Structural Rounding: Approximation Algorithms for Graphs Near an Algorithmically Tractable Class

Erik D. Demaine
MIT, Cambridge, MA, USA
edemaine@mit.edu

Timothy D. Goodrich
NC State University, Raleigh, NC, USA
tdgoodri@ncsu.edu

Kyle Kloster
NC State University, Raleigh, NC, USA
kakloste@ncsu.edu

Brian Lavallee
NC State University, Raleigh, NC, USA
blavall@ncsu.edu

Quanquan C. Liu
MIT, Cambridge, MA, USA
quanquan@mit.edu

Blair D. Sullivan
NC State University, Raleigh, NC, USA
blair_sullivan@ncsu.edu

Ali Vakilian
MIT, Cambridge, MA, USA
vakilian@mit.edu

Andrew van der Poel
NC State University, Raleigh, NC, USA
ajvande4@ncsu.edu

Abstract

We develop a framework for generalizing approximation algorithms from the structural graph algorithm literature so that they apply to graphs somewhat close to that class (a scenario we expect is common when working with real-world networks) while still guaranteeing approximation ratios. The idea is to **edit** a given graph via vertex- or edge-deletions to put the graph into an algorithmically tractable class, apply known approximation algorithms for that class, and then **lift** the solution to apply to the original graph. We give a general characterization of when an optimization problem is amenable to this approach, and show that it includes many well-studied graph problems, such as Independent Set, Vertex Cover, Feedback Vertex Set, Minimum Maximal Matching, Chromatic Number, (ℓ-)Dominating Set, Edge (ℓ-)Dominating Set, and Connected Dominating Set.

To enable this framework, we develop new editing algorithms that find the approximately-fewest edits required to bring a given graph into one of a few important graph classes (in some cases these are bicriteria algorithms which simultaneously approximate both the number of editing operations and the target parameter of the family). For bounded degeneracy, we obtain an $O(r \log n)$-approximation and a bicriteria $(4, 4)$-approximation which also extends to a smoother bicriteria trade-off. For bounded treewidth, we obtain a bicriteria $(O(\log^{1.5} n), O(\sqrt{\log w}))$-approximation, and for bounded pathwidth, we obtain a bicriteria $(O(\log^{1.5} n), O(\sqrt{\log w} \cdot \log n))$-approximation. For treedepth 2 (related to bounded expansion), we obtain a 4-approximation. We also prove complementary hardness-of-approximation results assuming P $\neq$ NP: in particular, these problems are all log-factor inapproximable, except the last which is not approximable below some constant factor 2 (assuming UGC).

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases structural rounding, graph editing, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.37

Related Version `https://arxiv.org/abs/1806.02771`

Funding This research was supported in part by the Army Research Office under Grant Number W911NF-17-1-0271 to Blair D. Sullivan, the Gordon & Betty Moore Foundation's Data-Driven Discovery Initiative under Grant GBMF4560 to Blair D. Sullivan, as well as NSF grants CCF-1161626 and IIS-1546290 to Erik D. Demaine. Timothy D. Goodrich is partially supported by a National Defense Science & Engineering Graduate Program fellowship. Quanquan C. Liu is partially supported by a National Science Foundation Graduate Research Fellowship under grant 1122374.

27th Annual European Symposium on Algorithms (ESA 2019).
Editors: Michael A. Bender, Ola Svensson, and Grzegorz Herman; Article No. 37; pp. 37:1–37:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Acknowledgements We thank Abida Haque and Adam Hesterberg for helpful initial discussions, Nicole Wein for providing helpful references on bounded degeneracy problems, and Michael O'Brien for helpful comments on the manuscript.

1 Introduction

Network science has empirically established that real-world networks (social, biological, computer, etc.) exhibit sparse structure. Theoretical computer science has shown that graphs with certain structural properties enable significantly better approximation algorithms for hard problems. Unfortunately, the experimentally observed structures and the theoretically required structures are generally not the same: mathematical graph classes are rigidly defined, while real-world data is noisy and full of exceptions. This paper provides a framework for extending approximation guarantees from existing rigid classes to broader, more flexible graph families that are more likely to include real-world networks.

Specifically, we hypothesize that most real-world networks are in fact **small perturbations of graphs from a structural class**, that is, a family of graphs which adhere to some specified structure (e.g. treewidth at most w) [9, 29]. Intuitively, these perturbations may be exceptions caused by unusual/atypical behavior (e.g., weak links rarely expressing themselves), natural variation from an underlying model, or noise caused by measurement error or uncertainty. Formally, a graph is **γ-close** to a structural class $\mathcal{C}$, where $\gamma \in \mathbb{N}$, if some γ edits (e.g., vertex deletions, edge deletions, or edge contractions) bring the graph into class $\mathcal{C}$[1]. (Other papers call this the "noisy setting" [44, 12, 4].)

Our goal is to extend existing approximation algorithms for a structural class $\mathcal{C}$ to apply more broadly to graphs γ-close to $\mathcal{C}$. To achieve this goal, we need two algorithmic ingredients:

1. **Editing algorithms.** Given a graph G that is γ-close to a structural class $\mathcal{C}$, find a sequence of $f(\gamma)$ edits that result in a member of $\mathcal{C}$. When the structural class is parameterized (e.g., treewidth $\leq w$), we may also approximate those parameters.
2. **Structural rounding algorithms.** Develop approximation algorithms for optimization problems on graphs γ-close to a structural class $\mathcal{C}$ by converting ρ-approximate solutions on an edited graph in class $\mathcal{C}$ into $g(\rho, \gamma)$-approximate solutions on the original graph.

1.1 Our Results: Structural Rounding

In Section 4, we present a general metatheorem giving sufficient conditions for an optimization problem to be amenable to the structural rounding framework. Specifically, if a problem Π has an approximation algorithm in structural class $\mathcal{C}$, the problem and its solutions are "stable" under an edit operation, and there is an α-approximate algorithm for editing to $\mathcal{C}$, then we get an approximation algorithm for solving Π on graphs γ-close to $\mathcal{C}$. The new approximation algorithm incurs an additive error of $O(\gamma)$, so we preserve PTAS-like $(1+\varepsilon)$ approximation factors provided $\gamma \leq \delta\,\mathrm{OPT}_\Pi$ for a suitable constant $\delta = \delta(\varepsilon, \alpha) > 0$. Our metatheorem generalizes previous analysis of two specific problems [4].

[1] Note that the number of these edits could be super-constant. The number of edits could be as large as $O(m+n)$, the size of the graph.

For example, we obtain $(1 + O(\delta \log^{1.5} n))$-approximation algorithms for Vertex Cover, Feedback Vertex Set, Minimum Maximal Matching, and Chromatic Number on graphs $(\delta \cdot \mathrm{OPT}_\Pi(G))$-close to having treewidth w via vertex deletions (generalizing exact algorithms for bounded treewidth graphs); and we obtain a $(1 - 4\delta)/(4r + 1)$-approximation algorithm for Independent Set on graphs $(\delta \cdot \mathrm{OPT}_\Pi(G))$-close to having degeneracy r (generalizing a $1/r$-approximation for degeneracy-r graphs). These results use our new algorithms for editing to treewidth-w and degeneracy-r graph classes as summarized next.

1.2 Our Results: Editing

We develop editing approximation algorithms and/or hardness-of-approximation results for six well-studied graph classes: bounded degeneracy, bounded treewidth and pathwidth, bounded clique number, bounded treedepth, bounded weak c-coloring number, and bounded degree. Refer to the full version of this paper ([18]) for details about these classes. Table 1 summarizes our results for the bounded degeneracy and bounded treewidth classes which we use in our structural rounding framework to find approximate solutions for some classic problems. Refer to the full version of this paper ([18]) for an overview of our results for the rest of the aforementioned graph classes.

Table 1 Summary of results for $(\mathcal{C}_\lambda, \psi)$-Edit problems, i.e. finding the minimum number of ψ-edits needed to obtain a graph in class $\mathcal{C}_\lambda$ (including abbreviations and standard parameter notation). For each combination we give a shorthand problem name in bold (e.g. r-DE-V). "Approx." denotes a polynomial-time approximation or bicriteria approximation algorithm (see Section 3); "inapprox." denotes inapproximability assuming P $\neq$ NP unless otherwise specified.

Graph Family $\mathcal{C}_\lambda$	**Edit Operation ψ**	
	Vertex Deletion	**Edge Deletion**
Bounded Degeneracy (r)	**r-DE-V** $O(r \log n)$-approx. $\left(\frac{4m-\beta rn}{m-rn}, \beta\right)$-approx. $\left(\frac{1}{\varepsilon}, \frac{4}{1-2\varepsilon}\right)$-approx. ($\varepsilon < 1/2$) $o(\log(n/r))$-inapprox.	**r-DE-E** $O(r \log n)$-approx. – $\left(\frac{1}{\varepsilon}, \frac{4}{1-\varepsilon}\right)$-approx. ($\varepsilon < 1$) $o(\log(n/r))$-inapprox.
Bounded Treewidth (w)	**w-TW-V** $(O(\log^{1.5} n), O(\sqrt{\log w}))$-approx. $o(\log n)$-inapprox. for $w \in \Omega(n^{1/2})$	**w-TW-E** $(O(\log n \log\log n), O(\log w))$-approx. [4] –

1.3 Related Work

Editing to approximate optimization problems. The most closely related results are in the "noisy setting" introduced by Magen and Moharrami [44], which imagines that the "true" graph lies in the structural graph class that we want, and any extra edges observed in the given graph are "noise." In this model, Magen and Moharrami [44] developed a PTAS for estimating the *size* of Independent Set (IS) in graphs that are δn edits away from a minor-closed graph family (for sufficiently small values of δ). However, they provide no method for actually finding a solution set of vertices that achieves this approximation [44]. Later, Chan and Har-Peled [12] developed a PTAS that returns a $(1 + \varepsilon)$-approximation to IS in noisy planar graphs. More recently, Bansal et al. [4] developed an LP-based approach for noisy minor-closed IS whose runtime and approximation factor achieve better dependence on δ but only for edge edits. Moreover, they provide a similar guarantee for noisy Max k-CSPs also for edge edits [4]. Their approximation analysis resembles our general analysis of the structural rounding framework.

Another related set of approximation algorithms near a graph class are **parameterized approximations**, meaning that they run in polynomial time only when the number of edits is very small (constant or logarithmic input size). This research direction was initiated by Cai [11]; see the survey and results of Marx [46, Section 3.2] and e.g. [30, 45]. An example of one such result is a $\frac{7}{3}$-approximation algorithm to Chromatic Number in graphs that become planar after γ vertex edits, with a running time of $f(\gamma) \cdot O(n^2)$, where $f(\gamma)$ is at least $2^{2^{2^{\Omega(\gamma)}}}$ (from the use of Courcelle's Theorem), limiting its polynomial-time use to when the number of edits satisfies $\gamma = O(\lg\lg\lg\lg n)$. In contrast, our algorithms allow up to $\delta\,\mathrm{OPT}_\Pi$ edits.

Editing algorithms. Editing graphs into a desired graph class is an active field of research and has various applications outside of graph theory, including computer vision and pattern matching [28]. In general, the editing problem is to delete a minimum set X of vertices (or edges) in an input graph G such that the result $G[V \setminus X]$ has a specific property. Previous work studied this problem from the perspective of identifying the maximum induced subgraph of G that satisfies a desired "nontrivial, hereditary" property [39, 41, 42, 56]. A graph property π is nontrivial if and only if infinitely many graphs satisfy π and infinitely many do not, and π is hereditary if G satisfying π implies that every induced subgraph of G satisfies π. The vertex-deletion problem for any nontrivial, hereditary property has been shown to be NP-complete [42] and even requires exponential time to solve, assuming the ETH [37]. Approximation algorithms for such problems have also been studied in this domain [27, 43, 52], but in general this problem requires additional restrictions on the input graph and/or output graph properties in order to develop fast algorithms [17, 20, 22, 33, 38, 48, 49, 55].

Much past work on editing is on parameterized algorithms. For example, Dabrowski et al. [17] found that editing a graph to have a given degree sequence is W[1]-complete, but if one additionally requires that the final graph be planar, the problem becomes Fixed Parameter Tractable (FPT). Mathieson [48] showed that editing to degeneracy d is W[P]-hard (even if the original graph has degeneracy $d+1$ or maximum degree $2d+1$), but suggests that classes which offer a balance between the overly rigid restrictions of bounded degree and the overly global condition of bounded degeneracy (e.g., structurally sparse classes such as H-minor-free and bounded expansion [51]) may still be FPT. Some positive results on the parameterized complexity of editing to classes can be found in Drange's 2015 PhD thesis [20]; in particular, the results mentioned include parameterized algorithms for a variety of NP-complete editing problems such as editing to threshold and chain graphs [22], star forests [22], multipartite cluster graphs [25], and $\mathcal{H}$-free graphs given finite $\mathcal{H}$ and bounded indegree [21].

Our approach differs from this prior work in that we focus on approximations of edit distance that are **polynomial-time approximation algorithms**. There are previous results about approximate edit distance by Fomin et al. [26] and, in a very recent result regarding approximate edit distance to bounded treewidth graphs, by Gupta et al. [31]. Fomin et al. [26] provided two types of algorithms for vertex editing to planar $\mathcal{F}$-minor-free graphs: a randomized algorithm that runs in $O(f(\mathcal{F}) \cdot mn)$ time with an approximation constant $c_{\mathcal{F}}$ that depends on $\mathcal{F}$, as well as a fixed-parameter algorithm parameterized by the size of the edit set whose running time thus has an exponential dependence on the size of this edit set. Agrawal et al. [1] recently provided a $O(\log^{1.5} n)$-approximation via a parameterized algorithm for the Weighted $\mathcal{F}$ Vertex Deletion problem (among some other problems) where $\mathcal{F}$ is a minor-closed family excluding at least one planar graph.

Gupta et al. [31] strengthen the results in [26] but only in the context of **parameterized approximation algorithms**. Namely, they give a deterministic fixed-parameter algorithm for Planar $\mathcal{F}$-Deletion that runs in $f(\mathcal{F}) \cdot n \log n + n^{O(1)}$ time and an

$O(\log k)$-approximation where k is the maximum number of vertices in any planar graph in $\mathcal{F}$; this implies a fixed-parameter $O(\log w)$-approximation algorithm with running time $2^{O(w^2 \log w)} \cdot n \log n + n^{O(1)}$ for w-TW-V and w-PW-V. They also show that w-TW-E and w-PW-E have parameterized algorithms that give an absolute constant factor approximation but with running times parameterized by w and the maximum degree of the graph [31]. Finally, they show that when $\mathcal{F}$ is the set of all connected graphs with three vertices, deleting the minimum number of edges to exclude $\mathcal{F}$ as a subgraph, minor, or immersion is APX-hard for bounded degree graphs [31]. Again, these running times are weaker than our results, which give bicriteria approximation algorithms that are polynomial without any parameterization on the treewidth or pathwidth of the target graphs. Here, bicriteria relates to the number of editing operations and the target parameter.

In a similar regime, Bansal et al. [4] studied w-TW-E (which implies an algorithm for w-PW-E) and designed an LP-based bicriteria approximation for this problem. For a slightly different set of problems in which the goal is to exclude a single graph H of size k as a subgraph (H-Vertex-Deletion), there exists a simple k-approximation algorithm. On the hardness side, Guruswami and Lee [32] proved that whenever H is 2-vertex-connected, it is NP-hard to approximate H-Vertex-Deletion within a factor of $(|V(H)| - 1 - \varepsilon)$ for any $\varepsilon > 0$ ($|V(H)| - \varepsilon$ assuming UGC). Moreover, when H is a star or simple path with k vertices, $O(\log k)$-approximation algorithms with running time $2^{O(k^3 \log k)} \cdot n^{O(1)}$ are known [32, 40].

An important special case of the problem of editing graphs into a desired class is the *minimum planarization* problem, in which the target class is planar graphs, and the related application is approximating the well-known *crossing number* problem [15]. Refer to [7, 13, 14, 34, 36, 35, 47, 54] for the recent developments on minimum planarization and crossing number.

2 Techniques

This section summarizes the main techniques, ideas, and contributions in the paper.

2.1 Structural Rounding Framework

The main contribution of our structural rounding framework (Section 4) is establishing the right definitions that make for a broadly applicable framework with precise approximation guarantees. Our framework supports arbitrary graph edit operations and both minimization and maximization problems, provided they jointly satisfy two properties: a combinatorial property called "stability" and an algorithmic property called "structural lifting". Roughly, these properties bound the amount of change that OPT can undergo from each edit operation, but they are also parameterized to enable us to derive tighter bounds when the problem has additional structure. With the right definitions in place, the framework is simple: edit to the target class, apply an existing approximation algorithm, and lift.

The rest of Section 4 shows that this framework applies to many different graph optimization problems. In particular, we verify the stability and structural lifting properties, and combine all the necessary pieces, including our editing algorithms from Section 5 and existing approximation algorithms for structural graph classes. We summarize all of these results in Table 2 and formally define the framework in Section 4.1.

Table 2 Problems for which structural rounding (Theorem 4.4) results in approximation algorithms for graphs near the structural class $\mathcal{C}$, where the problem has a $\rho(\lambda)$-approximation algorithm. We also give the associated stability (c') and lifting (c) constants, which are class-independent. The last column shows the running time of the $\rho(\lambda)$-approximation algorithm for each problem provided an input graph from class $\mathcal{C}_\lambda$. We remark that vertex* is used to emphasize the rounding process has to pick the set of annotated vertices in the edited set carefully to achieve the associated stability and lifting constants. We provide precise problem statements in the full version of this paper ([18]).

Problem	Edit ψ	c'	c	Class $\mathcal{C}_\lambda$	$\rho(\lambda)$	runtime
Independent Set (IS)	vertex del.	1	0	degeneracy r	$\frac{1}{r+1}$	polytime
Annotated Dominating Set (ADS)	vertex* del.	0	1	degeneracy r	$O(r)$	polytime [5][a]
Independent Set (IS)	vertex del.	1	0	treewidth w	1	$O(2^w n)$ [2]
Annotated Dominating Set (ADS)	vertex* del.	0	1	treewidth w	1	$O(3^w n)$
Annotated (ℓ-)Dominating Set (ADS)	vertex* del.	0	1	treewidth w	1	$O((2\ell+1)^w n)$ [10]
Connected Dominating Set (CDS)	vertex* del.	0	3	treewidth w	1	$O(n^w)$[b]
Vertex Cover (VC)	vertex del.	0	1	treewidth w	1	$O(2^w n)$ [2]
Feedback Vertex Set (FVS)	vertex del.	0	1	treewidth w	1	$2^{O(w)} n^{O(1)}$ [16]
Minimum Maximal Matching (MMM)	vertex del.	0	1	treewidth w	1	$O(3^w n)$[c]
Chromatic Number (CRN)	vertex del.	0	1	treewidth w	1	$w^{O(w)} n^{O(1)}$
Independent Set (IS)	edge del.	0	1	degeneracy r	$\frac{1}{r+1}$	polytime
Dominating Set (DS)	edge del.	1	0	degeneracy r	$O(r)$	polytime [5]
(ℓ-)Dominating Set (DS)	edge del.	1	0	treewidth w	1	$O((2\ell+1)^w n)$ [10]
Edge (ℓ-)Dominating Set (EDS)	edge del.	1	1	treewidth w	1	$O((2\ell+1)^w n)$ [10]
Max-Cut (MC)	edge del.	1	0	treewidth w	1	$O(2^w n)$ [19]

[a] The approximation algorithm of [5] is analyzed only for DS; however, it is straightforward to show that the same algorithm achieves $O(r)$-approximation for ADS as well.
[b] Our rounding framework needs to solve an annotated version of CDS which can be solved in $O(n^w)$ by modifying the $O(w^w n)$ dynamic-programming approach of DS.
[c] The same dynamic-programming approach of DS can be modified to solve ADS and MMM in $O(3^w n)$.

2.2 Editing to Bounded Degeneracy and Degree

We first present a $O(r \log n)$-approximation algorithm for finding the fewest vertex or edge deletions to reduce the **degeneracy** to a target threshold r. The algorithm is a greedy algorithm over a type of **min-degree ordering** computed via the classic algorithm for finding the degeneracy of a graph G given by Matula and Beck [50]. In addition, we present two constant-factor bicriteria approximation algorithms for the same editing problem to degeneracy r. We provide a summary of the techniques used to obtain our results here; refer to the full version of this paper to see a detailed description of our techniques ([18]). The first approach uses the local ratio technique by Bar-Yehuda et al. [6] to establish that good-enough local choices result in a guaranteed approximation. The second approach is based on rounding a linear-programming relaxation of an integer linear program and works even when the input graph is **weighted** (both vertices and edges are weighted) and the goal is to minimize the total weight of the edit set.

On the lower bound side, we show $o(\log(n/r))$-approximation is impossible for vertex or edge edits when we forbid bicriteria approximation, i.e., when we must match the target degeneracy r exactly. This result is based on a reduction from Set Cover. A similar reduction proves $o(\log d)$-inapproximability of editing to maximum degree d, which proves tightness (up to constant factors) of a known $O(\log d)$-approximation algorithm [23].

2.3 Editing to Bounded Treewidth

We present a bicriteria approximation algorithm in the full version ([18]) for finding the fewest vertex edits to reduce the **treewidth** to a target threshold w. Our approach builds on the deep separator structure inherent in treewidth. We combine ideas from Bodlaender's $O(\log n)$-approximation algorithm for treewidth with Feige et al.'s $O(\sqrt{\log w})$-approximation algorithm for vertex separators [24] (where w is the target treewidth). In the end, we obtain a bicriteria $(O(\log^{1.5} n), O(\sqrt{\log w}))$-approximation that runs in polynomial time on all graphs (in contrast to many previous treewidth algorithms). The tree decompositions that we generate are guaranteed to have $O(\log n)$ height. As a result, we also show a bicriteria $(O(\log^{1.5} n), O(\sqrt{\log w} \cdot \log n))$-approximation result for pathwidth, based on the fact that the **pathwidth** is at most the width times the height of a tree decomposition.

On the lower bound side, we prove a $o(\log w)$-inapproximability result by another reduction from Set Cover. By a small modification, this lower bound also applies to editing to **bounded clique number**.

3 Preliminaries

This section defines several standard notions and graph classes, and is probably best used as a reference. The one exception is Section 3.1, which formally defines the graph-class editing problem $(\mathcal{C}_\lambda, \psi)$-Edit introduced in this paper.

Graph notation. We consider finite, loopless, simple graphs. Unless otherwise specified, we assume that graphs are undirected and unweighted. We denote a graph by $G = (V, E)$, and set $n = |V|$, $m = |E|$. Given $G = (V, E)$ and two vertices $u, v \in V$ we denote edges by $e(u, v)$ or (u, v). We write $N(v) = \{u \,|\, (u, v) \in E\}$ for the set of neighbors of a vertex v; the degree of v is $\deg(v) = |N(v)|$. In digraphs, in-neighbors and out-neighbors of a vertex v are defined using edges of the form (u, v) and (v, u), respectively, and we denote in- and out-degree by $\deg^-(v), \deg^+(v)$, respectively. For the maximum degree of G we use $\Delta(G)$, or just Δ if context is clear. The clique number of G, denoted $\omega(G)$, is the size of the largest clique in G. Given some subset E' of the edges in G, we define $G[E']$ to be the subgraph of G induced on the edge set E'. Note that if every edge adjacent to some vertex v is in $E \setminus E'$, then v does not appear in the vertex set of $G[E']$.

We present below our definitions of editing problems that we consider in this paper. Please refer to the full version of this paper ([18]) for complete definitions of the structural graph classes, hardness reduction techniques, and hard optimization problems for which we provide approximation algorithms.

3.1 Editing Problems

This paper is concerned with algorithms that edit graphs into a desired structural class, while guaranteeing an approximation ratio on the size of the edit set. Besides its own importance, editing graphs into structural classes plays a key role in our structural rounding framework for approximating optimization problems on graphs that are "close" to structural graph classes (see Section 4). The basic editing problem is defined as follows relative to an edit operation ψ such as vertex deletion, edge deletion, or edge contraction:

$(\mathcal{C}, \psi)$-EDIT parametrised by

Input:	An input graph $G = (V, E)$, family $\mathcal{C}$ of graphs, edit operation ψ
Problem:	Find k edits $\psi_1, \psi_2, \ldots, \psi_k$ such that $\psi_k \circ \psi_{k-1} \circ \cdots \circ \psi_2 \circ \psi_1(G) \in \mathcal{C}$.
Objective:	Minimize k

We can also loosen the graph class we are aiming for, and approximate the parameter value λ for the family $\mathcal{C}_\lambda$. Thus we obtain a **bicriteria problem** which can be formalized as follows:

$(\mathcal{C}_\lambda, \psi)$-EDIT parametrised by

Input:	An input graph $G = (V, E)$, parameterized family $\mathcal{C}_\lambda$ of graphs, a target parameter value λ^*, edit operation ψ
Problem:	Find k edits $\psi_1, \psi_2, \ldots, \psi_k$ such that $\psi_k \circ \psi_{k-1} \circ \cdots \circ \psi_2 \circ \psi_1(G) \in \mathcal{C}_\lambda$ where $\lambda \geq \lambda^*$.
Objective:	Minimize k.

▶ **Definition 3.1.** *An algorithm for $(\mathcal{C}_\lambda, \psi)$-EDIT is a **(bicriteria) (α, β)-approximation** if it guarantees that the number of edits is at most α times the optimal number of edits into $\mathcal{C}_\lambda$, and that $\lambda \leq \beta \cdot \lambda^*$.*

See the full version of this paper for a complete list of the problems considered, along with their abbreviations. Recall that $\rho(\lambda)$ is the approximation factor for a problem in class $\mathcal{C}$. We assume that $\mathcal{C}_i \subseteq \mathcal{C}_j$ for $i \leq j$, or equivalently, that $\rho(\lambda)$ is monotonically increasing in λ.

4 Structural Rounding

In this section, we show how approximation algorithms for a structural graph class can be extended to graphs that are near that class, provided we can find a certificate of being near the class. These results thus motivate our results in later sections about editing to structural graph classes. Our general approach, which we call **structural rounding**, is to apply existing approximation algorithms on the edited ("rounded") graph in the class, then "lift" that solution to solve the original graph, while bounding the loss in solution quality throughout.

4.1 General Framework

First we define our notion of "closeness" in terms of a general family ψ of allowable graph edit operations (e.g., vertex deletion, edge deletion, edge contraction):

▶ **Definition 4.1.** *A graph G' is **γ-editable** from a graph G under edit operation ψ if there is a sequence of $k \leq \gamma$ edits $\psi_1, \psi_2, \ldots, \psi_k$ of type ψ such that $G' = \psi_k \circ \psi_{k-1} \circ \cdots \circ \psi_2 \circ \psi_1(G)$. A graph G is **γ-close** to a graph class $\mathcal{C}$ under ψ if some $G' \in \mathcal{C}$ is γ-editable from G under ψ.*

To transform an approximation algorithm for a graph class $\mathcal{C}$ into an approximation algorithm for graphs γ-close to $\mathcal{C}$, we will need two properties relating the optimization problem and the type of edits:

▶ **Definition 4.2.** *A graph minimization (resp. maximization) problem* Π *is* ***stable*** *under an edit operation* ψ *with constant* c' *if* $\mathrm{OPT}_\Pi(G') \leq \mathrm{OPT}_\Pi(G) + c'\gamma$ *(resp.* $\mathrm{OPT}_\Pi(G') \geq \mathrm{OPT}_\Pi(G) - c'\gamma$*) for any graph* G' *that is* γ*-editable from* G *under* ψ*. In the special case where* $c' = 0$*, we call* Π ***closed*** *under* ψ*. When* ψ *is vertex deletion, closure is equivalent to the graph class defined by* $\mathrm{OPT}_\Pi(G) \leq \lambda$ *(resp.* $\mathrm{OPT}_\Pi(G) \geq \lambda$*) being* ***hereditary****; we also call* Π ***hereditary****.*

▶ **Definition 4.3.** *A minimization (resp. maximization) problem* Π *can be* ***structurally lifted*** *with respect to an edit operation* ψ *with constant* c *if, given any graph* G' *that is* γ*-editable from* G *under* ψ*, and given the corresponding edit sequence* $\psi_1, \psi_2, \ldots, \psi_k$ *with* $k \leq \gamma$*, a solution* S' *for* G' *can be converted in polynomial time to a solution* S *for* G *such that* $\mathrm{Cost}_\Pi(S) \leq \mathrm{Cost}_\Pi(S') + c \cdot k$ *(resp.* $\mathrm{Cost}_\Pi(S) \geq \mathrm{Cost}_\Pi(S') - c \cdot k$*).*

Now we can state the main result of structural rounding:

▶ **Theorem 4.4** (Structural Rounding Approximation)**.** *Let* Π *be a minimization (resp. maximization) problem that is stable under the edit operation* ψ *with constant* c' *and that can be structurally lifted with respect to* ψ *with constant* c*. If* Π *has a polynomial-time* $\rho(\lambda)$*-approximation algorithm in the graph class* $\mathcal{C}_\lambda$*, and* $(\mathcal{C}_\lambda, \psi)$*-*Edit *has a polynomial-time* (α, β)*-approximation algorithm, then there is a polynomial-time* $((1 + c'\alpha\delta) \cdot \rho(\beta\lambda) + c\alpha\delta)$*-approximation (resp.* $((1 - c'\alpha\delta) \cdot \rho(\beta\lambda) - c\alpha\delta)$*-approximation) algorithm for* Π *on any graph that is* $(\delta \cdot \mathrm{OPT}_\Pi(G))$*-close to the class* $\mathcal{C}_\lambda$*.*

Proof. We write $\mathrm{OPT}(G)$ for $\mathrm{OPT}_\Pi(G)$. Let G be a graph that is $(\delta \cdot \mathrm{OPT}(G))$-close to the class $\mathcal{C}_\lambda$. By Definition 3.1, the polynomial-time (α, β)-approximation algorithm finds edit operations $\psi_1, \psi_2, \ldots, \psi_k$ where $k \leq \alpha\delta \cdot \mathrm{OPT}(G)$ such that $G' = \psi_k \circ \psi_{k-1} \circ \cdots \circ \psi_2 \circ \psi_1(G) \in \mathcal{C}_{\beta\lambda}$.Let $\rho = \rho(\beta\lambda)$ be the approximation factor we can attain on the graph $G' \in \mathcal{C}_{\beta\lambda}$.

We prove the case when Π is a minimization problem. The proof of the maximization case can be found in the full version of this paper. Because Π has a ρ-approximation in $\mathcal{C}_{\beta\lambda}$ (where $\rho > 1$), we can obtain a solution S' with cost at most $\rho \cdot \mathrm{OPT}(G')$ in polynomial time. Applying structural lifting (Definition 4.3), we can use S' to obtain a solution S for G with $\mathrm{Cost}(S) \leq \mathrm{Cost}(S') + ck \leq \mathrm{Cost}(S') + c\alpha\delta \cdot \mathrm{OPT}(G)$ in polynomial time. Because Π is stable under ψ with constant c',

$$\mathrm{OPT}(G') \leq \mathrm{OPT}(G) + c'k \leq \mathrm{OPT}(G) + c'\alpha\delta \cdot \mathrm{OPT}(G) = (1 + c'\alpha\delta)\,\mathrm{OPT}(G),$$

and we have

$$\mathrm{Cost}(S) \leq \rho \cdot \mathrm{OPT}(G') + c\alpha\delta \cdot \mathrm{OPT}(G) = (\rho + \rho c'\alpha\delta + c\alpha\delta)\,\mathrm{OPT}(G),$$

proving that we have a polynomial time $(\rho + (c + c'\rho)\alpha\delta)$-approximation algorithm as required. ◀

To apply Theorem 4.4, we need four ingredients: (a) a proof that the problem of interest is stable under some edit operation (Definition 4.2); (b) a polynomial-time (α, β)-approximation algorithm for editing under this operation (Definition 3.1); (c) a structural lifting algorithm (Definition 4.3); and (d) an approximation algorithm for the target class $\mathcal{C}$.

In the remainder of this section, we show how this framework applies to many problems and graph classes, as summarized in Table 2 on page 6. Most of our approximation algorithms depend on our editing algorithms described in Section 5.

Structural rounding for annotated problems. We refer to graph optimization problems where the input consists of both a graph and subset of annotated vertices/edges as **annotated** problems. Hence, in our rounding framework, we have to carefully choose the set of annotated vertices/edges in the edited graph to guarantee small **lifting** and **stability** constants. To emphasize the difference compared to "standard" structural rounding, we denote the edit operations as vertex* and edge* in the annotated cases. Moreover, we show that we can further leverage the flexibility of annotated rounding to solve **non-annotated** problems that cannot normally be solved via structural rounding. In the full version of this paper ([18]), we consider applications of annotated rounding for both annotated problems such as Annotated Dominating Set and non-annotated problems such as Connected Dominating Set.

4.2 Applications: Vertex and Edge Deletions

For each problem, we show stability and structural liftability, and use these to conclude approximation algorithms. Using our structural rounding framework above, we obtain the following results on a broad set of problems for a number of different target classes. We point out that these problems are hard-to-solve on general graphs. Table 2 shows a summary of the set of problems we can obtain efficient approximation algorithms using structural rounding. The full version of this paper ([18]) contains the stability and structural liftability proofs used to obtain the corresponding results stated below.

We first use our structural rounding framework with vertex deletions to obtain the following approximation results.

▶ **Theorem 4.5.** *For graphs $(\delta \cdot \mathrm{OPT}(G))$-close to degeneracy r via vertex deletions,*

- Independent Set *has a $(1-4\delta)/(4r+1)$-approximation.*
- Annotated Dominating Set *has $O(r+\delta)$-approximation.*

For graphs $(\delta \cdot \mathrm{OPT}(G))$-close to treewidth w via vertex deletions:

- Annotated $(\ell$-$)$Dominating Set *has a $(1+O(\delta \log^{1.5} n))$-approximation for the case $w\sqrt{\log w} = O(\log_\ell n)$.*
- Independent Set *has a $(1-O(\delta \log^{1.5} n))$-approximation when $w\sqrt{\log w} = O(\log n)$.*
- *The problems* Vertex Cover*,* Chromatic Number*, and* Feedback Vertex Set *have $(1+O(\delta \log^{1.5} n))$-approximations when $w\sqrt{\log w} = O(\log n)$.*
- Minimum Maximal Matching *has a $(1+O(\delta \log^{1.5} n))$-approximation for the case $w \log^{1.5} w = O(\log n)$.*
- Connected Dominating Set *has a $(1+O(\delta \log^{1.5} n))$-approximation when $w = O(1)$.*

Finally, for graphs $(\delta \cdot \mathrm{OPT}(G))$-close to planar-$H$-minor-free via vertex deletions,

- Independent Set *has a $(1-c_H\delta)$-approximation.*
- *The problems* Vertex Cover*,* Minimum Maximal Matching*,* Chromatic Number*, and* Feedback Vertex Set *have $(1+c_H\delta)$-approximations.*

We now use our structural rounding framework with edge deletions to obtain the following approximation results.

▶ **Theorem 4.6.** *For graphs $(\delta \cdot \mathrm{OPT}(G))$-close to degeneracy r via edge deletions,*

- Independent Set *has a $(1/(3r+1)-3\delta)$-approximation.*
- Dominating Set *has an $O((1+\delta)r)$-approximation.*

For graphs $(\delta \cdot \mathrm{OPT}(G))$-close to treewidth w via edge deletions,

- $(\ell$-$)$Dominating Set *and* Edge $(\ell$-$)$Dominating Set *have $(1+O(\delta \log n \log\log n))$-approximations when $w \log w = O(\log_\ell n)$.*
- Max-Cut *has a $(1-O(\delta \log n \log\log n))$-approximation when $w \log w = O(\log n)$.*

Although we do not present any editing algorithms for edge contractions, we point out that such an editing algorithm would enable our framework to apply to additional problems such as (Weighted) TSP Tour, and to apply more efficiently to other problems such as Dominating Set (reducing c' from 1 to 0).

5 Editing Algorithms

5.1 Degeneracy: Greedy $O(r \log n)$-Approximation

In this section, we give a polytime $O(\log n)$-approximation for reducing the degeneracy of a graph by one using either vertex deletions or edge deletions. More specifically, given a graph $G = (V, E)$ with degeneracy $r + 1$, we produce an edit set X such that $G' = G \setminus X$ has degeneracy r and $|X|$ is at most $O(\log |V|)$ times the size of an optimal edit set. Note that this complements an $o(\log \frac{n}{r})$-inapproximability result for the same problem.

In general, the algorithm works by computing a vertex ordering and greedily choosing an edit to perform based on that ordering. In our algorithm, we use the **min-degree ordering** of a graph. The **min-degree ordering** is computed via the classic greedy algorithm given by Matula and Beck [50] that computes the degeneracy of the graph by repeatedly removing a minimum degree vertex from the graph. The degeneracy of G, degen(G), is the maximum degree of a vertex when it is removed. In the following proofs, we make use of the observation that given a min-degree ordering L of the vertices in $G = (V, E)$ and assuming the edges are oriented from smaller to larger indices in L, $\deg^+(u) \leq \text{degen}(G)$ for any $u \in L$.

The first ordering L_0 is constructed by taking a min-degree ordering on the vertices of G where ties may be broken arbitrarily. Using L_0, an edit is greedily chosen to be added to X. Each subsequent ordering L_i is constructed by taking a min-degree ordering on the vertices of $G \setminus X$ where ties are broken based on L_{i-1}. Specifically, if the vertices u and v have equal degree at the time of removal in the process of computing L_i, then $L_i(u) < L_i(v)$ if and only if $L_{i-1}(u) < L_{i-1}(v)$. The algorithm terminates when the min-degree ordering L_j produces a witness that the degeneracy of $G \setminus X$ is r.

In order to determine which edit to make at step i, the algorithm first computes the forward degree of each vertex u based on the ordering L_i (equivalently, $\deg^+(u)$ when edges are oriented from smaller to larger index in L_i). Each vertex with forward degree $r + 1$ is marked, and similarly, each edge that has a marked left endpoint is also marked. The algorithm selects the edit that **resolves** the largest number of marked edges. We say that a marked edge is **resolved** if it will not be marked in the subsequent ordering L_{i+1}.

We observe that given an optimal edit set (of size k), removing the elements of the set in any order will resolve every marked edge after k rounds (assuming that at most one element from the optimal edit set is removed in each round). If it does not, then the final ordering L_k must have a vertex with forward degree $r + 1$, a contradiction. Let m_i be the number of marked edges based on the ordering L_i. We show that we can always resolve at least $\frac{m_i}{k}$ marked edges in each round, giving our desired approximation (all proofs of this section are deferred to the full version of this paper).

By repeatedly applying the $O(\log n)$ approximation given above, we can edit a graph with arbitrary degeneracy to the class of graphs with degeneracy r.

▶ **Theorem 5.1.** *There exists an $O(r \cdot \log n)$-approximation for finding the minimum size edit set to reduce the degeneracy of a graph to r.*

5.2 Treewidth: bicriteria-approximation for vertex deletion

Our method for editing to bounded treewidth exploits the general recursive approach of the approximation algorithms for constructing a tree decomposition [3, 8, 24, 53]. Our algorithm iteratively subdivides the graph, considering $G[V_i]$ in iteration i. We first apply the result of [8, 24] (see Theorem 5.2) to determine if $G[V_i]$ has a tree decomposition with "small" width; if yes, the algorithm removes nothing and terminates. Otherwise, we compute an approximate vertex (3/4)-separator S of $G[V_i]$, remove it from the graph, and recurse on the connected components of $G[V_i \setminus S]$. The full exposition of our results for editing to bounded treewidth and pathwidth graphs are given in the full version of this paper.

Algorithm 1 Approximation for Vertex Editing to Bounded Treewidth Graphs.

procedure TREEWIDTHNODEEDIT($G = (V, E)$, w)
  $t \leftarrow$ compute $\mathsf{tw}(G)$ ▷ refer to Theorem 5.2
  if $t \leq 32c_1 \cdot w\sqrt{\log w}$ **then**
    return $\emptyset$
  else
    $S \leftarrow$ compute a vertex $(\frac{3}{4})$-separator of G by invoking the algorithm of [24]
    let $G[V_1], \cdots, G[V_\ell]$ be the connected components of $G[V \setminus S]$.
    return $\left(\bigcup_{i \leq \ell} \text{TREEWIDTHNODEEDIT}(G[V_i], w)\right) \cup S$
  end if
end procedure

▶ **Theorem 5.2** ([8, 24]). *There exists an algorithm that, given an input graph G, in polynomial time returns a tree decomposition of G of width at most $c_2 \cdot \mathsf{tw}(G)\sqrt{\log \mathsf{tw}(G)}$ and height $O(\log |V(G)|)$ for a sufficiently large constant c_2.*

Next, we analyze the performance of Algorithm 1. Our approach relies on known results for **vertex c-separators**, structures which are used extensively in many other algorithms for finding an approximate tree decomposition.

▶ **Definition 5.3.** *For a subset of vertices W, a set of vertices $S \subseteq V(G)$ is a **vertex c-separator** of W in G if each component of $G[V \setminus S]$ contains at most $c|W|$ vertices of W. The minimum sized vertex c-separator of a graph is a separator with size k, denoted $\mathsf{sep}_c(G)$, where k is the minimum integer such that for any subset $W \subseteq V$ there exists a vertex c-separator of W in G of size k.*

▶ **Theorem 5.4.** *Algorithm 1 removes at most $O(\log^{1.5} n)\,\text{OPT}_{w\text{-TW-V}}(G)$ vertices from any n-vertex graph G. The treewidth of the subgraph of G returned by Algorithm 1 is $O(w \cdot \sqrt{\log w})$.*

References

1 Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Polylogarithmic Approximation Algorithms for Weighted-F-Deletion Problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, pages 1:1–1:15, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.1.

2 Jochen Alber and Rolf Niedermeier. Improved Tree Decomposition Based Algorithms for Domination-like Problems. In *Proceedings of the 5th Latin American Symposium on Theoretical Informatics*, pages 613–627. Springer, 2001.

3 Eyal Amir. Approximation Algorithms for Treewidth. *Algorithmica*, 56(4):448–479, 2010.

4 Nikhil Bansal, Daniel Reichman, and Seeun William Umboh. LP-based robust algorithms for noisy minor-free and bounded treewidth graphs. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1964–1979. Society for Industrial and Applied Mathematics, 2017.

5 Nikhil Bansal and Seeun William Umboh. Tight approximation bounds for dominating set on graphs of bounded arboricity. *Information Processing Letters*, 122:21–24, 2017.

6 Reuven Bar-Yehuda, Keren Bendel, Ari Freund, and Dror Rawitz. Local ratio: A unified framework for approximation algorithms. *ACM Computing Surveys*, 36(4):422–463, 2004.

7 Therese Biedl, Markus Chimani, Martin Derka, and Petra Mutzel. Crossing number for graphs with bounded pathwidth. In *28th International Symposium on Algorithms and Computation, (ISAAC)*, pages 13:1–13:13, Phuket, Thailand, December 2017.

8 Hans L. Bodlaender, John R. Gilbert, Hjálmtyr Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995.

9 Geoff Boeing. Planarity and Street Network Representation in Urban Form Analysis. *arXiv preprint arXiv:1806.01805*, 2018. `arXiv:1806.01805`.

10 Glencora Borradaile and Hung Le. Optimal Dynamic Program for r-Domination Problems over Tree Decompositions. In *Proceedings of the 11th International Symposium on Parameterized and Exact Computation*, volume 63 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:23, 2017.

11 Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.

12 Timothy M Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete & Computational Geometry*, 48(2):373–392, 2012.

13 Chandra Chekuri and Anastasios Sidiropoulos. Approximation algorithms for Euler genus and related problems. In *Proceedings of the IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 167–176. IEEE, 2013.

14 Julia Chuzhoy. An algorithm for the graph crossing number problem. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 303–312, 2011.

15 Julia Chuzhoy, Yury Makarychev, and Anastasios Sidiropoulos. On graph crossing number and edge planarization. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1050–1069, 2011.

16 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Joham MM van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Foundations of computer science (focs), 2011 ieee 52nd annual symposium on*, pages 150–159. IEEE, 2011.

17 Konrad K Dabrowski, Petr A Golovach, Pim van't Hof, Daniël Paulusma, and Dimitrios M Thilikos. Editing to a planar graph of given degrees. In *Computer Science–Theory and Applications*, pages 143–156. Springer, 2015.

18 Erik D. Demaine, Timothy D. Goodrich, Kyle Kloster, Brian Lavallee, Quanquan C. Liu, Blair D. Sullivan, Ali Vakilian, and Andrew van der Poel. Structural Rounding: Approximation Algorithms for Graphs Near an Algorithmically Tractable Class. *CoRR*, abs/1806.02771, 2018. `arXiv:1806.02771`.

19 Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

20 Pål Grønås Drange. *Parameterized Graph Modification Algorithms*. PhD thesis, The University of Bergen, 2015.

21 Pål Grønås Drange, Markus S. Dregi, and R. B. Sandeep. Compressing Bounded Degree Graphs. In *LATIN*, volume 9644 of *Lecture Notes in Computer Science*, pages 362–375. Springer, 2016.

22 Pål Grønås Drange, Markus Sortland Dregi, Daniel Lokshtanov, and Blair D. Sullivan. On the Threshold of Intractability. In *Proceedings of the 23rd Annual European Symposium on Algorithms*, pages 411–423, Patras, Greece, September 2015.

23 Tomáš Ebenlendr, Petr Kolman, and Jirı Sgall. An Approximation Algorithm for Bounded Degree Deletion. *Preprint*, unpublished. URL: `http://kam.mff.cuni.cz/~kolman/papers/star.pdf`.

24 Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved Approximation Algorithms for Minimum Weight Vertex Separators. *SIAM Journal on Computing*, 38(2):629–657, 2008.

25 Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Yngve Villanger. Tight bounds for parameterized complexity of Cluster Editing with a small number of clusters. *Journal of Computer and System Sciences*, 80(7):1430–1447, 2014.

26 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-deletion: Approximation, kernelization and optimal FPT algorithms. In *Proceedings of the IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 470–479, 2012.

27 Toshihiro Fujito. A unified approximation algorithm for node-deletion problems. *Discrete Applied Mathematics*, 86(2-3):213–231, 1998.

28 Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and Applications*, 13(1):113–129, 2010.

29 Michael T Gastner and Mark EJ Newman. The spatial structure of networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 49(2):247–252, 2006.

30 Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In *International Workshop on Parameterized and Exact Computation*, pages 162–173. Springer, 2004.

31 Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michał Włodarczyk. Losing Treewidth by Separating Subsets. *CoRR*, abs/1804.01366, 2018. `arXiv:1804.01366`.

32 Venkatesan Guruswami and Euiwoong Lee. Inapproximability of H-transversal/packing. *SIAM Journal on Discrete Mathematics*, 31(3):1552–1571, 2017.

33 Falk Hüffner, Christian Komusiewicz, and André Nichterlein. Editing Graphs Into Few Cliques: Complexity, Approximation, and Kernelization Schemes. In *Proceedings of the 14th International Symposium on Algorithms and Data Structures*, pages 410–421. Springer, 2015.

34 Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1802–1811, 2014.

35 Ken-ichi Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 639–648. IEEE, 2009.

36 Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Polylogarithmic Approximation for Minimum Planarization (Almost). In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science*, pages 779–788, Berkeley, CA, October 2017.

37 Christian Komusiewicz. Tight Running Time Lower Bounds for Vertex Deletion Problems. *TOCT*, 10(2):6:1–6:18, 2018.

38 Michal Kotrbčík, Rastislav Královič, and Sebastian Ordyniak. Edge-Editing to a Dense and a Sparse Graph Class. In *Proceedings of the 12th Latin American Symposium on Theoretical Informatics*, pages 562–575. Springer, 2016.

39 Mukkai S Krishnamoorthy and Narsingh Deo. Node-deletion NP-complete problems. *SIAM Journal on Computing*, 8(4):619–625, 1979.

40 Euiwoong Lee. Partitioning a graph into small pieces with applications to path transversal. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1546–1558. Society for Industrial and Applied Mathematics, 2017.

41 John M. Lewis. On the complexity of the maximum subgraph problem. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 265–274. ACM, 1978.

42 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

43 Carsten Lund and Mihalis Yannakakis. The approximation of maximum subgraph problems. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pages 40–51. Springer, 1993.

44 Avner Magen and Mohammad Moharrami. Robust algorithms for on minor-free graphs based on the Sherali-Adams hierarchy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 258–271. Springer, 2009.

45 Dániel Marx. Parameterized coloring problems on chordal graphs. *Theoretical Computer Science*, 351(3):407–424, 2006.

46 Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.

47 Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012.

48 Luke Mathieson. The parameterized complexity of editing graphs for bounded degeneracy. *Theoretical Computer Science*, 411(34):3181–3187, 2010.

49 Luke Mathieson and Stefan Szeider. Parameterized graph editing with chosen vertex degrees. In *Combinatorial Optimization and Applications*, pages 13–22. Springer, 2008.

50 David W. Matula and Leland L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM*, 30(3):417–427, 1983.

51 J. Nešetřil and P. O. de Mendez. *Sparsity: Graphs, Structures, and Algorithms*. Algorithms and Combinatorics. Springer Berlin Heidelberg, 2012.

52 Michael Okun and Amnon Barak. A new approach for approximating node deletion problems. *Information Processing Letters*, 88(5):231–236, 2003.

53 Bruce A. Reed. Finding Approximate Separators and Computing Tree Width Quickly. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 221–228, 1992.

54 Marcus Schaefer. The graph crossing number and its variants: A survey. *The Electronic Journal of Combinatorics*, 1000:21–22, 2013.

55 Mingyu Xiao. A Parameterized Algorithm for Bounded-Degree Vertex Deletion. In *Proceedings of the 22nd International Conference on Computing and Combinatorics*, pages 79–91, 2016.

56 Mihalis Yannakakis. Node-and edge-deletion NP-complete problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 253–264, 1978.