Boundaried Kernelization via Representative Sets

Leonid Antipov
Humboldt-Universität zu Berlin, Germany

Stefan Kratsch
Humboldt-Universität zu Berlin, Germany

Abstract

A kernelization is an efficient algorithm that given an instance of a parameterized problem returns an equivalent instance of size bounded by some function of the input parameter value. It is quite well understood which problems do or (conditionally) do not admit a kernelization where this size bound is polynomial, a so-called polynomial kernelization. Unfortunately, such polynomial kernelizations are known only in fairly restrictive settings where a small parameter value corresponds to a strong restriction on the global structure on the instance. Motivated by this, Antipov and Kratsch [WG 2025] proposed a local variant of kernelization, called boundaried kernelization, that requires only local structure to achieve a local improvement of the instance, which is in the spirit of protrusion replacement used in meta-kernelization [Bodlaender et al. JACM 2016]. They obtain polynomial boundaried kernelizations as well as (unconditional) lower bounds for several well-studied problems in kernelization.

In this work, we leverage the matroid-based techniques of Kratsch and Wahlström [JACM 2020] to obtain randomized polynomial boundaried kernelizations for s-MULTIWAY CUT, DELETABLE TERMINAL MULTIWAY CUT, ODD CYCLE TRANSVERSAL, and VERTEX COVER[oct], for which randomized polynomial kernelizations in the usual sense were known before. A priori, these techniques rely on the global connectivity of the graph to identify reducible (irrelevant) vertices. Nevertheless, the separation of the local part by its boundary turns out to be sufficient for a local application of these methods.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases Parameterized complexity, boundaried kernelization, local preprocessing, representative sets method

Digital Object Identifier 10.4230/LIPIcs.IPEC.2025.6

Related Version *Full Version*: `https://arxiv.org/abs/2510.00832` [2]

Funding *Leonid Antipov*: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 526215872.

1 Introduction

(Polynomial) kernelization is a very successful notion for rigorously studying efficient preprocessing for hard problems. A kernelization is an efficient algorithm that given an instance of a parameterized problem returns an equivalent instance of size bounded by some function of the input parameter value. It is a polynomial kernelization if this function is polynomially bounded. By now, it is quite well understood which problems do or (conditionally) do not admit a polynomial kernelization. In this way, we have learned what structural properties are helpful for a provable size reduction through efficient preprocessing, which is unlikely in general unless $\mathsf{P} = \mathsf{NP}$. Unfortunately, most polynomial kernelizations are for parameters that take low values only on instances with very restricted *global structure*, e.g., only parameter many vertex deletions away from a known tractable special case of the problem. This is an issue because the size guarantee of a kernelization is only nontrivial when the parameter is small, otherwise we could just leave the instance as is.

20th International Symposium on Parameterized and Exact Computation (IPEC 2025).
Editors: Akanksha Agrawal and Erik Jan van Leeuwen; Article No. 6; pp. 6:1–6:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Motivated by this, Antipov and Kratsch [1] proposed a local variant of kernelization, called *boundaried kernelization*, inspired by protrusion replacement in meta-kernelization (Bodlaender et al. [3]). Roughly, this expects as input a boundaried graph G_B and will return a boundaried graph G'_B of size some function of the chosen parameter plus the boundary size (and possibly a shift Δ in solution value). The two graphs are equivalent in the sense that gluing any boundaried graph H_B will result in equivalent instances (up to shift of Δ for optimization). Intuitively, such a preprocessing only needs a local part of a large graph to have beneficial structure and sufficiently small connection to the rest of the input, for the size bound to imply a size reduction. In this sense, boundaried kernelization relaxes the required structural conditions (from global to local), though of course also the size bound is only local (namely only for the boundaried local part). Since we can always run a boundaried kernelization with empty boundary, however, it is by itself in fact a strengthening of kernelization (modulo some technical aspects due to the variety of different parameterizations). Thus, it is natural and interesting to ask, which known polynomial kernelizations can be strengthened to polynomial boundaried kernelizations.

In [1], polynomial boundaried kernelizations were obtained for VERTEX COVER[vc], VERTEX COVER[fvs], FEEDBACK VERTEX SET[fvs], LONG CYCLE[vc], LONG PATH[vc], HAMILTONIAN CYCLE[vc], and HAMILTONIAN PATH[vc].[1] In contrast, CLUSTER EDITING[cvd], CLUSTER EDITING[ce], TREE DELETION SET[vc], TREE DELETION SET[tds], and DOMINATING SET[vc] were shown to (unconditionally) not admit a polynomial boundaried kernelization.[2] Existence of unconditional lower bounds is not surprising and relies on exhibiting an unbounded (or at least too large) family of non-equivalent boundaried graphs. That being said, it is somewhat surprising that simple local reduction rules can sometimes be leveraged also for boundaried kernelization, while they otherwise fail completely.

There are of course plenty of other graph problems with polynomial kernelizations to explore. Next to well-studied problems (as above) and cases with somewhat special kernelization (as TREE DELETION SET[tds]), it would be interesting to know how inherently global tools fare in the boundaried setting. Here, the matroid-based techniques of Kratsch and Wahlström [13] come to mind, as they rely on the properties of certain matroids defined on the entire graph, while also enabling the first (and so far only) polynomial kernelizations for certain cut and feedback problems. Another interesting one would be the randomized polynomial compression for STEINER CYCLE[$|T|$] due to Wahlström [18] but likely one would first have to strengthen it to a kernelization, as it outputs a somewhat contrived matrix problem not known to be in NP.

Our work. Motivated by the question of whether the global matroid-based techniques of Kratsch and Wahlström [13] can also be adapted for local preprocessing, we study the same problems in the boundaried setting. We are able to strengthen several results to polynomial boundaried kernelizations.

▶ **Theorem 1.** *The following parameterized problems admit randomized polynomial boundaried kernelization: s-*MULTIWAY CUT[local solution] *(with a minor restriction described below),* DELETABLE TERMINAL MULTIWAY CUT[local solution]*,* ODD CYCLE TRANSVERSAL[local solution]*, and* VERTEX COVER[oct]*.*

---

[1] Parameterized problems are denoted as problem name followed by parameter in brackets: These include parameterization by the size of a given vertex cover (vc), feedback vertex set (fvs), cluster vertex deletion set (cvd), cluster editing set (ce), and tree deletion set (tds), respectively.

[2] It was known that DOMINATING SET[vc] has no polynomial kernelization unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, unlike all other problems listed here, but exclusion of polynomial boundaried kernelization is unconditional.

The main idea lies in the power of the gammoids and the resulting cut covers Z which contain the vertices of an exponential number of min-cut queries. While Kratsch and Wahlström applied this on gammoids with sources being the terminal set, or an approximate odd cycle transversal, we additionally put in the boundary vertices as sources. As a result, this family of min-cut queries whose optimal answers are all covered by Z, contain in particular all possible combinations needed in order to complete an optimal global solution for any forced behavior of the boundary vertices in the solution. For instance, in s-MULTIWAY CUT, where the solution is a vertex set X such that each of the s given terminals $t \in T$ is contained in its own component in $G - X$, we are able to preserve an optimal solution for every choice of putting any boundary vertex either into the solution, the component of some terminal, or some unrelated component. Or in ODD CYCLE TRANSVERSAL, where $G - X$ has to be a bipartite graph, the set Z contains the local part of an optimal solution for every choice of putting B vertices in the solution or one of the two parts of the bipartite graph $G - X$.

Note however for the problem s-MULTIWAY CUT, that, for G_B being the local graph, T_G a set of terminals among G-vertices, and H_B, T_H the rest of the global graph with the remaining terminals among the H-vertices, if we allow T_H to also contain B-vertices, this forces us to be able to adapt to any pair of the boundary vertices to be terminals, and thus we need to store at least some information about the disjoint path sets between any such pair. Since the sizes of these path sets are not bounded, this means that we need to be able to store an unbounded amount of information, which directly rules out any effective preprocessing.

▶ **Theorem 2.** *In general, the parameterized problem s-*MULTIWAY CUT[local solution] *does not admit any boundaried kernelization.*

So, the slightly restricted case that we are referring to in Theorem 1 is that we forbid the terminal-part T_H that is unknown for the kernelization, to contain any boundary vertices. This restriction is enough in order to obtain a boundaried kernelization, and it fits with the concept of local kernelization because the terminals in B are known in that setting. (That being said, in general, it is appealing to push to the most general setting for getting a polynomial boundaried kernelization. This includes having the preprocessing be fully independent of possible glued graphs H_B.)

Related work. Most closely related is the already mentioned work on meta-kernelization via protrusion replacement (subgraphs with constant treewidth and constant boundary size). Inherently, the machinery for that does not lead to any polynomial size bounds, and that is not to be expected. Generally, the perspective of boundaried graphs is essential for dynamic programming on path and tree decompositions (cf. [4]). In that setting, there is no hope for polynomial boundaried kernelizations because it would generalize the (likely) infeasible case of problems parameterized by path/treewidth regarding polynomial kernelization. Instead, our settings have more restrictive structure than small treewidth (as is usual for polynomial kernelization) and we obtain polynomial size bounds.

Fomin et al. [6] use a different form of protrusion for (CONNECTED) DOMINATING SET[k] where they require constant boundary size and a constant local solution size. It is used very differently than our (parameter) local cost, but is certainly reminiscent. Clearly, a small or constant local cost is a natural quality of interest amongst more explicit structural parameters like the size of modulators.

Work by Jansen and Kratsch [10] for preprocessing the feasibility problem of integer linear programs showed how to shrink subsystems with constant boundary size and either constant treewidth or total unimodularity to size polynomial in the domain. This predates the present notion of boundaried kernelization, like protrusion-replacement, but similarly does not yield size polynomial in the boundary size.

Organization. We give preliminaries regarding graph problems and boundaried kernelization in Section 2 and restate the tools from representative sets for matroids in Section 3. In Sections 4, 5, 6, and 7 we describe the randomized polynomial boundaried kernelizations for s-MULTIWAY CUT[local solution], DELETABLE TERMINAL MULTIWAY CUT[local solution], ODD CYCLE TRANSVERSAL[local solution], and VERTEX COVER[oct], respectively. Finally, we conclude in Section 8. Throughout the paper, we denote results by $\star$, if the corresponding proof is omitted and to be found in the full version [2].

2 Preliminaries

Graphs and graph problems. We use the usual notion of decision and optimization problems as well as standard graph theoretic notation mostly following Diestel [5]. Our graphs are finite, simple, undirected, loopless, and unweighted, unless explicitly stated otherwise. A *mixed* graph contains both directed and undirected edges. For a vertex set $V' \subseteq V(G)$, we denote by $N_{G,V'}(v)$ the set of neighbors of v that are contained in V', i.e., $N_{G,V'}(v) := N_G(v) \cap V'$. For vertex set $W \subseteq V(G)$, we extend this notion to $N_{G,V'}(W)$.

A *(vertex) annotated graph* is a tuple $(G, \mathcal{T})$, where G is an undirected graph and $\mathcal{T}$ itself is a tuple of subsets of $V(G)$. A *decision or optimization problem on (annotated) graphs* consists of instances encoding (annotated) graphs, respectively. A problem on graphs, i.e., without annotation $\mathcal{T}$, is also called a *pure graph problem*. We define the pure graph optimization problems VERTEX COVER and ODD CYCLE TRANSVERSAL in the usual way.

A set $S \subseteq V(G)$ is called a *(multiway) cut* for a graph G and a set $T = \{t_1, \ldots, t_k\} \subseteq V(G)$ of given *terminals* if in graph $G' = G - S$ for every pair $u, v \in T$ there is no path between u and v in $G - S$. More generally, if we are given graph G and a tuple of terminal sets $\mathcal{T} = (T_1, \ldots, T_k)$, then a cut for $(G, \mathcal{T})$ is a set $S \subseteq V(G)$ such that for any $i \neq j \in [k]$ and any $u \in T_i, v \in T_j$ there is no path between u and v in $G - S$. We remark that unless stated otherwise, a cut S might intersect the set T of terminals, respectively the set $\bigcup_{T \in \mathcal{T}} T$ in the general case. A set $S \subseteq V(G)$ is said to be *closest* to some set $T \subseteq V$ in G if S is the unique (T, S)-min cut in G, and hence, there exist $|S|$-many vertex-disjoint paths from T to S. A T*-closest cut* between T and S is a (T, S)-min cut that is closest to T. Kratsch and Wahlström [13] describe a simple efficient algorithm for computing a T-closest cut. In the minimization problem s-MULTIWAY CUT the value of a solution x for undirected graph G and set $T = \{t_1, \ldots, t_s\}$ of terminals, is equal to $|S|$ if x encodes a cut S for (G, T), which is furthermore disjoint from T, and $+\infty$ otherwise. In contrast to this, in the minimization problem DELETABLE TERMINAL MULTIWAY CUT we are given an arbitrarily large set T of terminals, but now x is allowed to encode a cut S for (G, T) which might intersect the terminal set T, in which case the value of x is also equal to $|S|$.

Parameterized complexity. A *parameterized problem* is a set $Q \subseteq \Sigma^* \times \mathbb{N}$. For a tuple $(x, \ell) \in \Sigma^* \times \mathbb{N}$ the number ℓ is called the *parameter*. For a problem Π (decision or optimization) and minimization problem ρ we define the *structurally parameterized* problem $\Pi[\rho]$, for which it holds that $(x, s, \ell) \in \Pi[\rho]$ if and only if s is a feasible solution of value at most ℓ for ρ on

x and: (i) for Π being a decision problem, $x \in \Pi$; (ii) for Π being a minimization problem, $x = (x', k)$ for some $k \in \mathbb{N}$ and $\mathrm{OPT}_\Pi(x') \leq k$; (iii) for Π being a maximization problem, $x = (x', k)$ and $\mathrm{OPT}_\Pi(x') \geq k$.

A *kernelization* for a parameterized problem Q is a polynomial-time algorithm $\mathcal{A}$ which is given as input an instance (x, ℓ) and outputs an *equivalent* instance (x', ℓ'), i.e., $(x, \ell) \in Q \Leftrightarrow (x', \ell') \in Q$, such that $|x'|$ and ℓ' are upper bounded by $f(\ell)$ for some computable function f. The function f is called the *size* of the kernelization $\mathcal{A}$, and if f is polynomially bounded, then $\mathcal{A}$ is called a *polynomial kernelization*. If there is a probability that the output instance (x', ℓ') is not equivalent to (x, ℓ), then $\mathcal{A}$ is called a *randomized kernelization*. In our cases, the error probability will be upper bounded by $\mathcal{O}(2^{-\Theta(|x|)})$. The output instance (x', ℓ') is called a *kernel* of (x, ℓ) and we say that the problem Q *admits a (randomized) (polynomial) kernel*, if there exists a (randomized) (polynomial) kernelization for Q.

Boundaried graphs and boundaried kernelization. A *boundaried graph* is a special kind of annotated graph $G_B = (G, B)$, where we call G the *underlying graph* and B the *boundary*. We define the earlier mentioned problems on boundaried graphs in the same way as on their underlying graphs, i.e., we ignore the boundary for these problems. For two boundaried graphs G_B, H_C we define the operation of *gluing* these graphs together, denoted by $G_B \oplus H_C$, which results in a new boundaried graph with boundary $B \cup C$ and consisting of the disjoint union of the vertex and edge sets of G and H, under the identification of the vertices in $B \cap C$. Throughout this paper we will tacitly assume that $V(G)$ and $V(H)$ intersect exactly at $B \cap C$, in which case the underlying graph of $G_B \oplus H_C$ can simply be seen as the union of each the vertex sets and the edge sets of G and H. Clearly, graph gluing is commutative and associative.

Based on regular graph gluing, we additionally define gluing of boundaried graphs with additional vertex annotations, which we will call *(vertex) annotated boundaried graphs*: For two such annotated boundaried graphs $(G_B, T_1, \ldots, T_r)$ and $(H_C, U_1, \ldots, U_r)$ for some $r \in \mathbb{N}$ and with $V(G) \cap V(H) \subseteq B \cap C$, the result of $(G_B, T_1, \ldots, T_r) \oplus (H_C, U_1, \ldots, U_r)$ is the annotated boundaried graph $(G_B \oplus H_C, W_1, \ldots, W_r)$ where $W_i = T_i \cup U_i$ for every $i \in [r]$.

We say that the instances $(G_B, \mathcal{T})$ and $(G'_B, \mathcal{T}')$ are *gluing equivalent* with respect to an optimization problem Π on annotated graphs and boundary B, denoted as $(G_B, \mathcal{T}) \equiv_{\Pi,B} (G'_B, \mathcal{T}')$, if there exists some constant $\Delta \in \mathbb{Z}$ such that for every boundaried graph H_B and tuple $\mathcal{U}$ of $V(H)$-subsets, it holds that $\mathrm{OPT}_\Pi((G_B, \mathcal{T}) \oplus (H_B, \mathcal{U})) = \mathrm{OPT}_\Pi((G'_B, \mathcal{T}') \oplus (H_B, \mathcal{U})) + \Delta$. If Π and B are clear from context, we will omit them and write $\equiv$ instead. The optimization problem Π has *finite integer index* if the number of equivalence classes of $\equiv_{\Pi,B}$ is upper bounded by some function $f(|B|)$. We refer to the full version of the paper for definitions of gluing equivalence and finite index for Π being a decision problem.

Let Π be an optimization or decision problem and ρ a minimization problem, both on annotated graphs. A *boundaried kernelization* for the parameterized problem $\Pi[\rho]$ is a polynomial-time algorithm $\mathcal{A}$ which is given as input an annotated boundaried graph $(G_B, \mathcal{T})$ and feasible solution s for ρ on $(G_B, \mathcal{T})$ together with the value ℓ of s; and $\mathcal{A}$ outputs an annotated boundaried graph $(G'_B, \mathcal{T}')$, such that $(G'_B, \mathcal{T}')$ is gluing equivalent to $(G_B, \mathcal{T})$ and the size of $(G'_B, \mathcal{T}')$ is bounded by $f(|B| + \ell)$ for some computable function f. Additionally, if Π is an optimization problem, then $\mathcal{A}$ also needs to output the offset Δ by which the optimum value for $(G_B, \mathcal{T}) \oplus (H_B, \mathcal{U})$ differs from that for $(G'_B, \mathcal{T}') \oplus (H_B, \mathcal{U})$ for any boundaried graph H_B and tuple $\mathcal{U}$ of $V(H)$-subsets. Analogously to (regular) kernelization, we define the size of $\mathcal{A}$ and the notions of a (randomized) (polynomial) boundaried kernelization and kernel. By minor adaptation of a result by Antipov and Kratsch [1, Lemma 4] and under the assumption that Π and ρ are NP-optimization problems, the existence of a (randomized) (polynomial) boundaried kernelization for parameterized problem $\Pi[\rho]$ also implies the existence of a (randomized) (polynomial) kernelization for $\Pi[\rho]$. Additionally we define a boundaried kernelization for Π[local solution] to be a boundaried kernelization for Π parameterized by itself, i.e., the input is an annotated boundaried graph $(G_B, \mathcal{T})$ and a feasible solution s for Π on $(G_B, \mathcal{T})$, together with the value ℓ of s.

We will construct boundaried kernelization by the use of reduction rules. We say that a reduction rule is *gluing safe*, if it gets as input an instance $(G_B, \mathcal{T})$ and outputs a gluing equivalent instance $(G'_B, \mathcal{T}')$ and the corresponding shift Δ in solution value, if Π is an optimization problem and $\Delta \neq 0$.

▶ **Lemma 3** ([1, Lemma 1]). *Let Π be a pure graph problem, G and G' graphs, and B, C vertex subsets of both $V(G)$ and $V(G')$ such that $B \subseteq C$. Then $G_C \equiv_{\Pi,C} G'_C$ implies $G_B \equiv_{\Pi,B} G'_B$, with the same offset Δ for these two equivalences, if Π is an optimization problem.*

3 Matroid tools for kernelization

In this section we recall some of the results of Kratsch and Wahlström [13] regarding the use of matroids for polynomial kernelization, as well as the corresponding definitions.

A *matroid* is a pair $M = (E, \mathcal{I})$ of a *ground set* E and a collection of *independent sets* $\mathcal{I} \subseteq 2^E$, which are subsets of E, such that: (i) $\emptyset \in \mathcal{I}$; (ii) if $I_1 \subseteq I_2$ and $I_2 \in \mathcal{I}$, then also $I_1 \in \mathcal{I}$; 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 $I_1 \cup \{x\} \in \mathcal{I}$. An independent set $B \in \mathcal{I}$ is called a *basis* of M if no superset of B is independent. One can also define matroid M by its set of bases. The *rank* $r(X)$ of a subset $X \subseteq E$ is the largest cardinality of an independent set $I \subseteq X$. The rank of M is $r(M) := r(E)$. Some matroid M is said to be *linear*, if there exists some $m \times n$ matrix over a field $\mathbb{F}$, such that $M = (E, \mathcal{I})$, where E is the set of columns of A, and $\mathcal{I}$ contains those subsets of E that are linearly independent over $\mathbb{F}$. In such a case we also say that A *represents* M, and A *is a representation of* M.

Let $D = (V, A)$ be a directed graph and let $S, T \subseteq V$. We say that T is *linked* to S in D if there exist $|T|$-many vertex-disjoint paths from S to T, also allowing paths of length zero. In particular, any set is linked to itself. Given a directed graph $D = (V, A)$, a set $S \subseteq V$ of source vertices, and a set $U \subseteq V$ of sink vertices, we define a special case of matroids, called *gammoid*, by the sets $T \subseteq U$ that are linked to S in D. We will only use a further special case thereof with $U = V$, called *strict gammoid*, and slightly abuse notation by simply calling these gammoids. Due to Perfect [15] and Marx [14], a representation of a (strict) gammoid can be computed in randomized polynomial time, with one-sided error that can be made exponentially small in the size of the graph.

Let $M = (V, \mathcal{I})$ be a matroid and let X be an independent set in M. A set $Y \subseteq V$ is said to *extend* X in M if it holds that $X \cap Y = \emptyset$ and $X \cup Y \in \mathcal{I}$. For a collection of V-subsets $\mathcal{Y} \subseteq 2^V$ we say that a subset $\mathcal{Y}' \subseteq \mathcal{Y}$ is r-representative for $\mathcal{Y}$ if for every set $X \subseteq V$ of size at most r, the existence of a set $Y \in \mathcal{Y}$ that extends X in M, implies the existence of a set $Y' \in \mathcal{Y}'$ that also extends X in M.

Next, we state a result by Marx [14] dubbed as the *representative sets lemma* by Kratsch and Wahlström, and follow that by results that build up on it.

▶ **Lemma 4** ([14, 13]). *Let M be a linear matroid of rank $r + s$, and let $\mathcal{Y}$ be a collection of independent sets of M, each of size s. There exists a set $\mathcal{Y}' \subseteq \mathcal{Y}$ of size at most $\binom{r+s}{s}$ that is r-representative for $\mathcal{Y}$. Furthermore, given a representation A of M, we can find such a set $\mathcal{Y}'$ in time $(m + ||A||)^{\mathcal{O}(s)}$, where $m = |\mathcal{Y}|$ and $||A||$ denotes the encoding size of A.*

▶ **Lemma 5** ([11, Theorem 5]). *Let M be a gammoid and let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be a collection of independent sets, each of size p. We can find in randomized polynomial time a set $\mathcal{A}' \subseteq \mathcal{A}$ of size at most $\binom{p+q}{p}$ that is q-representative for $\mathcal{A}$.*

▶ **Lemma 6** ([13, Theorem 1.2]). *Let $G = (V, E)$ be a directed graph and let $S, T \subseteq V$. Let r denote the size of a minimum (S, T)-vertex cut (which may intersect S and T). There exists a set $Z \subseteq V$ with $|Z| = \mathcal{O}(|S| \cdot |T| \cdot r)$, such that for any $A \subseteq S$ and $B \subseteq T$, it holds that Z contains a minimum (A, B)-vertex cut as a subset. We can find such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-|V|})$.*

▶ **Lemma 7** ([13, Corollary 1.5]). *Let $G = (V, E)$ be an undirected graph and $X \subseteq V$. Let s be a constant. Then there is a set $Z \subseteq V$ of $\mathcal{O}(|X|^{s+1})$ vertices such that for every partition $\mathcal{X} = (X_0, X_X, X_1, \ldots, X_s)$ of X into $s+2$ possibly empty parts, the set Z contains a minimum multiway cut of $(X_1, \ldots, X_s)$ in the graph $G - X_X$ as a subset. We can compute such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-|V|})$.*

Such a set Z, as described in the preceding two lemmas, is called a *cutset* for the pair of vertex sets (S, T) (Lemma 6), respectively for the vertex set X (Lemma 7).

The main operation for reducing our graphs will be that of *bypassing* a vertex v in a graph G (also called making vertex v *undeletable* in G). In this operation one removes the vertex v from G and adds shortcut edges between the neighbors of v. In other words, for all paths (u, v, w) in G, one adds the edge (u, w) or $\{u, w\}$ (depending on whether the graph is (un)directed or mixed) unless already present. Effectively, this operation forbids to take v into a cut, while preserving the separation achieved by any vertex cuts that avoid v. For more detail, see [13]. Under the term *bypassing a set* $W \subseteq V(G)$, we mean the repeated operation of bypassing vertices of W one after another, in any order. Observe that by repeated use, the following lemma also holds when bypassing a vertex set W, if $X \subseteq V \setminus W$.

▶ **Lemma 8** ([13, Proposition 2.3]). *Let $G = (V, E)$ be an undirected, directed, or mixed graph and let G' be the result of bypassing some vertex $v \in V$ in G. Then, for any set $X \subseteq V \setminus \{v\}$ and any vertices $s, t \in V \setminus (X \cup \{v\})$, there is a path from s to t in $G - X$ if and only if there is a path from s to t in $G' - X$.*

4 s-Multiway Cut[local solution]

As our first result, we construct a boundaried kernelization variant of the randomized polynomial kernel for the s-MULTIWAY CUT problem by Kratsch and Wahlström [13]. For this, we leverage the argument of Kratsch and Wahlström, that for any optimal solution X it holds that each neighbor of a terminal t is either contained in the solution or in the same connected component of $G - X$ as t. In the boundaried setting this is translated to the observation (for G_B, T_G being the given local graph and terminal set, H_B, T_H any possible "rest" of the global instance, and X an optimal solution) that additionally any boundary vertex $x \in B$ is contained either in the solution, shares the component of some terminal in $G_B \oplus H_B - X$, or is contained in a component without any terminals. Then, using Lemma 7, we can compute a set Z that contains the needed vertices for any multiway cut that corresponds to any combination of these restrictions, and bypass all vertices of $F \setminus Z$ with $F := V(G) \setminus B$. Furthermore, we substitute the size bound of $|N(T)|$ of Kratsch and Wahlström due to an LP argument by Guillemot [8], and instead we apply Reduction Rule 1.

▶ **Theorem 9** ($\star$). *In general, s-MULTIWAY CUT[local solution] does not admit a boundaried kernelization.*

▶ **Theorem 10.** *The parameterized problem s-*MULTIWAY CUT[local solution] *admits a randomized polynomial boundaried kernelization with $\mathcal{O}((|B|+\ell)^{s+1})$ vertices, with failure probability $\mathcal{O}(2^{-|V|})$, and under the restriction that externally glued instances do not contain B-vertices as terminals.*

As partly indicated above the input to the boundaried kernelization is a tuple (G_B, T_G, S, ℓ) such that $|T_G| \leq s$ and S is a multiway cut of size at most ℓ for T_G in G. In particular it follows that $B' := S \cup (B \setminus T_G)$ is also a multiway cut for T_G in G, i.e., every terminal $t \in T_G$ is contained in its own component C_t in $G - B'$.

▶ **Reduction Rule 1.** *For any $t \in T_G$, if $|N(t)| > |B'|$, then choose a minimum size cut X between $N(t)$ and B' (allowing terminal-deletion) which is closest to $N(t)$. Remove the edges incident with t, and make X the new neighborhood of t.*

▶ **Lemma 11** (⋆)**.** *Reduction Rule 1 is gluing-safe.*

▶ **Reduction Rule 2.** *Assume that Reduction Rule 1 cannot be applied. Let $Z \subseteq V(G)$ be a vertex set such that for every partition $\mathcal{X} = (X_0, X_X, X_1, \dots, X_s)$ of $X := B \cup \bigcup_{t \in T_G} N_G(t)$ into $s+2$ possibly empty parts, the set Z contains a minimum multiway cut of $(X_1, \dots, X_s)$ in the graph $G - X_X$. Bypass every vertex from $V(G) \setminus (Z \cup B \cup T_G \cup N_G(T_G))$.*

▶ **Lemma 12.** *Reduction Rule 2 is gluing-safe.*

Proof. Let G_B denote the graph before applying the reduction rule, and G'_B the result thereof. Let H_B be the graph for gluing. Let T_G be the given set of terminals for G (resp. for G'), and $T_H \subseteq V(H) \setminus B$ the set of terminals given together with H_B. Let $T = T_G \cup T_H$ and assume without loss of generality that $|T| = s$. If $|T| < s$, we can add isolated terminals to H, while $|T| > s$ directly leads to $(G_B \oplus H_B, T)$ having no feasible solution. Likewise, we can assume that no two terminals are adjacent. Let $s_G = |T_G|$ and assume some arbitrary ordering of the terminals, such that the vertices of T_G can be denoted by $t_1, \dots, t_{s_G}$ and those from T_H by $t_{s_G+1}, \dots, t_s$.

“$\mathrm{OPT}(G_B \oplus H_B, T) \leq \mathrm{OPT}(G'_B \oplus H_B, T)$”. Let Y' be a multiway cut for T in $G'_B \oplus H_B$. Since Y' does not contain any bypassed vertices, it follows from Lemma 8 that Y' is a multiway cut for T in $G_B \oplus H_B$.

“$\mathrm{OPT}(G_B \oplus H_B, T) \geq \mathrm{OPT}(G'_B \oplus H_B, T)$”. Let Y be a multiway cut for T in $G_B \oplus H_B$ without terminal deletion. Thus, no component in $(G_B \oplus H_B) - Y$ contains more than one terminal and, obviously, every terminal $t \in T_G$ is in the same component as $N_G(t) \setminus Y$. Furthermore, any vertex in $B \setminus Y$ shares its component with at most one terminal. Based on this we partition the set X into $(X_0, X_X, X_1, \dots, X_s)$ as follows: X_X contains those $B \setminus T$-vertices that are contained in Y and X_0 those that are not connected with any terminal in $(G_B \oplus H_B) - Y$; for each $t_i \in T$ put into X_i every $B \setminus Y$-vertex that is connected with t_i in $(G_B \oplus H_B) - Y$, and if further t_i is contained in T_G then also put all vertices from $N_G(t_i) \setminus Y$ into X_i.

Let $F = V(G) \setminus B$. Clearly, $Y \cap F$ is a multiway cut for $(X_1, \dots, X_s)$ in $G - X_X$. By choice of Z, construction of G' and Lemma 8, there is also a multiway cut $Y^* \subseteq Z$ of size at most $|Y \cap F|$ for $(X_1, \dots, X_s)$ in $G' - X_X$. We will now verify that $Y' := (Y \setminus F) \cup Y^*$ is a multiway cut for T in $G'_B \oplus H_B$ (without terminal deletion). For that, assume for contradiction that there is a path P between distinct terminals $t_{i_1}, t_{i_2} \in T$ in $(G'_B \oplus H_B) - Y'$. Note that no two terminals from T are connected in either $H - Y' = H - Y$ or $G' - Y'$. Thus, the path cannot go only through $H - B$ or $G' - B$ and contains vertices from $B \setminus (T \cup Y')$. Give these vertices an order along the path in the direction from t_{i_1} to t_{i_2}, i.e., $P = (t_{i_1}, \dots, x_1, \dots, x_2, \dots, x_p, \dots, t_{i_2})$, where $x_1, \dots, x_p$ are the $B \setminus (T \cup Y')$-vertices contained in P and the vertices between x_i, x_j with distinct $i, j \in [p]$ are contained in $(V(G) \cup V(H)) \setminus (B \cup T \cup Y')$. If the path between x_i and x_j goes through $H - (B \cup Y')$, then they need to be in the same part of the earlier partition $(X_0, X_X, X_1, \dots, X_s)$: Since it holds that $H - (B \cup Y') = H - (B \cup Y)$, the vertex x_i is connected to $t \in T$ in $(G_B \oplus H_B) - Y$ if and only if x_j is connected to t. However, if the path goes through $G' - (B \cup Y')$ and if, say, x_i is in the part X_h with $t_h \in T_H$, then it might also be the case that x_j is in the part X_0, since Y^* does not have to separate X_h-vertices from X_0-vertices. Furthermore, if terminal $t_h \in T$ is connected with x_i (either through $H - (B \cup Y')$ or $G' - (B \cup Y')$), then x_i needs to be contained in the part X_h. Combining these points we come to the fact that $x_1 \in X_{i_1}$ and $x_p \in X_{i_2}$, while at the same time it holds that $x_p \in X_{i_1} \cup X_0$ by the transversal from x_1 to x_p through $H - (B \cup Y')$ and $G' - (B \cup Y')$ between consecutive B-vertices x_i, x_{i+1} for every $i \in [p-1]$. A contradiction. ◀

As Reduction Rules 1 and 2 are clearly applicable (the former exhaustively) in polynomial time, it remains to find the cut cover Z for $B \cup \bigcup_{t \in T_G} N_G(t)$, for which we use Lemma 7. Altogether, this proves Theorem 10, with small failure chance due to the computation of Z.

5 Deletable Terminal Multiway Cut[local solution]

Quite closely related to the s-MULTIWAY CUT problem is DELETABLE TERMINAL MULTIWAY CUT, in which the number of input terminals is arbitrary, but one is also allowed to take them into the solution. This time, for their kernel, Kratsch and Wahlström [13] define an auxiliary graph G^*, which adds two sink-only copies v', v'' for each $v \in V(G)$ to graph G, i.e., $G^*[V(G)] = G$ and for each $v \in V(G)$ and each neighbor $u \in N_G(v)$, add the directed edges $(u, v'), (u, v'')$. They show that for any solution X for graph G and terminal set T, and any $x \in X$, there are $|X| + 2$ many vertex-disjoint paths from T to $X \cup \{x', x''\}$ in G^*. Using a slightly stronger formulation by Kratsch [11], we get a similar connectivity result between $B \cup T_G$ and $(X \cap F) \cup \{x', x''\}$ in G^*, this time with G_B being only the given local graph-part, T_G a subset of the terminals, and $F = V(G) \setminus B$. This allows us to leverage another argument of Kratsch and Wahlström, and show that for gammoid $\mathcal{M}$ on G^* with sources $B \cup T_G$, set $\mathcal{Z} = \{\{x, x', x''\} \mid x \in F\}$ and every $x \in (X \setminus T) \cap F$, it holds that $\{x, x', x''\}$ is contained in every $(2|B| + |T_G| - 1)$-representative set $\mathcal{Z}' \subseteq \mathcal{Z}$ on $\mathcal{M}$. Again, while Kratsch and Wahlström use an LP argument by Guillemot [8] in order to bound size of the terminal set by a linear factor on the sought solution size k, we go another route, as the LP argument does not work without knowing the whole instance. Namely, use argumentation due to Razgon [16] in order to remove isolated components and vertices that have very high connection to the terminal set. This will give a bound of $|T_G| \in \mathcal{O}((|B| + \ell)^2)$, where ℓ is the size of a given local DELETABLE TERMINAL MULTIWAY CUT solution S on (G, T_G).

▶ **Theorem 13.** *The parameterized problem* DELETABLE TERMINAL MULTIWAY CUT[local solution] *admits a randomized polynomial boundaried kernel with $\mathcal{O}((|B| + \ell)^6)$ vertices.*

▶ **Lemma 14** ($\star$)**.** *There exists a set of polynomial-time exhaustively applicable and gluing safe reduction rules, after which it holds that $|T_G| \in \mathcal{O}((|B| + \ell)^2)$.*

▶ **Reduction Rule 3.** *Let $\mathcal{M}$ be the gammoid for G^* with sources $B \cup T_G$ and let $\mathcal{Z} := \{\{v, v', v''\} \mid v \in F\}$. Further, let $\mathcal{Z}'$ be a $(2|B| + |T_G| - 1)$-representative subset of $\mathcal{Z}$. Then bypass every vertex $v \in F \setminus T_G$ for which $\{v, v', v''\}$ is not contained in $\mathcal{Z}'$.*

▶ **Lemma 15.** *Reduction Rule 3 is gluing safe.*

Proof. Let G' be the resulting graph. Fix H_B, T_H. Let $T := T_G \cup T_H$.

"$\mathrm{OPT}(G_B \oplus H_B, T) \leq \mathrm{OPT}(G'_B \oplus H_B, T)$". Let X' be a solution for $(G'_B \oplus H_B, T)$. Obviously, X' does not contain any bypassed vertices and thus by Lemma 8 it is also a solution for $(G_B \oplus H_B, T)$.

"$\mathrm{OPT}(G_B \oplus H_B, T) \geq \mathrm{OPT}(G'_B \oplus H_B, T)$". Let X be a solution for $Q := G_B \oplus H_B$ with terminal set T, such that $|X|$ is minimized but among such $|X \cap T|$ is maximized. Let $X_F := X \cap F$, $X_0 := X \setminus T$, and $T_0 := T \setminus X$. Further, define the mixed graph Q^* on the base of Q, but with two additional sink-only copies v', v'' for every vertex $v \in V(Q)$. Kratsch [11] has shown that in $Q^* - (T \cap X)$ there exists a path packing with $|X_0| + 2$ paths from T_0 to $X_0 \cup \{x', x''\}$ for each $x \in X_0$. Note that this implies existence of a path packing in $G^* - (T \cap X)$ with $|X_0 \cap F| + 2$ paths from $(B \cup T_G) \setminus X$ to $(X_0 \cap F) \cup \{x', x''\}$ for each $x \in X_0 \cap F$: Take the path packing from T_0 to $X_0 \cup \{x', x''\}$ in $Q^* - (T \cap X)$, throw away any path going to $X_0 \setminus F$, and note that the remaining paths are either totally contained in $G^*[F \cup \{x', x''\}]$ or must intersect $B \setminus X$. For each path of the latter case we fix its last vertex u contained in $B \setminus X$ and use the subpath from u to $(X_0 \cap F) \cup \{x', x''\}$ for the path packing.

Now we show for any fixed $x \in X_0 \cap F$ that $\{x, x', x''\}$ is contained in $\mathcal{Z}'$. Fix the path packing of size $|X_0 \cap F| + 2$ from $(B \cup T_G) \setminus X$ to $(X_0 \cap F) \cup \{x', x''\}$ in $G^* - (T \cap X)$ and let U be the set of $B \setminus X$-vertices from which the paths go to $(X_0 \cap F) \cup \{x', x''\}$ without touching B any further. Set $Y := B \setminus U$ and observe that there exists a path packing of $|X_0 \cap F| + |Y| + 2$ paths from $B \cup T_G$ to $(X_0 \cap F) \cup Y \cup \{x', x''\}$ (including the length-0 path (y) for each $y \in Y$) in G^*, i.e., $\{x, x', x''\}$ extends $X_x := ((X_0 \cap F) - \{x\}) \cup Y$ in $\mathcal{M}$. Let us make sure that $|X_x|$ is upper bounded by $2|B| + |T_G| - 1$, thus leading to the fact that $\{x, x', x''\}$ is a correct candidate for $\mathcal{Z}'$: If it were true that $|X_x| > 2|B| + |T_G| - 1$, it would follow thereof that $|X_F| > |B| + |T_G|$ and thus that $\tilde{X} := (X \setminus V(G)) \cup B \cup T_G$ is a multiway cut for Q of smaller size, contradicting minimality of $|X|$. For this last point we remark that a potential path between terminals in $Q - \tilde{X}$ would need to be a path between terminals from $T_H \setminus B$ and contained entirely inside of $H - X$, contradicting the fact that X is a multiway cut for (Q, T).

Having seen that $\{x, x', x''\}$ is a candidate for $\mathcal{Z}'$ as an extension of X_x, we show that there is no other vertex $v \in F$ which extends X_x. Assume for contradiction that such a vertex v exists. First for any $v \in X_0 \setminus \{x\}$ it holds that $\{v, v', v''\} \cap X_x = \{v\} \neq \emptyset$ and thus that $\{v, v', v''\}$ does not extend X_x. Hence we consider the case that $v \in F \setminus X$. We remark that by choice of U we have that $U \cap X = \emptyset$, and that each of its vertices $u \in U$ is reachable from some terminal $t \in T$ in $Q - X$, with no two U-vertices sharing the terminal. Since by assumption there exist three paths from $B \cup T_G$ to $\{v, v', v''\}$ and at most one of them goes through x, two of these paths go directly through $G - X_x$. In particular this means that these two paths go from $U \cup T_G$ to $\{v, v', v''\}$, implying that v is connected with two different terminals in $Q - X$, a contradiction to X being a multiway cut for (Q, T).

As a result, for any fixed $x \in X_0 \cap F$ it must hold that $\{x, x', x''\}$ is contained in $\mathcal{Z}'$ and thus no X-vertices were bypassed while constructing G'. By Lemma 8 this means that X is also a multiway cut for $(G'_B \oplus H_B, T)$. ◀

Now, Theorem 13 is proven by first exhaustively applying the reduction rules behind Lemma 14 in order to bound $|T_G|$ by $\mathcal{O}((|B| + \ell)^2)$; and afterwards applying Lemma 5 and Reduction Rule 3 in order to compute the representative set $\mathcal{Z}' \subseteq \mathcal{Z}$ of size $\mathcal{O}((|B| + |T_G|)^3)$ and bypass all vertices $v \in F \setminus T_G$ for which $\{v, v', v''\}$ is not contained in $\mathcal{Z}'$. Together this yields our wanted vertex size of $\mathcal{O}((|B| + \ell)^6)$.

6 Odd Cycle Transversal[local]

In this section, we work with the problem ODD CYCLE TRANSVERSAL, which is not directly defined through cuts, other than the previous two problems. However, Reed et al. [17] have used a connection between odd cycle transversals and cuts, and this connection is what allows us to use representative sets on gammoids. Basically, given some odd cycle transversal X for graph G, and defining auxiliary graph G^* on G with dedicated vertices X^* corresponding to X-vertices, an optimal solution for G can be deduced by computing for each partition $S \cup T \cup R$ of X^* a minimum (S, T)-cut in $G^* - R$.

Kratsch and Wahlström [12] used this connection in order to get a polynomial compression in the form of a matroid with $\mathcal{O}(|X|)$ elements and randomized encoding size of $\mathcal{O}(|X|^3)$, which includes information on the min-cuts for each partition $S \cup T \cup R$ of X^*. Later the same authors showed a randomized polynomial kernelization for the problem Γ-FEEDBACK VERTEX SET[k] [13], which includes ODD CYCLE TRANSVERSAL[k] as a special case. A written out kernel specifically for this problem can be found in [7].

▶ **Theorem 16.** *The parameterized problem* ODD CYCLE TRANSVERSAL[local solution] *admits a randomized polynomial boundaried kernel with $\mathcal{O}((|B| + \ell)^6)$ vertices.*

Let us be given boundaried graph G_B and odd cycle transversal S for G. By Lemma 3 and since the class of bipartite graphs is hereditary, we can simply assume that S is a subset of B and redefine $B := B \cup S$ otherwise. Let $F = V(G) \setminus B$ and note that by previous assumption it holds that $G[F] = G - B$ is a bipartite graph. Fix some arbitrary bi-coloring (L, R) of $G[F]$ and define an auxiliary graph G^* with vertices $B_L \cup B_R \cup F$, where $B_{\texttt{side}} := \{x_{\texttt{side}} \mid x \in B\}$ for $\texttt{side} \in \{L, R\}$. The graph G^* contains edges such that $G^*[F] = G[F]$ and for all $x \in B$ vertex x_L is adjacent to $N_G(x) \cap R$, while x_R is adjacent to $N_G(x) \cap L$. In addition, for any pair $y, x \in B$ that are adjacent in G, add the edges $\{y_L, x_R\}$ and $\{y_R, x_L\}$ to G^*.

▶ **Lemma 17** ($\star$)**.** *Let G_B and G^* be as above and let H_B be one more boundaried graph. The size of a minimum odd cycle transversal for $G_B \oplus H_B$ is equal to the minimum over the sum of the sizes of an odd cycle transversal X for H and of a minimum cut between $B_\sim := (B_L \cap K_L^*) \cup (B_R \cap K_R^*)$ and $B_\Delta := (B_L \cap K_R^*) \cup (B_R \cap K_L^*)$ in $G^* - \{x_L, x_R \mid x \in B \cap X\}$, where (K_L, K_R) is an arbitrary bi-coloring of $H - X$ and $K_{\texttt{side}}^* := \{x_L, x_R \mid x \in B \cap K_{\texttt{side}}\}$ for $\texttt{side} \in \{L, R\}$.*

▶ **Reduction Rule 4.** *Let G_B, G^* and the corresponding vertex sets be as above. Let Z be a cut cover for G^* and the set $B_L \cup B_R$. Then remove all vertices in $F \setminus Z$ in G.*

For any two vertices u, v in $B \cup (F \cap Z)$, if there was an odd-length path between u and v with all inner vertices in $F \setminus Z$, then add an edge between u and v; and if there was such an even-length path between u and v, then add a path of length two between u and v. It might be the case that we add both an edge and a length-2 path between u and v. Also, for any $x \in B$, if there was an odd path consisting only of x and vertices from $F \setminus Z$, then add a triangle with x one of its vertices.

▶ **Lemma 18.** *Reduction Rule 4 is gluing-safe.*

Proof. Let G' be the resulting graph. Observe that $G' - B$ is bipartite as we only added independent edges for the triangles, substituted odd-length paths by edges, and even-length paths by length-2 paths. Construct the graph $\tilde{G}$ from G' in the same way as G^* was constructed from G. The following claim is analogous to Lemma 8 and used for our proof.

▷ Claim 19 ($\star$). For any set $C \subseteq B_L \cup B_R \cup (F \cap Z)$ and any vertices $s, t \in B_L \cup B_R$, there is a path from s to t in $G^* - C$ if and only if there is a path from s to t in $\tilde{G} - C$.

We only show one direction of the proof here, as the other direction (to be found in the full version) works very similarly.

"$\mathbf{OPT}(G_B \oplus H_B) \leq \mathbf{OPT}(G'_B \oplus H_B)$". Let S' be an odd cycle transversal for $G'_B \oplus H_B$. By Lemma 17 we can divide this solution into an odd cycle transversal X' for H and a minimum size cut C' between $B_\sim$ and B_Δ in $\tilde{G} - \{x_L, x_R \mid x \in B \cap X'\}$. We can assume without loss of generality that none of the vertices created for the substituting paths and cycles are contained in C'. Hence C' only contains vertices from $B_L \cup B_R \cup (F \cap Z)$ and it follows from Claim 19 that C' is also a cut between $B_\sim$ and B_Δ in $G^* - \{x_L, x_R \mid x \in B \cap X'\}$. Hence by Lemma 17 there exists an odd cycle transversal of size at most $|S'|$ for $G_B \oplus H_B$. ◀

Hence, we get our randomized polynomial boundaried kernelization for Theorem 16 in the following way. Using Lemma 7, compute a cutset $Z \subseteq V(G^*)$ of size $\mathcal{O}((|B| + \ell)^3)$ for $B_L \cup B_R$. Afterwards apply Reduction Rule 4 in order to get a gluing equivalent graph G' with $\mathcal{O}((|B|+\ell)^3)$ vertices from $B \cup (F \cap Z)$ and at most $\mathcal{O}((|B|+\ell)^6)$ vertices for substituting odd cycles at vertices $x \in B$, and even-length paths between vertices from $B \cup (F \cap Z)$.

7 Vertex Cover[oct]

As our last problem, we consider VERTEX COVER[oct]. We only give a short overview here. For the complete set of reduction rules and used lemmas, we refer to the full version [2]. Similar to the previous section, we can assume to be given a boundaried graph G_B such that $G - B$ is bipartite. After some initial modifications on G_B, we perform some further reduction rules based on the increase of local solution size for $G - B$, when a certain subset $B' \subseteq B$ is explicitly forbidden, which is called the *conflict* caused by B' on F [9]. Namely, for some auxiliary graph G^*, we show a connection between conflicts in G and certain cuts in G^*, which will lead us to applying the representative sets framework in order to find a cut cover in G^*, using which we again choose vertices of G to bypass.

▶ **Theorem 20** (⋆). *The parameterized problem* VERTEX COVER[oct] *admits a polynomial boundaried kernelization with* $\mathcal{O}((|B| + \ell)^3)$ *vertices.*

8 Conclusion

Boundaried kernelization is a recently introduced model for efficient local preprocessing due to Antipov and Kratsch [1]. That work gave polynomial boundaried kernelizations for several pure graphs problems, all based on local reduction rules, as well as several unconditional lower bounds. We showed that also global tools like the matroid-based approach of Kratsch and Wahlström [13] can be leveraged for boundaried kernelization. Moreover, this required to extend the underlying definitions to work for annotated graphs, e.g., graphs with a distinguished set of terminal vertices, including a natural generalization of gluing.

We think that this should also be possible for other problems covered by these tools, e.g., VERTEX COVER[$k - LP$], SUBSET FEEDBACK VERTEX SET[k], and GROUP FEEDBACK VERTEX SET[k], possibly with some restriction as was necessary for s-MULTIWAY CUT. Similarly, using a natural notion of boundaried formula, it would be interesting to get, if possible, a polynomial boundaried kernelization for ALMOST 2-SAT[k].

References

1. Leonid Antipov and Stefan Kratsch. Boundaried kernelization. *CoRR*, abs/2504.18476, 2025. To appear in proceedings of WG 2025. doi:10.48550/arXiv.2504.18476.
2. Leonid Antipov and Stefan Kratsch. Boundaried kernelization via representative sets. *CoRR*, abs/2510.00832, 2025. doi:10.48550/arXiv.2510.00832.
3. Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (Meta) kernelization. *J. ACM*, 63(5):44:1–44:69, 2016. doi:10.1145/2973749.
4. 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.
5. Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Berlin, Heidelberg, 2025. doi:10.1007/978-3-662-70107-2.
6. Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Kernels for (connected) dominating set on graphs with excluded topological minors. *ACM Trans. Algorithms*, 14(1):6:1–6:31, 2018. doi:10.1145/3155298.
7. Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019.
8. Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011. doi:10.1016/j.disopt.2010.05.003.
9. Bart M. P. Jansen and Hans L. Bodlaender. Vertex Cover Kernelization Revisited: Upper and Lower Bounds for a Refined Parameter. *Theory of Computing Systems*, 53(2):263–299, 2013. doi:10.1007/s00224-012-9393-4.
10. Bart M. P. Jansen and Stefan Kratsch. A structural approach to kernels for ILPs: Treewidth and total unimodularity. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 779–791. Springer, 2015. doi:10.1007/978-3-662-48350-3_65.
11. Stefan Kratsch. Recent developments in kernelization: A survey. *Bull. EATCS*, 113, 2014. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/285.
12. Stefan Kratsch and Magnus Wahlström. Compression via Matroids: A Randomized Polynomial Kernel for Odd Cycle Transversal. *ACM Trans. Algorithms*, 10(4):20:1–20:15, 2014. doi:10.1145/2635810.
13. Stefan Kratsch and Magnus Wahlström. Representative Sets and Irrelevant Vertices: New Tools for Kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. doi:10.1145/3390887.
14. Dániel Marx. A parameterized view on matroid optimization problems. *Theoretical Computer Science*, 410(44):4471–4479, 2009. doi:10.1016/j.tcs.2009.07.027.
15. Hazel Perfect. Applications of Menger's graph theorem. *Journal of Mathematical Analysis and Applications*, 22(1):96–111, 1968. doi:10.1016/0022-247X(68)90163-7.
16. Igor Razgon. Large isolating cuts shrink the multiway cut. *CoRR*, abs/1104.5361, 2011. arXiv:1104.5361.
17. Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.
18. Magnus Wahlström. Abusing the tutte matrix: An algebraic instance compression for the k-set-cycle problem. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany*, volume 20 of *LIPIcs*, pages 341–352. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPICS.STACS.2013.341.