

# Compression via Matroids: A Randomized Polynomial Kernel for Odd Cycle Transversal

Stefan Kratsch<sup>\*†</sup>

Magnus Wahlström<sup>‡</sup>

## Abstract

The ODD CYCLE TRANSVERSAL problem (OCT) asks whether a given graph can be made bipartite by deleting at most  $k$  of its vertices. In a breakthrough result Reed, Smith, and Vetta (Operations Research Letters, 2004) gave a  $\mathcal{O}(4^k kmn)$  time algorithm for it, the first algorithm with polynomial runtime of uniform degree for every fixed  $k$ . It is known that this implies a polynomial-time compression algorithm that turns OCT instances into equivalent instances of size at most  $\mathcal{O}(4^k)$ , a so-called kernelization. Since then the existence of a polynomial kernel for OCT, i.e., a kernelization with size bounded polynomially in  $k$ , has turned into one of the main open questions in the study of kernelization. Despite the impressive progress in the area, including the recent development of lower bound techniques (Bodlaender et al., ICALP 2008; Fortnow and Santhanam, STOC 2008) and meta-results on kernelizations for graph problems on planar and other sparse graph classes (Bodlaender et al., FOCS 2009; Fomin et al., SODA 2010), the existence of a polynomial kernel for OCT has remained open, even when the input is restricted to be planar.

This work provides the first (randomized) polynomial kernelization for OCT. We introduce a novel kernelization approach based on matroid theory, where we encode all relevant information about a problem instance into a matroid with a representation of size polynomial in  $k$ . For OCT, the matroid is built to allow us to simulate the computation of the iterative compression step of the algorithm of Reed, Smith, and Vetta, applied (for only one round) to an approximate odd cycle transversal which it is aiming to shrink to size  $k$ . The process is randomized with one-sided error exponentially small in  $k$ , where the result can contain false positives but no false negatives, and the size guarantee is cubic in the size of the approximate solution. Combined with an  $\mathcal{O}(\sqrt{\log n})$ -approximation (Agarwal et al., STOC 2005), we get a reduction of the instance to size  $\mathcal{O}(k^{4.5})$ , implying a randomized polynomial kernelization. Interestingly, the known lower bound techniques can be seen to exclude ran-

domized kernels that produce no false negatives, as in fact they exclude even co-nondeterministic kernels (Dell and van Melkebeek, STOC 2010). Therefore, our result also implies that deterministic kernels for OCT cannot be excluded by the known machinery.

## 1 Introduction

One of the most successful (and natural) applications of parameterized complexity is the study of combinatorially hard problems for the case that one seeks a small solution. Such a problem is *fixed-parameter tractable* (FPT) if it can be checked in time  $f(k)n^c$  whether an instance of size  $n$  has a solution of size at most (or at least)  $k$ . When  $k$  is not too large, such an algorithm can be considered efficient. This can be especially important for minimization problems where the solution size corresponds to a real-world cost.

Curiously, for any decidable problem, having an FPT algorithm is known to coincide with having a polynomial-time data reduction algorithm that reduces any instance to an equivalent instance with size bounded by some function of  $k$ , a so-called *kernelization*.<sup>1</sup> However, the kernel size bound implied is the same function  $f(k)$  as occurs in the running time bound, which for non-trivial parameters will almost certainly be exponential in  $k$  unless  $P = NP$ .

A more useful notion of efficient data reduction is *polynomial kernels*, i.e., kernelizations with kernel size bounded polynomially in the parameter. For many problems, this can be achieved by a direct study of kernelization, e.g., the classic reduction of VERTEX COVER to  $2k$  vertices [50, 16], or the recent reduction of FEEDBACK VERTEX SET to size  $\mathcal{O}(k^2)$  by Thomassé [57], improving on work by Burrage et al. [13] and Bodlaender [7]. Having small (polynomial) kernels provides a formalization of efficient data reduction, and additionally, producing them often requires significant insight into the combinatorial structure of a problem.

<sup>\*</sup>Supported by the Netherlands Organization for Scientific Research (NWO), project “KERNELS: Combinatorial Analysis of Data Reduction”.

<sup>†</sup>Utrecht University, Utrecht, the Netherlands

<sup>‡</sup>Max-Planck-Institute for Informatics, Saarbrücken, Germany

<sup>1</sup>Running the algorithm for  $n^{c+1}$  steps will either solve the instance, or allow the conclusion that  $f(k)n^c > n^{c+1}$  and consequently that  $n$  is bounded by  $f(k)$ . The converse, a kernelization implying an FPT algorithm for any decidable problem, can be easily verified, although here the bound gets worse.

Accordingly, the search for more and better kernelizations has evolved into a main branch of parameterized complexity (in fact, the opinion has been raised that kernelization is what fixed-parameter tractability is really about [22]). In particular, the existence of a polynomial kernelization for a problem is seen as a significant threshold, comparable to the existence of an FPT algorithm in the first place. Recent seminal work of Bodlaender et al. [8] and Fortnow and Santhanam [27] enforced the importance of this threshold by providing techniques to show that certain problems do not admit polynomial kernels unless  $\text{NP} \subseteq \text{coNP/poly}$  and the polynomial hierarchy collapses to its third level (see also Harnik and Naor [35] for a related question). Furthermore, a paper by Dell and van Melkebeek [19] was the first work to provide lower bounds for the *degree* of a polynomial kernelization; among other things, their work implies an  $\mathcal{O}(k^2)$  lower bound for FEEDBACK VERTEX SET and ODD CYCLE TRANSVERSAL. Another recent focus has been *meta kernelizations*, i.e., meta-level results that provide kernelizations for a large range of problems, under restrictions on the input [9, 26]; see below.

Still, for all this work, some problems have so far resisted classification with respect to existence of polynomial kernels. Among these, emerging as the most important and most frequently raised questions – e.g., the two problems singled out as having the highest importance at the recent workshop on kernelization, WorKer 2010 – is the existence of polynomial kernelizations for the problems ODD CYCLE TRANSVERSAL (OCT) and DIRECTED FEEDBACK VERTEX SET (DFVS). Both problems are also open even in the restricted case of planar graphs [10]. In this paper, we focus on OCT, where the question was first raised in [33]; see also the recent survey on lower bounds for kernelization by Misra et al. [48].

**The Odd Cycle Transversal problem.** The ODD CYCLE TRANSVERSAL problem asks whether a given graph  $G$  can be made bipartite by deleting at most  $k$  of its vertices. Together with natural variants such as EDGE BIPARTIZATION, the edge deletion version, and BALANCED SUBGRAPH, the problem of removing odd-parity cycles in signed graphs, this problem has numerous applications (see, e.g., [37, 38]), and has received significant research attention [55, 33, 24, 31, 37, 44, 38, 40]. With respect to parameterized and exact computation the breakthrough result was the  $\mathcal{O}(4^k kmn)$  time algorithm by Reed, Smith, and Vetta [55].<sup>2</sup> This was the first occurrence of the tech-

nique now called *iterative compression*. (Note that this is entirely distinct from the notion of *instance compression*, which has been used as a term to generalize kernelization; see related work below.) This technique also led to the first FPT algorithm for DIRECTED FEEDBACK VERTEX SET by Chen et al. [18] and ALMOST 2-SAT by Razgon and O’Sullivan [54], and has become an important tool in parameterized complexity; see the survey by Guo et al. [34]. For EDGE BIPARTIZATION, an  $\mathcal{O}(2^k m^2)$  time algorithm exists due to Guo et al. [33]. The best known approximation result for OCT is  $\mathcal{O}(\sqrt{\log n})$  due to Agarwal et al. [1], improving on earlier results with a ratio of  $\mathcal{O}(\log n)$  [29]. For EDGE BIPARTIZATION there is also an  $\mathcal{O}(\log \text{OPT})$ -approximation by Avidor and Langberg [4]. Under the unique games conjecture, neither problem can have a constant-factor approximation [41].

**What makes OCT special?** The OCT problem belongs to the class of graph modification problems, i.e., finding a minimum number of modifications of vertices and/or edges in a given graph to achieve a given property (like bipartiteness). If the target property is sparse (e.g., being a forest for FEEDBACK VERTEX SET), or if it can be defined by a constant number of forbidden structures (e.g., induced paths on three vertices for CLUSTER EDITING), then many such problems are known to have a polynomial kernel, although the kernelization is by no means always easy. On the other hand, only few polynomial kernels are known for target properties that do not have either characteristic. One candidate is deleting at most  $k$  arcs to remove all directed cycles from a tournament<sup>3</sup>, but here it suffices to consider the directed triangles [5]. Also, CHORDAL COMPLETION has a polynomial kernel [49], but here, a large obstacle is helpful in that a chordless cycle of length  $t$  requires at least  $t - 2$  edges to be added. No such useful exception exists for OCT.

Additionally, there are involved meta-results for graph problems restricted to sparse inputs like planar, bounded genus, and  $H$ -minor free graphs [9, 26]. However, it can be easily seen that OCT is neither *compact* nor *quasi-compact* ([9]), as YES-instances do not have bounded treewidth (take an arbitrarily large grid graph and add a few edges that cause odd cycles); similarly it is not *minor bidimensional* ([26]) as the cost on a grid is zero. Thus, even for planar inputs, it is not covered by any known meta-result. Generally, similar statements as the above can be made about DFVS.

**Our work.** In this paper, we give a randomized polynomial kernelization for OCT, for unrestricted in-

<sup>2</sup>Hüffner [36] observed that the analysis could be improved to yield  $\mathcal{O}(3^k kmn)$ .

<sup>3</sup>A directed graph is a tournament if for every pair of its vertices, exactly one of the two possible arcs is present.

put. The kernel takes the form of a *compression* of the instance into a polynomial-sized matrix (with bounded entry lengths), such that the independent sets of columns in the matrix reveal whether the instance is positive or negative. By the NP-completeness of OCT, this then implies a randomized polynomial kernel. The result is produced by combining the iterative compression step of Reed, Smith, and Vetta [55] (in a suitable variant) with the theory of linearly represented matroids, specifically the class called gammoids [53]. We observe that, given a graph  $G$  and a set of terminals  $X$ , a single gammoid can be used to produce a matroid that encodes the flow from  $S$  to  $T$  in  $G - R$  for arbitrary  $S, T, R \subseteq X$ , and show (using results of Marx [46]) how to produce a matrix representing this matroid, of total size cubic in  $|X|$  and logarithmic in  $|G|$ , in randomized polynomial time. Having access to this information is sufficient to simulate the algorithm of [55]. Here,  $X$  is an initial approximate solution to the problem. To get this initial set  $X$ , we use the  $\mathcal{O}(\sqrt{\log n})$ -approximation of Agarwal et al. [1] to produce an  $\mathcal{O}(\sqrt{\text{OPT}})$ -approximation, giving an  $\mathcal{O}(k^{4.5})$ -size randomized polynomial compression and completing the kernelization. This approach of applying matroid theory to kernelization is a new tool in the field, and should prove useful for several future kernelization results as well. This is also one of very few randomized polynomial kernelizations (we are aware only of Harnik and Naor’s probabilistic compression for SUBSET SUM [35]). Our result also implies an  $\tilde{\mathcal{O}}(k^3)$  compression for the TANGLEGRAM LAYOUT problem from computational biology [23] via the reduction given in [6]; a polynomial kernel for this problem was left open in [6].

**Related work.** Generally, not much is known yet about excluding polynomial kernels for graph modification problems, compared to the wide range of problems that belong to this class. Although a few kernelization lower-bounds [20, 42, 19, 32] and FPT infeasibility results [43] exist, for many problems in this class, including ODD CYCLE TRANSVERSAL and DIRECTED FEEDBACK VERTEX SET, the question of polynomial kernels is open. Another related set of problems is graph separation problems; here as well, there are many FPT problems where the existence of polynomial kernels is unknown, such as MULTIWAY CUT [45, 17] and MULTICUT [12, 47].

Kernels for OCT for *non-standard* parameters are studied by Jansen and Kratsch [39]; a polynomial kernel is obtained for the case that one is given an OCT instance  $(G, k)$  as well as a set  $X$  such that  $G - X$  is both bipartite and bounded treewidth, with parameter  $|X|$ . The paper also contains related lower bounds (e.g.,

if  $G - X$  is treewidth 2 but not necessarily bipartite).

Harnik and Naor [35] raised the question of compression of NP instances with respect to the witness size (e.g., size  $\mathcal{O}(k \log n)$  for specifying a subset of size  $k$ ). As they note, polynomial kernelization with the witness size as parameter is equivalent to their notion of deterministic compression. See Fortnow and Santhanam [27] for further discussion comparing the approaches (and note also that the factor of  $\log n$  can be absorbed into  $k$  if a problem is FPT with running time  $2^{k^{\mathcal{O}(1)}} n^c$ ). Both Harnik and Naor [35] and Fortnow and Santhanam [27] also give notions of probabilistic compression; the one we use is closer to [27], though we restrict ourselves to one-sided error (see Section 2). However, as there are almost no further examples of randomized polynomial kernelization or compression, there will be plenty of time for settling notation later. In related work, compressions are instead called *bikernels* [3] or *generalized kernelizations* [8].

Connections between parameterized complexity and matroid theory have previously been studied by Marx [46], including a self-contained description of representation tools and issues for matroids. For more on matroid theory and algorithmic aspects see Oxley [52] as well as Schrijver [56].

## 2 Preliminaries

**2.1 Parameterized complexity and kernelization.** We use the following standard notation from parameterized complexity, for more background on this area we refer the reader to [21, 25, 51]. A *parameterized problem* over alphabet  $\Sigma$  is a language  $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ ; the second component of instances  $(x, k) \in \Sigma^* \times \mathbb{N}$  is called the parameter. A *kernelization* (or *kernel*) of  $\mathcal{Q}$  is a polynomial-time computable mapping  $K : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \times \mathbb{N} : (x, k) \mapsto (x', k')$  such that  $(x, k) \in \mathcal{Q}$  if and only if  $(x', k') \in \mathcal{Q}$  and with  $|x'|, k' \leq h(k)$  where  $h$  is a computable function;  $h$  is called the *size* of the kernel. A kernelization is a *polynomial kernelization* if the size  $h(k)$  is polynomially bounded.

We use the term (*parameterized*) *compression of  $\mathcal{Q}$  (into  $\mathcal{Q}'$ )* to denote the relaxed variant where  $K$  is allowed to map to a different language  $\mathcal{Q}'$  (also called *bikernel* [3] or *generalized kernelization* [8]). When  $\mathcal{Q}'$  is in NP and  $\mathcal{Q}$  with parameter coded in unary is NP-complete then, using the implicit Karp reduction, a polynomial compression of  $\mathcal{Q}$  into  $\mathcal{Q}'$  implies a polynomial kernelization for  $\mathcal{Q}$  [11].

We define a natural randomized version of kernelization with one-sided error, corresponding to the complexity class coRP (variants for RP and BPP could be defined similarly). Our notion of polynomial coRP-compression is essentially equivalent to that of prob-

abilistic compression in [27], except for our one-sided error, and that [27] defines the parameter to be given in unary.

**DEFINITION 2.1. (coRP-KERNELIZATION)** *Let  $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ . A randomized polynomial-time algorithm  $K$  with inputs and outputs in  $\Sigma^* \times \mathbb{N}$  is a randomized kernelization without false negatives, or coRP-kernelization, for  $\mathcal{Q}$  if there is a computable function  $h : \mathbb{N} \rightarrow \mathbb{N}$  such that for all  $(x, k) \in \Sigma^* \times \mathbb{N}$ :*

1. *if  $(x, k) \in \mathcal{Q}$  then  $\text{prob}[K((x, k)) \in \mathcal{Q}] = 1$ ,*
2. *if  $(x, k) \notin \mathcal{Q}$  then  $\text{prob}[K((x, k)) \notin \mathcal{Q}] \geq \frac{1}{2}$ , and*
3. *the size of  $x'$  and the value of  $k'$  are bounded by  $h(k)$ , where  $(x', k') := K((x, k))$ .*

*The notions of coRP-compression and polynomial coRP-compression are defined in the natural way.*

Note that unlike algorithms for BPP, RP, or coRP we cannot use majority, disjunction, or conjunction over the outputs of  $N$  independent runs to boost the success probability, since kernelizations and compressions typically do not solve instances. Harnik and Naor [35] observed that a similar effect may be attained by making a combined instance from the result of  $N$  independent runs of the compression (e.g., in our setting, creating an output which is to be interpreted as  $((x_1, k_1) \in \mathcal{Q}) \wedge \dots \wedge ((x_t, k_t) \in \mathcal{Q})$ ). Strictly speaking, this approach gives a compression, but it can be turned into a kernelization by the argument via the Karp reduction.

**2.2 Matroids.** Matroids are interesting combinatorial structures, generalizing the notion of independence from linear algebra, while also drawing from graph theory. There is an extensive theory of matroids, as well as several important algorithmic results; see e.g. [52, 56].

A *matroid* is a pair  $M = (E, \mathcal{I})$ , where  $E$  is the *ground set* and  $\mathcal{I} \subseteq 2^E$  a collection of *independent sets*, such that: (i)  $\emptyset \in \mathcal{I}$ ; (ii) if  $I_1 \subseteq I_2$  and  $I_2 \in \mathcal{I}$ , then  $I_1 \in \mathcal{I}$ ; and (iii) if  $I_1, I_2 \in \mathcal{I}$  and  $|I_2| > |I_1|$ , then there exists some  $x \in (I_2 \setminus I_1)$  such that  $I_1 \cup \{x\} \in \mathcal{I}$ . A set  $I \subseteq E$  is *independent* if  $I \in \mathcal{I}$ , and *dependent* otherwise. A set  $B \in \mathcal{I}$  is a *basis* of  $M$  if no superset of  $B$  is independent; a matroid may equivalently be defined by its set of bases (among other variants). Let  $\mathcal{B}$  be the set of bases of  $M$ , and  $\mathcal{B}^* = \{E \setminus B : B \in \mathcal{B}\}$ . Then  $\mathcal{B}^*$  is the set of bases of a matroid  $M^*$ , called the *dual* of  $M$ . Note that  $(M^*)^* = M$ .

Let  $A$  be a matrix over a field  $\mathbb{F}$  and  $E$  be the set of columns of  $A$ . Let  $\mathcal{I}$  be the set of all sets  $X \subseteq E$  of columns that are linearly independent over  $\mathbb{F}$  (as vectors). Then  $(E, \mathcal{I})$  defines a matroid  $M$ , and we say

that  $A$  *represents*  $M$ . A matroid is *representable* (over a field  $\mathbb{F}$ ) if there is a matrix (over  $\mathbb{F}$ ) that represents it. A matroid representable over some field is called *linear*. In this work, we will concern ourselves only with linear matroids. From a representation of  $M$ , one can easily get a representation of  $M^*$  over the same field.

Finally, we define *minors* of a matroid. For a matroid  $M = (E, \mathcal{I})$  and a set  $T \subseteq E$ , *deleting*  $T$  results in a matroid  $M \setminus T = (E \setminus T, \mathcal{I}')$  where  $\mathcal{I}' = \{I \in \mathcal{I} : I \subseteq E \setminus T\}$ . *Contracting*  $T$  results in a matroid  $M/T = (M^* \setminus T)^*$ ; if  $T \in \mathcal{I}$ , then the independent sets of  $M/T$  are the sets  $X \subseteq E \setminus T$  such that  $X \cup T \in \mathcal{I}$ . A *minor* of a matroid  $M$  is any matroid produced from  $M$  by deletions and contractions. Both operations can be performed with preserved representation.

### 3 Polynomial encoding of terminal cuts using gammoids

The basic situation that is handled in this section is the following. Let  $D = (V, A)$  be a directed graph, and let  $X \subseteq V$  be a set of terminals. We want to reduce the graph to a size polynomial in  $|X|$  and  $\log |V|$ , while preserving the size of a minimum vertex cut  $(S, T)$  for all sets  $S, T \subseteq X$ . Here, a vertex cut is understood as being allowed to delete vertices of  $S$  or  $T$  as well as other vertices of  $V$ ; thus the min-cut sizes are bounded by  $|X|$ . As an extension, we will also require that we may specify any set  $R \subseteq X$  as removed, i.e., we want to have also the cuts  $(S, T)$  in  $D - R$ .

Clearly, this question is closely connected to the search for polynomial kernels for FPT cut problems. However, a direct combinatorial reduction to achieve this, e.g., via edge contractions and vertex deletions or other direct simplifications on the graph, seems difficult. It is not even clear whether there always exists a graph of the required size, where every minimum  $(S, T)$ -cut for  $S, T \subseteq X$  has the same cardinality as in  $D - R$ . Instead, we here solve the question by introducing the use of matroids and matroid representations to the field of kernelization.

Let us recall a few helpful definitions. For  $S, T \subseteq V$ , the set  $T$  is *linked to*  $S$  if there exist  $|T|$  vertex-disjoint paths from  $S$  to  $T$ , where also the end points of the paths must be disjoint. The sets  $S$  and  $T$  do not need to be disjoint; a vertex is linked to itself by a path of length zero. By the cut-flow duality, it is clear that being able to find the linked subsets of  $X$  will suffice to answer all questions about cuts  $(S, T)$  in  $D$ . Perfect [53] showed, given any  $D = (V, A)$  and  $S \subseteq V$ , that the subsets of  $T$  which are linked to  $S$  in  $D$  form a matroid  $(D, S)$ , of a class now called *gammoids* (see [52, 56]). Marx [46] gave a randomized polynomial-time procedure for finding a representation of this matroid.

**THEOREM 3.1.** ([53, 46]) *Let  $D = (V, A)$  be a directed graph, and let  $S \subseteq V$ . The subsets  $T \subseteq V$  which are linked to  $S$  form the independent sets of a matroid over  $V$ . Furthermore, a representation of this matroid can be obtained in randomized polynomial time with one-sided error.*

Here, one-sided error means that dependent (non-linked) sets are preserved, but independent (linked) sets in the graph may not be, i.e., if the procedure returns a matrix  $A$ , then there may be some subsets of  $T$  which are linked to  $S$  but which are not independent in the matroid represented by  $A$ . However, the risk of this can be made arbitrarily small.

It remains for us only to bound the bit-length of the entries of the matrix (which would otherwise be polynomial in  $|V|$ ). This is easily done by standard methods.

**COROLLARY 3.1.** *Let  $D = (V, A)$  be a directed graph,  $\epsilon > 0$  a given real, and let  $S$  and  $T$  be possibly overlapping subsets of  $V$ . Let  $M$  be the gammoid formed by subsets of  $T$  linked to  $S$ . A representation of  $M$  as an  $|S| \times |T|$  matrix over the rationals with entries of bit-length  $\mathcal{O}(\min(|T|, |S| \log |T|) + \log(1/\epsilon) + \log |V|)$  can be computed in randomized polynomial time with one-sided error at most  $\epsilon$ .*

*Proof.* Theorem 3.1 can be made to return an  $|S| \times |V|$  matrix over the rationals, with arbitrarily small one-sided error  $\epsilon' > 0$  and individual entries being integers of bit-length polynomial in  $|V|$  and  $\log 1/\epsilon'$ . Let  $\epsilon' = \epsilon/2$ , and let  $A$  be the matrix returned, with columns not in  $T$  removed. To reduce the length of the entries, we take all entries in  $A$  modulo a sufficiently large random prime  $p$ . We argue that the matrix  $A'$  produced this way satisfies all conditions.

Consider an independent column set in  $A$ . Since independence corresponds to a square submatrix with non-zero determinant, we see that  $A$  and  $A'$  differ in this aspect only if  $p$  divides said determinant. The number of distinct prime factors in a number is bounded by the bit-length, which for our determinants is polynomially bounded in  $|V| + \log 1/\epsilon$ . Since the number of maximal independent sets in  $M$  is bounded by both  $|T|^{|S|}$  and by  $2^{|T|}$ , the total number of distinct primes is bounded as  $t = \min(|T|^{|S|}, 2^{|T|})(|V| + \log 1/\epsilon)^{\mathcal{O}(1)}$ . Thus, if we pick a random prime from a set of size at least  $t' = (2/\epsilon) \cdot t$ , the total risk of failure is bounded by  $\epsilon$ . By the Prime Number Theorem, primes of bit length  $\mathcal{O}(\log(t' \log t')) = \mathcal{O}(\log t + \log 1/\epsilon)$  are sufficient for this, which matches the statement of the corollary. By the AKS primality testing algorithm [2], finding a random prime can be done with high probability by repeated uniform sampling, and if this fails we may pick

an arbitrary fixed prime. It can be seen that errors throughout are one-sided.

The following proposition extends the available gammoid structure to allow any subset  $S$  of the terminals as sources (i.e., without fixing it in advance), and to support also deletion of terminals. This will be our interface for using Corollary 3.1 in the ODD CYCLE TRANSVERSAL kernelization. The argument is straightforward and works as well for a given directed graph.

**PROPOSITION 3.1.** *Let  $G = (V, E)$  be an undirected graph, let  $X \subseteq V$  be a set of terminals, and let  $X' := \{v' \mid v \in X\}$  be a set of new vertices. There is a polynomial-time construction of a directed graph  $D = (V \cup X', A)$  such that  $I \subseteq X \cup X'$  is an independent set of the gammoid  $(D, X')$  if and only if  $T$  is linked to  $S$  in  $G - R$  where*

- $S$  contains all vertices  $v \in X$  with  $v, v' \notin I$ ,
- $T$  contains all vertices  $v \in X$  with  $v, v' \in I$ , and
- $R$  contains all vertices  $v \in X$  with  $v \in I$  but  $v' \notin I$ .

*Proof.* The arc set  $A$  of the digraph  $D$  is defined as follows: For any two adjacent vertices  $u, v \in V$  add  $(u, v)$  and  $(v, u)$  to  $A$ . Then add an arc  $(v', v)$  for all  $v \in X$  (and corresponding  $v' \in X'$ ).

We consider first any independent set  $I \subseteq X \cup X'$  of the gammoid  $(D, X')$ . There are  $|I|$  vertex-disjoint directed paths from  $X'$  to  $I$  in  $D$ ; fix any such packing  $\mathcal{P}$  of directed paths. By the structure of  $D$  all paths of  $\mathcal{P}$  are either of form  $(u')$  with  $u' \in X'$ , or of form  $(u', u, \dots, v)$  (possibly with  $u = v$ ) and containing no vertices of  $X' \setminus \{u'\}$ . Let  $P = (u', u, \dots, v) \in \mathcal{P}$  with  $u' \in X'$  and  $v \in T$ , i.e., with  $v, v' \in I$ . Clearly,  $u' \neq v'$  since  $(v')$  is the unique directed path from  $X'$  to  $v'$  and must be contained in  $\mathcal{P}$  as  $v' \in I$  (and using vertex-disjointness). Since  $u$  and  $u'$  are on  $P$ , no other path of  $\mathcal{P}$  can end in  $u$  or  $u'$ , thus  $u, u' \notin I$  and hence  $u \in S$ . Finally, no vertex  $p \in R$  can be on  $\mathcal{P}$  since, by  $p \in I$ , that requires another path of  $\mathcal{P}$  to end in  $p$ . Now, the subpath  $(u, \dots, v)$  contained in  $D - X'$  corresponds to an undirected path from  $S$  to  $T$  in  $G - R$ , and all those paths are vertex-disjoint.

Now, let  $\mathcal{P}$  be a set of  $|T|$  vertex-disjoint paths from  $S$  to  $T$  in  $G - R$ . We construct a set  $\mathcal{P}'$  of directed vertex-disjoint paths from  $X'$  to  $I$  in the digraph  $D$  where  $I$  is obtained according to the statement of the proposition. For each path  $(u, \dots, v) \in \mathcal{P}$  we add the path  $(u', u, \dots, v)$  to  $\mathcal{P}'$ . Clearly, those paths exist in  $D$  and they are vertex-disjoint. We also require paths ending in the vertices of  $I \setminus T$ . This includes vertices  $r \in R$  and vertices  $v'$  with  $v \notin S \cup R$ . It is easy

to see that adding paths  $(r', r)$  and  $(v')$ , respectively, for all those vertices yields the required path packing  $\mathcal{P}'$  (key fact: in the initial  $|T|$  paths we only used  $v'$  when  $v \in S$ , and vertices  $r \in R$  were unused). Thus  $I$  is an independent set of the gammoid  $(D, X')$ .

Since it appears a very useful form, e.g., for obtaining polynomial kernels for other cut problems, we explicitly state the combination of Proposition 3.1 and Corollary 3.1 as a corollary.

**COROLLARY 3.2.** *Let  $G = (V, E)$  be an undirected graph,  $X \subseteq V$  a set of terminals, and  $\epsilon$  a positive real. There is a randomized polynomial-time algorithm computing an  $|X| \times 2|X|$  matrix with integer entries of bit length  $\mathcal{O}(|X| + \log |V| + \log 1/\epsilon)$ , such that with probability at least  $(1 - \epsilon)$  any set of columns  $I \subseteq X \cup X'$  is independent in  $M$  if  $T$  is linked to  $S$  in  $G - R$ , where  $S, T, R \subseteq X$  are defined as in Proposition 3.1. The error is one-sided: the number of disjoint paths as indicated by independence is a lower bound on the true value in  $G$ .*

#### 4 A randomized polynomial kernel for odd cycle transversal

In this section, using the results presented in the previous section, we will give our randomized polynomial kernelization for ODD CYCLE TRANSVERSAL. We will start by a presentation of the FPT algorithm of Reed, Smith, and Vetta [55], as understanding this algorithm is critical to understanding the kernelization. This is presented in Section 4.1. Then we present the kernelization in Section 4.2, and finally discuss the relation with lower bounds in Section 4.3.

**4.1 The Reed-Smith-Vetta algorithm.** The FPT algorithm of Reed, Smith, and Vetta [55] solves ODD CYCLE TRANSVERSAL by a recursive approach: Solve the problem for  $G - v$ , where  $v$  is an arbitrary vertex. If it returns a solution  $X_v$  of size at most  $k$ , then  $X := X_v \cup \{v\}$  is a solution of size at most  $k + 1$  for  $G$ , and the following *compression* version of the problem is solved. Otherwise  $(G - v, k)$  is NO, thus  $(G, k)$  must be NO.

**Input:** A graph  $G = (V, E)$ , an integer  $k$ , and a bipartization set  $X$  of size  $k + 1$ .

**Parameter:**  $k$ .

**Question:** Is there a bipartization set  $Y$  for  $G$  such that  $|Y| \leq k$ ?

The compression routine in the Reed-Smith-Vetta algorithm consists of trying exhaustively all ways of how the set  $X$  could interact with a smaller solution  $Y$ , each coming down to a maximum flow computation.

Concretely, we create a graph  $G' = (V', E')$  from  $G$  and  $X$  in the following way: let  $S_1 \cup S_2$  be a bipartition of  $G - X$ . Let  $V' = V - X \cup \{x_1, x_2 : x \in X\}$ , where  $x_1$  and  $x_2$  are new vertices. Connect  $x_1$  to all neighbors of  $x$  in  $S_2$  and  $x_2$  to all neighbors of  $x$  in  $S_1$ . By subdividing edges, we may assume that there are no edges inside  $X$ . Note that  $G'$  is bipartite with partitions  $S_1 \cup \{x_1 : x \in X\}$  and  $S_2 \cup \{x_2 : x \in X\}$ . For  $U \subseteq X$ , let  $X'(U) := \{x_1, x_2 : x \in U\}$ , and let  $X' = X'(X)$ .

The algorithm searches for cuts through  $X'$  in  $G'$ . For a subset  $U \subseteq X$ , let a pair  $(S, T)$  of disjoint subsets of  $X'$  be a *valid split* of  $U$  if for every  $x \in U$  we have  $|\{x_1, x_2\} \cap S| = |\{x_1, x_2\} \cap T| = 1$  and for every  $x \in (X \setminus U)$  we have  $|\{x_1, x_2\} \cap S| = |\{x_1, x_2\} \cap T| = 0$ . The following lemma is a direct consequence of [55].

**LEMMA 4.1.** ([55]) *Let  $G = (V, E)$  be a graph and let  $X \subseteq V$  such that  $G - X$  is bipartite. Let  $G'$  be constructed from  $G$  and  $X$  as above. Let  $\delta(H, S, T)$  denote the minimum size of an  $(S, T)$  vertex cut in  $H$ . The minimum size of  $Y \subseteq V$  such that  $G - Y$  is bipartite equals the minimum of  $|X \setminus U| + \delta(G' - X'(X \setminus U), S, T)$  over all subsets  $U$  of  $X$  and all valid splits  $(S, T)$  of  $U$ .*

Clearly, given this result, one can find an optimal bipartization set for  $G$  by looping over the  $3^{|X|}$  options for  $U$ ,  $S$ , and  $T$ . In particular, one is not limited to using  $X$  of size  $k + 1$  but, sacrificing runtime, a single run of the iterative compression routine on  $G$  and an approximate solution  $X$  for  $G$  suffices. In the next subsection, this setting will be used to give a polynomial kernelization of ODD CYCLE TRANSVERSAL.

For proofs of Lemma 4.1, see [55] or one of the presentations subsequently given by other authors [37, 44]. For variation, we will now sketch an alternative approach to showing the result.

Instead of a graph, we view the OCT instance as a 2-SAT formula  $F$  containing only constraints  $(x = y)$  and  $(x \neq y)$ . Clearly, for any graph  $G$ , if we replace every edge  $\{u, v\}$  in  $G$  by a constraint  $(u \neq v)$ , then we get a formula  $F$  over  $V$  which is satisfiable if and only if  $G$  is bipartite, and this holds for every induced subgraph of  $G$  as well. Thus, the problem reduces to deleting  $k$  variables  $Z$  of  $F$  and their incident constraints such that the remaining formula  $F - Z$  is satisfiable.

Now, observe that we can *negate* a variable  $v$  in  $F$  by changing  $(u = v)$ -constraints to  $(u \neq v)$ -constraints and vice versa. This does not affect the satisfiability of  $F - Y$  for any variable set  $Y$ . Thus, negate variables in  $F$  so that  $F - X$  is satisfied by the all-zero assignment. We now observe that the only remaining disequality constraints  $(x \neq y)$  of  $F$  are incident to  $X$ . By deleting or assigning every  $x \in X$ , we create smaller

formulas  $F'$  containing only equality constraints ( $u = v$ ) and assignments ( $u = 0$ ) or ( $u = 1$ ). This trivially reduces to a vertex cut problem in a graph, which would conclude the proof of the FPT result.

To get from here to the construction of  $G'$  above, consider the effects of splitting variables  $x \in X$  in  $F$  into distinct variables  $x$  and  $\neg x$  representing its literals, and replace constraints ( $x \neq y$ ) by ( $\neg x = y$ ). Clearly, in any assignment we must require ( $x \neq \neg x$ ). Applying all this to the bipartition  $S_1 \cup S_2$  of  $G - X$  will show the equivalence of the result.

**4.2 Kernelization.** Now, we give the kernelization. We begin by describing a compression procedure for OCT, by applying the matroid tools of Section 3 to Lemma 4.1 above. The result is a randomized polynomial-time compression procedure with one-sided error, consisting of the following steps. Let an instance  $(G, k)$  of OCT and an error parameter  $\epsilon > 0$  be given.

1. If  $k \leq \log n$ , run the Reed-Smith-Vetta algorithm in time  $\mathcal{O}(3^k kmn) = n^{\mathcal{O}(1)}$  (polynomial in  $n$ ) and return a constant-size YES- or NO-instance accordingly.
2. Otherwise, if  $k > \log n$ , let  $X$  be an approximate solution of ratio  $\mathcal{O}(\sqrt{\log n}) = \mathcal{O}(k^{1/2})$ , provided by an algorithm due to Agarwal et al. [1]. Unless  $|X| = \mathcal{O}(k^{3/2})$ , answer NO as there cannot be a solution of size at most  $k$ . (If  $|X| \leq k$  then answer YES.)
3. Create the auxiliary graph  $G'$  from  $G$  and  $X$ , as in Section 4.1. Let  $X' = \{x_1, x_2 \mid x \in X\}$ .
4. Apply Corollary 3.2 to  $G'$  with terminal set  $X'$  and error parameter  $\epsilon/2$ , creating a matrix  $A$ .
5. Output  $(A, k)$  as a polynomial-sized compression.

The total coding size of the matrix  $A$  is cubic in  $|X|$  (up to factors logarithmic in  $1/\epsilon$ ). By the size guarantee in Step 2, we get an  $\mathcal{O}(k^{4.5})$ -sized compression for OCT. Note that  $(A, k)$  may be interpreted as an instance of an (artificial) decision problem (implicitly defined in the proof of Lemma 4.2). Clearly, this problem is in NP, allowing us to reduce back to OCT to complete the kernelization (using the implicit Karp reduction as discussed in Section 2.1).

For EDGE BIPARTIZATION, we may replace Steps 1 and 2 by the  $\mathcal{O}(\log \text{OPT})$ -approximation of Avidor and Langberg [4], followed by an easy reduction from EDGE BIPARTIZATION to OCT, for a compression of size  $\tilde{\mathcal{O}}(k^3)$ . It is an interesting question whether a polylog(OPT)-approximation is possible for OCT, as this would give us a  $\tilde{\mathcal{O}}(k^3)$ -sized compression for OCT.

Now we give our central compression result.

**LEMMA 4.2.** *Let  $(G, k)$  be an instance of OCT,  $X$  a bipartition set for  $G$ , and  $\epsilon > 0$  be given. Then there is a randomized compression of  $(G, k)$  to size  $\mathcal{O}(|X|^2(|X| + \log 1/\epsilon))$  with one-sided error, producing no false negatives. The error probability is bounded by  $\epsilon$ , and the running time is polynomial in  $|G|$  and  $\log 1/\epsilon$ .*

*Proof.* The algorithm proceeds as Steps 3-5 of the kernelization algorithm, creating first an auxiliary graph  $G'$  with a terminal set  $X'$  of size  $2|X|$ , and then invoking Corollary 3.2 on  $G'$ ,  $X'$ , and  $\epsilon/2$ . Let the resulting matrix be  $A$ ; our compression output is then  $(A, k)$ . The running time and output size are given by Corollary 3.2; we only have to argue that  $(A, k)$  contains all the information needed to decide the status of the OCT instance  $(G, k)$ . By Step 2, we assume  $|X| > k$ .

Recall the definition  $X'(U) = \{x_1, x_2 : x \in U\}$  for  $U \subseteq X$ . By Lemma 4.1, we need for all  $U \subseteq X$  the minimum  $(S, T)$  vertex cut size in  $G' - X'(X \setminus U)$ , where  $S$  and  $T$  range over all valid splits of  $U$ . Clearly, if the minimum  $(S, T)$  vertex cut size in  $G' - R$  is  $\lambda$  then there is a set  $T' \subseteq T$  of size  $\lambda$  such that  $T'$  is linked to  $S$  in  $G' - R$ , i.e., such that there are  $\lambda = |T'|$  vertex disjoint paths from  $S$  to  $T'$  in  $G' - R$  (using cut-flow duality). This can be obtained from the matrix  $A$  by testing independence of all sets  $I$  which correspond (as in Corollary 3.2) to choices  $S, T', R$  with  $T' \subseteq T$  and  $R = X'(X \setminus U)$ . Note that whether an  $(S, T)$ -cut may delete  $S$  and  $T$  makes no difference for the algorithm using Lemma 4.1.

The behavior and one-sidedness of the error also follows from Corollary 3.2.

With the previously described algorithm and the approximation results we get the following.

**THEOREM 4.1.** *There is a randomized  $\mathcal{O}(k^{4.5})$ -compression for ODD CYCLE TRANSVERSAL and a randomized  $\tilde{\mathcal{O}}(k^3)$ -compression for EDGE BIPARTIZATION, with one-sided error with no false negatives and failure probability exponentially small in  $k$ .*

The target problems of the compressions are constrained minimization problems over the rank of a matroid. We omit the precise definitions, but it should be clear that the problems can be made well-defined, and that they are in NP. As discussed in Section 2.1, using NP-completeness of ODD CYCLE TRANSVERSAL and EDGE BIPARTIZATION, we get the following polynomial coRP-kernelization results.

**COROLLARY 4.1.** *ODD CYCLE TRANSVERSAL, EDGE BIPARTIZATION, BALANCED SUBGRAPH, and TANGLEGRAM LAYOUT (see [6]) have polynomial coRP-kernelizations.*

**4.3 A Note on Lower Bounds.** Finally, we remark that although kernelizations and kernelization lower bounds are usually expressed in terms of deterministic results, the type of randomized kernelizations we produce here (i.e., one-sided error with no false negatives) does in fact fit within the lower bounds framework of Bodlaender et al. [8] and Fortnow and Santhanam [27], as the proofs implicitly exclude also co-nondeterministic kernels. Since our coRP kernelization is a special case of this, our tools cannot be used to escape the lower bounds. Dell and van Melkebeek [19] noted the connection to co-nondeterministic kernels, and brought it further by giving lower bounds in terms of the amount of communication needed to solve a problem in an oracle setting, where a polynomially bounded but co-nondeterministic player communicates with an computationally unbounded oracle.<sup>4</sup> They provide concrete lower bounds for various problems, among others implying the following.

**THEOREM 4.2.** ([19]) *Let  $\epsilon > 0$ . No co-nondeterministic kernel or compression for OCT can achieve a total size of  $\mathcal{O}(k^{2-\epsilon})$  unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$  and the polynomial hierarchy collapses.*

It seems difficult to go below an  $\mathcal{O}(k^3)$  bound using our methods. Thus, we leave it as an open question whether the upper or the lower bound on the total compression size can be improved for ODD CYCLE TRANSVERSAL and EDGE BIPARTIZATION.

## 5 Conclusion

We have presented randomized polynomial kernelizations for ODD CYCLE TRANSVERSAL and EDGE BIPARTIZATION. The key contribution is the introduction of matroids into kernelization, by encoding the compression step of the Reed-Smith-Vetta algorithm for ODD CYCLE TRANSVERSAL [55], by means of a matroid. This leads to a compression of the problem into size  $\mathcal{O}(k^{4.5})$  for ODD CYCLE TRANSVERSAL and  $\tilde{\mathcal{O}}(k^3)$  for EDGE BIPARTIZATION, which is easily turned into a kernelization by back-reductions to the original problems. The kernelization has one-sided error, producing no false negatives, and the failure rate can be made exponentially small in  $k$  at only a constant factor cost to the size. While this essentially settles the question about existence of polynomial kernels, the more practical result seems to be the output of the compression. Not only is compression to any set the more robust notion (cf. [19]); the target problem is native in one of the most well-studied areas of mathematics and computer

science. The compression may also point the way for where to look for direct, combinatorial kernelizations for the problem.

It is interesting that the results of Fortnow and Santhanam [27] can be seen to exclude also co-nondeterministic compressions [19]. Thus, our technique is not a way of avoiding the lower bounds given by [8, 27], but a way of settling problems for which neither such lower bounds nor a polynomial kernelization or compression are known.

We close with some open problems. It is still interesting whether there exist deterministic polynomial kernelizations for ODD CYCLE TRANSVERSAL and EDGE BIPARTIZATION, either as a derandomization of our methods, or (which would have independent interest) as a properly combinatorial kernelization. Additionally, for both problems, there is the question of the correct size bound. We note that a  $\mathcal{O}(\text{polylog}(\text{OPT}))$ -approximation for OCT, which is consistent with approximation theory lower bounds, would improve our result to  $\tilde{\mathcal{O}}(k^3)$ , but this still leaves a gap to the  $\mathcal{O}(k^{2-\epsilon})$  lower bound given by Dell and van Melkebeek [19]. Finally, existence of (randomized) polynomial kernels is an exciting question for several related problems, including DIRECTED FEEDBACK VERTEX SET, MULTIWAY CUT, and EDGE and VERTEX MULTICUT. The robust way in which the introduction of matroids into kernelization helped to settle the question for ODD CYCLE TRANSVERSAL gives reason to believe that it will play a key role for some of these problems as well.

## References

- [1] A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev.  $\mathcal{O}(\sqrt{\log n})$  approximation algorithms for Min UnCut, Min 2CNF deletion, and directed cut problems. In H. N. Gabow and R. Fagin, editors, *STOC*, pages 573–581. ACM, 2005.
- [2] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Ann. of Math*, 2:781–793, 2002.
- [3] N. Alon, G. Gutin, E. Kim, S. Szeider, and A. Yeo. Solving MAX- $r$ -SAT above a tight lower bound. *Algorithmica*, pages 1–18, 2010.
- [4] A. Avidor and M. Langberg. The multi-multiway cut problem. *Theor. Comput. Sci.*, 377(1-3):35–42, 2007.
- [5] S. Bessy, F. V. Fomin, S. Gaspers, C. Paul, A. Perez, S. Saurabh, and S. Thomassé. Kernels for feedback arc set in tournaments. In R. Kannan and K. N. Kumar, editors, *FSTTCS*, volume 4 of *LIPICs*, pages 37–47. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2009.
- [6] S. Böcker, F. Hüffner, A. Truß, and M. Wahlström. A faster fixed-parameter approach to drawing binary tanglegrams. In Chen and Fomin [15], pages 38–49.

<sup>4</sup>Harnik and Naor [35] credit an extension of [27] to any one-sided error to unpublished work of Chen and Müller.

- [7] H. L. Bodlaender. A cubic kernel for feedback vertex set. In W. Thomas and P. Weil, editors, *STACS*, volume 4393 of *Lecture Notes in Computer Science*, pages 320–331. Springer, 2007.
- [8] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
- [9] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Peninkx, S. Saurabh, and D. M. Thilikos. (Meta) kernelization. In *FOCS*, pages 629–638. IEEE Computer Society, 2009.
- [10] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Peninkx, S. Saurabh, and D. M. Thilikos. (meta) kernelization. *CoRR*, abs/0904.0727, 2009.
- [11] H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. In A. Fiat and P. Sanders, editors, *ESA*, volume 5757 of *Lecture Notes in Computer Science*, pages 635–646. Springer, 2009.
- [12] N. Bousquet, J. Daligault, and S. Thomassé. Multicut is FPT. In Fortnow and Vadhan [28], pages 459–468.
- [13] K. Burrage, V. Estivill-Castro, M. R. Fellows, M. A. Langston, S. Mac, and F. A. Rosamond. The undirected feedback vertex set problem has a poly( $k$ ) kernel. In H. L. Bodlaender and M. A. Langston, editors, *IWPEC*, volume 4169 of *Lecture Notes in Computer Science*, pages 192–202. Springer, 2006.
- [14] M. Charikar, editor. *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*. SIAM, 2010.
- [15] J. Chen and F. V. Fomin, editors. *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, volume 5917 of *Lecture Notes in Computer Science*. Springer, 2009.
- [16] J. Chen, I. A. Kanj, and W. Jia. Vertex cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.
- [17] J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.
- [18] J. Chen, Y. Liu, S. Lu, B. O’Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.
- [19] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In L. J. Schulman, editor, *STOC*, pages 251–260. ACM, 2010.
- [20] M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. E. Nikolettseas, and W. Thomas, editors, *ICALP (1)*, volume 5555 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2009.
- [21] R. G. Downey and M. R. Fellows. *Parameterized Complexity (Monographs in Computer Science)*. Springer, November 1998.
- [22] V. Estivill-Castro, M. R. Fellows, M. A. Langston, and F. A. Rosamond. FPT is P-time extremal structure I. In H. Broersma, M. Johnson, and S. Szeider, editors, *ACiD*, volume 4 of *Texts in Algorithmics*, pages 1–41. King’s College, London, 2005.
- [23] H. Fernau, M. Kaufmann, and M. Poths. Comparing trees via crossing minimization. *J. Comput. Syst. Sci.*, 76(7):593–608, 2010.
- [24] S. Fiorini, N. Hardy, B. A. Reed, and A. Vetta. Planar graph bipartization in linear time. *Discrete Applied Mathematics*, 156(7):1175–1180, 2008.
- [25] J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer, March 2006.
- [26] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In Charikar [14], pages 503–510.
- [27] L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.
- [28] L. Fortnow and S. P. Vadhan, editors. *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*. ACM, 2011.
- [29] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. Comput.*, 25(2):235–251, 1996.
- [30] M. Grohe and R. Niedermeier, editors. *Parameterized and Exact Computation, Third International Workshop, IWPEC 2008, Victoria, Canada, May 14-16, 2008. Proceedings*, volume 5018 of *Lecture Notes in Computer Science*. Springer, 2008.
- [31] S. Guillemot. FPT algorithms for path-transversals and cycle-transversals problems in graphs. In Grohe and Niedermeier [30], pages 129–140.
- [32] S. Guillemot, C. Paul, and A. Perez. On the (non-)existence of polynomial kernels for  $P_t$ -free edge modification problems. In V. Raman and S. Saurabh, editors, *IPEC*, volume 6478 of *Lecture Notes in Computer Science*, pages 147–157. Springer, 2010.
- [33] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *J. Comput. Syst. Sci.*, 72(8):1386–1396, 2006.
- [34] J. Guo, H. Moser, and R. Niedermeier. Iterative compression for exactly solving NP-hard minimization problems. In J. Lerner, D. Wagner, and K. A. Zweig, editors, *Algorithmics of Large and Complex Networks*, volume 5515 of *Lecture Notes in Computer Science*, pages 65–80. Springer, 2009.
- [35] D. Harnik and M. Naor. On the compressibility of NP instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.
- [36] F. Hüffner. Algorithm engineering for optimal graph bipartization. In S. E. Nikolettseas, editor, *WEA*, volume 3503 of *Lecture Notes in Computer Science*, pages 240–252. Springer, 2005.
- [37] F. Hüffner. Algorithm engineering for optimal graph

- bipartization. *J. Graph Algorithms Appl.*, 13(2):77–98, 2009.
- [38] F. Hüffner, N. Betzler, and R. Niedermeier. Separator-based data reduction for signed graph balancing. *J. Comb. Optim.*, 20(4):335–360, 2010.
- [39] B. Jansen and S. Kratsch. Vertex and edge bipartization with nonstandard parameters: Upper and lower bounds for kernelization, 2011. Unpublished manuscript.
- [40] K. Kawarabayashi and B. A. Reed. An (almost) linear time algorithm for odd cycles transversal. In Charikar [14], pages 365–378.
- [41] S. Khot. On the power of unique 2-prover 1-round games. In *IEEE Conference on Computational Complexity*, page 25, 2002.
- [42] S. Kratsch and M. Wahlström. Two edge modification problems without polynomial kernels. In Chen and Fomin [15], pages 264–275.
- [43] D. Lokshtanov. Wheel-free deletion is  $W[2]$ -hard. In Grohe and Niedermeier [30], pages 141–147.
- [44] D. Lokshtanov, S. Saurabh, and S. Sikdar. Simpler parameterized algorithm for OCT. In J. Fiala, J. Kratochvíl, and M. Miller, editors, *IWOCA*, volume 5874 of *Lecture Notes in Computer Science*, pages 380–384. Springer, 2009.
- [45] D. Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.
- [46] D. Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009.
- [47] D. Marx and I. Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In Fortnow and Vadhan [28], pages 469–478.
- [48] N. Misra, V. Raman, and S. Saurabh. Lower bounds on kernelization. *Discrete Optimization*, 8(1):110–128, 2011.
- [49] A. Natanzon, R. Shamir, and R. Sharan. A polynomial approximation algorithm for the minimum fill-in problem. *SIAM J. Comput.*, 30(4):1067–1079, 2000.
- [50] G. Nemhauser and L. Trotter. Vertex packing: structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.
- [51] R. Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*. Oxford University Press, USA, March 2006.
- [52] J. Oxley. *Matroid Theory*. Oxford University Press, 2nd edition, 2011.
- [53] H. Perfect. Applications of menger’s graph theorem. *J. Math. Anal. Appl.*, 22:96–111, 1968.
- [54] I. Razgon and B. O’Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.
- [55] B. A. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.
- [56] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.
- [57] S. Thomassé. A  $4k^2$  kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.