

A Randomized Polynomial Kernelization for Vertex Cover with a Smaller Parameter

Stefan Kratsch

Universität Bonn, Institut für Informatik, Bonn, Germany
kratsch@cs.uni-bonn.de

Abstract

In the VERTEX COVER problem we are given a graph $G = (V, E)$ and an integer k and have to determine whether there is a set $X \subseteq V$ of size at most k such that each edge in E has at least one endpoint in X . The problem can be easily solved in time $\mathcal{O}^*(2^k)$, making it fixed-parameter tractable (FPT) with respect to k . While the fastest known algorithm takes only time $\mathcal{O}^*(1.2738^k)$, much stronger improvements have been obtained by studying *parameters that are smaller than k* . Apart from treewidth-related results, the arguably best algorithm for VERTEX COVER runs in time $\mathcal{O}^*(2.3146^p)$, where $p = k - LP(G)$ is only the excess of the solution size k over the best fractional vertex cover (Lokshtanov et al. TALG 2014). Since $p \leq k$ but k cannot be bounded in terms of p alone, this strictly increases the range of tractable instances.

Recently, Garg and Philip (SODA 2016) greatly contributed to understanding the parameterized complexity of the VERTEX COVER problem. They prove that $2LP(G) - MM(G)$ is a lower bound for the vertex cover size of G , where $MM(G)$ is the size of a largest matching of G , and proceed to study parameter $\ell = k - (2LP(G) - MM(G))$. They give an algorithm of running time $\mathcal{O}^*(3^\ell)$, proving that VERTEX COVER is FPT in ℓ . It can be easily observed that $\ell \leq p$ whereas p cannot be bounded in terms of ℓ alone. We complement the work of Garg and Philip by proving that VERTEX COVER admits a randomized polynomial kernelization in terms of ℓ , i.e., an efficient preprocessing to size polynomial in ℓ . This improves over parameter $p = k - LP(G)$ for which this was previously known (Kratsch and Wahlström FOCS 2012).

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Vertex cover, parameterized complexity, kernelization

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.59

1 Introduction

A *vertex cover* of a graph $G = (V, E)$ is a set $X \subseteq V$ such that each edge $e \in E$ has at least one endpoint in X . The VERTEX COVER problem of determining whether a given graph G has a vertex cover of size at most k has been an important benchmark problem in parameterized complexity for both *fixed-parameter tractability* and *(polynomial) kernelization*,¹ which are the two notions of tractability for parameterized problems. Kernelization, in particular, formalizes the widespread notion of efficient preprocessing, allowing a rigorous study (cf. [14]). We present a randomized polynomial kernelization for VERTEX COVER for the to-date smallest parameter, complementing a recent fixed-parameter tractability result by Garg and Philip [10].

¹ Definitions can be found in the full version. Note that we use ℓ , rather than k , as the default symbol for parameters and use VERTEX COVER(ℓ) to refer to the VERTEX COVER problem with parameter ℓ .

Let us first recall what is known for the so-called *standard parameterization* VERTEX COVER(k), i.e., with parameter $\ell = k$: There is a folklore $\mathcal{O}^*(2^k)$ time² algorithm for testing whether a graph G has a vertex cover of size at most k , proving that VERTEX COVER(k) is fixed-parameter tractable (FPT); this has been improved several times with the fastest known algorithm due to Chen et al. [4] running in time $\mathcal{O}^*(1.2738^k)$. Under the Exponential Time Hypothesis of Impagliazzo et al. [11] there is no algorithm with runtime $\mathcal{O}^*(2^{o(k)})$. The best known kernelization for VERTEX COVER(k) reduces any instance (G, k) to an equivalent instance (G', k') with $|V(G')| \leq 2k$; the total size is $\mathcal{O}(k^2)$ [3]. Unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ and the polynomial hierarchy collapses there is no kernelization to size $\mathcal{O}(k^{2-\varepsilon})$ [8].

At first glance, the FPT and kernelization results for VERTEX COVER(k) seem essentially best possible. This is true for parameter $\ell = k$, but there are *smaller parameters* ℓ' for which both FPT-algorithms and polynomial kernelizations are known. The motivation for this is that even when $\ell' = \mathcal{O}(1)$, the value $\ell = k$ may be as large as $\Omega(n)$, making both FPT-algorithm and kernelization for parameter k useless for such instances (time $2^{\Omega(n)}$ and size guarantee $\mathcal{O}(n)$). In contrast, for $\ell' = \mathcal{O}(1)$ an FPT-algorithm with respect to ℓ' runs in polynomial time (with only leading constant depending on ℓ'). Let us discuss the relevant type of smaller parameter, which relates to *lower bounds on the optimum* and was introduced by Mahajan and Raman [19]; two other types are discussed briefly under related work.

Two well-known lower bounds for the size of vertex covers for a graph $G = (V, E)$ are the maximum size of a matching of G and the smallest size of fractional vertex covers for G ; we (essentially) follow Garg and Philip [10] in denoting these two values by $MM(G)$ and $LP(G)$. Note that the notation $LP(G)$ comes from the fact that fractional vertex covers come up naturally in the linear programming relaxation of the VERTEX COVER problem, where we must assign each vertex a fractional value such that each edge is incident with total value of at least 1. In this regard, it is useful to observe that the LP relaxation of the MAXIMUM MATCHING problem is exactly the dual of this. Accordingly, we have $MM(G) \leq LP(G)$ since each integral matching is also a fractional matching, i.e., with each vertex incident to a total value of at most 1. Similarly, using $VC(G)$ to denote the minimum size of vertex covers of G we get $VC(G) \geq LP(G)$ and, hence, $VC(G) \geq LP(G) \geq MM(G)$.

A number of papers have studied vertex cover with respect to “above lower bound” parameters $\ell' = k - MM(G)$ or $\ell'' = k - LP(G)$ [24, 23, 7, 21, 17]. Observe that $k \geq k - MM(G) \geq k - LP(G)$. For the converse, note that k can be unbounded in terms of $k - MM(G)$ and $k - LP(G)$, whereas $k - MM(G) \leq 2(k - LP(G))$ holds [16, 12]. Thus, from the perspective of achieving fixed-parameter tractability (and avoiding large parameters) both parameters are equally useful for improving over parameter k . Razgon and O’Sullivan [24] proved fixed-parameter tractability of ALMOST 2-SAT(k), which implies that VERTEX COVER($k - MM$) is FPT due to a reduction to ALMOST 2-SAT(k) by Mishra et al. [20]. Using $k - MM(G) \leq 2(k - LP(G))$, this also entails fixed-parameter tractability of VERTEX COVER($k - LP$).

After several improvements [23, 7, 21, 17] the fastest known algorithm, due to Lokshtanov et al. [17], runs in time $\mathcal{O}^*(2.3146^{k-MM(G)})$. The algorithms of Narayanaswamy et al. [21] and Lokshtanov et al. [17] achieve the same parameter dependency also for parameter $k - LP(G)$. The first (and to our knowledge only) kernelization result for these parameters is a randomized polynomial kernelization for VERTEX COVER($k - LP$) by Kratsch and Wahlström [16], which of course applies also to the larger parameter $k - MM(G)$.

Recently, Garg and Philip [10] made an important contribution to understanding the

² We use \mathcal{O}^* notation, which suppresses polynomial factors.

parameterized complexity of the VERTEX COVER problem by proving it to be FPT with respect to parameter $\ell = k - (2LP(G) - MM(G))$. Building on an observation of Lovász and Plummer [18] they prove that $VC(G) \geq 2LP(G) - MM(G)$, i.e., that $2LP(G) - MM(G)$ is indeed a lower bound for the minimum vertex covers size of any graph G . They then design a branching algorithm with running time $\mathcal{O}^*(3^\ell)$ that builds on the well-known Gallai-Edmonds decomposition for maximum matchings to guide its branching choices.

VERTEX COVER($k - (2LP - MM)$)

Input: A graph $G = (V, E)$ and an integer $k \in \mathbb{N}$.

Parameter: $\ell = k - (2LP(G) - MM(G))$ where $LP(G)$ is the minimum size of fractional vertex covers for G and $MM(G)$ is the maximum cardinality of matchings of G .

Question: Does G have a vertex cover of size at most k , i.e., a set $X \subseteq V$ of size at most k such that each edge of E has at least one endpoint in X ?

Since $LP(G) \geq MM(G)$, we clearly have $2LP(G) - MM(G) \geq LP(G)$ and hence $\ell = k - (2LP(G) - MM(G))$ is indeed at most as large as the previously best parameter $k - LP(G)$. We can easily observe that $k - LP(G)$ cannot be bounded in terms of ℓ : For any odd cycle C of length $2s + 1$ we have $LP(C) = \frac{1}{2}(2s + 1)$, $VC(C) = s + 1$, and $MM(C) = s$. Thus, a graph G consisting of t vertex-disjoint odd cycles of length $2s + 1$ has $LP(G) = \frac{1}{2}t(2s + 1)$, $VC(G) = t(s + 1)$, and $MM(G) = ts$. For $k = VC(G) = t(s + 1)$ we get

$$\ell = k - (2LP(G) - MM(G)) = t(s + 1) - t(2s + 1) + ts = 0$$

whereas

$$k - LP(G) = t(s + 1) - \frac{1}{2}t(2s + 1) = \frac{1}{2}t(2s + 2) - \frac{1}{2}t(2s + 1) = \frac{1}{2}t.$$

Generally, it can be easily proved that $LP(G)$ and $2LP(G) - MM(G)$ differ by exactly $\frac{1}{2}$ on any *factor-critical* graph (cf. Proposition 4).

As always in parameterized complexity, when presented with a new fixed-parameter tractability result, the next question is whether the problem also admits a polynomial kernelization. It is well known that decidable problems are fixed-parameter tractable if and only if they admit a (not necessarily polynomial) kernelization.³ Nevertheless, not all problems admit polynomial kernelizations and, in the present case, both an extension of the methods for parameter $k - LP(G)$ [16] or a lower bound proof similar to Cygan et al. [6] or Jansen [12, Section 5.3] (see related work) are conceivable.

Our result. We give a randomized polynomial kernelization for VERTEX COVER($k - (2LP - MM)$). This improves upon parameter $k - LP(G)$ by giving a strictly smaller parameter for which a polynomial kernelization is known. At high level, the kernelization takes the form of a (randomized) polynomial parameter transformation from VERTEX COVER($k - (2LP - MM)$) to VERTEX COVER($k - MM$), i.e., a polynomial-time many-one (Karp) reduction with *output parameter polynomially bounded in the input parameter*. It is well known (cf. Bodlaender et

³ We sketch this folklore fact for VERTEX COVER($k - (2LP - MM)$): If the input is larger than 3^ℓ , where $\ell = k - (2LP(G) - MM(G))$, then the algorithm of Garg and Philip [10] runs in polynomial time and we can reduce to an equivalent small yes- or no-instance; else, the instance size is bounded by 3^ℓ ; in both cases we get size at most 3^ℓ in polynomial time. The converse holds since a kernelization followed by any brute-force algorithm on an instance of, say, size $g(\ell)$ gives an FPT running time in terms of ℓ .

al. [2]) that this implies a polynomial kernelization for the source problem, i.e., for VERTEX COVER($k - (2LP - MM)$) in our case. Let us give some more details of this transformation.

Since the transformation is between different parameterizations of the same problem, it suffices to handle parts of any input graph G where the input parameter $\ell = k - (2LP(G) - MM(G))$ is (much) smaller than the output parameter $k - MM(G)$. After the well-known LP-based preprocessing (cf. [10]), the difference in parameter values is equal to the number of vertices that are exposed (unmatched) by any maximum matching M of G . Consider the Gallai-Edmonds decomposition $V = A \dot{\cup} B \dot{\cup} D$ of $G = (V, E)$, where D contains the vertices that are exposed by at least one maximum matching, $A = N(D)$, and $B = V \setminus (A \cup D)$. Let M be a maximum matching and let t be the number of exposed vertices. There are t components of $G[D]$ that have exactly one exposed vertex each. The value $2LP(G) - MM(G)$ is equal to $|M| + t$ when $LP(G) = \frac{1}{2}|V|$, as implied by LP-based preprocessing.

To reduce the difference in parameter values we will remove all but $\mathcal{O}(\ell^4)$ components of $G[D]$ that have an exposed vertex; they are called *unmatched components* for lack of a matching edge to A and we can ensure that they are not singletons. It is known that any such component C is factor-critical and hence has no vertex cover smaller than $\frac{1}{2}(|C| + 1)$; this exactly matches its contribution to $|M| + t$: It has $\frac{1}{2}(|C| - 1)$ edges of M and one exposed vertex. Unless the instance is trivially **no** all but at most ℓ of these components C have a vertex cover of size $\frac{1}{2}(|C| + 1)$, later called a *tight vertex cover*. The only reason not to use a tight vertex cover for C can be due to adjacent vertices in A that are not selected; this happens at most ℓ times. A technical lemma proves that this can always be traced to at most three vertices of C and hence at most three vertices in A that are adjacent with C .

In contrast, there are (matched, non-singleton) components C of $G[C]$ that together with a matched vertex $v \in A$ contribute $\frac{1}{2}(|C| + 1)$ to the lower bound due to containing this many matching edges. To cover them at this cost requires not selecting vertex v . This in turn propagates along M -alternating paths until the cover picks both vertices of an M -edge, which happens at most ℓ times, or until reaching an unmatched component, where it may help prevent a tight vertex cover. We translate this effect into a two-way separation problem in an auxiliary directed graph. Selecting both vertices of an M -edge is analogous to adding a vertex to the separator. Relative to a separator the question becomes which sets of at most three vertices of A that can prevent tight vertex covers are still reachable by propagation. At this point we can apply representative set tools from Kratsch and Wahlström [16] to identify a small family of such triplets that works for all separators (and hence for all so-called *dominant* vertex covers) and keep only the corresponding components.

Related work. Let us mention some further kernelization results for VERTEX COVER with respect to nonstandard parameters. There are two further types of interesting parameters:

1. *Width-parameters*: Parameters such as treewidth allow dynamic programming algorithms running in time, e.g., $\mathcal{O}^*(2^{\text{tw}})$, independently of the size of the vertex cover. It is known that there are no polynomial kernels for VERTEX COVER (or most other NP-hard problems) under such parameters [1]. The treewidth of a graph is upper bounded by the smallest vertex cover, whereas graphs of bounded treewidth can have vertex cover size $\Omega(n)$.
2. *“Distance to tractable case”-parameters*: VERTEX COVER can be efficiently solved on forests. By a simple enumeration argument it is fixed-parameter tractable when ℓ is the minimum number of vertices to delete such that G becomes a forest. Jansen and Bodlaender [13] gave a polynomial kernelization to $\mathcal{O}(\ell^3)$ vertices. Note that the vertex cover size is an upper bound on ℓ , whereas trees can have unbounded vertex cover size. The FPT-result can be carried over to smaller parameters corresponding to distance from

larger graph classes on which VERTEX COVER is polynomial-time solvable, however, Cygan et al. [6] and Jansen [12, Section 5.3] ruled out polynomial kernels for some of them. E.g., if ℓ is the deletion-distance to an outerplanar graph then there is no kernelization for VERTEX COVER(ℓ) to size polynomial in ℓ unless the polynomial hierarchy collapses [12].

Organization. Section 2 gives some preliminaries. In Section 3 we discuss vertex covers of factor-critical graphs and prove the claimed lemma about critical sets. Section 4 introduces a relaxation of the Gallai-Edmonds decomposition, called *nice decomposition*, and Section 5 explores the relation between nice decompositions and vertex covers. The kernelization for VERTEX COVER($k - (2LP - MM)$) is given in Section 6. We conclude in Section 7.

2 Preliminaries

Parameterized complexity. We use standard definitions from parameterized complexity, with the difference of using ℓ as the default symbol for the parameter. We use VERTEX COVER(ℓ) to refer to the VERTEX COVER problem parameterized by ℓ , e.g., $\ell = k$ for the *standard parameterization* or $\ell = k - LP(G)$. For a detailed introduction to parameterized complexity we recommend the recent books by Downey and Fellows [9] and Cygan et al. [5].

Graphs. We require both directed and undirected graphs; all graphs are finite and simple, i.e., they have no parallel edges or loops. Accordingly, an undirected graph $G = (V, E)$ consists of a finite set V of vertices and a set $E \subseteq \binom{V}{2}$ of edges; a directed graph $H = (V, E)$ consists of a finite set V and a set $E \subseteq V^2 \setminus \{(v, v) \mid v \in V\}$. For clarity, all undirected graphs are called G and all directed graphs are called H (possibly with indices etc.). For a graph $G = (V, E)$ and vertex set $X \subseteq V$ we use $G - X$ to denote the graph induced by $V \setminus X$; we also use $G - v$ if $X = \{v\}$. Analogous definitions are used for directed graphs H .

Let $H = (V, E)$ be a directed graph and let S and T be two not necessarily disjoint vertex sets in H . A set $X \subseteq V$ is an S, T -separator if in $G - X$ there is no path from $S \setminus X$ to $T \setminus X$; note that X may overlap both S and T and that $S \cap T \subseteq X$ is required. The set T is *closest to S* if there is no S, T -separator X with $X \neq T$ and $|X| \leq |T|$, i.e., if T is the unique minimum S, T -separator in G (cf. [16]). Both separators and closeness have analogous definitions in undirected graphs but they are not required here.

► **Proposition 1** (cf. [16]). *Let $H = (V, E)$ be a directed graph and let $S, T \subseteq V$ such that T is closest to S . For any vertex $v \in V \setminus T$ that is reachable from S in $H - T$ there exist $|T| + 1$ (fully) vertex-disjoint paths from S to $T \cup \{v\}$.*

Proof. Assume for contradiction that such $|T| + 1$ directed paths do not exist. By Menger's Theorem there must be an $S, T \cup \{v\}$ -separator X of size at most $|T|$. Observe that $X \neq T$ since v is reachable from S in $H - T$. Thus, X is an S, T -separator of size at most $|T|$ that is different from T ; this contradicts closeness of T . ◀

For an undirected graph $G = (V, E)$, a *matching* is any set $M \subseteq E$ such that no two edges in M have an endpoint in common. If M is a matching in $G = (V, E)$ then we will say that a path is M -alternating if its edges are alternatingly from M and from $\overline{M} := E \setminus M$. An M, M -path is an M -alternating path whose first and last edge are from M ; it must have odd length. Similarly, we define \overline{M}, M -paths, M, \overline{M} -paths (both of even length), and $\overline{M}, \overline{M}$ -paths (of odd length). If M is a matching of G and v is incident with an edge of M then we use $M(v)$ to denote the other endpoint of that edge, i.e., the *mate* or *partner* of v . Say that a vertex v is *exposed by M* if it is not incident with an edge of M ; we say that

v is *exposable* if it is exposed by some maximum matching of G . A graph $G = (V, E)$ is *factor-critical* if for each vertex $v \in V$ the graph $G - v$ has a perfect matching (a *near-perfect matching* of G); observe that all factor-critical graphs must have an odd number of vertices.

We denote by $LP(G)$ the optimum value of fractional vertex covers, which are exactly the feasible solutions of the well-known LP-relaxation of VERTEX COVER. It is known that the extremal points x of that linear program are half-integral, i.e., $x \in \{0, \frac{1}{2}, 1\}^V$. With this in mind, we will tacitly assume that all considered fractional vertex covers are half-integral. We use the shorthand $[n] := \{1, \dots, n\}$. By $A \dot{\cup} B$ to denote the disjoint union of A and B .

Gallai-Edmonds decomposition. We will now recall the Gallai-Edmonds decomposition following the well-known book of Lovász and Plummer [18].⁴

► **Definition 2.** Let $G = (V, E)$ be a graph. The *Gallai-Edmonds decomposition* of G is a partition of V into three sets A , B , and D where

- D consists of all vertices v of G such that there is a maximum matching M of G that contains no edge incident with v , i.e., that leaves v exposed,
- A is the set of neighbors of D , i.e., $A := N(D)$, and
- B contains all remaining vertices, i.e., $B := V \setminus (A \cup D)$.

It is known (and easy to verify) that the Gallai-Edmonds decomposition of any graph G is unique and can be computed in polynomial time. The Gallai-Edmonds decomposition has a number of useful properties; the following theorem states some of them.

► **Theorem 3** (cf. [18, Theorem 3.2.1]). *Let $G = (V, E)$ be a graph and let $V = A \dot{\cup} B \dot{\cup} D$ be its Gallai-Edmonds decomposition. The following properties hold:*

1. *The connected components of $G[D]$ are factor-critical.*
2. *The graph $G[B]$ has a perfect matching.*
3. *Every maximum matching M of G consists of a perfect matching of $G[B]$, a near-perfect matching of each component of $G[D]$, and a matching of A into D .*

3 Tight vertex covers of factor-critical graphs

In this section we study vertex covers of factor-critical graphs, focusing on those that are of smallest possible size (later called tight vertex covers). We first recall the fact that any factor-critical graph with $n \geq 3$ vertices has no vertex cover of size less than $\frac{1}{2}(n + 1)$. By a similar argument such graphs have no fractional vertex cover of cost less than $\frac{1}{2}n$.

► **Proposition 4** (folklore). *Let $G = (V, E)$ be a factor-critical graph with at least three vertices. Every vertex cover X of G has cardinality at least $\frac{1}{2}(|V| + 1)$ and every fractional vertex cover $x: V \rightarrow \mathbb{R}_{\geq 0}$ of G has cost at least $\frac{1}{2}|V|$.*

Proof. Let $X \subseteq V$ be a vertex cover of G . Since G has at least three vertices and is factor-critical, it has a maximum matching M of size $\frac{1}{2}(|V| - 1) \geq 1$. It follows that X has size at least one. (This is not true for graphs consisting of a single vertex, which are also factor-critical. All other factor-critical graphs have at least three vertices.) Pick any vertex $v \in X$. Since G is factor-critical, there is a maximum matching M_v of $G - v$ of size $\frac{1}{2}(|V| - 1)$. It follows that X must contain at least one vertex from each edge of M_v , and no

⁴ We use B instead of C for $V \setminus (A \cup D)$ to leave the letter C for cycles and connected components.

vertex is contained in two of them. Together with v , which is not in any edge of M_v , this gives a lower bound of $1 + \frac{1}{2}(|V| - 1) = \frac{1}{2}(|V| + 1)$, as claimed.

Let $x: V \rightarrow \mathbb{R}_{\geq 0}$ be a fractional vertex cover of G . We use again the matching M of size at least one from the previous case; let $\{u, v\} \in M$. It follows that $x(u) + x(v) \geq 1$; w.l.o.g. we have $x(v) \geq \frac{1}{2}$. Let M_v be a maximum matching of $G - v$ of size $\frac{1}{2}(|V| - 1)$. For each edge $\{p, q\} \in M_v$ we have $x(p) + x(q) \geq 1$. Since the matching edges are disjoint we get a lower bound of $\sum_{p \in V \setminus \{v\}} x(p) \geq \frac{1}{2}(|V| - 1)$. Together with $x(v) \geq \frac{1}{2}$ we get the claimed lower bound of $\frac{1}{2}|V|$ for the cost of x . \blacktriangleleft

Note that Proposition 4 is tight for example for all odd cycles of length at least three, all of which are factor-critical. We now define tight vertex covers and critical sets.

► **Definition 5** (tight vertex covers, critical sets). Let $G = (V, E)$ be a factor-critical graph with $|V| \geq 3$. A vertex cover X of G is *tight* if $|X| = \frac{1}{2}(|V| + 1)$. Note that this is different from a minimum vertex cover, and a factor-critical graph need not have a tight vertex cover; e.g., odd cliques with at least five vertices are factor-critical but have no tight vertex cover.

A set $Z \subseteq V$ is called a *bad set* of G if there is no tight vertex cover of G that contains Z . The set Z is a *critical set* if it is a minimal bad set, i.e., no tight vertex cover of G contains Z but for all proper subsets Z' of Z there is a tight vertex cover containing Z' .

Observe that a factor-critical graph $G = (V, E)$ has no tight vertex cover if and only if $Z = \emptyset$ is a critical set of G . It may be interesting to note that a set $X \subseteq V$ of size $\frac{1}{2}(|V| + 1)$ is a vertex cover of G if and only if it contains no critical set. (We will not use this fact and hence leave its two line proof to the reader.) The following lemma proves that all critical sets of a factor-critical graph have size at most three; this is of central importance for our kernelization. For the special case of odd cycles, the lemma has a much shorter proof and we point out that all critical sets of odd cycles have size exactly three.

► **Lemma 6.** *Let $G = (V, E)$ be a factor-critical graph with at least three vertices. All critical sets Z of G have size at most three.*

Proof. Let $\ell \in \mathbb{N}$ with $\ell \geq 1$ such that $|V| = 2\ell + 1$; recall that all factor-critical graphs have an odd number of vertices.

Assume for contradiction that there is a critical set Z of G of size at least four. Let $w, x, y, z \in Z$ be any four pairwise different vertices from Z . Let M be a maximum matching of $G - w$. Since G is factor-critical, we get that M is a perfect matching of $G - w$ and has size $|M| = \ell$. Observe that any tight vertex cover of G that contains w must contain exactly one vertex from each edge of M , since its total size is $\frac{1}{2}(|V| + 1) = \ell + 1$. We will first analyze G and show that the presence of certain structures would imply that some proper subset Z' of Z is bad, contradicting the assumption that Z is critical. Afterwards, we will use the absence of these structures to find a tight vertex cover that contains Z , contradicting the fact that it is a critical set.

If there is an M, M -path from x to y then $\{w, x, y\}$ is a bad set, i.e., no tight vertex cover of G contains all three vertices w, x , and y , contradicting the choice of Z : Let $P = (v_1, v_2, \dots, v_{p-1}, v_p)$ denote an M, M -path from $v_1 = x$ to $v_p = y$. Accordingly, we have $\{v_1, v_2\}, \dots, \{v_{p-1}, v_p\} \in M$ and the path P has odd length. Assume that X is a tight vertex cover containing w, x , and y . It follows, since $w \in X$, that X contains exactly one vertex per edge in M ; in particular it contains exactly one vertex per matching edge on the path P . Since $v_1 = x \in X$ we have $v_2 \notin X$. Thus, as $\{v_2, v_3\}$ is an edge of G , we must have $v_3 \in X$ to cover this edge; this in turn implies that $v_4 \notin X$ since it already contains v_3 from the matching edge $\{v_3, v_4\}$. Continuing this argument along the path P we conclude

that $v_{p-1} \in X$ and $v_p \notin X$, contradicting the fact that $v_p = y \in X$. Thus, if there is an M, M -path from x to y then there is no tight vertex cover of G that contains w , x , and y , making $\{w, x, y\}$ a bad set and contradicting the assumption that Z is a critical set. It follows that there can be no M, M -path from x to y . The same argument can be applied also to x and z , and to y and z , ruling out M, M -paths connecting them.

Similarly, if there is an edge $\{u, v\} \in M$ such that z reaches both u and v by (different, not necessarily disjoint) M, \overline{M} -paths then no tight vertex cover of G contains both w and z , contradicting the choice of Z : Let $P = (v_1, v_2, \dots, v_{p-1}, v_p)$ denote an M, \overline{M} -path from $v_1 = z$ to $v_p = u$ with $\{v_1, v_2\}, \{v_3, v_4\}, \dots, \{v_{p-2}, v_{p-1}\} \in M$. Let X be a tight vertex cover of G that contains w and z . It follows (as above) that $v_1, v_3, \dots, v_{p-2} \in X$ and $v_2, v_4, \dots, v_{p-1} \notin X$, by considering the induced M, M -path from $z = v_1$ to v_{p-1} . The fact that $v_{p-1} \notin X$ directly implies that $v_p = u \in X$ in order to cover the edge $\{v_{p-1}, v_p\}$. Repeating the same argument on an M, \overline{M} -path from z to v we get that $v \in X$. Thus, we conclude that u and v are both in X , contradicting the fact that X must contain exactly one vertex of each edge in M . Hence, there is no tight vertex cover of G that contains both w and z . We conclude that $\{w, z\}$ is a bad set, contradicting the choice of Z . Hence, there is no edge $\{u, v\} \in M$ such that z has M, \overline{M} -paths (not necessarily disjoint) to both u and v .

Now we will complete the proof by using the established properties, i.e., the non-existence of certain M -alternating paths starting in z , to construct a tight vertex cover of G that contains all of Z , giving the final contradiction. Using minimality of Z , let X be a tight vertex cover of G that contains $Z \setminus \{z\}$; by choice of Z we have $z \notin X$. We construct the claimed vertex cover $X' \supseteq Z$ from $X' = X$ as follows:

1. Add vertex z to X and remove $M(z)$, i.e., remove the vertex that z is matched to.
2. Add all vertices v to X' that can be reached from z by an M, \overline{M} -path.
3. Remove all vertices from X' that can be reached from z by an M, M -path of length at least three. (There is a single such path of length one from z to $M(z)$ which, for clarity, was handled already above.)

We need to check four things: (1) The procedure above is well-defined, i.e., no vertex can be reached by both M, M - and M, \overline{M} -paths from z . (2) The size of X' is at most $|X| = \ell + 1$. (3) X' is a vertex cover. (4) The set X' contains w, x, y , and z .

(1) Assume that there is a vertex v such that z reaches v both by an M, M -path $P = (v_1, v_2, \dots, v_p)$ with $v_1 = z$ and $v_p = v$, and by an M, \overline{M} -path P' . Observe that $\{v_{p-1}, v_p\} \in M$ since P is an M, M -path and, hence, that $P'' = (v_1, \dots, v_{p-1})$ is an M, \overline{M} -path from v to v_{p-1} . Together, P' and P'' constitute two M, \overline{M} -paths from z to both endpoints v_{p-1} and v_p of the matching edge $\{v_{p-1}, v_p\}$; a contradiction (since we ruled out this case earlier).

(2) In the first step, we add z and remove $M(z)$. Note that $z \notin X$ implies that $M(z) \in X$ (we start with $X' = X$). Thus the size of X' does not change. Consider a vertex v that is added in the second step, i.e., with $v \notin X$: There is an M, \overline{M} -path P from z to v . Since $w \in X$ we know that $v \neq w$. Thus, since M is a perfect matching of $G - w$, there is a vertex u with $u = M(v)$. The vertex $u := M(v)$ must be in X to cover the edge $\{v, u\} \in M$, as $v \notin X$. Moreover, u cannot be on P since that would make it incident with a second matching edge other than $\{u, v\}$. Thus, by extending P with $\{v, u\}$ we get an M, M -path from z to u , implying that u is removed in the second step. Since $u \in X$ the total size change is zero. Observe that the vertex $u = M(v)$ used in this argument is not used for any other vertex v' added in the second step since it is only matched to v . Similarly, due to (1),

the vertex u is not also added in the second step since it cannot be simultaneously have an M, \overline{M} -path from z .

(3) Assume for contradiction that some edge $\{u, v\}$ is not covered by X' , i.e., that $u, v \notin X'$. Since $w \in X'$ is the only unmatched vertex it follows that both u and v are incident with some edge of M . We distinguish two cases, namely (a) $\{u, v\} \in M$ and (b) $\{u, v\} \notin M$.

(3.a) If $\{u, v\} \in M$ then without loss of generality assume $u \in X$ (as X is a vertex cover). By our assumption we have $u \notin X'$, which implies that we have removed it on account of having an M, M -path P from z to u . Since $\{u, v\} \in M$ the path P must visit v as its penultimate vertex; there is no other way for an M, M -path to reach u . This, however, implies that there is an M, \overline{M} -path from z to v , and that we have added v in the second step; a contradiction.

(3.b) In this case we have $\{u, v\} \notin M$. Again, without loss of generality, assume that $u \in X$. Since $u \notin X'$ there must be an M, M -path P from z to u . If P does not contain v then extending P by edge $\{u, v\} \notin M$ would give an M, \overline{M} -path from z to v and imply that $v \in X'$; a contradiction. In the remaining case, the vertex v is contained in P ; let P' denote the induced path from z to v (not containing u as it is the final vertex of P). Since $v \notin X'$ we know that P' cannot be an M, \overline{M} -path, or else we would have $v \in X'$, and hence it must be an M, M -path. Now, however, extending P' via $\{v, u\} \notin M$ yields an M, \overline{M} -path from z to u , contradicting (1). Altogether, we conclude that X' is indeed a vertex cover.

(4) Clearly, $z \in X'$ by construction. Similarly, $w \in X'$ since it is contained in X and it cannot be removed since there is no incident M -edge (i.e., no M, M -paths from z can end in w). Finally, regarding x and y , we proved earlier that there are no M, M -paths from z to x or from z to y . Thus, since both x and y are in X they must also be contained in X' .

We have showed that under the assumption of minimality of Z and using $|Z| \geq 4$ one can construct a vertex cover X' of optimal size $\ell + 1$ that contains Z entirely. This contradicts the choice of Z and completes the proof. ◀

4 Nice decompositions

The well-known Gallai-Edmonds decomposition plays an important role in the FPT-algorithm of Garg and Philip [10]. It is, in principle, also very useful for our kernelization result, but it is much more convenient to use a form that is both relaxed (in part) but also includes a certain maximum matching of the graph, called *nice decomposition*. Due to space restrictions, we give the definition directly rather than first defining a natural intermediate form.

► **Definition 7** (nice decomposition). Let $G = (V, E)$ be a graph. A *nice decomposition* of G is a tuple (A, B, D, M) where $V = A \dot{\cup} B \dot{\cup} D$ and M is a maximum matching of G such that

1. $A = N(D)$, i.e., all vertices not in D that are adjacent to D ,
2. each connected component of $G[D]$ is factor-critical,
3. M restricted to B is a perfect matching of $G[B]$,
4. M restricted to any component C of $G[D]$ is a near-perfect matching of $G[C]$,
5. each vertex of A is matched by M to a vertex of D , and
6. for each singleton component $\{v\}$ of $G[D]$ there is a vertex $u \in A$ with $\{u, v\} \in M$.

For a nice decomposition (A, B, D, M) of G it will be of importance for us which components of $G[D]$ are matched to a vertex in A . Since M induces a near-perfect matching on each component of $G[D]$, there is always at most one such vertex per component of $G[D]$.

► **Definition 8** (matched/unmatched components of $G[D]$). Let $G = (V, E)$ be a graph and let (A, B, D, M) be a nice decomposition of G . We say that a connected component C of $G[D]$ is *matched* if there are vertices $v \in C$ and $u \in N(C) \subseteq A$ such that $\{u, v\} \in M$; we will also say that u and C are matched to one another. Otherwise, we say that C is *unmatched*. Note that edges of M with both ends in C have no influence on whether C is matched or unmatched.

We use \mathcal{C}_1 to denote the set of matched singleton components in $G[D]$; a nice decomposition has no unmatched singleton components. We use \mathcal{C}_3 and $\hat{\mathcal{C}}_3$ for matched and unmatched non-singleton components. By A_1 and A_3 we denote the set of vertices in A that are matched to singleton respectively non-singleton components of $G[D]$; note that $A = A_1 \dot{\cup} A_3$.

Our main reason for preferring nice decompositions is captured in the following lemma, namely that deleting certain types of components, e.g., in a reduction rule, allows the obtained graph G' to effectively inherit a nice decomposition.

► **Lemma 9.** *Let $G = (V, E)$ be a graph, (A, B, D, M) a nice decomposition, and $C \in \hat{\mathcal{C}}_3$ an unmatched component of $G[D]$. Then (A, B, D', M') is a nice decomposition of $G' = G - C$ where M' is M restricted to $V(G') = V \setminus C$ and where $D' := D \setminus C$. The corresponding sets A_1, A_3, \mathcal{C}_1 , and \mathcal{C}_3 are the same as for G . For $\hat{\mathcal{C}}_3$ we have $\hat{\mathcal{C}}'_3 = \hat{\mathcal{C}}_3 \setminus \{C\}$.*

5 Nice decompositions and vertex covers

Due to space restrictions, this section gives a brief summary of the relation between a nice decomposition (A, B, D, M) of a graph G and (certain) vertex covers of G . We first prove a lower bound on $VC(G)$ in terms of (A, B, D, M) and relate it to $2LP(G) - MM(G)$. Note that Garg and Philip [10] proved that $2LP(G) - MM(G)$ is a lower bound for the vertex cover size for every graph G , but we require the bound of $|M| + |\hat{\mathcal{C}}_3|$ related to our decompositions, and the equality to $2LP(G) - MM(G)$ is “only” required to complete the kernelization later.

► **Lemma 10.** *Let $G = (V, E)$ be a graph and let (A, B, D, M) be a nice decomposition of G . Each vertex cover of G has size at least $|M| + |\hat{\mathcal{C}}_3| = 2LP(G) - MM(G)$.*

Intuitively, a vertex cover X of size at most $|M| + |\hat{\mathcal{C}}_3| + \ell$ can “overpay” only ℓ times as compared to spending one vertex per edge of M and $\frac{1}{2}(|C| + 1)$ for any component $C \in \hat{\mathcal{C}}_3$. Conversely, X must induce tight vertex covers of all but at most ℓ components $C \in \hat{\mathcal{C}}_3$.

► **Definition 11** (active component). Let $G = (V, E)$ be a graph, let (A, B, D, M) be a nice decomposition of G , and let X be a vertex cover of G . A component $C \in \hat{\mathcal{C}}_3$ is *active* (w.r.t. X) if X contains more than $\frac{1}{2}(|C| + 1)$ vertices of C , i.e., if $X \cap C$ is not a tight vertex cover of $G[C]$.

► **Definition 12** (set X_{op}). Let $G = (V, E)$ be a graph and let (A, B, D, M) be a nice decomposition of G . For $X \subseteq V$ define $X_{\text{op}} = X_{\text{op}}(A_1, A_3, M, X) \subseteq A \cap X$ to contain all vertices v that fulfill either of the following two conditions:

1. $v \in A_1$ and X contains both v and $M(v)$.
2. $v \in A_3$ and X contains v .

Both conditions of Definition 12 capture parts of the graph where X contains more vertices than implied by the lower bound. To see this for the second condition, note that if $v \in A_3 \cap X$ then X still needs at least $\frac{1}{2}(|C| + 1)$ vertices of the component $C \in \mathcal{C}_3$ that v is matched to; since there are $\frac{1}{2}(|C| + 1)$ matching edges that M has between vertices of $C \cup \{v\}$ we find that X (locally) exceeds the lower bound, as $|X \cap (C \cup \{v\})| \geq 1 + \frac{1}{2}(|C| + 1)$. Conversely, if X does match the lower bound on $C \cup \{v\}$ then it cannot contain v .

We now prove formally that a vertex cover X of size close to the lower bound of Lemma 10 has only few active components and only a small set $X_{\text{op}} \subseteq X$.

► **Lemma 13.** *Let $G = (V, E)$ be a graph, let (A, B, D, M) be a nice decomposition of G , let X be a vertex cover of G , and let $X_{\text{op}} = X_{\text{op}}(A_1, A_3, M, X)$. The set X_{op} has size at most ℓ and there are at most ℓ active components in $\hat{\mathcal{C}}_3$ with respect to X where $\ell = |X| - (|M| + |\hat{\mathcal{C}}_3|) = |X| - (2LP(G) - MM(G))$.*

The central question is of course how the different structures where X exceeds the lower bound interact. We are only interested in aspects that are responsible for not allowing a tight vertex cover for any (unmatched, non-singleton) components $C \in \hat{\mathcal{C}}_3$. This happens exactly due to vertices in A that are adjacent to C and that are not selected by X . Between components of $G[B]$ and non-singleton components of $G[D]$ there are M -alternating paths with vertices alternatingly from A and from singleton components of $G[D]$ since vertices in A are all matched to D and singleton components in $G[D]$ have all their neighbors in A . Unless X contains both vertices of a matching edge, it contains the A - or the D -vertices of such a path. Unmatched components of $G[D]$ and components of $G[B]$ have all neighbors in A . Matched components C in $G[D]$ with matched neighbor $v \in A$ enforce not selecting v for X unless X spends more than the lower bound; in this way, they lead to selection of D -vertices on M -alternating paths. Intuitively, this leads to two “factions” that favor either A - or D -vertices and that are effectively separated when X selects both A - and D -endpoint of a matching edge. An optimal solution need not separate all neighbors in A of any component $C \in \hat{\mathcal{C}}_3$, and C may still have a tight vertex cover or paying for a larger cover of C is overall beneficial. The following auxiliary directed graph H captures this situation and for certain vertex covers X reachability of $v \in A$ in $H - X_{\text{op}}$ will be proved to be equivalent with $v \notin X$.

► **Definition 14** (auxiliary directed graph H). Let $G = (V, E)$ be a graph and let (A, B, D, M) be a nice decomposition of G . Define a *directed graph* $H = H(G, A, B, D, M)$ on vertex set A by letting (u, v) be a directed edge of H , for $u, v \in A$, whenever there is a vertex $w \in D$ with $\{u, w\} \in E \setminus M$ and $\{w, v\} \in M$.

There are three important properties of vertex covers in relation to the corresponding graph H . Two of them hold only for what we call *dominant* vertex covers and their proofs build on a fairly technical replacement argument. We will define dominant vertex covers next and then summarize the properties in a single lemma.

► **Definition 15** (dominant vertex cover). Let $G = (V, E)$ be a graph and let (A, B, D, M) be a nice decomposition of G . A vertex cover $X \subseteq V$ of G is *dominant* if G has no vertex cover of size less than $|X|$ and no vertex cover of size $|X|$ contains fewer vertices of D .

► **Lemma 16.** *Let $G = (V, E)$ be a graph, let (A, B, D, M) be a nice decomposition of G , and X a vertex cover of G . Let $H = H(G, A, B, D, M)$ and let $X_{\text{op}} = X_{\text{op}}(A_1, A_3, M, X)$. The following properties hold:*

1. *If X is dominant then X_{op} is closest to A_3 in H .*
2. *If $v \in A$ is reachable from A_3 in $H - X_{\text{op}}$ then X does not contain v .*
3. *If X is dominant and $v \in A$ is not reachable from A_3 in $H - X_{\text{op}}$ then X contains v .*

All three properties are crucial for applying the matroid tools of Kratsch and Wahlström [16]. Closeness of X_{op} is needed to translate between reachability and independence in an appropriate matroid. The latter two properties are required to translate between the directed graph, where the tools are applied, and the undirected input graph G .

6 Randomized polynomial kernelization

In this section, we describe our randomized polynomial kernelization for VERTEX COVER($k - (2LP - MM)$). For convenience, let us fix an input instance (G, k, ℓ) , i.e., $G = (V, E)$ is a graph for which we want to know whether it has a vertex cover of size at most k ; the parameter is $\ell = k - (2LP(G) - MM(G))$, where $LP(G)$ is the minimum cost of a fractional vertex cover of G and $MM(G)$ is the size of a largest matching.

From previous work of Garg and Philip [10] we know that the well-known linear program-based preprocessing for VERTEX COVER (cf. [5]) can also be applied to VERTEX COVER($k - (2LP - MM)$); the crucial new aspect is that this operation does not increase the value $k - (2LP - MM)$. The LP-based preprocessing builds on the half-integrality of fractional vertex covers and a result of Nemhauser and Trotter [22] stating that all vertices with value 1 and 0 in an optimal fractional vertex cover $x: V \rightarrow \{0, \frac{1}{2}, 1\}$ are included respectively excluded in at least one minimum (integral) vertex cover. Thus, only vertices with value $x(v) = \frac{1}{2}$ remain and the best LP solution costs exactly $\frac{1}{2}$ times number of (remaining) vertices. For our kernelization we only require the fact that if G is reduced under this reduction rule then $LP(G) = \frac{1}{2}(|V(G)|)$; e.g., we do not require $x: V \rightarrow \{\frac{1}{2}\}$ to be the unique optimal fractional vertex cover. Without loss of generality, we assume that our given graph $G = (V, E)$ already fulfills $LP(G) = \frac{1}{2}|V|$.

► **Observation 17.** *If $LP(G) = \frac{1}{2}|V|$ then $2LP(G) - MM(G) = |V| - MM(G)$. In other words, if M is a maximum matching of G then the lower bound $2LP(G) - MM(G) = |V| - MM(G) = |V| - |M|$ is equal to cardinality of M plus the number of isolated vertices.*

As a first step, let us compute the Gallai-Edmonds decomposition $V = A \dot{\cup} B \dot{\cup} D$ of G ; this can be done in polynomial time.⁵ Using $LP(G) = \frac{1}{2}|V|$ we can find a maximum matching M of G such that (A, B, D, M) is a nice decomposition of G .

► **Lemma 18.** *Given $G = (V, E)$ with $LP(G) = \frac{1}{2}|V|$ and a Gallai-Edmonds decomposition $V = A \dot{\cup} B \dot{\cup} D$ of G one can in polynomial time compute a maximum matching M of G such that (A, B, D, M) is a nice decomposition of G .*

We fix a nice decomposition (A, B, D, M) of G obtained via Lemma 18. We have already learned about the relation of dominant vertex covers X , their intersection with the set A , and separation of A vertices from A_3 in $H - X_{\text{op}}$, where $H = H(G, A, B, D, M)$. It is safe to assume that solutions are dominant vertex covers as among minimum vertex covers there is a minimum intersection with D . We would now like to establish that most components of $\hat{\mathcal{C}}_3$ can be deleted (while reducing k by the cost for corresponding tight vertex covers). Clearly, since any vertex cover pays at least for tight covers of these components, we cannot turn a yes- into a no-instance this way. However, if the instance is no then it might become yes.

⁵ The main expenditure is finding the set D . A straightforward approach is to compute a maximum matching M_v of $G - v$ for each $v \in V$. If $|M_v| = MM(G)$ then v is in D as M_v is maximum and exposes v ; otherwise $v \notin D$ as no maximum matching exposes v .

In the following, we will try to motivate both the selection process for components of $\hat{\mathcal{C}}_3$ that are deleted as well as the high-level proof strategy for establishing correctness. We will tacitly ignore most technical details, like parameter values, getting appropriate nice decompositions, etc., and refer to the formal proof instead. Assume that we are holding a no-instance (G, k, ℓ) . Consider for the moment, the effect of deleting all components $C \in \hat{\mathcal{C}}_3$ that have tight vertex covers and updating the budget accordingly; for simplicity, say they all have such vertex covers. Let (G_0, k_0, ℓ) be the obtained instance; if this instance is no as well, then deleting any subset of $\hat{\mathcal{C}}_3$ also preserves the correct answer (namely: no). Else, if (G_0, k_0, ℓ) is yes then pick any dominant vertex cover X^0 for it. We could attempt to construct a vertex cover of G of size at most k by adding back the components of C and picking a tight vertex cover for each; crucially, these covers must also handle edges between C and A . Since (G, k, ℓ) was assumed to be a no-instance, there must be too many components $C \in \hat{\mathcal{C}}_3$ for which this approach fails. For any such component, the adjacent vertices in $A \setminus X^0$ force a selection of their neighbors $Z_A = N(A) \cap C$ that cannot be completed to a tight vertex cover of C . To avoid turning the no-instance (G, k, ℓ) into a yes-instance (G', k', ℓ) we have to keep enough components of $\hat{\mathcal{C}}_3$ in order to falsify any suggested solution X' of size at most k' for G . The crux is that there may be an exponential number of such solutions and that we do not know any of them. This is where the auxiliary directed graph and related technical lemmas as well as the matroid-based tools of Kratsch and Wahlström [16] are essential.

Let us outline how we arrive at an application of the matroid-based tools. Crucially, if C (as above) has no tight vertex cover containing $Z_A = N(A) \cap C$ then, by Lemma 6, there is a set $Z \subseteq Z_A$ of size at most three such that no tight vertex cover contains Z . Accordingly, there is a set $T \subseteq A \setminus X^0$ of size at most three whose neighborhood in C contains Z . Thus, the fact that X^0 contains no vertex of T is responsible for not allowing a tight vertex cover of C . This in turn, by Lemma 16 means that all vertices in T are reachable from A_3 in $H - X_{\text{op}}^0$. Recalling that a set X_{op}^0 corresponding to a dominant vertex cover is also closest to A_3 , we can apply a result from [16] that generates a sufficiently small representative set of sets T corresponding to components of $\hat{\mathcal{C}}_3$. If a dominant vertex cover has any reachable sets T then the lemma below guarantees that at least one such set is in the output. For each set we select a corresponding component $C \in \hat{\mathcal{C}}_3$ and then start over on the remaining components. After $\ell + 1$ iterations we can prove that for any not selected component C , which we delete, and any proposed solution X' for the resulting graph that does not allow a tight vertex cover for C , there are $\ell + 1$ other selected components on which X' cannot be tight. This is a contradiction as there are at most ℓ such active components by Lemma 13.

Concretely, we will use the following lemma about representative sets of vertex sets of size at most three regarding reachability in a directed graph (modulo deleting a small set of vertices). Notation of the lemma is adapted to the present application. The original result is for pairs of vertices in a directed graph (see [15, Lemma 2]) but extends straightforwardly to sets of fixed size q and to sets of size at most q . We provide a proof in the full version for completeness. Note that the lemma is purely about reachability of small sets in a directed graph (like the DIGRAPH PAIR CUT problem studied in [15, 16]) and we require the structural lemmas proved so far to negotiate between this an VERTEX COVER($k - (2LP - MM)$).

► **Lemma 19.** *Let $H = (V_H, E_H)$ be a directed graph, let $S_H \subseteq V_H$, let $\ell \in \mathbb{N}$, and let \mathcal{T} be a family of nonempty vertex sets $T \subseteq V_H$ each of size at most three. In randomized polynomial time, with failure probability exponentially small in the input size, we can find a set $\mathcal{T}^* \subseteq \mathcal{T}$ of size $\mathcal{O}(\ell^3)$ such that for any set $X_H \subseteq V_H$ of size at most ℓ that is closest to S_H if there is a set $T \in \mathcal{T}$ such that all vertices $v \in T$ are reachable from S_H in $H - X_H$ then there is a corresponding set $T^* \in \mathcal{T}^*$ satisfying the same properties.*

Using the lemma we will be able to identify a small set \mathcal{C}_{rel} of components of $\hat{\mathcal{C}}_3$ that contains for each dominant vertex cover X of G of size at most k all active components with respect to X . Conversely, if there is no solution of size k , we will have retained enough components of $\hat{\mathcal{C}}_3$ to preserve this fact. Concretely, the set \mathcal{C}_{rel} is computed as follows:

1. Let $\mathcal{C}_{\text{rel}}^0$ contain all components $C \in \hat{\mathcal{C}}_3$ that have no vertex cover of size at most $\frac{1}{2}(|C|+1)$. Clearly, these components are active for every vertex cover of G . We know from Lemma 13 that there are at most ℓ such components if the instance is **yes**. We can use the algorithm of Garg and Philip [10] to test in polynomial time whether any $C \in \hat{\mathcal{C}}_3$ has a vertex cover of size at most $k_C := \frac{1}{2}(|C|+1)$: We have parameter value

$$k_C - (2LP(G[C]) - MM(G[C])) = \frac{1}{2}(|C|+1) - (|C| - \frac{1}{2}(|C|-1)) = 0.$$

We could of course also use an algorithm for VERTEX COVER parameterized above maximum matching size, where we would have parameter value 1. If there are more than ℓ components C with no vertex cover of size $\frac{1}{2}(|C|+1)$ then we can safely reject the instance. Else, as indicated above, let $\mathcal{C}_{\text{rel}}^0$ contain all these components and continue.

2. Let $i = 1$. We will repeat the following steps for $i \in \{1, \dots, \ell+1\}$.
3. Let \mathcal{T}^i contain all nonempty sets $T \subseteq A$ of size at most three such that there is a component $C \in \hat{\mathcal{C}}_3 \setminus (\mathcal{C}_{\text{rel}}^0 \cup \dots \cup \mathcal{C}_{\text{rel}}^{i-1})$ such that:
 - a. There is a set $Z \subseteq N_G(T) \cap C$ of at most three neighbors of T in C such that no vertex cover of $G[C]$ of size $\frac{1}{2}(|C|+1)$ contains Z . Note that $Z \neq \emptyset$ since $C \notin \mathcal{C}_{\text{rel}}^0$ implies that it has at least some vertex cover of size $\frac{1}{2}(|C|+1)$.
 - b. For each C and $Z \subseteq C$ of size at most three, existence of a vertex cover of $G[C]$ of size $k_C := \frac{1}{2}(|C|+1)$ containing Z can be tested by the algorithm of Garg and Philip [10] since the parameter value is constant. Concretely, run the algorithm on $G[C \setminus Z]$ and solution size $k_C - |Z|$ and observe that the parameter value is

$$(k_C - |Z|) - (2LP(G[C \setminus Z]) - MM(G[C \setminus Z])).$$

Using that $LP(G[C \setminus Z]) \geq LP(G[C]) - |Z|$ and $MM(G[C \setminus Z]) \leq MM(G[C]) = \frac{1}{2}(|C|-1)$ this value can be upper bounded by

$$\begin{aligned} & k_C - |Z| - 2LP(G[C]) + 2|Z| + MM(G[C]) \\ &= \frac{1}{2}(|C|+1) - |Z| - |C| + 2|Z| + \frac{1}{2}(|C|-1) = |Z|. \end{aligned}$$

Since $|Z| \leq 3$ the parameter value is at most three and the FPT-algorithm of Garg and Philip [10] runs in polynomial time.

Intuitively, C must always be active for vertex covers not containing T , but for the formal correctness proof that we give later the above description is more convenient.

4. Apply Lemma 19 to graph $H = H(G, A, B, D, M)$ on vertex set $V_H = A$, set $S_H = A_3 \subseteq A$, integer ℓ , and family \mathcal{T}^i of nonempty subsets of A of size at most three to compute a subset \mathcal{T}^{i*} of \mathcal{T}^i in randomized polynomial time. The size of $|\mathcal{T}^{i*}|$ is $\mathcal{O}(\ell^3)$.
5. Select a set $\mathcal{C}_{\text{rel}}^i$ as follows: For each $T \in \mathcal{T}^{i*}$ add to $\mathcal{C}_{\text{rel}}^i$ a component $C \in \hat{\mathcal{C}}_3 \setminus (\mathcal{C}_{\text{rel}}^0 \cup \dots \cup \mathcal{C}_{\text{rel}}^{i-1})$ such that C fulfills the condition for T in Step 3, i.e., such that:
 - a. There is a set $Z \subseteq N_G(T) \cap C$ of at most three neighbors of T in C such that no vertex cover of $G[C]$ of size $\frac{1}{2}(|C|+1)$ contains Z . (We know that Z must be nonempty.)
Clearly, the size of $|\mathcal{C}_{\text{rel}}^i|$ is $\mathcal{O}(\ell^3)$. Note that the same component C can be chosen for multiple sets $T \in \mathcal{T}^{i*}$ but we only require an upper bound on $|\mathcal{C}_{\text{rel}}^i|$

6. If $i < \ell + 1$ then increase i by one and return to Step 3. Else return the set $\mathcal{C}_{\text{rel}} := \bigcup_{i=0}^{\ell+1} \mathcal{C}_{\text{rel}}^i$. The size of \mathcal{C}_{rel} is $\mathcal{O}(\ell^4)$ since it is the union of $\ell + 2$ sets that are each of size $\mathcal{O}(\ell^3)$.

In particular, we will be interested in the components $C \in \hat{\mathcal{C}}_3$ that are not in \mathcal{C}_{rel} . We call these *irrelevant components* and let $\mathcal{C}_{\text{irr}} := \hat{\mathcal{C}}_3 \setminus \mathcal{C}_{\text{rel}}$ denote the set of all irrelevant components. (Of course we still need to prove that they are true to their name.)

► **Lemma 20.** *Let G' be obtained by deleting from G all vertices of irrelevant components, i.e., $G' := G - \bigcup_{C \in \mathcal{C}_{\text{irr}}} C$, and let $k' = k - \sum_{C \in \mathcal{C}_{\text{irr}}} \frac{1}{2}(|C| + 1)$, i.e., k' is equal to k minus the lower bounds for vertex covers of the irrelevant components. Then G has a vertex cover of size at most k if and only if G' has a vertex cover of size at most k' . Moreover, $k - (2LP(G) - MM(G)) = k' - (2LP(G') - MM(G'))$, i.e., the instances (G, k, ℓ) and (G', k', ℓ') of VERTEX COVER($k - (2LP - MM)$) have the same parameter value $\ell = \ell'$.*

We can now complete our kernelization. According to Lemma 20 we may delete all irrelevant components and update k . We obtain a graph G' and integer k' such that:

1. G' has a vertex cover of size at most k' if and only if G has a vertex cover of size at most k , i.e., the instances (G, k) and (G', k') for VERTEX COVER are equivalent.
2. As a part of the proof of Lemma 20 we showed that $k' = |M'| + |\hat{\mathcal{C}}'_3| + \ell$ where $\hat{\mathcal{C}}'_3$ is the set of unmatched non-singleton components of $G'[D']$ with respect to M' .
3. From Lemma 9 we know that $\hat{\mathcal{C}}'_3$ is equal to the set $\hat{\mathcal{C}}_3$ minus the components $C \in \mathcal{C}_{\text{irr}}$ that were removed to obtain G' . In other words, $\hat{\mathcal{C}}'_3 = \hat{\mathcal{C}}_3 \setminus \mathcal{C}_{\text{irr}} = \mathcal{C}_{\text{rel}}$.
4. We know from Step 6 that $|\mathcal{C}_{\text{rel}}| = \mathcal{O}(\ell^4)$. Hence, $|\hat{\mathcal{C}}'_3| = \mathcal{O}(\ell^4)$.
5. Let us consider $p := k' - |M'|$, which is the parameter value of (G', k') when considered as an instance of VERTEX COVER parameterized above the size of a maximum matching. Clearly, $p = k' - |M'| = |M'| + |\hat{\mathcal{C}}'_3| + \ell - |M'| = \ell + \mathcal{O}(\ell^4) = \mathcal{O}(\ell^4)$.
6. We can now apply the randomized polynomial kernelization for VERTEX COVER($k - MM$) [16] to get a polynomial kernelization for VERTEX COVER($k - (2LP - MM)$). On input of (G', k', p) it returns an equivalent instance (G^*, k^*, p^*) of size $\mathcal{O}(p^c)$ for some constant c . We may assume that $k^* = \mathcal{O}(p^c)$ since else it would exceed the number of vertices in G^* and we may as well return a **yes**-instance of constant size. Let $\ell^* = k^* - (2LP(G^*) - MM(G^*))$, i.e., the parameter value of the instance (G^*, k^*, ℓ^*) of VERTEX COVER($k - (2LP - MM)$). Clearly, $\ell^* \leq k^* = \mathcal{O}(p^c)$. Thus, (G^*, k^*, ℓ^*) has size and parameter value $\mathcal{O}(p^c)$.

► **Theorem 21.** VERTEX COVER($k - (2LP - MM)$) has a randomized polynomial kernelization with error probability exponentially small in the input size.

7 Conclusion

We have presented a randomized polynomial kernelization for VERTEX COVER($k - (2LP - MM)$) by giving a (randomized) polynomial parameter transformation to VERTEX COVER($k - MM$). This improves upon the smallest parameter, namely $k - LP(G)$, for which such a result was known [16]. The kernelization for VERTEX COVER($k - MM$) [16] involves reductions to and from ALMOST 2-SAT(k), which can be done without affecting the parameter value (cf. [23]). We have not attempted to optimize the total size. Given an instance (G, k, ℓ) for VERTEX COVER($k - (2LP - MM)$) we get an equivalent instance of ALMOST 2-SAT(k) with $\mathcal{O}(k^{24})$ variables and size $\mathcal{O}(k^{48})$, which still needs to be reduced to a VERTEX COVER instance.

It seems likely that the kernelization can be improved if one avoids the blackbox use of the kernelization for VERTEX COVER($k - MM$) and the detour via ALMOST 2-SAT(k). In particular, the underlying kernelization for ALMOST 2-SAT(k) applies, in part, the same

representative set machinery to reduce the number of a certain type of clauses. Conceivably the two applications can be merged, thus avoiding the double blow-up in size. As a caveat, it appears to be likely that this would require a much more obscure translation into a directed separation problem. Moreover, the kernelization for $\text{ALMOST 2-SAT}(k)$ requires an approximate solution, and it is likely that the same would be true for this approach. It would of course also be interesting whether a deterministic polynomial kernelization is possible, but this is, e.g., already not known for $\text{ALMOST 2-SAT}(k)$ and $\text{VERTEX COVER}(k - \text{MM})$.

We find the appearance of a notion of critical sets of size at most three and the derived separation problem in the auxiliary directed graph quite curious. For the related problem of separating at least one vertex from each of a given set of triples from some source s by deleting at most ℓ vertices (a variant of DIGRAPH PAIRCUT [16]) there is a natural $\mathcal{O}^*(3^\ell)$ time algorithm that performs at most ℓ three-way branchings before finding a solution (if possible). It would be interesting whether a complete encoding of $\text{VERTEX COVER}(k - (2\text{LP} - \text{MM}))$ into a similar form would be possible, since that would imply an algorithm that exactly matches the running time of the algorithm of the algorithm by Garg and Philip [10].

References

- 1 Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009. doi:10.1016/j.jcss.2009.04.001.
- 2 Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011. doi:10.1016/j.tcs.2011.04.039.
- 3 Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001. doi:10.1006/jagm.2001.1186.
- 4 Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010. doi:10.1016/j.tcs.2010.06.026.
- 5 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 6 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On the hardness of losing width. *Theory Comput. Syst.*, 54(1):73–82, 2014. doi:10.1007/s00224-013-9480-1.
- 7 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3, 2013. doi:10.1145/2462896.2462899.
- 8 Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014. doi:10.1145/2629620.
- 9 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.
- 10 Shivam Garg and Gevarghese Philip. Raising the bar for vertex cover: Fixed-parameter tractability above a higher guarantee. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1152–1166. SIAM, 2016. doi:10.1137/1.9781611974331.ch80.
- 11 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

- 12 Bart M. P. Jansen. *The power of data reduction: Kernels for fundamental graph problems*. PhD thesis, Utrecht University, 2013.
- 13 Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013. doi:10.1007/s00224-012-9393-4.
- 14 Stefan Kratsch. Recent developments in kernelization: A survey. *Bulletin of the EATCS*, 113, 2014. URL: <http://eatcs.org/beatcs/index.php/beatcs/article/view/285>.
- 15 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *CoRR*, abs/1111.2195, 2011. URL: <http://arxiv.org/abs/1111.2195>.
- 16 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 450–459. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.46.
- 17 Daniel Lokshantov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms*, 11(2):15:1–15:31, 2014. doi:10.1145/2566616.
- 18 László Lovász and Michael D. Plummer. *Matching Theory*. North-Holland, 1986.
- 19 Meena Mahajan and Venkatesh Raman. Parameterizing above guaranteed values: Maxsat and maxcut. *J. Algorithms*, 31(2):335–354, 1999. doi:10.1006/jagm.1998.0996.
- 20 Sounaka Mishra, Venkatesh Raman, Saket Saurabh, Somnath Sikdar, and C. R. Subramanian. The complexity of König subgraph problems and above-guarantee vertex cover. *Algorithmica*, 61(4):857–881, 2011. doi:10.1007/s00453-010-9412-2.
- 21 N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. LP can be a cure for parameterized problems. In Christoph Dürr and Thomas Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*, volume 14 of *LIPICs*, pages 338–349. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012. doi:10.4230/LIPICs.STACS.2012.338.
- 22 George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Math. Program.*, 8(1):232–248, 1975. doi:10.1007/BF01580444.
- 23 Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Paths, flowers and vertex cover. In Camil Demetrescu and Magnús M. Halldórsson, editors, *Algorithms - ESA 2011 - 19th Annual European Symposium, Saarbrücken, Germany, September 5-9, 2011. Proceedings*, volume 6942 of *Lecture Notes in Computer Science*, pages 382–393. Springer, 2011. doi:10.1007/978-3-642-23719-5_33.
- 24 Igor Razgon and Barry O’Sullivan. Almost 2-sat is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009. doi:10.1016/j.jcss.2009.04.002.