trivially or-cross-composes, like the or-Minimum Fill-In which asks whether a graph has a connected component that can be completed to a chordal graph. This problem cannot have a polynomial kernel under NP $\not\subseteq$ coNP/poly, but does admit a subexponential time algorithm, simply by running the algorithm by Fomin and Villanger [30] for each connected component. However, the or-Minimum Fill-In problem is not of the $\mathcal{H}$-free Modification type and it turns out that for all such problems that we know to admit a subexponential time algorithm, we also have polynomial kernels—with the possible exception of Interval Completion for which existence of a polynomial kernel remains open.

Open problem 2.20. *Does there exist an $\mathcal{H}$ for which $\mathcal{H}$-free Completion or $\mathcal{H}$-free Editing is solvable in time $2^{o(k)}n^{\mathcal{O}(1)}$ but does not admit a polynomial kernel unless* NP $\subseteq$ coNP/poly*?*

Finally, another important question to address is about the tools used to show lower bounds. Many results are established on complexity hypotheses that are not as reliable as the assumption that P $\neq$ NP, such as, for example, the assumption that NP $\not\subseteq$ coNP/poly. It would be highly desirable to develop the necessary techniques to find more of these results on the sole hypothesis that P $\neq$ NP. Some exist already, for example, Chen, Flum, and Müller [133] showed that a rather artificial problem, Rooted Path, the problem of finding a simple path of length k in a graph that starts from a prespecified root vertex, parameterized by k has no strict polynomial kernel unless P $=$ NP. In this case, *strict polynomial kernel* is a polynomial kernel that does not increase the parameter value k. Building on the aforementioned work, Fernau et al. [134] showed that several other problems do not admit polynomial kernels under the same assumption. Their result include (but is not limited to) Multicolored Path parameterized by the length of the path, Clique and Biclique parameterized by maximum degree Δ, as well as treewidth tw, Colorful Graph Motif, and several problems relating to short NTM computation. We refer the reader to the article for the full list of problems.

Open problem 2.21. *Which lower bounds for edge modification problems, and graph modification in general, can be shown assuming only* P $\neq$ NP*?*

2.4. Related results

In some cases, the input graph is naturally a bipartite graph. The Chain Completion problem is the problem of making a bipartite graph a bipartite chain graph, that is, a bipartite graph with no induced $2K_2$. The problem was first shown to admit a polynomial kernel by Guo [53] when the bipartition is fixed. Fomin and Villanger showed that this version of the problem admits a subexponential time algorithm [30] and Bliznets et al. [120] showed that it cannot be solved in time $\mathcal{O}(2^{k^{1/4}})$ unless ETH fails. Drange et al. [38] relaxed the input requirements, showing that the problem still admits a quadratic kernel even when the bipartition is not fixed.

In the Minimum Flip Consensus Tree problem, we are asked to turn an input bipartite graph into a bipartite graph with same partition that contains no P_5 starting from any top vertex, called a *consensus tree*. This kind of graph, a consensus tree, arises in computational phylogenetics, with the bottom vertices being characters and the top vertices being the taxa. The problem is solvable in time $c^k n^{\mathcal{O}(1)}$ by Cai's theorem. Chen [135] proved that it is NP-complete and gave an $\mathcal{O}(6^k n^2)$ FPT algorithm, which was later improved to $\mathcal{O}(4.42^k n)$ by Böcker, Bui, and Truß [136]. Finally, Komusiewicz and Uhlmann [137] gave a $\mathcal{O}(3.68^k n^3)$ algorithm and a $\mathcal{O}(k^3)$ kernel for Minimum Flip Consensus Tree.

Several variants of cluster editing were introduced for the special case of bipartite graphs. The main one, called Bicluster Editing, aims at obtaining a union of complete bipartite graphs. It admits a $4k$ linear kernel [138] and an FPT algorithm running in $\mathcal{O}(2.636^k + n^{\mathcal{O}(1)})$ [139]. Drange et al. [119] considered the extension of p-Cluster Editing to Bicluster Editing and the more general t-Partite Cluster Editing, yielding the problems p-Bicluster Editing and t-Partite p-Cluster Editing. None of the classical parameterized versions are solvable in subexponential time, but fixing the number p of connected components in the solution, the problems become solvable in subexponential time. In [119], it is shown that a problem called p-Starforest Editing is solvable in time $\mathcal{O}(2^{3\sqrt{pk}} + m + n)$, whereas an algorithm of running time $2^{\mathcal{O}(p\sqrt{k}\log(pk))} + \mathcal{O}(m + n)$ is given for p-Bicluster Editing, as well as t-Partite p-Cluster Editing.

Let us also mention that for planar input graphs, Xia and Zhang [70] gave a linear kernel for 5-cycle Transversal and $_\leq$5-cycle Transversal, thereby showing that $\{C_3, C_4, C_5\}$-free Deletion, or Girth-6 Deletion admits a linear kernel on planar graphs. A problem in the area of *graph drawing* relating to both bipartite graphs and planar graphs is studied by Fernau [140]. A 2-layer drawing of a bipartite graph is a drawing where the vertices in one side are positioned on a line in the plane, which is parallel to another line containing the vertices in the other side, and the edges are drawn as straight line-segments. A biplanar graph is then a bipartite graph that admits a 2-layer drawing with no edge crossings. The two problems studied by Fernau are 2-Layer Planarization: whether k edges can be deleted from a given graph G so that the remaining graph is biplanar, and 1-Layer Planarization: whether k edges can be deleted from a given graph G so that the remaining graph is biplanar when the ordering of the first side is fixed. The problems can be solved in time $\mathcal{O}(k^2 \cdot 5.1926^k + n)$ and $\mathcal{O}(k^3 \cdot 2.5616^k + n^2)$, respectively, and the latter admits a cubic kernel [140]. Fernau et al. [141] showed that a related problem, p-One-Sided Crossing Minimization admits a subexponential time algorithm, which was later improved to $\mathcal{O}(k2^{\sqrt{k}} + n)$ by Kobayashi and Tamaki [142].

Open problem 2.22. *Do* 1-Layer Planarization *and* 2-Layer Planarization *admit subexponential time algorithms?*

We refer to the review on crossing minimization by Zehavi [143] for recent results in this area.

A non-hereditary variant of Edge Bipartization is edge deletion toward Kőnig graphs. A graph is a Kőnig graph if its vertex cover number is equal to the size of its maximum matching. This class contains all bipartite graphs, but not every Kőnig graph is bipartite; for example, a triangle with a pendant is Kőnig. The problem Kőnig Deletion, whose vertex deletion version was known to be FPT [144], was asked in the open problem set of the FPT school of 2014 [145] and was shown to be W[1]-hard by Majumdar et al. [146].

To alleviate the hardness, Majumdar et al. introduce a variation where, given a graph G with a maximum matching M, we instead are given the task of making G into a Kőnig graph by deleting edges that are not in M. The problem Kőnig Deletion Disjoint from Matching is reducible to Almost 2-Sat and thus admits a kernel (ibid.).

They conclude by raising the following question: A graph is *stable* if its maximum matching has size equal to the maximum fractional matching.[10] Vertex deletion to stable graphs is solvable in polynomial time whereas the edge deletion version is NP-complete [147].

Open problem 2.23 *([146]). Is* Stable Deletion, *the problem of deleting at most k edges to obtain a stable graph, in* FPT?

3. Connectivity, cuts, and clustering

In this section we consider problems around edge cuts and connectivity augmentations. By cut problems here we mean a wide class of problems where one wants for a given (directed) graph G to identify a minimum-sized set of edges X (edge-cut) such that in the new graph $G - X$ obtained by deleting X from G, some connectivity conditions change. For example, the condition can be that a set of specific terminals becomes separated or that at least one connected component in the new graph is of a certain size. Clustering problems can be seen as a hybrid of connectivity and cuts, where we want to identify highly connected areas of a graph that can be easily cut from each other. Most of these problems are NP-hard, except several notable exceptions, like Minimum s-t Cut and Minimum Multiway Cut in planar graphs with fixed number of terminals. Several interesting algorithmic techniques were developed in order to establish fixed-parameter tractability of various cut problems.

The "dual" set of problems is that of adding edges in order to augment some connectivity properties of the graph.

[10] A fractional matching is the optimum value of the maximum matching LP.

3.1. Cuts

Edge Multiway Cut. In the Edge Multiway Cut problem, we are given a graph G, a set $T \subseteq V(G)$ of terminal vertices, and an integer k. The task is to decide whether there exists a set X of at most k edges of G such that every element of T lies in a different connected component of $G - X$.

Edge Multiway Cut

Input: Graph G, $T \subseteq V(G)$, and integer k
Task: Does there exists a set X of at most k edges of G such that every element of T lies in a different connected component of $G - X$.

A related problem is Vertex Multiway Cut, where one wants to delete at most k vertices to separate terminals. For most of the variants of the cut problems, an FPT-algorithm for the edge deletion version can be obtained from the vertex deletion variant. This is why most of the work in the area was concentrated on vertex deletions.

For $|T| = 2$, Edge Multiway Cut is the classical Min-Cut and is solvable in polynomial time due to its duality with the maximum flow problem [148]. However, as it was shown by Dalhaus et al. [149], Edge Multiway Cut is NP-complete for $|T| = 3$.

In influential paper [150], Marx established fixed-parameter tractability of Edge Multiway Cut and Vertex Multiway Cut parameterized by k. For that, Marx developed the technique of important separators based on submodular properties of cuts. The technique appeared to be handy for many problems in this area. Algorithms for Edge Multiway Cut with running times $2^k \cdot n^{\mathcal{O}(1)}$ and $1.84^k \cdot n^{\mathcal{O}(1)}$ were given by Xiao [151] and Cao, Chen, and Fan [152], correspondingly. Chapter 8 of the textbook on parameterized algorithms [10] contains an overview of basic techniques around important separators and parameterized algorithms for finding cuts in graphs.

Edge Multiway Cut remains NP-complete on planar graphs but as it was shown by Dalhaus et al. [149], for a fixed number of terminals, the problem can be solved in time $n^{\mathcal{O}(|T|)}$ on planar graphs. The running time for planar graphs was improved to $2^{\mathcal{O}(|T|)} \cdot n^{\mathcal{O}(\sqrt{|T|})}$ by Klein and Marx [153].

Lokshtanov and Ramanujan [154] studied the version of Vertex Multiway Cut called Parity Multiway Cut. Here the terminal set T consists of two not necessarily disjoint subsets T_o and T_e. The objective is to decide whether there exists a k-sized vertex (or edge) subset S such that S intersects all odd-length paths from $v \in T_o$ to $T - v$ and all even-length paths from $v \in T_e$ to $T - v$. The edge deletion case with $T_o = T_e$ is exactly Edge Multiway Cut. Lokshtanov and Ramanujan proved that both edge- and vertex deletion versions of Parity Multiway Cut are FPT parameterized by k. Chandrasekaran and Mozaffari studied parity variants of these problem on directed acyclic graphs [155].

The random sampling of important separators technique developed by Lokshtanov and Ramanujan was generalized to directed graphs by Chitnis, Hajiaghayi, and Marx [156] who showed the fixed-parameter tractability of Directed Edge Multiway Cut and Directed Vertex Multiway Cut parameterized by the size of the solution.

The technique based on important separators was used by Chen et al. [157] in their FPT algorithm for Directed Feedback Vertex Set and Directed Feedback Arc Set, whose parameterized complexity was open for a long time. In these problems the task is to decide whether at most k vertices (respectively, arcs) can be removed from a directed graph such that the resulting graph is acyclic. It should be mentioned that Directed Feedback Arc Set and Directed Feedback Vertex Set are equivalent in the sense that they can be reduced to each other under the same parameter, as observed by Even et al. [158].

The generalization of the problem, namely Directed Subset Feedback Vertex Set, was studied by Chitnis et al. [159]. Xiao and Nagamochi gave an FPT algorithm for Subset Feedback Arc Set [160]. Kratsch et al. [161] define multi-budgeted variants of Directed Feedback Arc Set and some versions of Min-Cut and establish fixed-parameter tractability of these problems. The existence of a polynomial kernel for Directed Feedback Vertex Set and for Directed Feedback Arc Set is widely open.

Open problem 3.1. *Do* Directed Feedback Vertex Set *and* Directed Feedback Arc Set *admit a polynomial kernel?*

Lucchesi and Younger [162] proved that on planar graphs Directed Feedback Arc Set is solvable in polynomial time.

Open problem 3.2. *Does* Directed Feedback Vertex Set *admit a polynomial kernel on planar graphs?*

Whether the running time $k^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ of Chen et al. [157] for Directed Feedback Vertex Set is tight, is another open question.

Open problem 3.3. *Could* Directed Feedback Vertex Set *and* Directed Feedback Arc Set *be solved in time* $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$*? Could* Directed Feedback Vertex Set *be solved in time* $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ *on planar graphs?*

Edge Multicut. In the related Edge Multicut problem, we are given a graph G, a set of pairs $(s_i, t_i)_{i=1}^{\ell}$ of vertices of G, and an integer k. The question is if there exists a set X of at most k edges of G such that for every $1 \le i \le \ell$, vertices s_i and t_i lie in different connected components of $G - X$. The vertex deletion version of the problem is Vertex Multicut.

Edge Multicut is NP-hard on trees [163]. Guo and Niedermeier [164] obtained a $2^k \cdot n^{\mathcal{O}(1)}$-time algorithm for the problem on trees. For general graphs, the fixed-parameter tractability of Edge Multicut and Vertex Multicut parameterized by the solution size k was a long-standing open question, which was resolved independently by Bousquet, Daligault, and Thomassé [165] and Marx and Razgon [166].

On general directed graphs Edge Multicut is FPT parameterized by k for the special case with two terminal pairs (s_1, t_1), (s_2, t_2) [156] and is W[1]-hard for four terminal pairs, as show by Pilipczuk and Wahlström [167]. The complexity of the case with three terminal pairs is open.

Open problem 3.4 (*[167]*)**.** *What is the parameterized complexity of* Edge Multicut *on directed graphs when the three pairs of terminal sets* $(s_i, t_i)_{i=1}^{3}$ *parameterized by the cut-size k?*

Kratsch et al. [168] proved that Edge Multicut is FPT parameterized by k and $|T|$ on directed acyclic graphs and that it remains W[1]-hard parameterized by k even on DAGs. Chitnis and Feldmann [169] studied FPT inapproximability of Edge Multicut on directed graphs. The following question about kernelization of Edge Multicut is open.

Open problem 3.5 (*[145]*)**.** *Does* Edge Multicut *admit a polynomial kernel on directed acyclic graphs, when parameterized by k and* $|T|$*? Or when parameterized by k and when the number of terminal pairs is constant?*

Bringmann et al. [170] provide a detailed study of the following generalization of Edge Multicut. In the Steiner Multicut we are given an undirected graph G, a collection $\mathcal{T} = \{T_1, \dots, T_t\}$, $T_i \subseteq V(G)$, of terminal sets of size at most p, and an integer k. The task is to decide whether there is a set S of at most k edges such that of each set T_i at least one pair of terminals is in different connected components of $G - S$. Edge Multicut is the special case for $p = 2$.

The parameterized complexity of a variant of the cut problem called Length-Bounded Edge-Cut (delete at most k edges such that the resulting graph has no $s-t$ path of length shorter than ℓ) was studied by Golovach and Thilikos [171]. They showed that Length-Bounded Edge-Cut is in FPT for the combined parameter $k + \ell$. Fluschnik et al. [172] proved that it is unlikely to admit a polynomial kernel in $k + \ell$ even when the input graph is planar. When it concerns structural parameterized complexity, Dvorák and Knop [173] showed that the problem is W[1]-hard when parameterized by the pathwidth and is fixed-parameter tractable when parameterized by the treedepth of the input graph. Bazgan et al. [174] provided an XP algorithm for the parameter Δ, the maximum degree of the input graph G, and an FPT algorithm for the feedback edge number. Bentert, Heeger, and Knop [175] prove W[1]-hardness for the combined parameter pathwidth and maximum degree Δ of the input graph. They also prove that Length-Bounded Edge-Cut is W[1]-hard for the feedback vertex number.

Kolman [176] showed that Length-Bounded Edge-Cut is FPT on planar graphs when parameterized by ℓ. The parameterized complexity of the problem with parameter k on planar graphs is open.

Open problem 3.6. *What is the parameterized complexity of* Length-Bounded Edge-Cut *when the input graph G is planar and the parameter is the cardinality of the cut k?*

The problems Metric Violation Distance and Metric Repair, which generalizes both Edge Multicut and Length-Bounded Edge-Cut, were studied by Fan et al. [177–179]. In these problems we are given a weighted graph with the goal of finding a small set of edges that can be changed to make the graph *metric*, i.e., there does not exist an edge uv such that $w(uv) > w(P_{u,v})$, where the latter is the cost of the cheapest path from u to v.

Constrained cuts. Here we collected the results on the problems of the following type: is it possible to delete at most k edges from the graph such that some of the required constraints like on the size of a connected component or on the number of connected components hold. For a vertex set $X \subseteq V(G)$, we denote by $\partial(X)$ the set of edges between X and $V(G) \setminus X$.

A general framework for defining constrained cuts was suggested by Lokshtanov and Marx [180]. Let μ be a function that assigns a non-negative integer to each subset of vertices in the graph. Following the notation of Lokshtanov and Marx, we say that a vertex set $X \subseteq V(G)$ is a *(μ, p, q)-cluster*, if $|\partial(X)| \le q$ and $\mu(X) \le p$. For example, if $\mu(X)$ is the number of non-edges in the subgraph induced by X, then $(\mu, 0, q)$-cluster is a clique which can be cut from the graph by at most q edges.

Then in the (μ, p, q)-Cut problem, for a given graph G the task is to identify whether G contains a (μ, p, q)-cluster. In the Terminal (μ, p, q)-Cut problem, we are given graph G and vertex v, the task is to decide whether there is a (μ, p, q)-cluster containing v.

Lokshtanov and Marx proved that Terminal (μ, p, q)-Cut is solvable in time $2^{\mathcal{O}(q)}n^{\mathcal{O}(1)}$ (p being a part of the input) and in time $2^{\mathcal{O}(p)}n^{\mathcal{O}(1)}$ (q being a part of the input) for the following important special cases

- $\mu(X)$ is the number of non-edges in the subgraph induced by X;
- $\mu(X)$ is the maximum degree of $\overline{G}[X]$, the complement of the graph induced by X;
- $\mu(X)$ is the number of vertices of X.

Let us note that for each of the above cases the TERMINAL (μ, p, q)-CUT problem is NP-complete when both p and q are part of the input [181]. An FPT algorithm for TERMINAL (μ, p, q)-CUT trivially implies an FPT algorithm for (μ, p, q)-CUT; we just try all possible terminal vertices.

Open problem 3.7. *What is the parameterized complexity of the weighted versions of (μ, p, q)-CUT when parameterized by p and by q and function $\mu(X)$ being the number of non-edges in the subgraph induced by X and the maximum degree of $\overline{G}[X]$?*

Open problem 3.8. *What is the parameterized complexity of deciding for given graph G and integers p and q, if G contains a set of vertices X such that $|X| = p$ and $|\partial(X)| \leq q$ with parameter p or q?*

The related BISECTION problem, the problem of separating a graph into two equally large graphs cutting at most k edges, was shown to be solvable in time $2^{\mathcal{O}(k^3)} \cdot n^{\mathcal{O}(1)}$ by Cygan et al. [182]. The incompressibility of the problem was shown by van Bevern et al. [183].

Lokshtanov and Marx [180] also show that when μ is monotone, then the solution for TERMINAL (μ, p, q)-CUT can in polynomial time be transformed into a solution of the following (μ, p, q)-PARTITION problem. In this problem we are given a graph G and the task is to decide whether there is a partition of the vertex set $V(G)$ into (μ, p, q)-clusters. In particular, since for every monotone polynomial time computable function μ, TERMINAL (μ, p, q)-CUT is solvable in time $n^{\mathcal{O}(q)}$ by a brute-force algorithm trying all cuts of size at most q, this yields that (μ, p, q)-PARTITION is solvable in time $n^{\mathcal{O}(q)}$.

Kim et al. [184] considered a related problem, under name MIN–MAX MULTIWAY CUT, where we are given a graph G, a non-negative integer ℓ, and a set T of terminals, the question is whether we can partition the vertices of G into $|T|$ parts such that (a) each part contains one terminal and (b) there are at most ℓ edges with only one endpoint in this part. They gave an algorithm solving this problem in time $2^{\mathcal{O}((\ell|T|)^2 \log \ell|T|)} n^4 \log n$.

An interesting variant of EDGE MULTICUT was introduced by Chitnis, Egri, and Marx [185] under the name CHAIN SAT. In the graph version, the problem can be phrased as follows. The input to CHAIN SAT is a directed graph G, an integer k, and two vertices s and t. The edges of G are partitioned into sets $E_1, \ldots, E_m$, called *bundles*, such that each E_i is an edge-set of a path. If every such path has length at most ℓ, we call the problem ℓ-CHAIN SAT. The goal is to find an $s - t$-cut that intersects at most k bundles.

The parameterized complexity of the multi-budgeted variant of EDGE MULTICUT, where arcs are colored and the required cut should contain a certain amount of arcs of each color, was investigated by Kratsch et al. [161]. Chitnis et al. [185] conjectured that for every fixed $\ell \geq 1$, ℓ-CHAIN SAT is FPT parameterized by k. Using a new technique called *directed flow-augmentations*, Kim et al. [186] gave a randomized FPT algorithm for a more general version of CHAIN SAT (under the name BUNDLED CUT WITH ORDER) parameterized by $k + \ell$. Their algorithm also works in the weighted setting, where each bundle E_i has a weight $w : \{E_1, \ldots E_m\} \to \mathbb{N}$ and the $s - t$-cut should intersect bundles summing up to at most some total weight bound W.

Another problem of cutting a graph is MINIMUM K-WAY CUT OF BOUNDED SIZE, where we are given graph G and integers k and s. The task is to decide whether there is a set of at most s edges X such that $G - X$ has at least k connected components. Downey et al. [187] prove that the problem parameterized by k is W[1]-hard. Kawarabayashi and Thorup [188] show that the problem is fixed parameter tractable when parameterized by s.

A *matching cut* is an edge cut that is a matching. MATCHING CUT is fixed-parameter tractable parameterized by the size of the solution. This result, as well as tractability of cut problems with other various constraints, follows from the work of Marx, O'Sullivan, and Razgon [189]. Kernelization algorithms for various structural parameterization of MATCHING CUT and its generalization are further studied by Komusiewicz, Kratsch, and Bang Le [190] and Gomes and Sau [191].

Vulnerability measures. Before proceeding to connectivity related problems, we will briefly touch upon a class of problems relating to both cuts and connectivity, namely the class of *graph vulnerability measures*. A graph vulnerability measure is a problem which concerns itself about how badly communication in a network is disrupted when we remove edges (or vertices); in that way, it can be seen to generalize some cut and connectivity problems.

Vulnerability comes in many forms, the canonical vulnerability measure is called GRAPH INTEGRITY (sometimes VERTEX INTEGRITY), which asks for the relation between the size of a separator and the resulting components. However, as we are mainly concerned with edge modification, we only investigate the edge versions here. We note that several of these problems make sense in a weighted setting [192], where we look at the *weight* of a connected component (i.e., sum of weights) rather than its size.

Let $\mathrm{mc}(G)$ for a graph G to be the number of vertices in the largest connected components of G, $\omega(G)$ the number of connected components of G, and $\tau(G)$ to mean the number of edges in the largest component. The fundamental problem for parameterized analysis is the COMPONENT ORDER CONNECTIVITY [193] problem and specifically the COMPONENT ORDER EDGE CONNECTIVITY [194]:

COMPONENT ORDER EDGE CONNECTIVITY

Input: $G = (V, E)$, k, ℓ

Task: Does there exist a set of edges $F \subseteq E$ of size at most k such that if we remove F from the graph, the size of the largest component is at most ℓ, or using the above notation, $\mathrm{mc}(G - F) \leq \ell$?

Lemma 3.1. COMPONENT ORDER EDGE CONNECTIVITY *is* FPT *parameterized by $k + \ell$.*

Proof. We immediately get that the problem is (non-uniformly) FPT by Cai [33] if we take all connected graphs on $\ell + 1$ vertices as forbidden induced subgraphs. However, there is a quite simple faster algorithm:

Given a graph G on m edges together with integers k and ℓ, build a graph $G_0, G_1, \ldots, G_m$ where we introduce edges one by one. We call this the *introducing mode* of the algorithm. We alternate between the introducing mode and the *branching mode*. The branching mode works as follows. If $\mathrm{mc}(G_i) \leq \ell$, we do nothing and go back to the introducing mode. Otherwise, we know that $\ell < \mathrm{mc}(G_i) \leq 2\ell$, and that there is only one component with more than ℓ vertices. We branch into $2^{2\ell}$ branches, one for each vertex subset. For each subset, we decrease k accordingly, and continue in the branching mode. The correctness of the algorithm is immediate and its running time is $f(\ell)^k \cdot n^{\mathcal{O}(1)}$. □

We can also mention here that directly from this algorithm, it follows that the problem where we would like have at most ℓ *edges* in each connected component is FPT as well, i.e., get $\tau(G') \leq \ell$ by removing at most k edges.

There is much work that remains to be done in the parameterized area with regards to vulnerability measures. The only other vulnerability studied in a parameterized sense is the vertex version of GRAPH INTEGRITY, first shown NP-complete [195] and later fixed-parameter tractable by Fellows and Stueckle [196].

The edge version Edge Integrity was first studied by Barefoot et al. [197].

The classic (non-parameterized) statements of the most studied vulnerability measures are as follows:

Edge Integrity
Minimize, over sets of edges $F \subseteq E$, the target $|F| + \mathrm{mc}(G - F)$.
Edge Toughness
Minimize, over edge-cutsets $F \subseteq E$, the target $\frac{|F|}{\omega(G-F)-1}$.
Edge Scattering Number
Minimize, over sets of edges $F \subseteq E$, the target $\omega(G - F) - |F|$.
Edge Rupture Degree
Minimize, over sets of edges $F \subseteq E$, the target $\omega(G - F) - |F| - \mathrm{mc}(G - F)$.
Edge Tenacity
Minimize, over sets of edges $F \subset E$, the target $\frac{|F|+\tau(G-F)}{\omega(G-F)}$.

Open problem 3.9 (*Research Direction*)**.** *What are the natural parameters for graph vulnerability problems?*

Bang-Jensen et al. [198] study component order connectivity in the setting of directed graphs under several different parameterizations: ℓ, k, $\ell + k$, and $n - \ell$, arguing that if k is small, then ℓ may have to be large, in which case $n - \ell$ can be smaller.

Open problem 3.10 (*Research Direction*)**.** *How is the parameterized landscape for graph vulnerability problems?*

We refer to surveys by Bagga et al. [199], Kratsch et al. [200], and Gross et al. [194] for more on vulnerability measures.

3.2. Connectivity

In this subsection we discuss results around connectivity augmentation problems. In such problems the input is a (multi) graph and the objective is to increase edge or vertex connectivity by adding the minimum number (weight) of additional edges, called links.

This problem was first studied by Eswaran and Tarjan [7] who showed that increasing the edge connectivity of a given graph to 2 by adding a minimum number of links (also called an augmenting set) is polynomial time solvable. Subsequent work of Watanabe and Nakamura [201], Cai and Sun [202], and Frank [203] established that the problem is also polynomial time solvable for any given target value of edge connectivity to be achieved. However, if the set of links is restricted, that is, there are pairs of vertices in the graph which do not constitute a link, or if the links have (non-identical) weights on them, then the problem of computing the minimum size (or weight) augmenting set is NP-complete [7].

It is interesting to note that the vertex version of the problem is substantially less understood even when the set of links that can be added is unrestricted. Vegh [204] obtained a polynomial time solution for the special case when the connectivity of the graph is required to increase by 1, but the complexity of the general case is open. Jackson and Jordán [205] gave a $2^{\mathcal{O}(\lambda)} n^{\mathcal{O}(1)}$-time algorithm for the problem of finding a minimum number of edges to make a graph λ-vertex connected. Let us note that, according to the current knowledge, the problem might still be solvable in polynomial time; whether the general version of the problem is NP-complete is a long-standing open problem [204].

In the Weighted Minimum-Cost Edge-Connectivity Augmentation by One, we are given a graph G which is λ-edge connected, a set of links $\mathcal{L}$, an integer k, a weight function w on $\mathcal{L}$, and $p \in \mathbb{R}$. The task is to decide whether there is a link set $F \subseteq \mathcal{L}$ such that $w(F) \leq p$, $|F| \leq k$ and $G \cup F$ is $\lambda + 1$-edge connected?

The first parameterized algorithm for the connectivity augmentation problem was considered by Nagamochi [206], who gave a $2^{\mathcal{O}(k \log k)} |V|^{\mathcal{O}(1)}$ algorithm for the case when the weights on the links are identical and λ is odd. Guo and Uhlmann [207] gave a kernel with $\mathcal{O}(k^2)$ vertices and links for the same case. Marx and Vegh [208] studied the problem in its full generality and gave a kernel with $\mathcal{O}(k)$ vertices, $\mathcal{O}(k^3)$ links and weights of $(k^6 \log k)$ bit integers. Basavaraju et al. [209] gave an algorithm solving Weighted Minimum-Cost Edge-Connectivity Augmentation by One in time $9^k n^{\mathcal{O}(1)}$.

Another variant of connectivity concerns problems where one has to delete a set of edges while still keeping some connectivity requirements on the remaining graph. Basavaraju et al. [209] study the following Deletion with λ connectivity problem. In this problem we are given a triple $(G, \mathcal{L}, k)$ where G is a λ-edge connected, $\mathcal{L}$ is a set of edges, called links, $G + \mathcal{L}$ is $(\lambda + 1)$-edge connected, and k a positive integer. The task is to decide whether there is a set of k links in $\mathcal{L}$ whose deletion from $G + \mathcal{L}$ maintains $(\lambda + 1)$-edge connectivity. Basavaraju et al. gave an algorithm solving Deletion with λ connectivity in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$.

Hüffner, Komusiewicz, and Sorge [210] introduced the following edge deletion problem. We say that an n-vertex graph G is highly connected, if every vertex of G is of degree at least $\lfloor n/2 \rfloor + 1$. In the Seeded Highly Connected Deletion problem, the input is graph G, a vertex set $S \subseteq V(G)$, and integers k and α. The task is to decide whether there is an edge set $X \subseteq E(G)$ of at most α edges such that $G - X$ consists only of degree-zero vertices and a $(k + |S|)$-vertex highly connected subgraph containing S. They obtained a kernel with at most $2\alpha + 4\alpha/k$ vertices and $\binom{2\alpha}{2} + \alpha$ edges computable in $\mathcal{O}(\alpha^2 \, nm)$ time. They also gave a subexponential algorithm of running time $\mathcal{O}(2^{4 \cdot \alpha^{0.75}} + \alpha^2 \, nm)$.

Adler, Kolliopoulos, and Thilikos [211] introduced the problem of augmenting a planar graph G with a given set of k pairs of terminals. The task is to augment G with the minimum number of edges such that all edges are added within one face of G, the augmented graph is planar and all terminal-pairs are linked with vertex-disjoint paths. This problem is FPT parameterized by k [211].

Gutin, Ramanujan, Reidl, and Wahlström [212] studied problems relating to deleting edges in a biconnected graph while maintaining biconnectivity; recall that a graph is biconnected if it is connected and has no cut vertex. Two of the problems are a weighted variant and an unweighted variant. The weights are on the edges, and in Weighted Biconnectivity Deletion, we are given an edge-weighted graph G, and integer k, and a target weight w^* and asked to delete at most k edges whose weights sum to at least w^*, while maintaining biconnectivity. They show that this problem is in FPT by giving an algorithm with running time $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$. For the unweighted version, in which every edge has unit weight and $w^* = k$, they give a randomized kernel with k^9 vertices.

They go on to generalize the unweighted variant to the problem Vertex-ρ-Connectivity Deletion, parameterized by $k + \rho$; given a graph G and two integers ρ and k, the task is to decide whether there exists a set $S \subseteq E(G)$ of k edges such that $G - S$ is ρ-vertex connected. This problem is shown to be non-uniform FPT, i.e., for fixed k and ρ, the problem is solvable in polynomial time $\mathcal{O}(n^6)$ due to the fact that the problem is expressable in CMSOL (ibid.).

Open problem 3.11 (*Open Questions [212]*)**.** *The following four questions are raised:*

1. *Is there a single exponential time algorithm for* Weighted Biconnectivity Deletion*?*
2. *Can we solve* Vertex-ρ-Connectivity Deletion *in time* $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$ *for fixed* $\rho > 2$*?*

3. *Does* Biconnectivity Deletion *(the unweighted variant) admit a deterministic polynomial kernel?*
4. *Does* Weighted Biconnectivity Deletion *admit a polynomial kernel?*

3.3. Clustering

One of the simplest variants of clustering is Cluster Editing or Correlation Clustering, where the task is to delete/add in total at most k edges from/to graph G such that every connected component of the obtained graph is a clique. Since a clique is a graph containing no induced path P_3, Cluster Editing is also a special case of the problem of editing to a graph class characterized by a finite number of minimal forbidden subgraphs. This is why we discussed this problem in Section 2.1. However, different variants of clustering do not fit this scheme and we discuss them here.

One can generalize the concept of cluster graphs as follows. A graph is an s-club cluster if every connected component has diameter at most s. These graph classes are not hereditary, as adding a universal vertex will transform any graph into a 2-club cluster. Liu, Zhang, and Zhu [213] studied 2-Club Cluster Deletion as well as 2-Club Cluster Editing. They show that both these problems (and the vertex deletion version) are NP-complete and they give a $2.74^k \cdot n^{\mathcal{O}(1)}$ time algorithm for 2-Club Cluster Deletion. Whether the problem admits a polynomial kernel is open.

Open problem 3.12 (*[213]*)**.** *Does* 2-Club Cluster Deletion *admit a polynomial kernel?*

As mentioned earlier, Cluster Editing does not admit a subexponential time algorithm [118] unless ETH fails. On the other hand, the problem p-Cluster Editing, where the number of components in the target class is fixed to be exactly p—rather surprisingly—does indeed admit a subexponential parameterized time algorithm. This was shown by Fomin et al. [123], who designed an algorithm solving this problem in time $2^{\mathcal{O}(\sqrt{pk})} \cdot n^{\mathcal{O}(1)}$. The p-Cluster Editing problem, as well as p-Cluster Deletion was first studied by Shamir, Sharan, and Tsur [8], who showed that even 2-Cluster Deletion is NP-complete. Abu-Khzam [214] studied a different multi-parameterized version, called (a, d, s, k)-Cluster Editing (as well as subsets of $\{a, d, s, k\}$) where we want to transform an input graph G to a cluster graph, each of size at least s, using at most k edits such that for each vertex $v \in V(G)$ the number of newly added (respectively, deleted) edges incident to v is some fixed $\alpha(v) \leq a$ (respectively, $\delta(v) \leq d$). They show that the problems are already NP-complete for constant a and d, and also show some positive results, both in terms of kernelization, as well as settings where the problem is polynomial time solvable.

Motivated by the result of Fomin et al. [123], Misra, Panolan, and Saurabh [215] studied a similarly constrained version of another clustering problem: s-Club d-Cluster Editing. In this version, the clusters are s-club graphs and the number of clusters is *at most d*. They show that in this case, constraining the number of desired clusters does not help in obtaining a subexponential time algorithm: s-Club d-Cluster Editing does not admit a subexponential time fixed parameter algorithm $2^{o(k)}n^{\mathcal{O}(1)}$ when parameterized by k, even for any fixed $s \geq 2$ and $d \geq 2$.

Hüffner et al. [216] considered the parameterized complexity and kernelization of a clustering variant called Highly Connected Deletion. In this problem one seeks to delete at most k edges such that, in the resulting graph, each vertex in each connected component is adjacent to at least half of the vertices of this component. See also the work of Bliznets and Karpov for further improvements and other variants of this problem [217]. Golovach and Thilikos [218] studied a related notion of connectivity clustering, where the task is to delete at most k edges to obtain clusters of given size and of given connectivity.

Finally, let us mention that for weighted graphs, the Cluster Editing problem also admits a $2k$ kernel with integer weights [45] and an FPT algorithm in $\mathcal{O}(1.82^k)$ time [219]. Several "dynamic" versions of Cluster Editing have also been studied. We mention a few here: The parameterized complexity of Dynamic Cluster Editing, along with the deletion and completion versions, were studied by Luo et al. [220]. In this problem we are given two input graph over the same vertex set, G and G_c, where G_c is a cluster graph. The problem is to edit at most k edges in G to obtain a graph G' that is at most d "far away" from G_c. They have several different distance functions they study. The problem is already NP-complete when the input graph is a cluster graph. All versions are also W[1]-hard parameterized by either k or d, however, when parameterized by $k + d$, they obtain polynomial kernels for some of the problems. We refer to the article for the list of open problems relating to Dynamic Cluster Editing. Chen et al. [221] study two problems, called Multi-Layer Cluster Editing and Temporal Cluster Editing. They show some positive (FPT) results for the former and a W[1]-hardness result for the latter. Bocci et al. [222] took this framework and studied the problem in the context of *temporal cliques*, under the name of Editing to Temporal Cliques. They show that the problem is NP-complete, but that it admits an FPT algorithm when parameterized by $k+t$, where k is the budget and t the number of timesteps in the input temporal graph.

4. Degree constraints

In this section, we survey the advances in modifying graphs to have some specified degree constraints possibly together with other properties, e.g. connectivity. The degree constraints may be related to degrees of individual vertices or degree sequences. Such problems have a long history in the literature as they encompass classical problems like Perfect Matching, r-Factor, Hamiltonian Path, or Hamiltonian Cycle. Typically, whenever the parameterized complexity of problems of this kind was investigated, the authors also considered vertex deletions besides edge modification operations. Hence, to present the full spectra of the work, we extend our framework in this section to include the results about vertex deletion when appropriate.

4.1. Modification to satisfy individual degree constraints

The investigation of the parameterized complexity of the problems where the aim is to satisfy some degree restrictions for each vertex was initiated by Moser and Thilikos [223] and Mathieson and Szeider [224].

In particular, Moser and Thilikos [223] considered the k-Almost r-Regular Graph problem, which asks, given a graph G and a non-negative integer k, whether G can be made r-regular by deleting at most k vertices. They proved that the problem admits a kernel with $\mathcal{O}(kr(r + k)^2)$ vertices. Despite the fact that Thilikos and Moser were interested solely in vertex deletions, we discuss this result briefly, because the approach that was used is generic for similar problems. Since the deletion of a vertex decreases the degrees of its neighbors by one, a vertex of degree at most $r - 1$ or at least $r + k + 1$ should be deleted. Applying this straightforward reduction rule to the input graph, we obtain a graph G of bounded degree. For a set of vertices X of G of degree at least $r + 1$, one can observe that its size should be polynomially bounded in k and r for any yes-instance, since G has bounded degree. Further, we have that the vertices of $G - X$ have the same degrees r, and for each component H of $G - X$, it holds that if any vertex of H or a neighbor of a vertex of H is deleted, then H should be deleted completely. This observation allows us to construct reduction rules for the components of $G - X$. This way a polynomial kernel could be constructed. Thilikos and Moser complement these results by observing that because Cubic Subgraph is one of the fundamental NP-complete problems discussed by Garey and Johnson [225] (in fact, this problem is NP-complete for very restricted inputs [226–228]), k-Almost r-Regular Graph is para-NP-complete when parameterized by d.

The most general variant of the modification problem to satisfy degree constraints was introduced by Mathieson and Szeider [224] (see also the thesis of Mathieson [229] for more details). For a set of modification operations S, they defined

Weighted Degree Constraint Editing (S)(WDCE(S))

Input: A graph G, non-negative integers k and r, a weight function $\rho: V(G) \cup E(G) \rightarrow \mathbb{N}_0$, and a degree list function $\delta: V(G) \rightarrow 2^{\{0,\dots,r\}}$.

Question: Is it possible to obtain a graph G' from G such that for every $v \in V(G')$, $\sum_{vx \in E(G')} \rho(vx) \in \delta(v)$, using at most k modification operations from S?

Here, the aim is to obtain a graph such that the (weighted) degree of every vertex is in a given set defined by the list function. They considered WDCE(S) for various non-empty

$S \subseteq$ {vertex deletion, edge deletion, edge addition}.

Mathieson and Szeider also considered the unweighted variant of the problem that we call Degree Constraint Editing (S) (DCE(S)). In this variant the weight function $\rho \equiv 1$, but DCE(S) is not a restricted version of WDCE(S). For DCE(S), vertex and edge deletions and edge additions are defined in the standard way. For the weighted problem, it is more complicated. It is assumed that each modification operation has unit cost. If a vertex v has weight $\rho(v) = 1$, then the vertex deletion operation deletes v together with incident edges, but if $\rho(v) > 1$, then this operation just reduces its weight by 1. Similarly, the edge deletion operation deletes an edge e of weight 1 and reduces its weight by 1 if $\rho(e) > 1$. Hence, the edge addition operation can be applied to an existing edge, and it increases the weight of such an edge by 1.

Mathieson and Szeider [224] proved that WDCE(S) and DCE(S) are W[1]-hard when parameterized by k for any non-empty S. Moreover, the hardness for DCE(S) holds even if $\delta(v) = \{r\}$ if vertex deletion $\in S$, that is, for the case where the aim is to obtain an r-regular graph. It is interesting to observe that for the important case of degree lists of size 1, WDCE(S) and DCE(S) can be solved in polynomial time if vertex deletion $\notin S$ by the reduction to the Perfect Matching problem [224].

On the positive side, Mathieson and Szeider proved that WDCE (S) and DCE(S) are FPT when parameterized by $k + r$ for any non-empty S. To achieve this result, they showed that for any k and r, the problems can be expressed in first-order logic. Applying a similar technique to the one applied by Moser and Thilikos above [223], an instance of WDCE(S) or DCE(S) can be reduced to an equivalent instance with a graph of bounded degree. Then the meta-theorem of Frick and Grohe [230] gives the result (the same can be obtained without preprocessing by the meta-theorem of Bulian and Dawar [231]). Clearly, this approach only allows to classify WDCE(S) and DCE(S) to be in FPT. For the special case of DCE(S) with degree lists of size 1, Golovach [232] used the random separation technique (see the textbook on parameterized algorithms [10] for an introduction to this technique) to show that the problem can be solved in time $2^{\mathcal{O}(kr^2+k\log k)}n^{\mathcal{O}(1)}$. This gives rise to the following open problem.

Open problem 4.1. *Is it possible to give efficient* FPT *algorithms for* DCE(S) *and/or* WDCE(S) *parameterized by* $k + r$ *for general degree list functions?*

For the case S = {vertex deletion} and S = {vertex deletion, edge deletion} and single-element degree lists, Mathieson and Szeider [224] showed that WDCE(S) admits a kernel with $\mathcal{O}(kr(k+r))$ vertices. For general degree lists, they demonstrated a kernel with $\mathcal{O}(k^2r^{k+1}+kr^{k+2})$ vertices. These results were complemented by Froese, Nichterlein, and Niedermeier [233], who proved that if only edge additions are allowed (i.e., for the completion problem), then DCE(S) has kernels with $\mathcal{O}(kr^2)$ and $\mathcal{O}(r^5)$ vertices, that is, it admits a polynomial kernel whose size depends only on r. To obtain the latter result, they proved that the problem can be solved in polynomial time if k is sufficiently large (greater than some polynomial function of r). The latter result is based on a clever application of combinatorial results about existence of f-factors. Hence, the following win–win approach can be used: if k is large, then the problem is solved in polynomial time, and if k is bounded by a polynomial function of r, then the kernelization algorithm for the case where the parameter is $k + r$ is applied. Froese, Nichterlein, and Niedermeier [233] also gave lower bounds by proving that DCE(S) parameterized by $k + r$ has no polynomial kernel unless NP $\subseteq$ coNP/poly if S = {vertex deletion} or S = {edge addition}. Another lower bound for this parameterization was given by Golovach [232] who proved that DCE(S) with degree lists of size one has no polynomial kernel unless NP $\subseteq$ coNP/poly if {vertex deletion, edge addition} $\subseteq S$.

The variant of DCE(S) with degree lists of size one, where $S \subseteq$ {vertex deletion, edge deletion} and where we are given separate bounds k_v and k_e for the number of vertex and edge deletions, respectively, was considered by Dabrowski et al. [234] on planar graphs. They proved that the problem admits a polynomial kernel when parameterized by $k_v + k_e$.

Golovach [235] introduced the degree constrained modification problem with connectivity restrictions. There it was called Edge Editing to a Connected Graph of Given Degrees but later the other title Edge Editing to Connected f-Degree Graph was proposed and we use it in the survey.

Edge Editing to Connected f-Degree Graph (EECG)

Input: A graph G, non-negative integers d and k, and a function $f: V(G) \rightarrow \{0,\dots,d\}$.

Question: Is it possible to obtain a connected graph G' from G with $d_{G'}(v) = f(v)$ for every $v \in V(G')$ by at most k edge deletions and additions?

Recall [224] that if the degree lists have size 1, DCE(S) is polynomial time solvable if only edge deletions and additions are allowed. Contrary to this, EECG is NP-hard even if $f(v) = 2$ for all $v \in V(G)$: it is straightforward to see that EECG for $f(v) = 2$ for $v \in V(G)$ and $k = m - n$ is equivalent to the Hamiltonian Cycle problem that is well-known to be NP-complete [225]. We also mention here that Franzblau and Raychaudhuri [236] studied the problem of adding k edges to get a Hamiltonian graph, Hamiltonian Completion, which is equivalent to ask for a Path Partition (also known as a Path Cover) with $k+1$ paths [236] (this implies that Path Partition is para-NP-hard). Moran and Wolfstahl [237] gave a linear-time algorithm for the problem on the class of cacti graph.[11]

Golovach [235] proved that EECG has a kernel with $\mathcal{O}(kd^3(k+d)^2)$ vertices. The results is obtained using the generic approach of

[11] A cactus graph is a graph where no two cycles share an edge.

Moser and Thilikos [223], but due to allowing edge additions and connectivity restrictions, the reduction rules are great deal more complicated and are based on the following structural observations. It can be easily seen that if v is a vertex of G with $d_G(v) = f(v)$, then the number of deleted edges incident to v equals to the number of added edges incident to v. Therefore, if X is the set of vertices of G whose degrees are different from the values of f, then for any solution, the set of deleted edges D and the set of added edges A compose a graph which can be covered by edge disjoint walks without repeated edges joining vertices of X and closed walks that are alternating in the sense that if an edge of a walk is from D, then the next is from A and vice versa. Golovach [235] also constructed an algorithm running in time $k^{\mathcal{O}(k^3)}n^{\mathcal{O}(1)}$ for the case $f(v) = d$ for each $v \in V(G)$, that is, for the modification to a connected regular graph, but left open the question whether EECG if FPT when parameterized by k only. This question was resolved by Fomin, Golovach, Panolan, and Saurabh [238]. They show that EECG is solvable in $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ time. Fomin et al. [238] use the same structural properties of solutions as Golovach in [235], but the crucial new component is the application of the recently developed matroid representative sets techniques combined with color coding (we refer to the textbook on parameterized algorithms [10] for an introduction to these techniques). It is still open whether EECG has a polynomial kernel whose size depends on k only. For the special case of planar graphs, Dabrowski et al. [234] proved that the problem admits a polynomial kernel parameterized by number of vertex deletions, and parameterized by the number of edge deletions.

Open problem 4.2. *Does* Edge Editing to Connected f-Degree Graph *(*EECG*) parameterized by k have a polynomial kernel?*

For the weighted variant of EECG, Fomin et al. [238] proved that it is W[1]-hard when parameterized by $k + d$. Recall that in DCE(S) we require that each vertex has the degree from a given list but in DCE(S) these lists have size one. It leads to the following open problem:

Open problem 4.3. *Investigate the parameterized complexity of the variant of* EECG *where, instead of the degree function f, a degree list function $\delta : V(G) \to 2^{\{0,\dots,r\}}$ is given and we are asked whether it is possible to obtain a connected graph G' from G with $d_{G'}(v) \in \delta(v)$ for every $v \in V(G')$ by at most k edge deletions and additions.*

Notice that if we have a choice of degrees, then the structural properties of solutions used above [235,238] could not be applied any more. The problem is open even when the degree constraints are intervals of bounded size. Haarberg considered the special case where the degree constraints are given by inequalities. More precisely, they considered the Edge Editing to a Connected Upper (Lower) Bounded Degrees (EditUBD and EditLBD, respectively) problems [239]. EditUBD asks, given a (multi-) graph G, a non-negative integer k and a function $f : V(G) \to \mathbb{N}$, whether it is possible to obtain a connected graph G' from G with $d_{G'}(v) \le f(v)$ for every $v \in V(G')$ by at most k edge deletions and additions. They show that this problem is NP-complete, has a kernel with $\mathcal{O}(k^3)$ vertices and $\mathcal{O}(k^6)$ edges, and can be solved in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$. In EditLBD, we further require that $d_{G'}(v) \ge f(v)$ for every $v \in V(G')$, that is, the upper bounds on the degrees are replaced by lower bounds. Interestingly, EditLBD was shown to be solvable in polynomial time [239].

All aforementioned problems are stated for undirected graphs. The systematic study of the degree constraint modification problems for directed graphs was recently initiated by Bredereck et al. [240]. We will return to this paper in the next section where we consider degree sequence restriction, but here we mention only that they considered the Digraph Degree Constraint Completion problem that could be seen as a variant of DCE(S), where for each vertex, a degree list function that assigns to each vertex a set of pairs of non-negative integers from $\{0, \dots, r\}$ that specify the desired pairs of values of in- and out-degrees respectively are given and S = {edge/arc addition}. Bredereck et al. [240] show that this problem admits a kernel with $\mathcal{O}(r^5)$ vertices.

Open problem 4.4. *Investigate the parameterized complexity of variants of* DCF(S) *and* EECG *for directed graphs.*

Besides vertex degree constraints, it could be interesting to consider edge degree constraints or combined vertex and edge degree constraints. In particular, Mathieson [241] considered a number of problems of this type. For an edge weighted graph, the weighted degree of a vertex is defined as the sum of weights of incident edges. Respectively, the weighted sum of an edge is the sum of the vertex degrees of its end-points. Mathieson [241] considered the following problems for edge weighted graphs:

- Weighted Edge Degree Constraint Editing, where for each edge, it is given a list of weighted degrees, and the aim is to obtain a graph, by at most k modification operations, such that every edge has a degree from its list.
- Weighted Bounded Degree Editing, where a degree bound for each vertex is given, and the aim is to obtain a graph, by at most k modification operations, such that the weighted degree of a vertex does not exceed its bound.
- Weighted Edge Regularity Editing, where for each vertex, it is given a list of weighted degrees, and for each pair of vertices, it is given a set of feasible sizes of the set of common neighbors, and the aim is to obtain a graph, by at most k modification operations, such that every vertex has weighted degree from its list and for every edge, the size of the set of common neighbors is feasible.
- Weighted Strongly Regular Editing, that is a variant of Weighted Edge Regularity Editing, where additionally a second set of allowed sizes of the set of common neighbors is given for each pair of vertices, and for every pair of non-neighbors of the modified graph, the size of the set of common neighbors should belong to this set.

The allowed modification operations are vertex deletions, edge deletions and edge additions. Mathieson presented the essentially complete picture of the complexity of these problems parameterized by the number of modification operations k and/or the upper bound of the feasible degrees for various combinations of allowed operations: the cases when the problems are W[1]-hard, FPT, have polynomial kernels or do not have them up to some conjectures are distinguished. He also investigated special cases, in particular, the unweighted problems (i.e., the problems for unit weights) and the case when the sets of feasible degrees are singletons. Additionally, the structural parameterization by the treewidth of an input graph was considered. We do not discuss the details of these results, because they proved to be similar to the results about DCS(S) and are obtained by similar techniques.

Another direction of research would be to consider the discussed problems for graph classes. Up to now, a very little work was done in this direction. Dabrowski et al. [234] considered variants of DCF(S) and EECG for planar graphs. More precisely, they considered the problems that asks for a given planar graph G, a degree function $f : V(G) \to \mathbb{N}_0$ and two non-negative integers k_e and k_v, whether it is possible to obtain a (connected) graph G' from G with $d_{G'}(v) = f(v)$ for $v \in V(G')$ by deleting at most k_v vertices and at most k_e edges. They proved that these problems have polynomial kernels when parameterized by $k_v + k_e$. In fact, more general kernalization results were obtained as it is shown that it could be assumed that vertices and edges have costs and the task is to delete at most k_v and k_e to satisfy degree restriction and achieve the minimum total cost of deleted vertices and edges. This result is obtained via the protrusion decomposition/replacement techniques introduced by Bodlaender et al. [242].

Open problem 4.5. *Investigate the parameterized complexity of variants of* DCF(S) *and* EECG *for graph classes. In particular, what can be said about planar graph when edge additions are allowed and the graph obtained by the modification should stay planar?*

We conclude this section by discussing modification problems which deal with the parity constraints for degrees. These problems are the most investigated degree constraint modification problems. Already in 1977 Boesch, Suffel, and Tindell [243] (see also [244,245]) proved that Eulerian Completion, which is finding the minimum number of edges that should be added to make the input graph Eulerian, can be solved in polynomial time, and the same holds if multiple edges are allowed and for Even Graph Completion where the aim is to obtain a graph with vertices of even degrees. Recall that a (directed) graph G is *Eulerian* if it contains a closed walk without repeating edges (arcs) that goes through every edge (arc). By the classical Euler theorem, a connected graph is Eulerian if and only if its vertices have even degrees. Similarly, a (weakly) connected directed graph is Eulerian if and only if for every vertex its in-degree is the same as its out-degree (see, e.g., [246]). Following the same scheme as with the previous problems, we state the generalization of Eulerian Completion for a set of modification operations S as follows.

Connected Parity Constraint Editing (S)(CPCE(S))

Input:	A graph G, a parity function $f : V(G) \to \{0, 1\}$ and a non-negative integer k.
Question:	Is it possible to obtain a connected graph G' from G such that for every $v \in V(G')$, $d_{G'}(v) \equiv f(v) (\mathrm{mod}\ 2)$, using at most k modification operations from S?

When we do not require connectivity, we refer to the problem as Parity Constraint Editing (S) (PCE(S)). For directed graphs, we state the following problem.

Connected Degree Balance Editing (S)(CDBE(S))

Input:	A directed graph G, a function $f : V(G) \to \mathbb{Z}$ and a non-negative integer k.
Question:	Is it possible to obtain a weakly connected directed graph G' from G such that for every $v \in V(G')$, $d^+_{G'}(v) - d^-_{G'}(v) = f(v)$, using at most k modification operations from S?

Here $d^-_G(v)$ and $d^+_G(v)$ denote in- and out-degree of a vertex v in a graph G. Notice that if $f(v) = 0$, then the question is equivalent to asking whether we can obtain an Eulerian graph by at most k operations.

Generalizing the results of Boesch, Suffel and Tindell [243], Dabrowski, Golovach, van 't Hof and Paulusma [247] proved that CPCE(S) and CDBE(S) are polynomial if S = {edge/arc addition}. Moreover, the problems remain polynomial if S = {edge/arc addition, edge/arc deletion}. It can be also observed that the same holds for PCE(S). If vertex deletion $\in S$, then CPCE(S), PCE(S) and CDBE(S) are NP-hard and W[1]-hard by the results of Cai and Yang [248], Cygan et al. [249], and Dabrowski et al. [247]. From the parameterized complexity point of view, the most interesting case is the case when S = {edge/arc deletion} for CPCE(S) and CDBE(S) (PCE(S) is polynomial in this case as it was proven by Cygan et al. [249]).

Cygan et al. [249] observed that if S = {edge/arc deletion}, then CPCE(S) and CDBE(S) are NP-complete. They also proved that they can be solved in time $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$ and complemented these results by proving that these problems have no polynomial kernel unless NP $\subseteq$ coNP/poly. Their FPT result is based on the following structural observation. If G is an undirected graph and T is the set of vertices for which degree constraints are broken, then the edges of a solution form a T-join, that is, they induce a forest that could be decomposed into edge disjoint paths that connect $|T|/2$ pairs of vertices of T. Hence, the task is to find a T-join of size at most k such that the deletion of the edges of the join does not destroy connectivity. Cygan et al. [249] use a non-trivial application of the color coding technique to solve this problem. Similar techniques work also for directed graphs. Their results were improved by Goyal et al. [250]. They showed that CPCE(S) and CDBE(S) for S = {edge/arc deletion} can be solved in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$. They use the same structural observations as Cygan et al. [249], but instead of color coding, they apply matroid representative sets techniques. In particular, for undirected graphs, they use the fact that the set of edges of a T-join is an independent set of the cographic (bond) matroid. It should be noted that Cygan et al. [249] and Goyal et al. [250] gave their results for the Eulerian Edge Deletion problem, that is for the special cases of CPCE(S) and CDBE(S) where $f(v) = 0$ (Goyal et al. [250] considered also Connected Odd Edge Deletion), but the algorithm could be rewritten for CPCE(S) and CDBE(S) in a straightforward way.

Observe that in CPCE(S) and CDBE(S), the parameter k upper-bounds the number of modification operations. We can ask whether the modifications can be done by *exactly* k operations. In particular, Cai and Yang [248] left open the following problem.

Open problem 4.6 (*[248]*)**.** *Is* $(m - k)$-edge Eulerian Subgraph, *which asks whether a (directed) graph has an Eulerian subgraph with exactly* $m - k$ *edges (arcs),* FPT*?*

The same question can be asked for more general degree restriction given in CPCE(S) and CDBE(S).

Notice that in CDBE(S) we require G' to be weakly connected. It is natural to ask whether this condition could be strengthened.

Open problem 4.7. *Investigate the parameterized complexity of variant of CDBE(S) where the graph G' obtained by the modifications is required to be strongly connected.*

A more special question was asked by Cygan et al. [249] (see also [251]).

Open problem 4.8 (*[249,251]*)**.** *Is it* FPT *to decide whether it is possible to delete at most k arcs from a directed graph to obtain a graph where each strongly connected component is Eulerian?*

This problem was considered by Crowston et al. [252] for tournaments, which they called Min-Desc (minimum deletion to obtain Eulerian strong components), but we call Eulerian Deletion on Tournaments. They proved that Eulerian Deletion on Tournaments has a kernel with at most $4k \cdot (4k + 2)$ vertices.

Recall that Boesch, Suffel, and Tindell [243] proved that the Eulerian Completion problem can be solved in polynomial time, but the situation changes if we switch to the weighted variant of the problem. For directed graphs, NP-hardness was proved by Höhn, Jacobs, and Megow [253] for special cases that occur in scheduling problems. It is also easy to see that the problem is NP-hard for undirected graphs as well by a straightforward reduction from Eulerian Deletion. The parameterized complexity of the following problem was considered by Dorn, Moser, Niedermeier, and Weller [245].

Weighted Multigraph Eulerian Completion (WMEC)

Input: A directed multigraph G, a weight function $w: V(G) \times V(G) \rightarrow \mathbb{N}_0$, and a non-negative integer k.
Question: Is it possible to obtain an Eulerian multigraph G' from G by adding arcs of total weight at most k?

Since WMEC deals with multigraphs, the addition of parallel arcs is allowed. It can be noted that the classical Chinese Postman problem, where the aim is to find a shortest closed walk that visits all arcs of a given directed graph, and the more general Rural Postman, where it is required to find a shortest walk that visits a given set of arcs, can be seen as special cases of WMEC. Dorn et al. [245] showed that WMEC can be solved in time $\mathcal{O}(4^k \cdot n^3)$. This result immediately implies the respective FPT result for Rural Postman. They conjecture that similar results can be obtained for undirected graphs. They also leave open the question about the variant with arc deletion. Generalizing it, we obtain the following open problem.

Open problem 4.9. *Investigate the parameterized complexity of weighted variants of CPCE(S) and CDBE(S) for graphs and multigraphs for $S \subseteq$ {edge/arc deletion, edge/arc addition}.*

Another parameterization of WMEC was considered by Sorge et al. [254,255]. They proved that WMEC is FPT when parameterized by $b + c$, where

$$b = \sum_{v \in V(G)} \left| d_G^+(v) - d_G^-(v) \right| ,$$

and c in the number of weakly connected components of G. They complemented this result by showing that WMEC has no polynomial kernel when parameterized by b, c, k or b_c unless NP $\subseteq$ coNP/poly.

We conclude the section by the open problem stated in [247]. We considered the degree constraint modification problems with parity restrictions. What can be said if we replace parity constraints by the more complicated "modulo d constraints" for $d \geq 3$. It is observed in [247] that this variant of CPCE(S) is NP-hard if S = {edge deletion, edge deletion} and $d = 3$. Taking into account the W[1]-hardness of CPCE(S) if vertex deletion $\in S$ (see [247]), we ask the following.

Open problem 4.10 (*[247]*)**.** *Investigate the parameterized complexity of the variants of CPCE(S) for $S \subseteq$ {edge deletion, edge addition}, where a positive integer d is given, the parity function f is replaced by a function $f : V(G) \rightarrow \{0, \ldots, d-1\}$ and where the aim is to obtain a connected graph G' such that for every $v \in V(G')$, $d_{G'}(v) \equiv f(v) (\mathrm{mod}\ d)$?*

Additionally, what can be said if we remove the connectivity restriction?

4.2. Modification to satisfy degree sequence constraints

In this section we consider problems where the task is to modify a graph in order to satisfy constraints on degree sequences. Motivations for the problems considered here often come from applications like social networks. The *identity disclosure* is a specific type of privacy breach in social networks. It happens when an adversary is able to determine the identity of an entity in a network. One can weaken this to the *existence disclosure*, where one is able to identify whether an entity is present in a social network or not. *Affiliation link disclosure* is the problem to determine whether an entity belongs to a specific group in a social network. As Zheleva and Getoor [256] say in their survey,

> k-anonymity protection of data is met if the information for each person contained in the data cannot be distinguished from at least $k - 1$ other individuals in the data.

In *degree anonymization*, a graph is said to be s-degree-anonymous (or simply s-anonymous when it is clear from context that we are talking about degree anonymity) if for every vertex v, there are at least $s - 1$ vertices with the same degree as v. This leads to the modification problems where the aim is to achieve the desired level of anonymity by bounded number of operations. We refer to the survey of Casas-Roma, Herrera-Joancomartí and Torra [257] for the introduction to the edge modification techniques used in anonymization and focus on the parameterized complexity of the problems. Following the style used in the previous section, we define the following problem for a set of modification operations S.

Anonymization(S)

Input: A graph G, a positive integer s and a non-negative integer k.
Question: Is it possible to obtain an s-anonymous graph G' from G using at most k modification operations from S?

Degree anonymization is perhaps one of few places where the operation of adding vertices is a natural operation; a "dummy" vertex can be created in a social network. Hence, the case vertex addition $\in S$ was investigated.

Bazgan et al. [258] obtained a number of hardness results. They proved that if S = {edge deletion} or S = {vertex deletion}, then Anonymization(S) is already NP-hard for $s = 2$ even for trees and the problem is NP-hard if the maximum degree Δ of the input graph is 3 or 7, respectively, that is, the problem is para-NP-hard for the respective parameterizations. They also showed that Anonymization(S) is W[1]- or W[2]-hard when parameterized by $s + k$ if S = {edge deletion} or S = {vertex deletion}, respectively. Furthermore, they also proved that (vertex deletion) has no polynomial kernel unless NP $\subseteq$ coNP/poly when parameterized by $k + s + \Delta$. They obtained a number of inapproximability results. In particular, they initiated the investigation of the parameterized approximation/inapproximability for Anonymization(S). Observe that we obtain a bicriteria optimization problem here. First, it is possible to maximize the anonymity level s by performing at most k modification operations, and the second option is to minimize the number of modification operations to obtain a s-anonymous graph. Finally, for the maximization of the anonymity level, they showed that the problem is not FPT $n^{1/2-\varepsilon}$-approximable for every $0 < \varepsilon \leq 1/2$ when parameterized by k even on trees unless FPT = W[2] if S = {vertex deletion}, and it is not FPT $n^{1-\varepsilon}$-approximable for every $1/2 < \varepsilon \leq 1$ when parameterized by k unless FPT = W[1] if S = {edge deletion}. The following question is open.

Open problem 4.11 (*[258]*)**.** *Are there "reasonable" (parameterized) approximation algorithms for the optimization variants of* Anonymization(S) *parameterized by s and k if $S \subseteq$ {edge deletion, edge addition}?*

On the positive side, Bazgan et al. [258] considered a more general variant of the problem where non-negative integers k_-^v, k_+^v, k_-^e, k_+^e are given and the question is whether it is possible to obtain an s-anonymous graph by at most k_-^v vertex deletions, at most k_+^v vertex additions, at most k_-^e edge deletions and at most k_+^e edge additions. They prove that the problem is FPT when parameterized by $k + \Delta$ for $k = k_-^v + k_+^v + k_-^e + k_+^e$. The result is obtained via the first-order logic machinery using the meta-theorem of Frick and Grohe [230]. Bazgan et al. [258] also sketched a direct color coding algorithm for the problem.

Bazgan et al. [258] initiated an investigation of Anonymization(S) for graph classes and obtained a number of hardness results and distinguished some polynomial cases. To properly understand which graph structures can be exploited to create efficient algorithms, and on the other hand which structures are obstacles for efficient algorithms, this line of research should be extended. Recently, a link between complex networks on the one side and graphs with certain topological features on the other side, has been established [259], and since the problem of anonymization originates from social networks, it makes sense to study the problem on topological graphs.

Open problem 4.12. *Investigate the parameterized complexity of* Anonymization(S) *for graph classes. In particular, what can be said about planar graphs?*

Notice that here we can restrict only the input graphs or demand that both the input and the modified graph belong to a specific class.

Hartung et al. [260] considered the case $S = \{\text{edge addition}\}$. They proved that the problem is W[1]-hard even when $s = 2$ when parameterized by the number of edge additions, k. The main result of the paper is that Anonymization{edge addition}) admits a kernel with $\mathcal{O}(\Delta^7)$ vertices implying that the problem is FPT when parameterized by the maximum degree of the input graph. As their first step, they use the approach that is generic for similar problems. Namely, if the set of vertices of the input graph of degree $0 \le d \le \Delta$ is sufficiently large, then it is possible to select a block of such vertices of size that is bounded in k and assume that for every added edge in a solution, if it has its end-vertex (both end-vertices) in the set of vertices of degree d, then this end-vertex (these end-vertices) is (are) in the selected block. This observation leads to a kernel size that is polynomial in Δ, s and k. Hartung et al. [260] showed that it is possible to obtain a kernel whose size depends on Δ only by adjusting s and showing that if k is sufficiently large compared to Δ, then the problem can be solved in polynomial time.

To conclude the part about anonymization, Bredereck et al. [261] considered Anonymization(S) for the case $S = \{\text{vertex addition}\}$, and Talmon and Hartung [262] investigated the case where the modification operations allowed are various types of contractions.

The investigation of the modification problems with the aim to satisfy some general degree sequence properties was initiated by Froese, Nichterlein, and, Niedermeier [233]. Recall that the degree sequence of an n-vertex graph G is an n-tuple containing the degrees of the vertices. Froese et al. [233] introduced the following problem for a tuple property Π.

Π-Degree Sequence Completion (Π-DSC)

Input: A graph G and a non-negative integer k.
Question: Is it possible to obtain a graph G' with the degree sequence satisfying the property Π from G using at most k edge additions?

Notice that Π is a tuple property. In particular, DCE({edge addition}) is not a special case of Π-DSC, but Anonymization{edge addition}) is. They introduced the auxiliary Π-Decision problem that asks whether an n-tuple $T = (d_1, \ldots, d_n)$ of non-negative integers satisfies Π and proved, using the previous results about Anonymization{edge addition}) [260], that if Π-Decision is FPT when parameterized by $\Delta' = \max\{d_i \mid 1 \le i \le n\}$, then Π-DSC is FPT when parameterized by $\Delta + k$. Recall now that Bredereck et al. [261] proved that Anonymization({edge addition}) is FPT when parameterized by Δ and has a kernel with $\mathcal{O}(\Delta^7)$ vertices. Generalizing this result, Froese et al. [233] defined the Π-Number Sequence Completion (Π-NSC) problem that asks for a sequence $d_1, \ldots, d_n$ of non-negative integers and two non-negative integers k and Δ', whether there are non-negative integers $x_1, \ldots, x_n$ such that the n-tuple $T = (d_1 + x_1, \ldots, d_n + x_n)$ satisfies Π, $\sum_{i=1}^{k} x_i = k$ and $d_i + x_i \le \Delta'$ for $i \in \{1, \ldots, n\}$. They proved that if Π-NSC is FPT when parameterized by Δ', then Π-DSC if FPT when parameterized by Δ'' where Δ'' is the maximum degree of the output graph. It is also shown that if Π-NSC can be solved in polynomial time and Π-DSC has a polynomial in k and Δ kernel, then Π-DSC has a polynomial kernel when parameterized by Δ''. Froese et al. [233] were interested only in edge additions, but it is tempting to extend their results for other modification operations, like edge deletion, contraction, and vertex deletion.

Open problem 4.13. *Investigate the (parameterized) complexity of the modification problems with the aim to satisfy some general degree sequence properties for wider sets of permitted operations.*

Some steps in this direction were done by Golovach and Mertzios [263]. They were interested in the case when the aim is to obtain a graph with the degree sequence $T = (d_1, \ldots, d_n)$ by at most k modification operations from a set

$S \subseteq \{\text{vertex deletion, edge deletion, edge addition}\}$,

and called the corresponding problem Editing to a Graph with a Given Degree Sequence(S). They proved that for any non-empty S, the problem is W[1]-hard when parameterized by k. On the positive side, it can be decided in time $2^{\mathcal{O}(k(\Delta'+k)^2)} n^{\mathcal{O}(1)}$ whether a graph with the degree sequence T can be obtained by at most k_1 vertex deletions, at most k_2 edge additions, and at most k_3 edge additions where $k_1 + k_2 + k_3 \le k$ and $\Delta' = \max T$. They furthermore show that the problem has a polynomial kernel when parameterized by $k + \Delta'$ if $S = \{\text{edge addition}\}$ and has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ in all other cases.

Bredereck et al. [240] extended some results of Froese et al. [233], Golovach and Mertzios [263], and Hartung et al. [260] for directed graphs. We already mentioned Digraph Degree Constraint Completion in the previous section but, they also considered more general Digraph Degree Constraint Sequence Completion that combines individual degree and degree sequence constraints. In this problem, we are given a directed graph, a degree list function that assigns to each vertex a set of pairs of non-negative integers from $\{0, \ldots, r\}$ that specify the desired pairs of values of in- and out-degrees of vertices, and the degree sequence property Π, and the question is whether we can add at most k arcs to obtain a directed graph with vertices whose pairs of in- and out-degree are from their lists and whose degree sequence satisfies Π. Working with directed graphs demands a great deal more efforts, but it proves that the behavior of the problems for directed and undirected graphs is essentially the same. Again, it would be interesting to extend the set of considered operations.

Open problem 4.14. *Investigate the (parameterized) complexity of the modification problems with the aim to satisfy some general* *degree sequence properties of directed graphs for wider sets of permitted modification operations.*

The related DAG Realization problem that asks whether there is a directed acyclic graph that realizes a given degree sequence was considered by Hartung and Nichterlein [264]. In particular, they showed that the problem is NP-hard and proved that it is FPT when parameterized by the maximum value in the input degree sequence.

4.3. Modification to satisfy subgraph degree constraints

In the above part of this section, we considered problems where the modification aim is to make a graph satisfy some given degree constraints. In Section 2, we considered problems where the task is to obtain a graph that does not contain a given induced subgraph. Nevertheless, it is also possible to ask the question whether we can perform modifications to achieve the property that the obtained graph *has* an induced subgraph with certain properties. In particular, the desired properties of a subgraph can include degree constraints.

An induced subgraph H of a graph G is said to be a *k-core* for a non-negative integer k if the minimum degree $\delta(H)$ of H is at least k. The introduction of this notion by Seidman [265] is motivated by the importance of k-cores in (social) networks. Intuitively, a k-core for a sufficiently large k is a "stable" part of a network. Chitnis and Talmon asked in [266] whether it is possible to create a big k-core by edge additions. Formally, the Edge k-Core problem asks, given a graph G and nonnegative integers k, p and b, whether it is possible to add at most b edges to G in such a way that the obtained graph has a k-core with at least p vertices. Chitnis and Talmon proved that this problem is NP-complete and analyzed its behavior with respect to the parameterizations by k, p, b, and the treewidth of the input graph. It is shown that Edge k-Core is W[1]-hard when parameterized by $k + p + b$, but can be solved in time $(k + \mathrm{tw})^{\mathcal{O}(b+\mathrm{tw})}n^{\mathcal{O}(1)}$, where tw is the treewidth of the input graph.

5. Miscellaneous problems

In this section, we consider several types of edge modification problems that do not fit into the framework of Sections 2–4.

5.1. Diameter augmentation

Recall that the diameter of a graph G is the longest shortest path between two vertices in a graph, that is, if $d_G(u, v)$ is the distance in G from u to v defined as the minimum number of edges (or the minimum sum of weights of edges, in the weighted case) of a (u, v)-path, then

$$\mathrm{diam}(G) = \max_{u,v \in V(G)} d_G(u, v).$$

In this way, we obtain the following completion problem.

Diameter Augmentation

Input: A graph G and non-negative integers k and d.
Question: Is it possible to obtain a graph G' with $\mathrm{diam}(G'') \leq d$ from G by adding at most k edges?

Li, McCormick, and Simchi-Levi showed that the problem is NP-hard even for $d = 2$ [267], and later Gao, Hare, and Nastos [268] proved that the problem is W[1]-hard when parameterized by k even if $d = 2$. Frati et al. [269] studied a more general weighted optimization version of Diameter Augmentation, where we have a weighted graph with a weight function $w: V(G) \times V(G) \to \mathbb{N}$, a cost function $c: V(G) \times V(G) \to \mathbb{N}$, and an integer bound B. The goal is to add a set of edges F such that $c(F) = \sum_{e \in F} c(e) \leq B$, and the diameter of $G+F$ is minimum. Frati et al. [269] gave an FPT 4-approximation algorithm running in time $3^B(n + B)^{\mathcal{O}(1)}$. They also established some inapproximability results.

Diameter Augmentation was actively investigated for graph classes, and the most famous in the parameterized framework and notoriously hard variant of the problem called Planar Diameter Augmentation was introduced by Dejter and Fellows in 1993 [270]. In this variant of the problem, the input graph is planar, the value of k is unbounded (it can be assumed that $k = 3n - 6$), and the graph obtained by adding edges should remain planar. Despite a lot of efforts, it is still unknown whether this problem can be solved in polynomial time or is NP-hard, but the most interesting question is about the parameterized complexity of the problem. Already Dejter and Fellows [270] proved that Planar Diameter Augmentation is FPT when parameterized by d. This follows from the fact that for any d, the class of planar graph $\mathcal{C}_d$ containing all graphs that can be augmented to graphs of diameter at most d is closed under taking minors. By the classical Robertson and Seymour theorem [96], $\mathcal{C}_d$ can be characterized by a finite set of forbidden minors. Together with the minor-checking algorithm of Robertson and Seymour [271], it implies that Planar Diameter Augmentation is FPT. Unfortunately, this algorithm is not uniform, because it depends on the set of forbidden minors for $\mathcal{C}_d$ that are distinct for different d and, moreover, are unknown. This lead to the following long standing open problem.

Open problem 5.1 (*[270]*). *Give a uniform constructive* FPT *algorithm for* Planar Diameter Augmentation.

In the last years, some partial results have been obtained. Interestingly, it was unknown whether Planar Diameter Augmentation can be solved by a constructive algorithm running in XP time. Lokshtanov, de Oliveira Oliveira, and Saurabh [272] considered the Plane Diameter Augmentation problem that differs from Planar Diameter Augmentation by the assumption that we are given a plane embedding of the input graph and new edges should be inserted within the faces of the embedding. They constructed an algorithm running in $n^{\mathcal{O}(d)}$ time. For the version of Plane Diameter Augmentation, where the augmented graph should be h-outerplanar, an algorithm with running time $f(d)n^{\mathcal{O}(h)}$ was given. This extends the result of Cohen et al. [273] who proved that Outerplanar Diameter Augmentation, is polynomial time solvable. For the variant of Plane Diameter Augmentation where the budget parameter k is a part of the input, Golovach, Requilé, and Thilikos [274] proved that the problem is NP-hard and FPT when parameterized by $k + d$. They also considered the variant where each face of the input graph is bounded by at most f edges and proved that Plane Diameter Augmentation is FPT when parameterized by $d + f$.

5.2. Local edge modifications

In the previous sections, we were dealing with edge modification problems where the only constraint on the set of modified edges itself was its cardinality. Nevertheless, there are problems when the set of modified edges should satisfy some additional, usually local, combinatorial property. In this subsection, we consider such problems.

Seidel's switching is a graph operation which makes a given vertex adjacent to precisely those vertices to which it was non-adjacent before, while keeping the rest of the graph unchanged.

Kratochvíl, Nešetřil, and Zýka [275] initiated the study of the problem Switching to $\mathcal{C}$, which is to decide whether a graph can be modified to belong to a given graph class $\mathcal{C}$ by a series of Seidel's switchings. There are various algorithmic and hardness results for the problem, but since we are interested in the parameterized complexity, we only mention the results of Jelínková, Suchý, Hlinený, and Kratochvíl [276]. In particular, they proved that if $\mathcal{C}$ is the class of graphs of minimum (maximum, respectively) degree at least (at most, respectively) d or the class of d-regular graphs, then the problem is FPT when parameterized by d.

If Seidel's switching complements adjacencies of a vertex, the *local complementation* introduced by Kotzig [277] complements the edges between the neighbors of a vertex. More formally, let $G^{(v)} = G[N_G(v)]$ and $\overline{G}$ denote the complement of G. Then the graph G' is obtained from G by the local complementation with respect to a vertex v if

$$G' = G - E(G^{(v)}) + E(\overline{G^{(v)}}).$$

The study of this operation is mainly motivated by its importance for vertex minors and rank-width (we refer to the work by Oum for more on this topic [278]) but, similarly to Switching to $\mathcal{C}$, we can define the Local Complementation to $\mathcal{C}$ problem. The investigation of the parameterized complexity of this problem was initiated by Cattanéo and Perdrix in [279], where they proved that the problem is W[1]-hard if $\mathcal{C}$ is the class of graphs of minimum degree at most d when parameterized by d.

Fomin et al. [280] considered complementations with respect to vertex subsets. For a set $S \subseteq V(G)$, the *partial complement* of G with respect to S is the graph G' obtained by taking the complement of $G[S]$ in G, that is, $G' = G - E(G[S]) + E(\overline{G[S]})$. For a graph class $\mathcal{C}$, they defined the Partial Complement to $\mathcal{C}$ problem that asks whether there is a partial complement of a graph G belonging to $\mathcal{C}$. Among the obtained results, they proved that Partial Complement to $\mathcal{C}$ is FPT when parameterized by w for some subclasses $\mathcal{C}$ of the graphs of clique-width at most w.

Open problem 5.2 (*[280]*). *What is the complexity of* Partial Complement to $\mathcal{C}$ *when* $\mathcal{G}$ *is*

- *the class of chordal graphs,*
- *the class of interval graphs,*
- *the class of graphs excluding a path* P_5 *as an induced subgraph,*
- *the class of graphs with minimum degree* $\geq r$ *for some constant* r*?*

Fomin, Golovach, and Thilikos in [281,282] introduced problems where the structure of the modified edges is defined by a given pattern graph H. They defined the notion of graph superposition [281]: Let G and H be graphs such that $|V(G)| \geq |V(H)|$ and let $\varphi: V(H) \to V(G)$ be an injective mapping. The graph G' is the *superposition of* G *and* H *(with respect to* φ*)* if $V(G') = V(G)$ and two vertices $u, v \in V(G')$ are adjacent in this graph if and only $uv \in E(G)$ or $u, v \in \varphi(V(H))$ and $\varphi^{-1}(u)\varphi^{-1}(v) \in E(H)$. Informally, we select $|V(H)|$ vertices in G and "glue" a copy of H into G using these vertices. They considered the Structural Connectivity and 2-Connectivity Augmentation problems that ask, given graphs G and H, whether there is a superposition of G and H such that the obtained graph is connected and 2-connected respectively. They showed a computational complexity dichotomy for the problem depending on the properties of the graph class $\mathcal{C}$ containing H. If the vertex cover number of graphs in $\mathcal{C}$ is at most t, then Structural Connectivity and 2-Connectivity Augmentation can be solved in polynomial time, that is, they are in XP when parameterized by t, and the problems are NP-hard if $\mathcal{C}$ contains graphs with arbitrarily large vertex cover number.

They further proposed a very general edge modification model [282]; the allowed changes are defined through *replacement actions*. Let $\mathcal{L}$ be a mapping that assigns to every labeled k-vertex graph H a list $L(H)$ of labeled k-vertex graphs. Then the replacement action selects a subset of k vertices S in the graph G and replaces the subgraph $G[S]$ induced by S by a graph F from the list $L(G[S])$. More precisely, the action selects a k-sized vertex subset S of G labeled by numbers $\{1, \ldots, k\}$ and, given that H is the labeled k-vertex graph obtained from $G[S]$, we select a labeled k-vertex graph F from $L(H)$ and replace H by F. Thus, the vertex set of the new graph G' is $V(G)$ and it has the same adjacencies as in G except pairs of vertices from S. In the transformed graph, vertices $u, v \in S$ labeled by $i, j \in \{1, \ldots, k\}$ are adjacent in G' if and only if $\{i, j\}$ is an edge of F. Using replacement actions we can express various modification problems. For example, we can express the deletion of at most ℓ edges as a family of actions that map graphs with at most 2ℓ vertices into graphs that can be obtained from them by at most ℓ edge deletions. Similarly, we can express edge additions. Fomin et al. considered the $\mathcal{L}$-Replacement to a Planar Graph problem, whose task is to decide, given a graph G and a positive integer k, whether there is an action that makes G planar. They proved that this problem is FPT when parameterized by k and got a number of related results where it is required to obtain a planar graph with some specific properties.

It can be seen from our brief description that, up to now, we have just a scattered set of parameterized complexity results for the aforementioned problems. We believe that these problems are natural and their systematic study for various parameterizations may lead to interesting findings.

5.3. Flip distance

Here we briefly discuss the geometric Flip Distance problem which, strictly speaking, is not defined as a graph modification problem but is closely related to our subject. Let $\mathcal{T}$ be a triangulation of a set of points $\mathcal{P}$ on the Euclidean plane. Let ABC and BCD be triangles of $\mathcal{T}$ such that $ABCD$ is a convex quadrilateral. The *flip* operation for ABC and BCD replaces these triangles by ABD and ADC, that is, the diagonal BC in the quadrilateral $ABCD$ is replaced by AD. The *flip* distance between two triangulations $\mathcal{T}_1$ and $\mathcal{T}_2$ of $\mathcal{P}$ is the minimum number of flips needed to transform $\mathcal{T}_1$ into $\mathcal{T}_2$. The Flip Distance problem asks, given two triangulations $\mathcal{T}_1$ and $\mathcal{T}_2$ of a set of points $\mathcal{P}$ and a non-negative integer k, whether the flip distance between $\mathcal{T}_1$ and $\mathcal{T}_2$ is at most k. Note that this problem can be considered as an edge modification problem on triangulated plane graphs. We refer to the survey of Bose and Hurtado [283] for the discussion of the relations between geometric and graph variants.

Lubiw and Pathak [284], and Pilz [285], independently proved that Flip Distance is NP-complete; The complexity of the problem was first stated as an open problem by Hanke, Ottmann, and Schuierer [286]. Cleary and St. John [287] initiated the study of the parameterized complexity of the problem. They considered the case when $\mathcal{P}$ defines a convex polygon and gave a kernel with $5k$ points using the relation between the flip distance and the *rotation distance* between two rooted binary trees. Rooted binary trees correspond naturally to triangulations of polygons via a standard equivalence pointed out by Sleator, Tarjan, and Thurston [288], and the flip operation described above is equivalent to a rotation of rooted trees [287]. The latter problem is studied under the name Rotation Distance. The kernel size for convex polygons was improved to $2k$ by Lucas [289]. The first FPT algorithm for the general case running in $\mathcal{O}(n + k \cdot c^k)$ time for $c \approx 2 \cdot 14^{11}$ was given by Kanj, Sedgwick, and Xia in [290]. The running time was recently improved by Li, Feng, Meng, and Wang [291].

Open problem 5.3 (*[290,291]*)**.** *Does* Flip Distance *admit a polynomial kernel when parameterized by k?*

5.4. Strong triadic closure and related problems

In the classical setting for graph editing problems, the task is to delete and/or add some edges to satisfy a certain property. There are closely related variants where the aim is to label edges of a graph to achieve a given property of labeled graphs. Considering all problems of this type is far beyond the scope of the survey and here we mention only a few of them that are related to our subject.

The notion of *triadic closure* was introduced in social network theory (see the book of Easley and Kleinberg [292] for details). In terms of graphs, this property is stated as follows. Let G be a graph, whose edges are labeled *strong* and *weak*. It is said that G satisfies the *strong triadic closure* property if for every two distinct strong edges uv and uw with a common end-vertex, $vw \in E(G)$. Informally, this means that if there are strong connections between v and u and between u and w, then there is a connection (either strong or weak) between v and w. The task of the Strong Triadic Closure problem is, given a graph G and a non-negative integer k, to decide whether there is a strong/weak labeling of the edges of G with at most k weak edges such that the labeled graph satisfies the strong triadic closure property. Observe that this problem is closely related to Cluster Deletion or P_3-Free Deletion, because for every induced path on three vertices at least one of its edges should be labeled weak.

Strong Triadic Closure is known to be NP-complete [293] and the parameterized complexity of the problem was considered by several authors [293–295]. In particular, Sintos and Tsaparas [293] observed that the problem is FPT when parameterized by k by a reduction to Vertex Cover. Golovach et al. [294] and Grüttemeier and Komusiewicz [295] observed that it admits a polynomial kernel for this parameterization. Cao and Ke [40] later proved that there is indeed a linear kernel, on $2k$ vertices, for Strong Triadic Closure.

For the dual parameterization by $\ell = |E(G)| - k$, that is, by the number of strong edges, Strong Triadic Closure is FPT but does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly [294,295]. Notice that the kernelization lower bound holds for Cluster Deletion as well.

It was observed in [294] that if M is a matching of a graph G, the edges of M are strong and the remaining edges are weak, then the labeled graph satisfies the strong triadic closure property. This means that the maximum size of a matching $\mu(G)$ gives a lower bound for the maximum number of strong edges. This lead to the following open problem.

Open problem 5.4 (*[294]*)**.** *Is* Strong Triadic Closure FPT *when parameterized by* $h = |E(G)| - k - \mu(G)$*, that is, by the number of strong edges above the maximum matching size?*

In that same article, Golovach et al. [294] proved that the problem is FPT on graph of maximum degree at most four. Notice that the question for the same parameterization is also open for Cluster Deletion. They also considered the more general variant called Strong F-Closure that is related to F-Free Deletion. Here, F is a fixed graph and the task is to label the edges of an input graph G in such a way that if the subgraph of G composed by strong edges contains a copy of F as an induced subgraph, then there is a weak edge with both end-vertices in this copy. Bulteau et al. [296] introduced another generalization, where there are c strong labels (or colors) and the constraint is that if uv and uw are distinct edges with a common end-vertex and the same strong label, then $uv \in E(G)$. In both of the aforementioned articles [294,296], the authors obtain various results that generalize the aforementioned results for Strong Triadic Closure.

Grüttemeier et al. [297] considered the Bicolored P_3-Deletion problem: given a graph G, whose edges are partitioned into two sets E_r and E_b of *red* and *blue* edges respectively, and a non-negative integer k, the task is to decide whether it is possible to delete at most k edges in such a way that the obtained graph has no bicolored induced P_3. It was proved that Bicolored P_3-Deletion can be solved in time $\mathcal{O}(1.85^k n^5)$ and has a polynomial kernel when parameterized by k and the maximum degree Δ of the input graph.

5.5. Beyond forbidden subgraphs

In Section 2, we considered editing problems whose task is to obtain a graph belonging to a given hereditary graph class, that is, a graph class defined by a family of forbidden induced subgraphs. Here we survey some variations and generalizations of these problems.

Besides forbidding induced subgraphs, it is possible to forbid other structures. In particular, there is a plethora of results for graph classes defined by families of forbidden minors or topological minors. However, these problems have been mainly investigated for vertex deletions and the results about edge deletions are corollaries. Therefore, we do not consider them in this survey. The situation is different if we forbid containment of some graphs as *immersions*. A graph H is an *immersion* of G if there is an injective mapping of the vertices of H to the vertices of G and a mapping of the edges of H to pairwise edge-disjoint paths of G such that for every two adjacent vertices u and v of H, the edge uv is mapped to a path of G whose end-vertices are the images of u and v. For a family of graphs $\mathcal{F}$, a graph G is *$\mathcal{F}$-immersion free* if H is not an immersion of G for every $H \in \mathcal{F}$. Giannopoulou et al. [298] initiated the study of the $\mathcal{F}$-Immersion Deletion problem. Given a (finite) family of graphs $\mathcal{F}$, the task is to decide whether a graph G can be made $\mathcal{F}$-immersion free by at most k edge deletions. They proved that if $\mathcal{F}$ consists of connected graphs and at least one graph in the family is planar, then $\mathcal{F}$-Immersion Deletion admits a linear kernel when parameterized by k and can be solved in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$.

Fomin, Golovach, and Thilikos [299] considered a generalization of another type in which the property that a graph G does not contain an induced subgraph isomorphic to H is *local*. The most general way to express local properties is via the *first-order logic* (FOL) formulas on graphs. Recall that the syntax of FOL-formulas on graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, variables for vertices, the quantifiers $\forall$, $\exists$ that are applied to these variables, and the adjacency and equality predicates. An FOL-formula φ is in *prenex normal form* if it is written as $\varphi = \mathsf{Q}_1 x_1 \mathsf{Q}_2 x_2 \cdots \mathsf{Q}_t x_t \chi$ where each $\mathsf{Q}_i \in \{\forall, \exists\}$ is a quantifier, x_i is a variable, and χ is a quantifier-free part. Let φ be a FOL-formula.

Edge Deletion to φ

Input: A graph G and non-negative integers k
Question: Is there a set of at most k edges $S \subseteq E(G)$, such that $G - S \models \varphi$?

The corresponding completion and editing versions are defined in the natural way, with the goal $G + F \models \varphi$, and $G \triangle F \models \varphi$, respectively. Fomin et al. [299] characterized the complexity of Edge Deletion (Completion, Editing) to φ (and the vertex deletion analogue) with respect to the prefix structure of φ, assuming that φ is in prenex normal form. More precisely, they obtained the following parameterized complexity dichotomy depending on the quantifier alternations in the prefix. If φ can be written in the form $\exists x_1 \ldots \exists x_s \forall y_1 \ldots \forall y_t \psi$ (we assume that either of the *universal* and *existential* quantification part may be empty), where ψ is a quantifier-free part, then Edge Deletion (Completion, Editing) to φ can be solved in time $|\varphi|^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(|\varphi|)}$, that is, the problem is FPT when parameterized by k. If we allow at least two quantifier alternations or one alternation but $\forall$ occurs first, then there is φ with the corresponding structure of the prefix for which the problem is W[2]-hard. Notice that the property that G has no induced subgraph isomorphic to H can be expressed in FOL. Hence, these results indeed generalize the results of Cai [33]. For kernelization, Fomin et al. [299] established a similar dichotomy: if $\varphi = \exists x_1 \ldots \exists x_s \psi$, then Edge Deletion (Completion, Editing) to φ admits a trivial kernel when parameterized by k, and for every other prefix structure, there is a formula such that the problem has no polynomial kernel unless NP $\subseteq$ coNP/poly.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

We thank the anonymous reviewers for many helpful corrections and suggestions that improved this survey article immensely. We also would like to thank Marcin Pilipczuk, William Lochet, and Dekel Tsur for helpful comments.

References

[1] John M. Lewis, Mihalis Yannakakis, The node-deletion problem for hereditary properties is NP-complete, J. Comput. System Sci. 20 (2) (1980) 219–230.
[2] Mihalis Yannakakis, Edge-deletion problems, SIAM J. Comput. 10 (2) (1981) 297–309.
[3] Pablo Burzyn, Flavia Bonomo, Guillermo Durán, NP-completeness results for edge modification problems, Discrete Appl. Math. 154 (13) (2006) 1824–1844.
[4] Federico Mancini, Graph modification problems related to graph classes (Ph.D. dissertation), University of Bergen, 2008.
[5] Assaf Natanzon, Ron Shamir, Roded Sharan, Complexity classification of some edge modification problems, Discrete Appl. Math. 113 (1) (2001) 109–128.
[6] Mihalis Yannakakis, Computing the minimum fill-in is NP-complete, SIAM J. Algebr. Discrete Methods 2 (1) (1981) 77–79.
[7] Kapali P. Eswaran, Robert E. Tarjan, Augmentation problems, SIAM J. Comput. 5 (4) (1976) 653–665.
[8] Ron Shamir, Roded Sharan, Dekel Tsur, Cluster graph modification problems, Discrete Appl. Math. 144 (1–2) (2004) 173–182.
[9] Peter L. Hammer, Bruno Simeone, The splittance of a graph, Combinatorica (1981) 275–284.
[10] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, Saket Saurabh, Parameterized Algorithms, Springer, 2015.
[11] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, Meirav Zehavi, Kernelization. Theory of Parameterized Preprocessing, Cambridge University Press, 2019.
[12] Petr A. Golovach, Pim van 't Hof, Daniël Paulusma, Obtaining planarity by contracting few edges, Theoret. Comput. Sci. 476 (2013) 38–46.
[13] Sylvain Guillemot, Dániel Marx, A faster FPT algorithm for bipartite contraction, Inform. Process. Lett. 113 (22–24) (2013) 906–912.
[14] Chengwei Guo, Leizhen Cai, Obtaining split graphs by edge contraction, Theoret. Comput. Sci. 607 (2015) 60–67.
[15] Pinar Heggernes, Pim van 't Hof, Daniel Lokshtanov, Christophe Paul, Obtaining a bipartite graph by contracting few edges, SIAM J. Discrete Math. 27 (4) (2013) 2143–2156.
[16] Leizhen Cai, Parameterized complexity of vertex colouring, Discrete Appl. Math. 127 (3) (2003) 415–429.
[17] Rodney G. Downey, Michael R. Fellows, Fixed-parameter tractability and completeness, in: Proceedings of the 21st Manitoba Conference on Numerical Mathematics and Computing, in: Congressus Numerantium, vol. 87, 1992, pp. 161–178.
[18] Jörg Flum, Martin Grohe, Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.
[19] Rodney G. Downey, Michael R. Fellows, Fundamentals of Parameterized Complexity. Texts in Computer Science, Springer, 2013.
[20] Martin Grohe, Logic, graphs, and algorithms, in: Logic and Automata—History and Perspectives, Amsterdam University Press, 2007.
[21] Rolf Niedermeier, Invitation to fixed-parameter algorithms, in: Oxford Lecture Series in Mathematics and Its Applications, vol. 31, Oxford University Press, Oxford, 2006.
[22] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, Danny Hermelin, On problems without polynomial kernels, J. Comput. System Sci. 75 (8) (2009) 423–434.
[23] Russell Impagliazzo, Ramamohan Paturi, Francis Zane, Which problems have strongly exponential complexity? J. Comput. System Sci. 63 (4) (2001) 512–530.
[24] Martin C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[25] Andreas Brandstädt, Van Bang Le, Jeremy P. Spinrad, Graph classes. A survey, in: SIAM Monographs on Discrete Mathematics and Applications, SIAM, 1999.
[26] Ken-ichi Kawarabayashi, Bruce A. Reed, Computing crossing number in linear time, in: Proceedings of the 39th Annual ACM Symposium on Theory of Computing, STOC, ACM, 2007, pp. 382–390.
[27] Daniel Lokshtanov, Wheel-free deletion is W[2]-hard, in: Proceedings of the 3rd International Workshop on Parameterized and Exact Computation, IWPEC, in: Lecture Notes in Computer Science, vol. 5018, Springer, 2008, pp. 141–147.
[28] Sylvain Guillemot, Frédéric Havet, Christophe Paul, Anthony Perez, On the (non-)existence of polynomial kernels for P_l-free edge modification problems, Algorithmica 65 (4) (2013) 900–926.
[29] Yufei Cai, Polynomial kernelisation of H-free edge modification problems (Master's thesis), The Chinese University of Hong Kong, 2012.
[30] Fedor V. Fomin, Yngve Villanger, Subexponential parameterized algorithm for minimum fill-in, SIAM J. Comput. 42 (6) (2013) 2197–2216.
[31] Pål Grønås Drange, Fedor V. Fomin, Michał Pilipczuk, Yngve Villanger, Exploring the subexponential complexity of completion problems, ACM Trans. Comput. Theory 7 (4) (2015) 14:1–14:38.
[32] Pål Grønås Drange, Michał Pilipczuk, A polynomial kernel for trivially perfect editing, Algorithmica 80 (12) (2018) 3481–3524.
[33] Leizhen Cai, Fixed-parameter tractability of graph modification problems for hereditary properties, Inform. Process. Lett. 58 (4) (1996) 171–176.
[34] Yunlong Liu, Jianxin Wang, Jiong Guo, An overview of kernelization algorithms for graph modification problems, Tsinghua Sci. Technol. 19 (4) (2014) 346–357.
[35] Faisal N. Abu-Khzam, A kernelization algorithm for d-hitting set, J. Comput. System Sci. 76 (7) (2010) 524–531.
[36] Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, Johannes Uhlmann, A more relaxed model for graph-based data clustering: s-plex cluster editing, SIAM J. Discrete Math. 24 (4) (2010) 1662–1683.
[37] Stéphane Bessy, Anthony Perez, Polynomial kernels for proper interval completion and related problems, Inform. and Comput. 231 (2013) 89–108.
[38] Pål Grønås Drange, Markus F. Dregi, Daniel Lokshtanov, Blair D. Sullivan, On the threshold of intractability, J. Comput. System Sci. 124 (2022) 1–25.
[39] Gabriel Bathie, Nicolas Bousquet, Yixin Cao, Yuping Ke, Théo Pierron, (Sub)linear kernels for edge modification problems toward structured graph classes, Algorithmica 84 (11) (2022) 3338–3364.
[40] Yixin Cao, Yuping Ke, Improved kernels for edge modification problems, in: Proceedings of the 16th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 214, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2021, pp. 13:1–13:14.
[41] Maël Dumas, Anthony Perez, Ioan Todinca, A cubic vertex-kernel for trivially perfect editing, in: Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science, MFCS, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 202, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2021, pp. 45:1–45:14.
[42] Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, Erik Jan van Leeuwen, Marcin Wrochna, Polynomial kernelization for removing induced claws and diamonds, Theory Comput. Syst. 60 (4) (2017) 615–636.
[43] Pål Grønås Drange, Parameterized Graph Modification Algorithms (Ph.D. dissertation), University of Bergen, 2015.
[44] Jianer Chen, Jie Meng, A $2k$ kernel for the cluster editing problem, J. Comput. System Sci. 78 (1) (2012) 211–220.

[45] Yixin Cao, Jianer Chen, Cluster editing: Kernelization based on edge cuts, Algorithmica 64 (1) (2012) 152–169.
[46] Daniel Brügmann, Christian Komusiewicz, Hannes Moser, On generating triangle-free graphs, Electron. Notes Discrete Math. 32 (2009) 51–58.
[47] Hanchun Yuan, Yuping Ke, Yixin Cao, Polynomial kernels for paw-free edge modification problems, Theoret. Comput. Sci. 891 (2021) 1–12.
[48] Eduard Eiben, William Lochet, Saket Saurabh, A polynomial kernel for paw-free editing, in: Proceedings of the 15th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 180, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2020, pp. 10:1–10:15.
[49] R.B. Sandeep, Naveen Sivadasan, Parameterized lower bound and improved kernel for diamond-free edge deletion, in: Proceedings of the 10th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 43, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2015, pp. 365–376.
[50] Yixin Cao, Ashutosh Rai, R.B. Sandeep, Junjie Ye, A polynomial kernel for diamond-free editing, Algorithmica 84 (1) (2022) 197–215.
[51] Dekel Tsur, Kernel for K_t-free edge deletion, Inform. Process. Lett. 167 (2021) 106082.
[52] Leizhen Cai, Yufei Cai, Incompressibility of H-free edge modification problems, Algorithmica 71 (3) (2015) 731–757.
[53] Jiong Guo, Problem kernels for NP-complete edge deletion problems: Split and related graphs, in: Proceedings of the 18th International Symposium on Algorithms and Computation, ISAAC, in: Lecture Notes in Computer Science, vol. 4835, Springer, 2007, pp. 915–926.
[54] Jens Gramm, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Data reduction and exact algorithms for clique cover, ACM J. Exp. Algorithmics 13 (2009) 2.2.2–2:2.15.
[55] Michael R. Fellows, Michael A. Langston, Frances A. Rosamond, Peter Shaw, Efficient parameterized preprocessing for cluster editing, in: Proceedings of the International Symposium on Fundamentals of Computation Theory, FCT, in: Lecture Notes in Computer Science, vol. 4639, Springer, 2007, pp. 312–321.
[56] Stefan Kratsch, Magnus Wahlström, Two edge modification problems without polynomial kernels, Discrete Optim. 10 (3) (2013) 193–199.
[57] James Nastos, Yong Gao, Bounded search tree algorithms for parametrized cograph deletion: Efficient branching rules by exploiting structures of special graph classes, Discrete Math. Algorithms Appl. 4 (01) (2012) 1250008.
[58] Yunlong Liu, Jianxin Wang, Jiong Guo, Jianer Chen, Complexity and parameterized algorithms for cograph editing, Theoret. Comput. Sci. 461 (2012) 45–54.
[59] Michael R. Fellows, Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, Johannes Uhlmann, Graph-based data clustering with overlaps, Discrete Optim. 8 (1) (2011) 2–17.
[60] Marek Cygan, Łukasz Kowalik, Marcin Pilipczuk, Open problems from Workshop on Kernels. Worker, 2013.
[61] Lowell W. Beineke, Characterizations of derived graphs, J. Combin. Theory 9 (1970) 129–135.
[62] N.R. Aravind, R.B. Sandeep, Naveen Sivadasan, On polynomial kernelization of $\mathcal{H}$-free edge deletion, Algorithmica 79 (3) (2017) 654–666.
[63] Sebastian Böcker, Jan Baumbach, Cluster editing, in: Proceedings of the 9th Conference on Computability in Europe (CiE), in: Lecture Notes in Computer Science, vol. 7921, Springer, 2013, pp. 33–44.
[64] Jens Gramm, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Graph-modeled data clustering: Exact algorithms for clique generation, Theory Comput. Syst. 38 (4) (2005) 373–392.
[65] Jiong Guo, A more effective linear kernelization for cluster editing, Theoret. Comput. Sci. 410 (8–10) (2009) 718–726.
[66] Sebastian Böcker, Peter Damaschke, Even faster parameterized cluster deletion and cluster editing, Inform. Process. Lett. 111 (14) (2011) 717–721.
[67] Sebastian Böcker, A golden ratio parameterized algorithm for cluster editing, J. Discrete Algorithms 16 (2012) 79–89.
[68] René van Bevern, Vincent Froese, Christian Komusiewicz, Parameterizing edge modification problems above lower bounds, Theory Comput. Syst. 62 (3) (2018) 739–770.
[69] Shaohua Li, Marcin Pilipczuk, Manuel Sorge, Cluster editing parameterized above modification-disjoint P_3-packings, in: Proceedings of the 38th International Symposium on Theoretical Aspects of Computer Science, STACS, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 187, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2021, pp. 49:1–49:16.
[70] Ge Xia, Yong Zhang, Kernelization for cycle transversal problems, Discrete Appl. Math. 160 (7–8) (2012) 1224–1231.
[71] Yunlong Liu, Jianxin Wang, Jie You, Jianer Chen, Yixin Cao, Edge deletion problems: Branching facilitated by modular decomposition, Theoret. Comput. Sci. 573 (2015) 63–70.
[72] Esha Ghosh, Sudeshna Kolay, Mrinal Kumar, Pranabendu Misra, Fahad Panolan, Ashutosh Rai, M.S. Ramanujan, Faster parameterized algorithms for deletion to split graphs, Algorithmica 71 (4) (2015) 989–1006.
[73] James Nastos, Yong Gao, Familial groups in social networks, Social Networks 35 (3) (2013) 439–450.
[74] Haim Kaplan, Ron Shamir, Robert E. Tarjan, Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs, SIAM J. Comput. 28 (5) (1999) 1906–1922.
[75] Assaf Natanzon, Ron Shamir, Roded Sharan, A polynomial approximation algorithm for the minimum fill-in problem, SIAM J. Comput. 30 (4) (2000) 1067–1079.
[76] Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, Meirav Zehavi, Feedback vertex set inspired kernel for chordal vertex deletion, ACM Trans. Algorithms 15 (1) (2019) 11:1–11:28.
[77] Bart M.P. Jansen, Marcin Pilipczuk, Approximation and kernelization for chordal vertex deletion, SIAM J. Discrete Math. 32 (3) (2018) 2258–2301.
[78] Yixin Cao, Dániel Marx, Chordal editing is fixed-parameter tractable, Algorithmica 75 (1) (2016) 118–137.
[79] Dániel Marx, Chordal deletion is fixed-parameter tractable, Algorithmica 57 (4) (2010) 747–768.
[80] Yngve Villanger, Pinar Heggernes, Christophe Paul, Jan Arne Telle, Interval completion is fixed parameter tractable, SIAM J. Comput. 38 (5) (2009) 2007–2020.
[81] Yixin Cao, An efficient branching algorithm for interval completion, 2013, CoRR, abs/1306.3181.
[82] Yixin Cao, Linear recognition of almost interval graphs, in: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, 2016, pp. 1096–1115.
[83] Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, Michał Pilipczuk, Subexponential parameterized algorithm for interval completion, ACM Trans. Algorithms 14 (3) (2018) 35:1–35:62.
[84] Akanksha Agrawal, Pranabendu Misra, Saket Saurabh, Meirav Zehavi, Interval vertex deletion admits a polynomial kernel, in: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, 2019, pp. 1711–1730.
[85] Yixin Cao, Dániel Marx, Interval deletion is fixed-parameter tractable, ACM Trans. Algorithms 11 (3) (2015) 21:1–21:35.
[86] Yunlong Liu, Jianxin Wang, Chao Xu, Jiong Guo, Jianer Chen, An effective branching strategy based on structural relationship among multiple forbidden induced subgraphs, J. Comb. Optim. 29 (1) (2015) 257–275.
[87] Naomi Nishimura, Prabhakar Ragde, Dimitrios M. Thilikos, On graph powers for leaf-labeled trees, J. Algorithms 42 (1) (2002) 69–108.
[88] Michael Dom, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Error compensation in leaf power problems, Algorithmica 44 (4) (2006) 363–381.
[89] Stéphane Bessy, Christophe Paul, Anthony Perez, Polynomial kernels for 3-leaf power graph modification problems, Discrete Appl. Math. 158 (16) (2010) 1732–1744.
[90] Michael Dom, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Extending the tractability border for closest leaf powers, in: Proceedings of the 31st International Workshop on Graph-Theoretic Concepts in Computer Science, WG, in: Lecture Notes in Computer Science, vol. 3787, Springer, 2005, pp. 397–408.
[91] Michael Dom, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Closest 4-leaf power is fixed-parameter tractable, Discrete Appl. Math. 156 (18) (2008) 3345–3361.
[92] Maël Dumas, Anthony Perez, Ioan Todinca, Polynomial kernels for strictly chordal edge modification problems, in: Proceedings of the 16th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 214, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2021, pp. 17:1–17:16.
[93] Christophe Crespelle, Benjamin Gras, Anthony Perez, Completion to chordal distance-hereditary graphs: A quartic vertex-kernel, in: Proceedings of the 47th International Workshop on Graph-Theoretic Concepts in Computer Science, WG, in: Lecture Notes in Computer Science, vol. 12911, Springer, 2021, pp. 156–168.
[94] Bruno Courcelle, Johann A. Makowsky, Udi Rotics, On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic, Discrete Appl. Math. 108 (1–2) (2001) 23–52.
[95] Eun Jung Kim, O-joung Kwon, A polynomial kernel for distance-hereditary vertex deletion, Algorithmica 83 (7) (2021) 2096–2141.
[96] Neil Robertson, Paul D. Seymour, Graph minors. XX. Wagner's conjecture, J. Combin. Theory Ser. B 92 (2) (2004) 325–357.
[97] Neil Robertson, Paul D. Seymour, Graph minors. XI. Circuits on a surface, J. Combin. Theory Ser. B 60 (1) (1994) 72–106.
[98] Isolde Adler, Martin Grohe, Stephan Kreutzer, Computing excluded minors, in: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, 2008, pp. 641–650.

[99] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Saket Saurabh, Planar F-deletion: Approximation, kernelization and optimal FPT algorithms, in: Proceedings of the 53rd Annual Symposium on Foundations of Computer Science, FOCS, IEEE, 2012, pp. 470–479.
[100] Bruce A. Reed, Kaleigh Smith, Adrian Vetta, Finding odd cycle transversals, Oper. Res. Lett. 32 (4) (2004) 299–301.
[101] Sebastian Wernicke, On the algorithmic tractability of single nucleotide polymorphism (SNP) analysis and related problems, in: Diplomarbeit, WSI für Informatik, Universität Tübingen, 2003.
[102] Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, Sebastian Wernicke, Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization, J. Comput. System Sci. 72 (8) (2006) 1386–1396.
[103] Marcin Pilipczuk, Michał Pilipczuk, Marcin Wrochna, Edge bipartization faster than 2^k, Algorithmica 81 (3) (2019) 917–966.
[104] Stefan Kratsch, Magnus Wahlström, Compression via matroids: A randomized polynomial kernel for odd cycle transversal, ACM Trans. Algorithms 10 (4) (2014) 20.
[105] Qilong Feng, Qian Zhou, Shaohua Li, Randomized parameterized algorithms for co-path set problem, in: Proceedings of the Annual International Workshop Frontiers in Algorithmics, FAW, in: Lecture Notes in Computer Science, vol. 8497, Springer, 2014, pp. 82–93.
[106] Pim van 't Hof, Yngve Villanger, Proper interval vertex deletion, Algorithmica 65 (4) (2013) 845–867.
[107] Pinar Heggernes, Pim van 't Hof, Bart M.P. Jansen, Stefan Kratsch, Yngve Villanger, Parameterized complexity of vertex deletion into perfect graph classes, Theoret. Comput. Sci. 511 (2013) 172–180.
[108] Dániel Marx, What's next? Future directions in parameterized complexity, in: The Multivariate Algorithmic Revolution and beyond — Essays Dedicated To Michael R. Fellows on the Occasion of His 60th Birthday, in: Lecture Notes in Computer Science, vol. 7370, Springer, 2012, pp. 469–496.
[109] Erik D. Demaine, Fedor V. Fomin, MohammadTaghi Hajiaghayi, Dimitrios M. Thilikos, Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs, J. ACM 52 (6) (2005) 866–893.
[110] Hans L. Bodlaender, Erik D. Demaine, Michael R. Fellows, Jiong Guo, Danny Hermelin, Daniel Lokshtanov, Moritz Müller, Venkatesh Raman, Johan van Rooij, Frances A. Rosamond, Open problems in parameterized and exact computation from IWPEC 2008, Technical Report UU-CS-2008-017, Department of Information and Computing Sciences, Utrecht University, 2008.
[111] Noga Alon, Daniel Lokshtanov, Saket Saurabh, Fast FAST, in: Proceedings of the 36th International Colloquium of Automata, Languages and Programming, ICALP, in: Lecture Notes in Computer Science, vol. 5555, Springer, 2009, pp. 49–58.
[112] Michael Dom, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Anke Truß, Fixed-parameter tractability results for feedback set problems in tournaments, J. Discrete Algorithms 8 (1) (2010) 76–86.
[113] Stéphane Bessy, Fedor V. Fomin, Serge Gaspers, Christophe Paul, Anthony Perez, Saket Saurabh, Stéphan Thomassé, Kernels for feedback arc set in tournaments, J. Comput. System Sci. 77 (6) (2011) 1071–1078.
[114] Hans L. Bodlaender, Pinar Heggernes, Yngve Villanger, Faster parameterized algorithms for minimum fill-in, Algorithmica 61 (4) (2011) 817–838.
[115] Vincent Bouchitté, Ioan Todinca, Treewidth and minimum fill-in: Grouping the minimal separators, SIAM J. Comput. 31 (1) (2001) 212–232.
[116] Vincent Bouchitté, Ioan Todinca, Listing all potential maximal cliques of a graph, Theoret. Comput. Sci. 276 (1–2) (2002) 17–32.
[117] Fedor V. Fomin, Dieter Kratsch, Ioan Todinca, Yngve Villanger, Exact algorithms for treewidth and minimum fill-in, SIAM J. Comput. 38 (3) (2008) 1058–1079.
[118] Christian Komusiewicz, Johannes Uhlmann, Cluster editing with locally bounded modifications, Discrete Appl. Math. 160 (15) (2012) 2259–2270.
[119] Pål Grønås Drange, Felix Reidl, Fernando S. Villaamil, Somnath Sikdar, Fast biclustering by dual parameterization, in: Proceedings of the 10th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 43, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2015, pp. 402–413.
[120] Ivan Bliznets, Marek Cygan, Paweł Komosa, Michał Pilipczuk, Lukáš Mach, Lower bounds for the parameterized complexity of minimum fill-in and other completion problems, ACM Trans. Algorithms 16 (2) (2020) 1–31.
[121] Peter Damaschke, Olof Mogren, Editing simple graphs, J. Graph Algorithms Appl. 18 (4) (2014) 557–576.
[122] N.R. Aravind, R.B. Sandeep, Naveen Sivadasan, Dichotomy results on the hardness of H-free edge modification problems, SIAM J. Discrete Math. 31 (1) (2017) 542–561.
[123] Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, Yngve Villanger, Tight bounds for parameterized complexity of cluster editing with a small number of clusters, J. Comput. System Sci. 80 (7) (2014) 1430–1447.
[124] Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, Michał Pilipczuk, A subexponential parameterized algorithm for proper interval completion, SIAM J. Discrete Math. 29 (4) (2015) 1961–1987.
[125] Hans L. Bodlaender, Pinar Heggernes, Daniel Lokshtanov, Graph modification problems (Dagstuhl Seminar 14071), Dagstuhl Rep. 4 (2) (2014) 38–59.
[126] Stephen P. Borgatti, Martin G. Everett, Models of core/periphery structures, Social Networks 21 (4) (2000) 375–395.
[127] Ivan Kovác, Ivana Selecéniová, Monika Steinová, On the clique editing problem, in: Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science, MFCS, in: Lecture Notes in Computer Science, vol. 8635, Springer, 2014, pp. 469–480.
[128] Syed Mohammad Meesum, Pranabendu Misra, Saket Saurabh, Reducing rank of the adjacency matrix by graph modification, Theoret. Comput. Sci. 654 (2016) 70–79.
[129] Syed Mohammad Meesum, Saket Saurabh, Rank reduction of oriented graphs by vertex and edge deletions, Algorithmica 80 (10) (2018) 2757–2776.
[130] Yixin Cao, R.B. Sandeep, Minimum fill-in: Inapproximability and almost tight lower bounds, Inform. and Comput. 271 (2020) 104514.
[131] Yixin Cao, Unit interval editing is fixed-parameter tractable, Inform. and Comput. 253 (2017) 109–126.
[132] Ivan Bliznets, Marek Cygan, Paweł Komosa, Michał Pilipczuk, Hardness of approximation for H-free edge modification problems, ACM Trans. Comput. Theory 10 (2) (2018) 9:1–9:32.
[133] Yijia Chen, Jörg Flum, Moritz Müller, Lower bounds for kernelizations and other preprocessing procedures, Theory Comput. Syst. 48 (4) (2011) 803–839.
[134] Henning Fernau, Till Fluschnik, Danny Hermelin, Andreas Krebs, Hendrik Molter, Rolf Niedermeier, Diminishable parameterized problems and strict polynomial kernelization, Computability 9 (1) (2020) 1–24.
[135] Duhong Chen, Oliver Eulenstein, David Fernández-Baca, Michael J. Sanderson, Minimum-flip supertrees: Complexity and algorithms, IEEE/ACM Trans. Comput. Biol. Bioinform. 3 (2) (2006) 165–173.
[136] Sebastian Böcker, Quang Bao Anh Bui, Anke Truß, Improved fixed-parameter algorithms for minimum-flip consensus trees, ACM Trans. Algorithms 8 (1) (2012).
[137] Christian Komusiewicz, Johannes Uhlmann, A cubic-vertex kernel for flip consensus tree, Algorithmica 68 (1) (2014) 81–108.
[138] Jiong Guo, Falk Hüffner, Christian Komusiewicz, Yong Zhang, Improved algorithms for bicluster editing, in: Proceedings of Theory and Applications of Models of Computation, TAMC, in: Lecture Notes in Computer Science, vol. 4978, Springer, 2008, pp. 445–456.
[139] Dekel Tsur, Faster parameterized algorithm for bicluster editing, Inform. Process. Lett. 168 (2021) 106095.
[140] Henning Fernau, Two-layer planarization: Improving on parameterized algorithmics, J. Graph Algorithms Appl. 9 (2) (2005) 205–238.
[141] Henning Fernau, Fedor V. Fomin, Daniel Lokshtanov, Matthias Mnich, Geevarghese Philip, Saket Saurabh, Social choice meets graph drawing: how to get subexponential time algorithms for ranking and drawing problems, Tsinghua Sci. Technol. 19 (4) (2014) 374–386.
[142] Yasuaki Kobayashi, Hisao Tamaki, A fast and simple subexponential fixed parameter algorithm for one-sided crossing minimization, Algorithmica 72 (3) (2015) 778–790.
[143] Meirav Zehavi, Parameterized analysis and crossing minimization problems, Comp. Sci. Rev. 45 (2022) 100490.
[144] Daniel Lokshtanov, N.S. Narayanaswamy, Venkatesh Raman, M.S. Ramanujan, Saket Saurabh, Faster parameterized algorithms using linear programming, ACM Trans. Algorithms 11 (2) (2014) 15.
[145] Marek Cygan, Łukasz Kowalik, Marcin Pilipczuk, Open problems from Update Meeting on Graph Separation Problems. Graph Cuts, 2013.
[146] Diptapriyo Majumdar, Rian Neogi, Venkatesh Raman, S. Vaishali, Tractability of Kőnig edge deletion problems, Theoret. Comput. Sci. 796 (2019) 207–215.
[147] Adrian Bock, Karthekeyan Chandrasekaran, Jochen Könemann, Britta Peis, Laura Sanitá, Finding small stabilizers for unstable graphs, Math. Program. 154 (1) (2015) 173–196.
[148] Lester R. Ford Jr., Delbert R. Fulkerson, Maximal flow through a network, Canad. J. Math. 8 (1956) 399–404.
[149] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, Mihalis Yannakakis, The complexity of multiterminal cuts, SIAM J. Comput. 23 (4) (1994) 864–894.
[150] Dániel Marx, Parameterized graph separation problems, Theoret. Comput. Sci. 351 (3) (2006) 394–406.
[151] Mingyu Xiao, Simple and improved parameterized algorithms for multiterminal cuts, Theory Comput. Syst. 46 (4) (2010) 723–736.
[152] Yixin Cao, Jianer Chen, Jia-Hao Fan, An $O^*(1.84^k)$ parameterized algorithm for the multiterminal cut problem, Inform. Process. Lett. 114 (4) (2014) 167–173.

[153] Philip N. Klein, Dániel Marx, Solving planar k-terminal cut in $\mathcal{O}(n^{c\sqrt{k}})$ time, in: Proceedings of the 39th International Colloquium of Automata, Languages and Programming, ICALP, in: Lecture Notes in Computer Science, vol. 7391, Springer, 2012, pp. 569–580.
[154] Daniel Lokshtanov, M.S. Ramanujan, Parameterized tractability of multiway cut with parity constraints, in: Proceedings of the 39th International Colloquium of Automata, Languages and Programming, ICALP, in: Lecture Notes in Computer Science, vol. 7391, Springer, 2012, pp. 750–761.
[155] Karthekeyan Chandrasekaran, Matthias Mnich, Sahand Mozaffari, Odd multiway cut in directed acyclic graphs, SIAM J. Discrete Math. 34 (2) (2020) 1385–1408.
[156] Rajesh H. Chitnis, MohammadTaghi Hajiaghayi, Dániel Marx, Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset, SIAM J. Comput. 42 (4) (2013) 1674–1696.
[157] Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, Igor Razgon, A fixed-parameter algorithm for the directed feedback vertex set problem, J. ACM 55 (5) (2008).
[158] Guy Even, Joseph (Seffi) Naor, Baruch Schieber, Madhu Sudan, Approximating minimum feedback sets and multicuts in directed graphs, Algorithmica 20 (2) (1998) 151–174.
[159] Rajesh H. Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Dániel Marx, Directed subset feedback vertex set is fixed-parameter tractable, ACM Trans. Algorithms 11 (4) (2015) 28.
[160] Mingyu Xiao, Hiroshi Nagamochi, An FPT algorithm for edge subset feedback edge set, Inform. Process. Lett. 112 (1–2) (2012) 5–9.
[161] Stefan Kratsch, Shaohua Li, Dániel Marx, Marcin Pilipczuk, Magnus Wahlström, Multi-budgeted directed cuts, Algorithmica 82 (8) (2020) 2135–2155.
[162] L. Lucchesi Claudio, Daniel Haven Younger, A minimax theorem for directed graphs, J. Lond. Math. Soc. 2–17 (3) (1978) 369–374.
[163] Naveen Garg, Vijay V. Vazirani, Mihalis Yannakakis, Primal–dual approximation algorithms for integral flow and multicut in trees, Algorithmica 18 (1) (1997) 3–20.
[164] Jiong Guo, Rolf Niedermeier, Fixed-parameter tractability and data reduction for multicut in trees, Networks 46 (3) (2005) 124–135.
[165] Nicolas Bousquet, Jean Daligault, Stéphan Thomassé, Multicut is FPT, SIAM J. Comput. 47 (1) (2018) 166–207.
[166] Dániel Marx, Igor Razgon, Fixed-parameter tractability of multicut parameterized by the size of the cutset, SIAM J. Comput. 43 (2) (2014) 355–388.
[167] Marcin Pilipczuk, Magnus Wahlström, Directed multicut is W[1]-hard, even for four terminal pairs, ACM Trans. Comput. Theory 10 (3) (2018) 1–18.
[168] Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, Magnus Wahlström, Fixed-parameter tractability of multicut in directed acyclic graphs, SIAM J. Discrete Math. 29 (1) (2015) 122–144.
[169] Rajesh H. Chitnis, Andreas E. Feldmann, FPT inapproximability of directed cut and connectivity problems, in: Proceedings of the 14th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 148, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019, pp. 8:1–8:20.
[170] Karl Bringmann, Danny Hermelin, Matthias Mnich, Erik Jan van Leeuwen, Parameterized complexity dichotomy for Steiner Multicut, J. Comput. System Sci. 82 (6) (2016) 1020–1043.
[171] Petr A. Golovach, Dimitrios M. Thilikos, Paths of bounded length and their cuts: Parameterized complexity and algorithms, Discrete Optim. 8 (1) (2011) 72–86.
[172] Till Fluschnik, Danny Hermelin, André Nichterlein, Rolf Niedermeier, Fractals for kernelization lower bounds, SIAM J. Discrete Math. 32 (1) (2018) 656–681.
[173] Pavel Dvorák, Dusan Knop, Parameterized complexity of length-bounded cuts and multicuts, Algorithmica 80 (12) (2018) 3597–3617.
[174] Cristina Bazgan, Till Fluschnik, André Nichterlein, Rolf Niedermeier, Maximilian Stahlberg, A more fine-grained complexity analysis of finding the most vital edges for undirected shortest paths, Networks 73 (1) (2019) 23–37.
[175] Matthias Bentert, Klaus Heeger, Dušan Knop, Length-bounded cuts: Proper interval graphs and structural parameters, J. Comput. System Sci. 126 (2022) 21–43.
[176] Petr Kolman, On algorithms employing treewidth for L-bounded cut problems, J. Graph Algorithms Appl. 22 (2) (2018) 177–191.
[177] Chenglin Fan, Benjamin Raichel, Gregory Van Buskirk, Metric violation distance: Hardness and approximation, Algorithmica 84 (5) (2022) 1441–1465.
[178] Chenglin Fan, Benjamin Raichel, Gregory Van Buskirk, Metric violation distance: Revisited and extended, 2018, CoRR, abs/1807.08078.
[179] Chenglin Fan, Anna C. Gilbert, Benjamin Raichel, Rishi Sonthalia, Gregory Van Buskirk, Generalized metric repair on graphs, in: Proceedings of the 17th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 162, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2020, pp. 25:1–25:22.
[180] Daniel Lokshtanov, Dániel Marx, Clustering with local restrictions, Inform. and Comput. 222 (2013) 278–292.
[181] Fedor V. Fomin, Petr A. Golovach, Janne H. Korhonen, On the parameterized complexity of cutting a few vertices from a graph, in: Proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science, MFCS, in: Lecture Notes in Computer Science, vol. 8087, Springer, 2013, pp. 421–432.
[182] Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, Saket Saurabh, Minimum bisection is fixed-parameter tractable, SIAM J. Comput. 48 (2) (2019) 417–450.
[183] René van Bevern, Andreas E. Feldmann, Manuel Sorge, Ondrej Suchý, On the parameterized complexity of computing graph bisections, in: Proceedings of the 39th International Workshop on Graph-Theoretic Concepts in Computer Science, WG, in: Lecture Notes in Computer Science, vol. 8165, Springer, 2013, pp. 76–87.
[184] Eun Jung Kim, Christophe Paul, Ignasi Sau, Dimitrios M. Thilikos, Parameterized algorithms for min–max multiway cut and list digraph homomorphism, J. Comput. System Sci. 86 (2017) 191–206.
[185] Rajesh H. Chitnis, László Egri, Dániel Marx, List H-coloring a graph by removing few vertices, Algorithmica 78 (1) (2017) 110–146.
[186] Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, Magnus Wahlström, Directed flow-augmentation, in: Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC), 2022, pp. 938–947.
[187] Rodney G. Downey, Vladimir Estivill-Castro, Michael R. Fellows, Elena Prieto, Frances A. Rosamond, Cutting up is hard to do: the parameterised complexity of k-cut and related problems, Electron. Notes Theor. Comput. Sci. 78 (2003) 209–222.
[188] Ken ichi Kawarabayashi, Mikkel Thorup, The minimum k-way cut of bounded size is fixed-parameter tractable, in: Proceedings of the 52nd Annual Symposium on Foundations of Computer Science, FOCS, IEEE Computer Society, 2011, pp. 160–169.
[189] Dániel Marx, Barry O'Sullivan, Igor Razgon, Finding small separators in linear time via treewidth reduction, ACM Trans. Algorithms 9 (4) (2013) 30:1–30:35.
[190] Christian Komusiewicz, Dieter Kratsch, Van Bang Le, Matching cut: Kernelization, single-exponential time FPT, and exact exponential algorithms, Discrete Appl. Math. 283 (2020) 44–58.
[191] Guilherme Gomes, Ignasi Sau, Finding cuts of bounded degree: Complexity, FPT and exact algorithms, and kernelization, Algorithmica 83 (6) (2021) 1677–1706.
[192] Sibabrata Ray, Jitender S. Deogun, Computational complexity of weighted integrity, J. Combin. Math. Combin. Comput. 16 (1994) 65–73.
[193] Pål Grønås Drange, Markus F. Dregi, Pim van 't Hof, On the computational complexity of vertex integrity and component order connectivity, Algorithmica 76 (4) (2016) 1181–1202.
[194] Daniel Gross, Monika Heinig, Lakshmi Iswara, L. William Kazmierczak, Kristi Luttrell, John T. Saccoman, Charles Suffel, A survey of component order connectivity models of graph theoretic networks, WSEAS Trans. Math. 12 (9) (2013) 895–910.
[195] Lane H. Clark, Roger C. Entringer, Michael R. Fellows, Computational complexity of integrity, J. Combin. Math. Combin. Comput. 2 (1987) 179–191.
[196] Michael R. Fellows, Sam Stueckle. The immersion order, Forbidden subgraphs and the complexity of network integrity, J. Combin. Math. Combin. Comput. 6 (1989) 23–32.
[197] Curtis A. Barefoot, Roger Entringer, Henda Swart, Vulnerability in graphs — a comparative survey, J. Combin. Math. Combin. Comput. 1 (38) (1987) 13–22.
[198] Jørgen Bang-Jensen, Eduard Eiben, Gregory Gutin, Magnus Wahlström, Anders Yeo, Component order connectivity in directed graphs, Algorithmica 84 (9) (2022) 2767–2784.
[199] Kunwarjit S. Bagga, Lowell W. Beineke, Marc J. Lipman, Raymond E. Pippert, Edge-integrity: a survey, Discrete Math. 124 (1–3) (1994) 3–12.
[200] Dieter Kratsch, Ton Kloks, Haiko Müller, Measuring the vulnerability for classes of intersection graphs, Discrete Appl. Math. 77 (3) (1997) 259–270.
[201] Toshimasa Watanabe, Akira Nakamura, Edge-connectivity augmentation problems, J. Comput. System Sci. 35 (1) (1987) 96–144.
[202] Guo-Ray Cai, Yu-Geng Sun, The minimum augmentation of any graph to a k-edge-connected graph, Networks 19 (1) (1989) 151–172.
[203] András Frank, Augmenting graphs to meet edge-connectivity requirements, SIAM J. Discrete Math. 5 (1) (1992) 25–53.
[204] László A. Vegh, Augmenting undirected node-connectivity by one, SIAM J. Discrete Math. 25 (2) (2011) 695–718.
[205] Bill Jackson, Tibor Jordán, Independence free graphs and vertex connectivity augmentation, J. Combin. Theory Ser. B 94 (1) (2005) 31–77.
[206] Hiroshi Nagamochi, An approximation for finding a smallest 2-edge-connected subgraph containing a specified spanning tree, Discrete Appl. Math. 126 (1) (2003) 83–113.
[207] Jiong Guo, Johannes Uhlmann, Kernelization and complexity results for connectivity augmentation problems, Networks 56 (2) (2010) 131–142.

[208] Dániel Marx, László A. Végh, Fixed-parameter algorithms for minimum-cost edge-connectivity augmentation, ACM Trans. Algorithms 11 (4) (2015) 27.
[209] Manu Basavaraju, Fedor V. Fomin, Petr A. Golovach, Pranabendu Misra, M.S. Ramanujan, Saket Saurabh, Parameterized algorithms to preserve connectivity, in: Proceedings of the 41st International Colloquium on Automata, Languages, and Programming, ICALP, in: Lecture Notes in Computer Science, vol. 8572, Springer, 2014, pp. 800–811.
[210] Falk Hüffner, Christian Komusiewicz, Manuel Sorge, Finding highly connected subgraphs, in: Proceedings of the Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM, in: Lecture Notes in Computer Science, vol. 8939, Springer, 2015, pp. 254–265.
[211] Isolde Adler, Stavros G. Kolliopoulos, Dimitrios M. Thilikos, Planar disjoint-paths completion, Algorithmica 76 (2) (2016) 401–425.
[212] Gregory Z. Gutin, M.S. Ramanujan, Felix Reidl, Magnus Wahlström, Path-contractions, edge deletions and connectivity preservation, J. Comput. System Sci. 101 (2019) 1–20.
[213] Hong Liu, Peng Zhang, Daming Zhu, On editing graphs into 2-club clusters, in: Proceedings of Joint International Conference Frontiers in Algorithmics and Algorithmic Aspects in Information and Management (FAW-AAIM), vol. 7285, Springer, 2012, pp. 235–246.
[214] Faisal N. Abu-Khzam, On the complexity of multi-parameterized cluster editing, J. Discrete Algorithms 45 (2017) 26–34.
[215] Neeldhara Misra, Fahad Panolan, Saket Saurabh, Subexponential algorithm for d-cluster edge deletion: Exception or rule? J. Comput. System Sci. 113 (2020) 150–162.
[216] Falk Hüffner, Christian Komusiewicz, Adrian Liebtrau, Rolf Niedermeier, Partitioning biological networks into highly connected clusters with maximum edge coverage, IEEE/ACM Trans. Comput. Biol. Bioinform. 11 (3) (2014) 455–467.
[217] Ivan Bliznets, Nikolay Karpov, Parameterized algorithms for partitioning graphs into highly connected clusters, in: Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 83, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2017, pp. 6:1–6:14.
[218] Petr A. Golovach, Dimitrios M. Thilikos, Clustering to given connectivities, in: Proceedings of the 14th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 115, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019, pp. 17:1–19:17.
[219] Sebastian Böcker, Sebastian Briesemeister, Quang Bao Anh Bui, Anke Truß, Going weighted: Parameterized algorithms for cluster editing, Theoret. Comput. Sci. 410 (52) (2009) 5467–5480.
[220] Junjie Luo, Hendrik Molter, André Nichterlein, Rolf Niedermeier, Parameterized dynamic cluster editing, Algorithmica 83 (1) (2021) 1–44.
[221] Jiehua Chen, Hendrik Molter, Manuel Sorge, Ondrej Suchý, Cluster editing in multi-layer and temporal graphs, in: Proceedings of the 29th International Symposium on Algorithms and Computation, ISAAC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 123, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2018, pp. 24:1–24:13.
[222] Cristiano Bocci, Chiara Capresi, Kitty Meeks, John Sylvester, A new temporal interpretation of cluster editing, in: Proceedings of the 33rd International Workshop on Combinatorial Algorithms, IWOCA, in: Lecture Notes in Computer Science, vol. 13270, Springer, 2022, pp. 214–227.
[223] Hannes Moser, Dimitrios M. Thilikos, Parameterized complexity of finding regular induced subgraphs, J. Discrete Algorithms 7 (2) (2009) 181–190.
[224] Luke Mathieson, Stefan Szeider, Editing graphs to satisfy degree constraints: A parameterized approach, J. Comput. System Sci. 78 (1) (2012) 179–191.
[225] Michael R. Garey, David S. Johnson, in: W.H. Freeman (Ed.), Computers and Intractability: A Guide To the Theory of NP-Completeness, 1979.
[226] Iain A. Stewart, Deciding whether a planar graph has a cubic subgraph is NP-complete, Discrete Math. 126 (1–3) (1994) 349–357.
[227] Iain A. Stewart, Finding regular subgraphs in both arbitrary and planar graphs, Discrete Appl. Math. 68 (3) (1996) 223–235.
[228] Iain A. Stewart, On locating cubic subgraphs in bounded-degree connected bipartite graphs, Discrete Math. 163 (1–3) (1997) 319–324.
[229] Luke Mathieson, The Parameterized Complexity of Degree Constrained Editing Problems (Ph.D. dissertation), Durham University, 2009.
[230] Markus Frick, Martin Grohe, Deciding first-order properties of locally tree-decomposable structures, J. ACM 48 (6) (2001) 1184–1206.
[231] Jannis Bulian, Anuj Dawar, Fixed-parameter tractable distances to sparse graph classes, Algorithmica 79 (1) (2017) 139–158.
[232] Petr A. Golovach, Editing to a graph of given degrees, Theoret. Comput. Sci. 591 (2015) 72–84.
[233] Vincent Froese, André Nichterlein, Rolf Niedermeier, Win-win kernelization for degree sequence completion problems, J. Comput. System Sci. 82 (6) (2016) 1100–1111.
[234] Konrad Kazimierz Dabrowski, Petr A. Golovach, Pim van 't Hof, Daniël Paulusma, Dimitrios M. Thilikos, Editing to a planar graph of given degrees, J. Comput. System Sci. 85 (2017) 168–182.
[235] Petr A. Golovach, Editing to a connected graph of given degrees, Inform. and Comput. 256 (2017) 131–147.
[236] S. Franzblau Deborah, Arundhati Raychaudhuri, Optimal Hamiltonian completions and path covers for trees, and a reduction to maximum flow, ANZIAM J. 44 (2) (2002) 193–204.
[237] Shlomo Moran, Yaron Wolfstahl, Optimal covering of cacti by vertex-disjoint paths, Theoret. Comput. Sci. 84 (2) (1991) 179–197.
[238] Fedor V. Fomin, Petr A. Golovach, Fahad Panolan, Saket Saurabh, Editing to connected F-degree graph, SIAM J. Discrete Math. 33 (2) (2019) 795–836.
[239] Øyvind Stette Haarberg, Complexity of edge editing to a connected graph of bounded degrees (Master's thesis), Department of Informatics, University of Bergen, 2019.
[240] Robert Bredereck, Vincent Froese, Marcel Koseler, Marcelo G. Millani, André Nichterlein, Rolf Niedermeier, A parameterized algorithmics framework for degree sequence completion problems in directed graphs, Algorithmica 81 (4) (2019) 1584–1614.
[241] Luke Mathieson, Graph editing problems with extended regularity constraints, Theoret. Comput. Sci. 677 (2017) 56–68.
[242] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, Dimitrios M. Thilikos, (Meta) kernelization, J. ACM 63 (5) (2016) 44:1–44:69.
[243] Francis T. Boesch, Charles L. Suffel, Ralph Tindell, The spanning subgraphs of Eulerian graphs, J. Graph Theory 1 (1) (1977) 79–84.
[244] Linda M. Lesniak, Ortrud R. Oellermann, An Eulerian exposition, J. Graph Theory 10 (3) (1986) 277–297.
[245] Frederic Dorn, Hannes Moser, Rolf Niedermeier, Mathias Weller, Efficient algorithms for Eulerian extension and rural postman, SIAM J. Discrete Math. 27 (1) (2013) 75–94.
[246] Reinhard Diestel, Graph theory, in: Graduate Texts in Mathematics, fourth ed., vol. 173, Springer, 2012.
[247] Konrad Kazimierz Dabrowski, Petr A. Golovach, Pim van 't Hof, Daniël Paulusma, Editing to Eulerian graphs, J. Comput. System Sci. 82 (2) (2016) 213–228.
[248] Leizhen Cai, Boting Yang, Parameterized complexity of even/odd subgraph problems, J. Discrete Algorithms 9 (3) (2011) 231–240.
[249] Marek Cygan, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, Ildikó Schlotter, Parameterized complexity of Eulerian deletion problems, Algorithmica 68 (1) (2014) 41–61.
[250] Prachi Goyal, Pranabendu Misra, Fahad Panolan, Geevarghese Philip, Saket Saurabh, Finding even subgraphs even faster, J. Comput. System Sci. 97 (2018) 1–13.
[251] Katarina Cechlárová, Ildikó Schlotter, Computing the deficiency of housing markets with duplicate houses, in: Proceedings of the 5th International Symposium on Parameterized and Exact Computation, IPEC, in: Lecture Notes in Computer Science, vol. 6478, Springer, 2010, pp. 72–83.
[252] Robert Crowston, Gregory Gutin, Mark Jones, Anders Yeo, Parameterized Eulerian strong component arc deletion problem on tournaments, Inform. Process. Lett. 112 (6) (2012) 249–251.
[253] Wiebke Höhn, Tobias Jacobs, Nicole Megow, On Eulerian extensions and their application to no-wait flowshop scheduling, J. Sched. 15 (3) (2012) 295–309.
[254] Manuel Sorge, René van Bevern, Rolf Niedermeier, Mathias Weller, From few components to an Eulerian graph by adding arcs, in: Proceedings of the 37th International Workshop on Graph-Theoretic Concepts in Computer Science, WG, in: Lecture Notes in Computer Science, vol. 6986, Springer, 2011, pp. 307–318.
[255] Manuel Sorge, René van Bevern, Rolf Niedermeier, Mathias Weller, A new view on rural postman based on Eulerian extension and matching, J. Discrete Algorithms 16 (2012) 12–33.
[256] Elena Zheleva, Lise Getoor, Social network data analytics, in: Social Network Data Analytics, in: chapter Privacy in Social Networks: A Survey, Springer US, 2011, pp. 277–306.
[257] Jordi Casas-Roma, Jordi Herrera-Joancomartí, Vicenç Torra, A survey of graph-modification techniques for privacy-preserving on networks, J. Artificial Intelligence Res. 47 (3) (2017) 341–366.
[258] Cristina Bazgan, Robert Bredereck, Sepp Hartung, André Nichterlein, Gerhard J. Woeginger, Finding large degree-anonymous subgraphs is hard, Theoret. Comput. Sci. 622 (2016) 90–110.
[259] Erik D. Demaine, Felix Reidl, Peter Rossmanith, Fernando S. Villaamil, Somnath Sikdar, Blair D. Sullivan, Structural sparsity of complex networks: Bounded expansion in random models and real-world graphs, J. Comput. System Sci. 105 (2019) 199–241.

[260] Sepp Hartung, André Nichterlein, Rolf Niedermeier, Ondrej Suchý, A refined complexity analysis of degree anonymization in graphs, Inform. and Comput. 243 (2015) 249–262.
[261] Robert Bredereck, Vincent Froese, Sepp Hartung, André Nichterlein, Rolf Niedermeier, Nimrod Talmon, The complexity of degree anonymization by vertex addition, Theoret. Comput. Sci. 607 (2015) 16–34.
[262] Nimrod Talmon, Sepp Hartung, The complexity of degree anonymization by graph contractions, Inform. and Comput. 256 (2017) 212–225.
[263] Petr A. Golovach, George B. Mertzios, Graph editing to a given degree sequence, Theoret. Comput. Sci. 665 (2017) 1–12.
[264] Sepp Hartung, André Nichterlein, NP-hardness and fixed-parameter tractability of realizing degree sequences with directed acyclic graphs, SIAM J. Discrete Math. 29 (4) (2015) 1931–1960.
[265] Stephen B. Seidman, Network structure and minimum degree, Social Networks 5 (3) (1983) 269–287.
[266] Rajesh H. Chitnis, Nimrod Talmon, Can we create large k-cores by adding few edges? in: Proceedings of the 13th International Computer Science Symposium in Russia (CSR), in: Lecture Notes in Computer Science, vol. 10846, Springer, 2018, pp. 78–89.
[267] Chung-Lun Li, S. Thomas McCormick, David Simchi-Levi, On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems, Oper. Res. Lett. 11 (5) (1992) 303–308.
[268] Yong Gao, Donovan R. Hare, James Nastos, The parametric complexity of graph diameter augmentation, Discrete Appl. Math. 161 (10–11) (2013) 1626–1631.
[269] Fabrizio Frati, Serge Gaspers, Joachim Gudmundsson, Luke Mathieson, Augmenting graphs to minimize the diameter, Algorithmica 72 (4) (2015) 995–1010.
[270] Italo J. Dejter, Michael R. Fellows, Improving the diameter of a planar graph, Manuscript (1993).
[271] Neil Robertson, Paul D. Seymour, Graph minors. XIII. The disjoint paths problem, J. Combin. Theory Ser. B 63 (1) (1995) 65–110.
[272] Daniel Lokshtanov, Mateus de Oliveira Oliveira, Saket Saurabh, A strongly-uniform slicewise polynomial-time algorithm for the embedded planar diameter improvement problem, in: Proceedings of the 13th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 115, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019, pp. 25:1–25:13.
[273] Nathann Cohen, Daniel Gonçalves, Eun Jung Kim, Christophe Paul, Ignasi Sau, Dimitrios M. Thilikos, Mathias Weller, A polynomial-time algorithm for outerplanar diameter improvement, J. Comput. System Sci. 89 (2017) 315–327.
[274] Petr A. Golovach, Clément Requilé, Dimitrios M. Thilikos, Variants of plane diameter completion, in: Proceedings of the 10th International Symposium on Parameterized and Exact Computation, IPEC, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 43, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2015, pp. 30–42.
[275] Jan Kratochvíl, Jaroslav Nešetřil, Ondřej Zýka, On the computational complexity of seidel's switching, in: Proceedings of the 4th Czechoslovakian Symposium on Combinatorics, Graphs and Complexity, in: Annals of Discrete Mathematics, vol. 51, North-Holland, Amsterdam, 1992, pp. 161–166.
[276] Eva Jelínková, Ondrej Suchý, Petr Hliněný, Jan Kratochvíl, Parameterized problems related to Seidel's switching, Discrete Math. Theor. Comput. Sci. 13 (2) (2011) 19–44.
[277] Anton Kotzig, Eulerian lines in finite 4-valent graphs and their transformations, in: Proceedings of the Tihany Theory of Graphs Colloquium, Academic Press, New York, 1968, pp. 219–230.
[278] Sang-il Oum, Rank-width and vertex-minors, J. Combin. Theory Ser. B 95 (1) (2005) 79–100.
[279] David Cattanéo, Simon Perdrix, Minimum degree up to local complementation: Bounds, parameterized complexity, and exact algorithms, in: Proceedings of the 26th International Symposium on Algorithms and Computation, ISAAC, in: Lecture Notes in Computer Science, vol. 9472, Springer, 2015, pp. 259–270.
[280] Fedor V. Fomin, Petr A. Golovach, Torstein J.F. Strømme, Dimitrios M. Thilikos, Partial complementation of graphs, in: Proceedings of the 16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 101, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2018, pp. 21:1–21:13.
[281] Fedor V. Fomin, Petr A. Golovach, Dimitrios M. Thilikos, Structured connectivity augmentation, SIAM J. Discrete Math. 32 (4) (2018) 2612–2635.
[282] Fedor V. Fomin, Petr A. Golovach, Dimitrios M. Thilikos, Modification to planarity is fixed parameter tractable, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 126, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019, pp. 28:1–28:17.
[283] Prosenjit Bose, Ferran Hurtado, Flips in planar graphs, Comput. Geom. 42 (1) (2009) 60–80.
[284] Anna Lubiw, Vinayak Pathak, Flip distance between two triangulations of a point set is NP-complete, Comput. Geom. 49 (2015) 17–23.
[285] Alexander Pilz, Flip distance between triangulations of a planar point set is APX-hard, Comput. Geom. 47 (5) (2014) 589–604.
[286] Sabine Hanke, Thomas Ottmann, Sven Schuierer, The edge-flipping distance of triangulations, J. UCS 2 (8) (1996) 570–579.
[287] Sean Cleary, Katherine St. John, Rotation distance is fixed-parameter tractable, Inform. Process. Lett. 109 (16) (2009) 918–922.
[288] Daniel D. Sleator, Robert E. Tarjan, William P. Thurston, Rotation distance, triangulations, and hyperbolic geometry, AMS J. Am. Math. Soc. 1 (3) (1988) 647–681.
[289] Joan M. Lucas, An improved kernel size for rotation distance in binary trees, Inform. Process. Lett. 110 (12–13) (2010) 481–484.
[290] Iyad A. Kanj, Eric Sedgwick, Ge Xia, Computing the flip distance between triangulations, Discrete Comput. Geom. 58 (2) (2017) 313–344.
[291] Qilong Feng, Shaohua Li, Xiangzhong Meng, Jianxin Wang, An improved FPT algorithm for the flip distance problem, Inform. and Comput. 281 (2021) 104708.
[292] David A. Easley, Jon M. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.
[293] Stavros Sintos, Panayiotis Tsaparas, Using strong triadic closure to characterize ties in social networks, in: Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2014, pp. 1466–1475.
[294] Petr A. Golovach, Pinar Heggernes, Athanasios L. Konstantinidis, Paloma T. Lima, Charis Papadopoulos, Parameterized aspects of strong subgraph closure, Algorithmica 82 (7) (2020) 2006–2038.
[295] Niels Grüttemeier, Christian Komusiewicz, On the relation of strong triadic closure and cluster deletion, Algorithmica 82 (4) (2020) 853–880.
[296] Laurent Bulteau, Niels Grüttemeier, Christian Komusiewicz, Manuel Sorge, Your rugby mates don't need to know your colleagues: Triadic closure with edge colors, J. Comput. System Sci. 120 (2021) 75–96.
[297] Niels Grüttemeier, Christian Komusiewicz, Jannik Schestag, Frank Sommer, Destroying bicolored P_3s by deleting few edges, Discrete Math. Theor. Comput. Sci. 23 (1) (2021).
[298] Archontia Giannopoulou, Michał Pilipczuk, Jean-Florent Raymond, Dimitrios M. Thilikos, Marcin Wrochna, Linear kernels for edge deletion problems to immersion-closed graph classes, SIAM J. Discrete Math. 35 (1) (2021) 105–151.
[299] Fedor V. Fomin, Petr A. Golovach, Dimitrios M. Thilikos, On the parameterized complexity of graph modification to first-order logic properties, Theory Comput. Syst. 64 (2) (2020) 251–271.