# Editing to a Planar Graph of Given Degrees ⋆

Konrad K. Dabrowski[1], Petr A. Golovach[2], Pim van 't Hof[3],
Daniël Paulusma[1], and Dimitrios M. Thilikos[4]

[1] School of Engineering and Computing Sciences, Durham University, United Kingdom. E-mail:{konrad.dabrowski,daniel.paulusma}@durham.ac.uk
[2] Department of Informatics, University of Bergen, Norway. E-mail: petr.golovach@ii.uib.no
[3] School of Built Environment, Rotterdam University of Applied Sciences, Rotterdam, the Netherlands. E-mail:p.van.t.hof@hr.nl
[4] Computer Technology Institute and Press "Diophantus", Patras, Greece, Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece and AlGCo project-team, CNRS, LIRMM, Montpellier, France. E-mail: sedthilk@thilikos.info

**Abstract.** We consider the following graph modification problem. Let the input consist of a graph $G = (V, E)$, a weight function $w\colon V \cup E \to \mathbb{N}$, a cost function $c\colon V \cup E \to \mathbb{N}$ and a degree function $\delta\colon V \to \mathbb{N}_0$, together with three integers $k_v$, $k_e$ and $C$. The question is whether we can delete a set of vertices of total weight at most $k_v$ and a set of edges of total weight at most $k_e$ so that the total cost of the deleted elements is at most $C$ and every non-deleted vertex $v$ has degree $\delta(v)$ in the resulting graph $G'$. We also consider the variant in which $G'$ must be connected. Both problems are known to be NP-complete and W[1]-hard when parameterized by $k_v + k_e$. We prove that, when restricted to planar graphs, they stay NP-complete but have polynomial kernels when parameterized by $k_v + k_e$.

## 1 Introduction

Graph modification problems capture a variety of graph-theoretic problems and are well studied in algorithmic graph theory. The aim is to modify some given graph $G$ into some other graph $H$ that satisfies a *certain property* by applying a bounded number of operations from a set $S$ of *prespecified graph operations*. Well-known graph operations are the edge addition, edge deletion and vertex deletion, denoted by ea, ed and vd, respectively. For example, if $S = \{\mathsf{vd}\}$ and $H$ must be a clique or independent set then we obtain the basic problems Clique

⋆ The first and fourth author were supported by EPSRC Grant EP/K025090/1. The research of the second author has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement n. 267959. The research of the fifth author was co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: ARISTEIA II.

and Independent Set, respectively. To give a few more examples, if $H$ must be a forest and $S = \{\mathsf{ed}\}$ or $S = \{\mathsf{vd}\}$ then we obtain the problems Feedback Edge Set and Feedback Vertex Set, respectively. As discussed in detail later, it is also common to consider sets $S$ consisting of more than one graph operation.

A property is *hereditary* if it holds for any induced subgraph of a graph that satisfies it, and a property is *non-trivial* if it is both true for infinitely many graphs and false for infinitely many graphs. A classic result of Lewis and Yannakakis [21] is that a vertex deletion problem is NP-hard for any property that is hereditary and non-trivial. In an earlier paper Yannakakis [27] also showed that the edge deletion problem is NP-complete for several properties, such as being planar or outer-planar. Natanzon, Shamir and Sharan [24] and Burzyn, Bonomo and Durán [5] proved that the graph modification problem is NP-complete when $S = \{\mathsf{ea}, \mathsf{ed}\}$ and the desired property is to belong to some hereditary graph class for a variety of such graph classes.

When a problem turns out to be NP-hard, a possible next step might be to consider it in the more refined framework offered by *parameterized complexity*. This is certainly an appropriate direction to follow for graph modification problems, because the bound on the total number of permitted operations is a natural parameter $k$. Cai [6] proved that for this parameter the graph modification problem is FPT if $S = \{\mathsf{ea}, \mathsf{ed}, \mathsf{vd}\}$ and the desired property is to belong to any fixed graph class characterized by a finite set of forbidden induced subgraphs. Khot and Raman [19] determined all non-trivial hereditary properties for which the vertex deletion problem is FPT on $n$-vertex graphs with parameter $n - k$ and proved that for all other such properties the problem is W[1]-hard (when parameterized by $n - k$).

From the aforementioned results we conclude that the graph modification problem has been thoroughly studied for hereditary properties. However, for other types of properties, much less is known. Dabrowski et al. [9] combined previous results [4, 7, 8] with new results to classify the (parameterized) complexity of the problem of modifying the input graph into a connected graph where each vertex has some prescribed degree parity for all $S \subseteq \{\mathsf{ea}, \mathsf{ed}, \mathsf{vd}\}$.

In this paper we consider the case when the vertices of the resulting graph must satisfy some prespecified degree constraints (note that such properties are non-hereditary, so the results of Lewis and Yannakakis do not apply to this case). Before presenting our results, we briefly discuss the known results and the general framework they fall under.

Moser and Thilikos in [23] and Mathieson and Szeider [22] initiated an investigation into the parameterized complexity of such graph modification problems. In particular, Mathieson and Szeider [22] introduced the following general problem.

Degree Constraint Editing($S$)
*Instance:* A graph $G$, integers $d, k$ and a function $\delta\colon V(G) \to \{1, \ldots, d\}$.
*Question:* Can $G$ be modified into a graph $G'$ such that $d_{G'}(v) = \delta(v)$ for each $v \in V(G')$ using at most $k$ operations from the set $S$?

Mathieson and Szeider [22] classified the parameterized complexity of this problem for $S \subseteq \{\mathsf{ea}, \mathsf{ed}, \mathsf{vd}\}$. In particular they showed the following results. If $S \subseteq \{\mathsf{ea}, \mathsf{ed}\}$ then the problem is polynomial-time solvable. If $\mathsf{vd} \in S$ then the problem is $\mathsf{NP}$-complete, $\mathsf{W}[1]$-hard with parameter $k$ and $\mathsf{FPT}$ with parameter $d + k$. Moreover, they proved that the latter result holds even for a more general version, in which the vertices and edges have costs and the desired degree for each vertex should be in some given subset of $\{1, \ldots, d\}$. If $S \subseteq \{\mathsf{ed}, \mathsf{vd}\}$, they proved that the problem has a polynomial kernel when parameterized by $d + k$. Golovach [18] considered the cases $S = \{\mathsf{ea}, \mathsf{vd}\}$ and $S = \{\mathsf{ea}, \mathsf{ed}, \mathsf{vd}\}$ and proved (amongst other results) that for these cases the problem has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP}/\text{poly}$. Froese, Nichterlein and Niedermeier [13] gave more kernelization results for Degree Constraint Editing($S$). Golovach [17] introduced a variant of Degree Constraint Editing($S$) in which we additionally insist that the resulting graph must be *connected*. He proved that, for $S = \{\mathsf{ea}\}$, this variant is $\mathsf{NP}$-complete, $\mathsf{FPT}$ when parameterized by $k$, and has a polynomial kernel when parameterized by $k + d$. The connected variant is readily seen to be $\mathsf{W}[1]$-hard when $\mathsf{vd} \in S$ by a straightforward modification of the proof of the $\mathsf{W}[1]$-hardness result for Degree Constraint Editing($S$), when $\mathsf{vd} \in S$, as given by Mathieson and Szeider [22].

In the light of the above $\mathsf{NP}$-completeness and $\mathsf{W}[1]$-hardness results (when $\mathsf{vd} \in S$) it is natural to restrict the input graph $G$ to a special graph class. Hence, inspired by the above results, we consider the set $S = \{\mathsf{ed}, \mathsf{vd}\}$ and study weighted versions of both variants (where we insist that the resulting graph is connected and where we don't) of these problems for *planar* input graphs. In fact the problems we study are even more general. The problem variant not demanding connectivity is defined as follows.

| Deletion to a Planar Graph of Given Degrees (DPGGD) | |
|---|---|
| *Instance:* | A planar graph $G = (V, E)$, integers $k_v, k_e, C$ and functions $\delta\colon V \to \mathbb{N}_0$, $w\colon V \cup E \to \mathbb{N}$, $c\colon V \cup E \to \mathbb{N}_0$. |
| *Question:* | Can $G$ be modified into a graph $G'$ by deleting a set $U \subseteq V$ with $w(U) \leq k_v$ and a set $D \subseteq E$ with $w(D) \leq k_e$ such that $c(U \cup D) \leq C$ and $d_{G'}(v) = \delta(v)$ for $v \in V(G')$? |

In the above problem, $w$ is the *weight* and $c$ is the *cost* function. The question is whether it is possible to delete vertices and edges of total weight at most $k_v$ and $k_e$, respectively, so that the total cost of the deleted elements is at most $C$ and the obtained graph satisfies the degree restrictions prescribed by the given function $\delta$.

The second problem we consider is the variant of DPGGD, in which the desired graph $G'$ must be connected. We call this variant the Deletion to a Connected Planar Graph of Given Degrees problem (DCPGGD).

**Our Results**. We note that DPGGD is $\mathsf{NP}$-complete even if $\delta \equiv 3$, $w \equiv 1$, $c \equiv 0$ and $k_v = |V(G)| - 1$, and DCPGGD is $\mathsf{NP}$-complete even if $\delta \equiv 2$, $w \equiv 1$, $c \equiv 0$ and $k_v = 0$. These observations follow directly from the respective facts that both testing whether a planar graph of degree at most 7 has a non-trivial cubic subgraph [26] and testing whether a cubic planar graph has a Hamiltonian cycle [14] is NP-complete. In contrast to the aforementioned W[1]-hardness results for general graphs, our two main results are that both DPGGD and DCPGGD have polynomial kernels when parameterized by $k_v + k_e$. Note that the integer $C$ is neither a constant nor a parameter but part of the input. In order to obtain our results we first show that both problems are polynomial-time solvable for any graph class of bounded treewidth. We then use the *protrusion decomposition/replacement* techniques introduced by Bodlaender at al. [2] (see [3] for the full text). These techniques were successfully used for various problems on sparse graphs [12, 15, 16, 20]. We stress that DPGGD and DCPGGD do not fit in the meta-kernelization framework of Bodlaender at al. [2]. Hence our approach is, unavoidably, problem-specific.

## 2 Preliminaries

All graphs in this paper are finite, undirected and without loops or multiple edges. The vertex set of a graph $G$ is denoted by $V(G)$ and the edge set is denoted by $E(G)$. For a set $X \subseteq V(G)$, we let $G[X]$ denote the subgraph of $G$ induced by $X$. We write $G - X = G[V(G) \setminus X]$; we allow the case where $X \not\subseteq V(G)$. If $X = \{x\}$, we may write $G - x$ instead. For a set $L \subseteq E(G)$, we let $G - L$ be the graph obtained from $G$ by deleting all edges of $L$. If $L = \{e\}$ then we write $G - e$ instead. For $v \in V(G)$, let $E_G(v) = \{e \in E(G) \mid e \text{ is incident to } v\}$. For $X \subseteq V(G)$, let $E_G(X) = \bigcup_{v \in X} E_G(v)$. For $e \in E(G)$ with $e = uv$, let $V(e) = \{u, v\}$. For a set $L \subseteq E(G)$ let $V(L) = \cup_{e \in L} V(e)$.

Let $G$ be a graph. For a vertex $v$, we let $N_G(v)$ denote its *(open) neighbourhood*, that is, the set of vertices adjacent to $v$. The *degree* of a vertex $v$ is denoted by $d_G(v) = |N_G(v)|$. For a set $X \subseteq V(G)$, we write $N_G(X) = (\bigcup_{v \in X} N_G(v)) \setminus X$. The *closed neighbourhood* $N_G[v] = N_G(v) \cup \{v\}$, and for a positive integer $r$, $N_G^r[v]$ is the set of vertices at distance at most $r$ from $v$; note that $N_G^0[v] = \{v\}$ and that $N_G^1[v] = N_G[v]$. For a set $X \subseteq V(G)$ and a positive integer $r$, let $N_G^r[X] = \bigcup_{v \in X} N_G^r[v]$. For a positive integer $r$, a set $X \subseteq V(G)$ is an *$r$-dominating* set of $G$ if $V(G) \subseteq N_G^r[X]$. For a set $X \subseteq V(G)$, $\partial_G(X) = X \cap N_G(V(G) \setminus X)$ is the *boundary* of $X$ in $G$.

A *tree decomposition* of a graph $G$ is a pair $(\mathcal{X}, T)$ where $T$ is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a collection of subsets (called *bags*) of $V(G)$ such that

(i) $\bigcup_{i \in V(T)} X_i = V(G)$,
(ii) for each edge $xy \in E(G)$, $x, y \in X_i$ for some $i \in V(T)$, and
(iii) for each $x \in V(G)$, the set $\{i \mid x \in X_i\}$ induces a connected subtree of $T$.

The *width* of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is $\max_{i \in V(T)} \{|X_i| - 1\}$. The *treewidth* of a graph $G$ (denoted $\mathbf{tw}(G)$) is the minimum width over all tree decompositions of $G$.

We need the following known observation, which is valid for every planar bipartite graph $G$ in which the vertices of one partition class $V_2$ have degree at least 3 (in order to prove this, note that $3|V_2| \leq \sum_{v \in V_2} d_G(v) = |E(G)| \leq 2|V(G)| - 4$, as $G$ is bipartite and planar).

**Lemma 1.** *Let $V_1$ and $V_2$ be bipartition classes of a planar bipartite graph $G$ such that $d_G(v) \geq 3$ for every $v \in V_2$ and $V_2$ is non-empty. Then $|V_2| \leq 2|V_1| - 4$.*

**Protrusion decompositions.** For a graph $G$ a positive integer $r$, a set $X \subseteq V(G)$ is an *$r$-protrusion* of $G$ if $|\partial_G(X)| \leq r$ and $\mathbf{tw}(G[X]) \leq r$. For positive integers $s$ and $s'$, an $(s, s')$-protrusion decomposition of a graph $G$ is a partition $\Pi = \{R_0, \ldots, R_p\}$ of $V(G)$ such that

(i) $\max\{p, |R_0|\} \leq s$,
(ii) for each $i \in \{1, \ldots, p\}$, $R_i^+ = N_G[R_i]$ is an $s'$-protrusion of $G$, and
(iii) for each $i \in \{1, \ldots, p\}$, $N_G(R_i) \subseteq R_0 \cap \partial_G[R_i^+]$.

Originally, condition (iii) only demanded that $N_G(R_i) \subseteq R_0$ holds for each $i \in \{1, \ldots, p\}$. However, we can move every vertex in $N_G(R_i) \setminus \partial_G[R_i^+]$ to $R_i$ without affecting any of the other properties. Hence we assume without loss of generality that such vertices do not exist and may indeed state condition (iii) as above (which is convenient for our purposes). The sets $R_1^+, \ldots, R_p^+$ are called the *protrusions* of $\Pi$.

The following statement is implicit in [3] (see Lemmas 6.1 and 6.2).

**Lemma 2 ([3]).** *Let $r$ and $k$ be positive integers and let $G$ be a planar graph that has an $r$-dominating set of size at most $k$. Then $G$ has an $(O(kr), O(r))$-protrusion decomposition, which can be constructed in polynomial time.*

**Parameterized Complexity.** Parameterized complexity is a two dimensional framework for studying the computational complexity of a problem. One dimension is the input size $n$ and another one is a parameter $k$. It is said that a problem is *fixed parameter tractable* (or $\mathsf{FPT}$) if it can be solved in time $f(k) \cdot n^{O(1)}$ for some function $f$. A *kernelization* for a parameterized problem is a polynomial algorithm that maps each instance $(x, k)$ with the input $x$ and the parameter $k$ to an instance $(x', k')$ such that

(i) $(x, k)$ is a yes-instance if and only if $(x', k')$ is a yes-instance, and
(ii) the size of $x'$ is bounded by $f(k)$ for a computable function $f$.

The output $(x', k')$ is called a *kernel*. The function $f$ is said to be the *size* of the kernel. A kernel is *polynomial* if $f$ is polynomial. We refer to the books of Downey and Fellows [10], Flum and Grohe [11], and Niedermeier [25] for detailed introductions to parameterized complexity.

## 3 The Polynomial Kernels

In this section we construct polynomial kernels for DPGGD and DCPGGD. We say that a pair $(U, D)$ with $U \subseteq V(G)$ and $D \subseteq E(G)$ is a *solution* for an instance $(G, k_v, k_e, C, \delta, w, c)$ of DPGGD if $w(U) \leq k_v$, $w(D) \leq k_e$ and $c(U \cup D) \leq C$ and

$G' = G - U - D$ satisfies $d_{G'}(v) = \delta(v)$ for all $v \in V(G')$. If $(G, k_v, k_e, C, \delta, w, c)$ is an instance of DCPGGD then $(U, D)$ is a solution if in addition $G'$ is connected. Notice that it can happen that $U = V(G)$ for a solution $(U, D)$.

In order to prove our main results, we first need to introduce some additional terminology and prove some structural results. We say that a solution $(U, D)$ for an instance of DPGGD or DCPGGD is *efficient* if $D$ has no edges incident to the vertices of $U$. We say that a solution $(U, D)$ is of *minimum cost* if $c(\hat{U}, \hat{D}) \geq c(U, D)$ for every solution $(\hat{U}, \hat{D})$. We make two observations.

**Observation 1** *Any yes-instance of* DPGGD *or* DCPGGD *has an efficient solution of minimum cost.*

**Observation 2** *Let* $(G, k_v, k_e, C, \delta, w, c)$ *be instance of* DPGGD *or* DCPGGD *that has an efficient solution* $(U, D)$*. If* $d_G(v) = \delta(v)$ *for some* $v \in V(G)$ *then* $v$ *is not incident to an edge of* $D$*.*

We say that an instance $(G, k_v, k_e, C, \delta, w, c)$ of DPGGD (DCPGGD respectively) is *normalized* if

(i) for every $v \in V(G)$, $\delta(v) \leq d_G(v) \leq \delta(v) + k_v + k_e$, and
(ii) every vertex $v$ in the set $S = \{u \in V(G) \mid d_G(u) = \delta(u)\}$ is adjacent to a vertex in $\overline{S} = V(G) \setminus S$.

**Lemma 3.** *There is a polynomial-time algorithm that for each instance of* DPGGD *or* DCPGGD *either solves the problem or returns an equivalent normalized instance.*

*Proof.* Let $(G, k_v, k_e, C, \delta, w, c)$ be an instance of DPGGD. To simplify notation, we keep the same notation for the functions $\delta, w, c$ if we delete vertices or edges and do not modify the values of the functions for the remaining elements if this does not create confusion.

We say that a reduction rule is *safe* if by applying the rule we either solve the problem or obtain an equivalent instance. It is straightforward to see that the following reduction rules are safe.

> **Yes-instance rule.** If $S = V(G)$, then $(\emptyset, \emptyset)$ is a solution, return a yes-answer and stop.

> **Vertex deletion rule.** If $G$ has a vertex $v$ with $d_G(v) < \delta(v)$ or $d_G(v) > \delta(v) + k_v + k_e$, then delete $v$ and set $k_v = k_v - w(v)$, $C = C - c(v)$. If $k_v < 0$ or $C < 0$, then stop and return a no-answer.

Observe that by the exhaustive application of the **vertex deletion rule** and applying the **yes-instance rule** whenever possible, we either solve the problem or we obtain an instance which satisfies (i) of the definition of normalized instances, but where $S \neq V(G)$. Notice that, in particular, the **yes-instance rule** is applied if the set of vertices becomes empty. To ensure (ii), we apply the following two rules.

**Contraction rule.** If $G$ has two adjacent vertices $u, v \in S = \{x \in V(G) \mid d_G(x) = \delta(x)\}$ such that $N_G(v) \subseteq S$, then we construct the instance $(G', k_v, k_e, C, \delta', w', c')$ as follows.

- Contract $uv$. Denote the obtained graph $G' = G/uv$ and let $z$ be the vertex obtained from $u$ and $v$.
- Set $\delta'(z) = d_{G'}(z)$ and set $\delta'(x) = d_{G'}(x)$ for any $x \in S \setminus \{u, v\}$. For each $x \in \overline{S}$, set $\delta'(x) = \delta(x)$.
- Set $w'(z) = w(u)+w(v)$ and $c'(z) = c(u)+c(v)$. For $x \in V(G)\setminus\{u, v\}$, set $w'(x) = w(x)$ and $c'(x) = c(x)$.
- For each $xz \in E(G')$, set $w'(xz) = k_e + 1$ and $c'(xz) = 0$. For all other edges $xy \in E(G')$, set $w'(xy) = w(xy)$ and $c'(xy) = c(xy)$.

Let $(U, D)$ be an efficient solution for $(G, k_v, k_e, C, \delta, w, c)$. By Observation 2, $D$ has no edges incident to $u$ or $v$. Also either $u, v \in U$ or $u, v \notin U$, because $u$ and $v$ are adjacent and $d_G(u) = \delta(u)$ and $d_G(v) = \delta(v)$. Let $U' = (U \setminus \{u, v\}) \cup \{z\}$ if $u, v \in U$ and $U' = U$ otherwise. We have that $(U', D)$ is a solution for $(G', k_v, k_e, C, \delta', w', c')$. If $(U', D')$ is an efficient solution for $(G', k_v, k_e, C, \delta', w', c')$, then $D'$ has no edges incident to $z$ by Observation 2. If $z \in U'$, let $U = (U' \setminus \{z\}) \cup \{u, v\}$ and $U = U'$ otherwise. We obtain that $(U, D)$ is a solution for the original instance.

We exhaustively apply the above rule. Assume that it cannot be applied for $(G, k_v, k_e, C, \delta, w, c)$. Then we have that this instance satisfies (i) and the following holds: for any $v \in S \neq V(G)$, either $v$ is adjacent to a vertex in $\overline{S}$ or $v$ is an isolated vertex. It remains to deal with isolated vertices.

**Isolates removal rule.** If $G$ has an isolated vertex $v$, then delete $v$.

To see that above rule is safe, notice that, because the considered instance satisfies (i), it follows that $v \in S$. Clearly, by the exhaustive application of the **isolates removal rule**, we either solve the problem or obtain an instance that satisfies (i) and (ii).

Now consider an instance $(G, k_v, k_e, C, \delta, w, c)$ of DCPGGD.

We replace the **yes-instance rule** by the following variant.

**Yes-instance rule (connected).** If $S = V(G)$ and $G$ is connected, then $(\emptyset, \emptyset)$ is a solution, return a yes-answer and stop.

It is straightforward to verify that the **vertex deletion rule** and the **contraction rule** are safe for this problem. By applying these rules and by the application of the connected variant of the **yes-instance rule** whenever possible, we either solve the problem or obtain an equivalent instance that satisfies (i) and has the property that for any $v \in S$, either $v$ is adjacent to a vertex in $\overline{S}$ or $v$ is an isolated vertex. Suppose that $(G, k_v, k_e, C, \delta, w, c)$ satisfies these properties. Observe that if $H$ is a component of $G$, then for any solution $(U, D)$, either $V(H) \subseteq U$ or $V(G) \setminus V(H) \subseteq U$. Therefore, it is safe to apply the following variant of the **isolates removal rule**.

**Isolates removal rule (connected).** If $G$ has an isolated vertex $v$, then if $w(V(G) \setminus \{v\}) \leq k_v$ and $c(V(G) \setminus \{v\}) \leq C$, then $(V(G) \setminus \{v\}, \emptyset)$ is a solution, return a yes-answer and stop. Otherwise, if $w(V(G) \setminus \{v\}) > k_v$ or $c(V(G) \setminus \{v\}) > C$, delete $v$ and set $k_v = k_v - w(v)$ and $C = C - c(v)$; if $k_v < 0$ or $C < 0$, then stop and return a no-answer.

It is easy to see that if the input graph was planar then the graph formed after applying the rules above will also be planar. □

**Lemma 4.** *If* $(G, k_v, k_e, C, \delta, w, c)$ *is a normalized yes-instance of* DPGGD *(*DCPGGD *respectively) then* $G$ *has a* 2*-dominating set of size at most* $k_v + 2k_e$.

*Proof.* We prove the lemma for DPGGD; the proof for DCPGGD is the same. Let $(G, k_v, k_e, C, \delta, w, c)$ be a normalized yes-instance of the problem. Let $(U, D)$ be a solution and $W = U \cup V(D)$. Clearly, $|W| \leq k_v + 2k_e$, because the weights are positive integers. We show that $W$ is a 2-dominating set of $G$.

Let $S = \{v \in V(G) \mid d_G(v) = \delta(v)\}$ and $\overline{S} = V(G) \setminus S$. For any vertex $v \in \overline{S}$, either $v \in U$ or $v$ is adjacent to a vertex of $U$ or $v$ is incident to an edge of $D$. Hence, $\overline{S} \subseteq N_G[W]$. Let $v \in S$. Because the considered instance is normalized, $v$ is adjacent to a vertex $u \in \overline{S}$. It implies, that $S \subseteq N_G^2[W]$. □

The following is a direct consequence of Lemmas 2 and 4.

**Lemma 5.** *There is a fixed constant* $\alpha$ *such that, if* $(G, k_v, k_e, C, \delta, w, c)$ *is a normalized yes-instance of* DPGGD *(*DCPGGD *respectively), then* $G$ *has an* $(\alpha(k_v + 2k_e), \alpha)$*-protrusion decomposition. Moreover, if there is such a decomposition, one can be constructed in* $O(n^2)$ *steps.*

The next lemma states that, for both DPGGD and DCPGGD, an optimal solution can be found in polynomial time on graphs of bounded treewidth. The proof is based on the standard techniques for dynamic programming over tree decompositions and is omitted due to the space restrictions.

**Lemma 6.** DPGGD *(*DCPGGD *respectively) can be solved and an efficient solution* $(U, D)$ *of minimum cost can be obtained in* $(k_v + k_e)^{O(q)} \cdot \mathrm{poly}(n)$ *time (in* $(q(k_v + k_e))^{O(q)} \cdot \mathrm{poly}(n)$ *time respectively) for instances* $(G, k_v, k_e, C, \delta, w, c)$ *where* $G$ *is an* $n$*-vertex graph of treewidth at most* $q$ *and* $\delta(v) \leq d_G(v) \leq \delta(v) + k_v + k_e$ *for* $v \in V(G)$.

We are now ready to present our two main results, starting with the one for DPGGD.

**Theorem 1.** DPGGD *has a polynomial kernel when parameterized by* $k_v + k_e$.

*Proof.* Let $(G, k_v, k_e, C, \delta, w, c)$ be an instance of DPGGD. By Lemma 3, we may assume that this instance is normalized. By Lemma 4, if $(G, k_v, k_e, C, \delta, w, c)$ is a yes-instance, then $G$ has a 2-dominating set of size at most $k_v + 2k_e$. By Lemma 5, there is a fixed constant $\alpha$ such that $G$ has an $(\alpha(k_v + 2k_e), \alpha)$-protrusion decomposition, and such a decomposition, if it exists, can be constructed in polynomial time. To simplify later arguments, we may assume $\alpha \geq 3$. Clearly, if we fail to obtain such a decomposition, we return a no-answer and stop. Hence, from now on we assume that an $(\alpha(k_v+2k_e),\alpha)$-protrusion decomposition $\Pi=\{R_0,\ldots,R_p\}$ of $G$ is given. As before, we keep the same notation $\delta, w, c$ for the restrictions of these functions. Again, we will introduce new reduction rules. We will keep the notation for $G$ and for the parameters unchanged where this is well-defined. We also assume that if we consider sets of vertices or edges associated with the considered instance and delete vertices or edges from the graph, then we also delete these elements from the associated sets.

For each $i \in \{1,\ldots,p\}$, we construct $W_i \subseteq R_i$ and $L_i \subseteq E_G(R_i)$. To do this, we consider the set $\mathcal{Q}$ of all possible quintuples $\mathbf{q}=(h_v,h_e,X,Y,\delta')$ such that

- $0 \leq h_v \leq k_v$ and $0 \leq h_e \leq k_e$,
- $X \subseteq N_G(R_i)$ and $Y \subseteq E(G[N_G(R_i)\setminus X])$, and
- We define $F=G[R_i^+]-X-Y$ and require that $\delta'\colon V(F)\to\mathbb{N}_0$ is a function such that $\delta'(v)\leq d_F(v)\leq \delta'(v)+k_v+k_e$ for $v\in N_G(R_i)\setminus X$ and $\delta'(v)=\delta(v)$ for $v\in R_i$

Observe that there are at most $2^\alpha$ sets $X$, at most $2^{3\alpha-6}$ sets $Y$, at most $(k_v+1)(k_e+1)$ pairs $h_v,h_e$, and for each $X$, there are at most $(k_v+k_e+1)^\alpha$ possibilities for $\delta'$. Therefore $|\mathcal{Q}|\leq 2^\alpha 2^{3\alpha-6}(k_v+1)(k_e+1)(k_v+k_e+1)^\alpha=(k_v+k_e)^{O(\alpha)}$.

For each $\mathbf{q}=(h_v,h_e,X,Y,\delta')\in\mathcal{Q}$, we construct an instance $I_{\mathbf{q}}=(F,h_v,h_e,C,\delta',w',c)$ of DPGGD such that

- $w'(v)=k_v+1$, for $v\in N_G(R_i)\setminus X$ and $w'(v)=w(v)$, for $v\in R_i$ and
- $w'(e)=k_e+1$, for $e\in E(G[N_G(R_i)\setminus X])\setminus Y$ and $w'(e)=w(e)$, for all other edges of $F$.

By Lemma 6, we can solve the problem for this instance in polynomial time. Let $(U_{\mathbf{q}},D_{\mathbf{q}})$ denote the obtained solution of minimum cost and set $U_{\mathbf{q}}=D_{\mathbf{q}}=\emptyset$ if no solution exists for $I_{\mathbf{q}}$. Let

$$W_i=\bigcup_{\mathbf{q}\in\mathcal{Q}}U_{\mathbf{q}} \text{ and } L_i=\bigcup_{\mathbf{q}\in\mathcal{Q}}D_{\mathbf{q}}.$$

Because each $U_{\mathbf{q}}$ has at most $k_v$ vertices and each $D_{\mathbf{q}}$ has at most $k_e$ edges, we obtain that $|W_i|\leq|\mathcal{Q}|k_v\leq(k_v+1)(k_e+1)\cdot 2^\alpha\cdot 2^{3\alpha-6}\cdot(k_v+k_e+1)^\alpha\cdot k_v$ and $|L_i|\leq|\mathcal{Q}|k_e\leq(k_v+1)(k_e+1)\cdot 2^\alpha\cdot 2^{3\alpha-6}\cdot(k_v+k_e+1)^\alpha\cdot k_e$. Hence, the size of $W_i$ and $L_i$ is $(k_v+k_e)^{O(\alpha)}$.

Let $W=R_0\cup\bigcup_{i\in[p]}W_i$ and $L=E(G[R_0])\cup\bigcup_{i\in[p]}L_i$. Because $\max\{p,|R_0|\}\leq\alpha(k_v+2k_e)$, we have that $|W|=(k_v+k_e)^{O(\alpha)}$ and $|L|=(k_v+k_e)^{O(\alpha)}$. We prove the following claim.

**Claim A.** *If $(G,k_v,k_e,C,\delta,w,c)$ is a yes-instance of* DPGGD*, then it has an efficient solution $(U,D)$ of minimum cost such that $U\subseteq W$ and $D\subseteq L$.*

We prove Claim A as follows. Let $(U,D)$ be an efficient solution for $(G,k_v,k_e,C,\delta,w,c)$ of minimum cost such that $s=|U\setminus W|+|D\setminus L|$ is minimum. If $s=0$, then the claim is fulfilled. Suppose, for a contradiction, that

$s > 0$. This means that there is an $i \in \{1, \ldots, p\}$ such that $(U \cap R_i) \setminus W_i \neq \emptyset$ or $(D \cap E_G(R_i)) \setminus L_i \neq \emptyset$. Let $X = U \cap N_G(R_i)$, $Y = D \cap E(N_G(R_i))$ and $F = G[R_i^+] - X - Y$. Let $h_v = |U \cap V(F)|$ and $h_e = |D \cap E(F)|$. For a vertex $v \in N_G(R_i) \setminus X$, let $d_v$ be the total number of vertices in $U \setminus V(F)$ adjacent to $v$ and edges in $D \setminus E(F)$ incident to $v$. Let $\delta'(v) = d_F(v) - (d_G(v) - \delta(v) - d_v)$ for $v \in N_G(R_i) \setminus X$ and $\delta'(v) = \delta(v)$ for other vertices of $F$.

Clearly, $(F, h_v, h_e, C, \delta', w', c) = I_{\mathbf{q}}$ is the instance of DPGGD when $\mathbf{q} = (F, h_v, h_e, C, \delta')$. Let $U' = U \cap V(F)$ and $D' = D \cap E(F)$. Then $(U', D')$ is a solution for the instance $I_{\mathbf{q}}$ and, therefore $I_{\mathbf{q}}$ is a yes-instance. In particular, this means that there is a solution $(U'', D'')$ for $I_{\mathbf{q}} = (F, h_v, h_e, C, \delta', w', c)$ that was constructed by the aforementioned procedure for the construction of $W_i$ and $L_i$. Clearly, $U'' \subseteq W_i \subseteq W$ and $D'' \subseteq L_i \subseteq L$. Because our algorithm for graphs of bounded treewidth finds a solution of minimum cost, it follows that $c(U'' \cup D'') \leq c(U' \cup D')$. It remains to observe that $(\hat{U}, \hat{D})$, where $\hat{U} = (U \setminus U') \cup U''$ and $\hat{D} = (D \setminus D') \cup D''$, is a solution for $(G, k_v, k_e, C, \delta, w, c)$ with $c(\hat{U} \cup \hat{D}) \leq c(U \cup D)$, but this contradicts the choice of $(U, D)$ because $|\hat{U} \setminus W| + |\hat{D} \setminus L| < s$. This completes the proof of Claim A.

Let $S = \{v \in V(G) \mid d_G(v) = \delta(v)\} \setminus W$ and $T = \{v \in V(G) \mid d_G(v) > \delta(v)\} \setminus W$; because the instance we consider is normalized, these sets form a partition of $V(G) \setminus W$ (note that these sets may be empty). If $v \in S$, then for any efficient solution $(U, D)$ such that $U \subseteq W$ and $D \subseteq L$, $v$ is not adjacent to a vertex of $U$. This implies that it is safe to exhaustively apply the following rule without destroying the statement of Claim A.

> **Set adjustment rule.** If there is a vertex $v \in S$ that is adjacent to a vertex $u \in W$, then set $W = W \setminus \{u\}$ and set $S = S \cup \{u\}$ if $d_G(u) = \delta(u)$ and set $T = T \cup \{u\}$ if $d_G(u) > \delta(u)$. If $v \in S$, remove any edge incident to $v$ from $L$.

By Claim A, it is safe to modify the weights as follows.

> **Weight adjustment rule.** Set $w(v) = k_v + 1$ for $v \in V(G) \setminus W$ and set $w(e) = k_e + 1$ for $e \in E(G) \setminus L$.

After the exhaustive application of the **set adjustment rule**, we have that $N_G(S) \subseteq T$. Now it is safe to remove $S$.

> ***S*-reduction rule.** If $v \in S$, then remove $v$ and set $\delta(u) = \delta(u) - 1$ for $u \in N_G(v)$. If $\delta(u) < 0$ for some $u \in N_G(v)$, then return a no-answer and stop.

To show that the above rule is safe, let $G' = G - S$ and let $\delta'$ be the function obtained from $\delta$ by the application of the rule. Suppose that $(G, k_v, k_e, C, \delta, w, c)$ is a yes-instance. Then we have a solution $(U, D)$ such that $U \subseteq W$ and $D \subseteq L$ by Claim A. Because $N_G(S) \subseteq T$, $T \cap W = \emptyset$ and the vertices of $S$ are not incident to edges of $L$, it follows that we do not stop and $(U, D)$ is a solution for $(G', k_v, k_e, C, \delta', w, c)$. Let now $(U, D)$ is a solution for $(G', k_v, k_e, C, \delta', w, c)$. Because of the application of the **weight adjustment rule**, $U \subseteq W$ and $D \subseteq L$.

Because $N_G(S) \subseteq T$, $T \cap W = \emptyset$ and the vertices of $S$ are not incident to edges of $L$, we have that $(U, D)$ is a solution for $(G, k_v, k_e, C, \delta, w, c)$. This completes the proof that the $S$**-reduction rule** is safe.

Let $W' = W \cup V(L)$ and $T' = T \setminus V(L)$. Clearly, $|W'| \leq |W| + 2|L| = (k_v + k_e)^{O(\alpha)}$.

Using similar arguments to those for the $S$**-reduction rule**, the following rule is also safe.

> $T'$**-reduction rule.** If $uv \in E(G[T'])$, then remove $uv$ and set $\delta(u) = \delta(u) - 1$ and $\delta(v) = \delta(v) - 1$. If $\delta(u) < 0$ or $\delta(v) < 0$, then return a no-answer and stop.

After the exhaustive application of the above rule, $T'$ is an independent set in the obtained graph $G$. Some of the vertices of this independent set may have the same neighbourhoods. We deal with them using the next rule.

> **Twin reduction rule.** Suppose there are $u, v \in T'$ with $N_G(u) = N_G(v)$. If $\delta(u) = \delta(v)$, then remove $v$ and set $\delta(x) = \max\{0, \delta(x) - 1\}$ for $x \in N_G(u)$. If $\delta(u) \neq \delta(v)$ then return a no-answer and stop.

To prove that the above rule is safe, consider a pair of vertices $u, v \in T'$ with $N_G(u) = N_G(v)$ and $\delta(u) = \delta(v)$. Let $G' = G - v$ and let $\delta'$ denote the function obtained from $\delta$ by the rule. Suppose that $(G, k_v, k_e, C, \delta, w, c)$ is a yes-instance. Then we have a solution $(U, D)$ such that $U \subseteq W$ and $D \subseteq L$. Notice that $T' \cap U = \emptyset$ and the vertices of $T'$ are not incident to the edges of $L$. Note that $u, v \notin U$ and if $x \in N_G(u)$ then $ux, vx \notin D$. We have that $U$ contains exactly $d_G(u) - \delta(u)$ vertices that are adjacent to $u$. Therefore, $(U, D)$ is a solution for $(G', k_v, k_e, C, \delta', w, c)$. Assume now that $(U, D)$ is a solution for $(G', k_v, k_e, C, \delta', w, c)$. By the same arguments, $U$ contains exactly $d_{G'}(u) - \delta'(u)$ vertices that are adjacent to $u$. Also if $x \in N_G(u)$ and $\delta'(x) = 0$, then $x \in U$, because $u \notin U$ and $ux \notin D$. Because $N_G(u) = N_G(v)$, $\delta(u) = \delta(v)$ and $T'$ is an independent set, $U$ contains $d_G(u) - \delta(u)$ vertices that are adjacent to $u$ and $d_G(v) - \delta(v)$ vertices that are adjacent to $v$. It follows that $(U, D)$ is a solution for $(G, k_v, k_e, C, \delta, w, c)$. Now consider the case when $N_G(u) = N_G(v)$ and $\delta(u) \neq \delta(v)$. Suppose, for contradiction that there is a solution $(U, D)$. By the above arguments, $U$ contains exactly $d_G(u) - \delta(u)$ vertices that are adjacent to $u$ and $d_G(v) - \delta(v)$ vertices that are adjacent to $v$. Since $N_G(u) = N_G(v)$ and $\delta(u) \neq \delta(v)$, this is a contradiction, so there cannot be such a solution.

After the exhaustive application of the above rule for any two vertices $u, v \in T'$, we have that $N_G(u) \neq N_G(v)$. Let $T'_0, T'_1, T'_2, T'_{\geq 3}$ denote the sets of vertices in $T'$ that are of degree 0, 1, 2 and at least 3 respectively. Observe that $d_G(v) > \delta(v) \geq 0$ for $v \in T'$. Therefore, $T'_0 = \emptyset$ and $T'_1, T'_2, T'_{\geq 3}$ form a partition of $T'$ (note that these sets may be empty). By the **twin reduction rule** $|T'_1| = |N_G(T'_1)| \leq |W'|$ and $|T'_2| \leq \binom{|N_G(T'_2)|}{2} \leq \frac{1}{2}|W'|(|W'| - 1)$. By Lemma 1, $|T'_{\geq 3}| \leq 2|N_G(T')| - 4 \leq 2|W'| - 4$ (or $|T'_{\geq 3}| = 0$). We have that $|V(G)| = |W'| + |T'| = |W'| + |T'_1| + |T'_2| + |T'_{\geq 3}| \leq \frac{1}{2}|W'|^2 + \frac{7}{2}|W'|$. Since, $W'$

has $(k_v + k_e)^{O(\alpha)}$ vertices, we obtain that the obtained graph $G$ has size $k^{O(1)}$ where $k = k_v + k_e$, i.e. we have a polynomial kernel for DPGGD.

To complete the proof, it remains to observe that the construction of the normalized instance can be done in polynomial time by Lemma 3, the construction of $W$ and $L$ can be done in polynomial time by Lemma 6, and all the subsequent reduction rules can be applied in polynomial time. $\square$

The proof of our second main result is based on the same approach as the proof of Theorem 1, but it is more technically involved because we have to ensure connectivity of the graph obtained by the editing. Hence, the proof is omitted here and will appear in the journal version of our paper.

**Theorem 2.** DCPGGD *has a polynomial kernel when parameterized by* $k_v + k_e$.

## 4 Conclusions

We proved that DPGGD and DCPGGD are NP-complete but allow polynomial kernels when parameterized by $k_v + k_e$. These problems generalize the Degree Constrained Editing($S$) problem and its connected variant for $S = \{\mathsf{ed}, \mathsf{vd}\}$; this can be seen, for instance, by testing all possible pairs $k_v, k_e$ with $k_v + k_e = k$ or by a slight adjustment of our algorithms. Note that by setting $k_v = 0$ or $k_e = 0$ we obtain the same results for $S = \{\mathsf{ed}\}$ and $S = \{\mathsf{vd}\}$, respectively (recall though that for $S = \{\mathsf{ed}\}$ this is not so surprising, as the less general problem Degree Constrained Editing($\{\mathsf{ed}\}$) is polynomial-time solvable for general graphs).

Several open problems remain. We note that graph modification problems that permit edge additions are less natural to consider for planar graphs, because the class of planar graphs is not closed under edge addition. However, we could allow other, more appropriate, operations such as edge contractions and vertex dissolutions when considering planar graphs. Belmonte et al. [1] considered the setting in which only edge contractions are allowed and obtained initial results for general graphs that extend the work of Mathieson and Szeider [22] on Degree Constrained Editing($S$) in this direction.

## References


1. Belmonte, R., Golovach, P.A., van 't Hof, P., Paulusma, D.: Parameterized complexity of three edge contraction problems with degree constraints. Acta Informatica 51(7), 473–497 (2014)
2. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (meta) kernelization. In: FOCS 2009. pp. 629–638. IEEE Computer Society (2009)
3. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (meta) kernelization. CoRR abs/0904.0727 (2009)
4. Boesch, F.T., Suffel, C.L., Tindell, R.: The spanning subgraphs of Eulerian graphs. Journal of Graph Theory 1(1), 79–84 (1977)

5. Burzyn, P., Bonomo, F., Durán, G.: NP-completeness results for edge modification problems. Discrete Applied Mathematics 154(13), 1824–1844 (2006)
6. Cai, L.: Fixed-parameter tractability of graph modification problems for hereditary properties. Inf. Process. Lett. 58(4), 171–176 (1996)
7. Cai, L., Yang, B.: Parameterized complexity of even/odd subgraph problems. Journal of Discrete Algorithms 9(3), 231–240 (2011)
8. Cygan, M., Marx, D., Pilipczuk, M., Pilipczuk, M., Schlotter, I.: Parameterized complexity of Eulerian deletion problems. Algorithmica 68(1), 41–61 (2014)
9. Dabrowski, K.K., Golovach, P.A., van 't Hof, P., Paulusma, D.: Editing to Eulerian graphs. In: FSTTCS 2014. LIPIcs, vol. 29, pp. 97–108. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2014)
10. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science, Springer (2013)
11. Flum, J., Grohe, M.: Parameterized complexity theory. Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin (2006)
12. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Linear kernels for (connected) dominating set on $H$-minor-free graphs. In: SODA 2012. pp. 82–93. SIAM (2012)
13. Froese, V., Nichterlein, A., Niedermeier, R.: Win-win kernelization for degree sequence completion problems. In: SWAT 2014. Lecture Notes in Computer Science, vol. 8503, pp. 194–205. Springer (2014)
14. Garey, M.R., Johnson, D.S., Tarjan, R.E.: The planar hamiltonian circuit problem is NP-complete. SIAM J. Comput. 5(4), 704–714 (1976)
15. Garnero, V., Paul, C., Sau, I., Thilikos, D.M.: Explicit linear kernels via dynamic programming. In: STACS 2014. LIPIcs, vol. 25, pp. 312–324. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2014)
16. Garnero, V., Sau, I., Thilikos, D.M.: A linear kernel for planar red-blue dominating set. CoRR abs/1408.6388 (2014)
17. Golovach, P.A.: Editing to a connected graph of given degrees. In: MFCS 2014, Part II. Lecture Notes in Computer Science, vol. 8635, pp. 324–335. Springer (2014)
18. Golovach, P.A.: Editing to a graph of given degrees. In: IPEC 2014. Lecture Notes in Computer Science, vol. 8894, pp. 196–207. Springer (2014)
19. Khot, S., Raman, V.: Parameterized complexity of finding subgraphs with hereditary properties. Theor. Comput. Sci. 289(2), 997–1008 (2002)
20. Kim, E.J., Langer, A., Paul, C., Reidl, F., Rossmanith, P., Sau, I., Sikdar, S.: Linear kernels and single-exponential algorithms via protrusion decompositions. In: ICALP 2013. Lecture Notes in Computer Science, vol. 7965, pp. 613–624. Springer (2013)
21. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. J. Comput. Syst. Sci. 20(2), 219–230 (1980)
22. Mathieson, L., Szeider, S.: Editing graphs to satisfy degree constraints: A parameterized approach. J. Comput. Syst. Sci. 78(1), 179–191 (2012)
23. Moser, H., Thilikos, D.M.: Parameterized complexity of finding regular induced subgraphs. J. Discrete Algorithms 7(2), 181–190 (2009)
24. Natanzon, A., Shamir, R., Sharan, R.: Complexity classification of some edge modification problems. Discrete Applied Mathematics 113(1), 109–128 (2001)
25. Niedermeier, R.: Invitation to fixed-parameter algorithms, Oxford Lecture Series in Mathematics and its Applications, vol. 31. Oxford University Press, Oxford (2006)
26. Stewart, I.A.: Deciding whether a planar graph has a cubic subgraph is NP-complete. Discrete Mathematics 126(1-3), 349–357 (1994)

27. Yannakakis, M.: Node- and edge-deletion NP-complete problems. In: STOC 1978. pp. 253–264. ACM (1978)