Meta-kernelization with Structural Parameters*

Robert Ganian, Friedrich Slivovsky, and Stefan Szeider

Institute of Information Systems, Vienna University of Technology, Vienna, Austria
{rganian,fslivovsky}@gmail.com, stefan@szeider.net

Abstract. Meta-kernelization theorems are general results that provide polynomial kernels for large classes of parameterized problems. The known meta-kernelization theorems, in particular the results of Bodlaender et al. (FOCS'09) and of Fomin et al. (FOCS'10), apply to optimization problems parameterized by *solution size*. We present meta-kernelization theorems that use *structural parameters* of the input and not the solution size. Let $\mathcal{C}$ be a graph class. We define the *$\mathcal{C}$-cover number* of a graph to be the smallest number of modules the vertex set can be partitioned into such that each module induces a subgraph that belongs to the class $\mathcal{C}$.

We show that each graph problem that can be expressed in Monadic Second Order (MSO) logic has a polynomial kernel with a linear number of vertices when parameterized by the $\mathcal{C}$-cover number for any fixed class $\mathcal{C}$ of bounded rank-width (or equivalently, of bounded clique-width, or bounded Boolean-width). Many graph problems such as c-COLORING, c-DOMATIC NUMBER and c-CLIQUE COVER are covered by this meta-kernelization result.

Our second result applies to MSO expressible optimization problems, such as MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, and MAXIMUM CLIQUE. We show that these problems admit a polynomial annotated kernel with a linear number of vertices.

1 Introduction

Kernelization is an algorithmic technique that has become the subject of a very active field in parameterized complexity, see, e.g., the references in [14,21,27]. Kernelization can be considered as a *preprocessing with performance guarantee* that reduces an instance of a parameterized problem in polynomial time to a decision-equivalent instance, the *kernel*, whose size is bounded by a function of the parameter alone [14,21,17]; if the reduced instance is an instance of a different problem, then it is called a *bikernel*. Once a kernel or bikernel is obtained, the time required to solve the original instance is bounded by a function of the parameter and therefore independent of the input size. Consequently one aims at (bi)kernels that are as small as possible.

Every fixed-parameter tractable problem admits a kernel, but the size of the kernel can have an exponential or even non-elementary dependence on the parameter [16]. Thus research on kernelization is typically concerned with the question of whether a fixed-parameter tractable problem under consideration admits a small, and in particular a *polynomial*, kernel. For instance, the parameterized MINIMUM VERTEX COVER

* Research supported by the European Research Council (ERC), project COMPLEX REASON 239962.

K. Chatterjee and J. Sgall (Eds.): MFCS 2013, LNCS 8087, pp. 457–468, 2013.

problem (does a given graph have a vertex cover consisting of k vertices?) admits a polynomial kernel containing at most $2k$ vertices.

There are many fixed-parameter tractable problems for which no polynomial kernels are known. Recently, theoretical tools have been developed to provide strong theoretical evidence that certain fixed-parameter tractable problems do not admit polynomial kernels [3]. In particular, these techniques can be applied to a wide range of graph problems parameterized by treewidth and other width parameters such as clique-width, or rank-width (see e.g., [3,5]). Thus, in order to get polynomial kernels, structural parameters have been suggested that are somewhat weaker than treewidth, including the vertex cover number, max-leaf number, and neighborhood diversity [15,23]. While these parameters do allow polynomial kernels for some problems, no meta-kernelization theorems are known. The general aim here is to find a parameter that admits a polynomial kernel for the given problem while being as general as possible.

We extend this line of research by using results from modular decompositions and rank-width to introduce new structural parameters for which large classes of problems have polynomial kernels. Specifically, we study the *rank-width-d cover number*, which is a special case of a *$\mathcal{C}$-cover number* (see Section 3 for definitions). We establish the following result which is an important prerequisite for our kernelization results.

Theorem 1. *For every constant d, a smallest rank-width-d cover of a graph can be computed in polynomial time.*

Hence, for graph problems parameterized by rank-width-d cover number, we can always compute the parameter in polynomial time. The proof of Theorem 1 relies on a combinatorial property of modules of bounded rank-width that amounts to a variant of partitivity [9].

Our kernelization results take the shape of *algorithmic meta-theorems*, stated in terms of the evaluation of formulas of monadic second order logic (MSO) on graphs. Monadic second order logic over graphs extends first order logic by variables that may range over sets of vertices (sometimes referred to as MSO_1 logic). Specifically, for an MSO formula φ, our first meta-theorem applies to all problems of the following shape, which we simply call *MSO model checking* problems.

MSO-MC_φ
Instance: A graph G.
Question: Does $G \models \varphi$ hold?

Many NP-hard graph problems can be naturally expressed as MSO model checking problems, for instance c-COLORING, c-DOMATIC NUMBER and c-CLIQUE COVER.

Theorem 2. *Let $\mathcal{C}$ be a graph class of bounded rank-width. Every MSO model checking problem, parameterized by the $\mathcal{C}$-cover number of the input graph, has a polynomial kernel with a linear number of vertices.*

While MSO model checking problems already capture many important graph problems, there are some well-known optimization problems on graphs that cannot be captured in this way, such as MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, and MAXIMUM CLIQUE. Many such optimization graph problems can be equivalently stated as decision problems, in the following way. Let $\varphi = \varphi(X)$ be an MSO formula with one free set variable X and $\diamondsuit \in \{\leq, \geq\}$.

MSO-OPT$_{\varphi}^{\diamondsuit}$
Instance: A graph G and an integer $r \in \mathbb{N}$.
Question: Is there a set $S \subseteq V(G)$ such that $G \models \varphi(S)$ and $|S| \diamondsuit r$?

We call problems of this form *MSO optimization problems*. MSO optimization problems form a large fragment of the so-called *LinEMSO* problems [2]. There are dozens of well-known graph problems that can be expressed as MSO optimization problems.

We establish the following result (cf. Section 2 for the definition of a *bikernel*)

Theorem 3. *Let $\mathcal{C}$ be a graph class of bounded rank-width. Every MSO optimization problem, parameterized by the $\mathcal{C}$-cover number of the input graph, has a polynomial bikernel with a linear number of vertices.*

In fact, the obtained bikernel is an instance of an annotated variant of the original MSO optimization problem [1]. Hence, Theorem 3 provides a polynomial kernel for an annotated version of the original MSO optimization problem.

We would like to point out that a class of graphs has bounded rank-width iff it has bounded clique-width iff it has bounded Boolean-width [7]. Hence, we could have equivalently stated the theorems in terms of clique-width or Boolean-width. Furthermore we would like to point out that the theorems hold also for some classes $\mathcal{C}$ where we do not know whether $\mathcal{C}$ can be recognized in polynomial time, and where we do not know how to compute the partition in polynomial time. For instance, the theorems hold if $\mathcal{C}$ is a graph class of bounded clique-width (it is not known whether graphs of clique-width at most 4 can be recognized in polynomial time).

Note: Some proofs were omitted due to space constraints. A full version of this paper is available on arxiv.org (arXiv:1303.1786).

2 Preliminaries

The set of natural numbers (that is, positive integers) will be denoted by $\mathbb{N}$. For $i \in \mathbb{N}$ we write $[i]$ to denote the set $\{1, \ldots, i\}$.

Graphs. We will use standard graph theoretic terminology and notation (cf. [12]). A *module* of a graph $G = (V, E)$ is a nonempty set $X \subseteq V$ such that for each vertex $v \in V \setminus X$ it holds that either no element of X is a neighbor of v or every element of X is a neighbor of v. We say two modules $X, Y \subseteq V$ are *adjacent* if there are vertices $x \in X$ and $y \in Y$ such that x and y are adjacent. A *modular partition* of a graph G is a partition $\{U_1, \ldots, U_k\}$ of its vertex set such that U_i is a module of G for each $i \in [k]$.

Monadic Second-Order Logic on Graphs. We assume that we have an infinite supply of individual variables, denoted by lowercase letters x, y, z, and an infinite supply of set variables, denoted by uppercase letters X, Y, Z. *Formulas* of *monadic second-order logic* (MSO) are constructed from atomic formulas $E(x, y)$, $X(x)$, and $x = y$ using the connectives $\neg$ (negation), $\wedge$ (conjunction) and existential quantification $\exists x$ over individual variables as well as existential quantification $\exists X$ over set variables. Individual variables range over vertices, and set variables range over sets of vertices. The atomic formula $E(x, y)$ expresses adjacency, $x = y$ expresses equality, and $X(x)$ expresses that vertex x in the set X. From this, we define the semantics of monadic second-order logic in the standard way (this logic is sometimes called MSO_1).

Free and bound variables of a formula are defined in the usual way. A *sentence* is a formula without free variables. We write $\varphi(X_1, \ldots, X_n)$ to indicate that the set of free variables of formula φ is $\{X_1, \ldots, X_n\}$. If $G = (V, E)$ is a graph and $S_1, \ldots, S_n \subseteq V$ we write $G \models \varphi(S_1, \ldots, S_n)$ to denote that φ holds in G if the variables X_i are interpreted by the sets S_i, for $i \in [n]$.

We review MSO *types* roughly following the presentation in [24]. The *quantifier rank* of an MSO formula φ is defined as the nesting depth of quantifiers in φ. For non-negative integers q and l, let $\text{MSO}_{q,l}$ consist of all MSO formulas of quantifier rank at most q with free set variables in $\{X_1, \ldots, X_l\}$.

Let $\varphi = \varphi(X_1, \ldots, X_l)$ and $\psi = \psi(X_1, \ldots, X_l)$ be MSO formulas. We say φ and ψ are equivalent, written $\varphi \equiv \psi$, if for all graphs G and $U_1, \ldots, U_l \subseteq V(G)$, $G \models \varphi(U_1, \ldots, U_l)$ if and only if $G \models \psi(U_1, \ldots, U_l)$. Given a set F of formulas, let $F/{\equiv}$ denote the set of equivalence classes of F with respect to $\equiv$. A system of representatives of $F/{\equiv}$ is a set $R \subseteq F$ such that $R \cap C \neq \emptyset$ for each equivalence class $C \in F/{\equiv}$. The following statement has a straightforward proof using normal forms (see Proposition 7.5 in [24] for details).

Fact 1. *Let q and l be fixed non-negative integers. The set $MSO_{q,l}/{\equiv}$ is finite, and one can compute a system of representatives of $MSO_{q,l}/{\equiv}$.*

We will assume that for any pair of non-negative integers q and l the system of representatives of $\text{MSO}_{q,l}/{\equiv}$ given by Fact 1 is fixed.

Definition 4 (MSO Type). *Let q, l be a non-negative integers. For a graph G and an l-tuple $\boldsymbol{U}$ of sets of vertices of G, we define $type_q(G, \boldsymbol{U})$ as the set of formulas $\varphi \in MSO_{q,l}$ such that $G \models \varphi(\boldsymbol{U})$. We call $type_q(G, \boldsymbol{U})$ the MSO* rank-q type of $\boldsymbol{U}$ in G.

It follows from Fact 1 that up to logical equivalence, every type contains only finitely many formulas. This allows us to represent types using MSO formulas as follows.

Lemma 5. *Let q and l be non-negative integer constants, let G be a graph, and let $\boldsymbol{U}$ be an l-tuple of sets of vertices of G. One can compute a formula $\Phi \in MSO_{q,l}$ such that for any graph G' and any l-tuple $\boldsymbol{U}'$ of sets of vertices of G' we have $G' \models \Phi(\boldsymbol{U}')$ if and only if $type_q(G, \boldsymbol{U}) = type_q(G', \boldsymbol{U}')$. Moreover, if $G \models \varphi(\boldsymbol{U})$ can be decided in polynomial time for any fixed $\varphi \in MSO_{q,l}$ then Φ can be computed in time polynomial in $|V(G)|$.*

Proof. Let R be a system of representatives of $\text{MSO}_{q,l}/{\equiv}$ given by Fact 1. Because q and l are constant, we can consider both the cardinality of R and the time required to compute it as constants. Let $\Phi \in \text{MSO}_{q,l}$ be the formula defined as $\Phi = \bigwedge_{\varphi \in S} \varphi \wedge \bigwedge_{\varphi \in R \setminus S} \neg\varphi$, where $S = \{\, \varphi \in R : G \models \varphi(\boldsymbol{U}) \,\}$. We can compute Φ by deciding $G \models \varphi(\boldsymbol{U})$ for each $\varphi \in R$. Since the number of formulas in R is a constant, this can be done in polynomial time if $G \models \varphi(\boldsymbol{U})$ can be decided in polynomial time for any fixed $\varphi \in \mathrm{MSO}_{q,l}$.

Let G' be an arbitrary graph and $\boldsymbol{U}'$ an l-tuple of subsets of $V(G')$. We claim that $\mathit{type}_q(G, \boldsymbol{U}) = \mathit{type}_q(G', \boldsymbol{U'})$ if and only if $G' \models \Phi(\boldsymbol{U}')$. Since $\Phi \in \mathrm{MSO}_{q,l}$ the forward direction is trivial. For the converse, assume $\mathit{type}_q(G, \boldsymbol{U}) \neq \mathit{type}_q(G', \boldsymbol{U'})$. First suppose $\varphi \in \mathit{type}_q(G, \boldsymbol{U}) \setminus \mathit{type}_q(G', \boldsymbol{U'})$. The set R is a system of representatives of $\mathrm{MSO}_{q,l}/\!\equiv$, so there has to be a $\psi \in R$ such that $\psi \equiv \varphi$. But $G' \models \Phi(\boldsymbol{U}')$ implies $G' \models \psi(\boldsymbol{U}')$ by construction of Φ and thus $G' \models \varphi(\boldsymbol{U}')$, a contradiction. Now suppose $\varphi \in \mathit{type}_q(G', \boldsymbol{U}') \setminus \mathit{type}_q(G, \boldsymbol{U})$. An analogous argument proves that there has to be a $\psi \in R$ such that $\psi \equiv \varphi$ and $G' \models \neg\psi(\boldsymbol{U}')$. It follows that $G' \not\models \varphi(\boldsymbol{U}')$, which again yields a contradiction. □

Fixed-Parameter Tractability and Kernels. A *parameterized problem* P is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. For a problem instance $(x, k) \in \Sigma^* \times \mathbb{N}$ we call x the main part and k the parameter. A parameterized problem P is *fixed-parameter tractable* (FPT in short) if a given instance (x, k) can be solved in time $O(f(k) \cdot p(|x|))$ where f is an arbitrary computable function of k and p is a polynomial function.

A *bikernelization* for a parameterized problem $P \subseteq \Sigma^* \times \mathbb{N}$ into a parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, outputs in time polynomial in $|x| + k$ a pair $(x', k') \in \Sigma^* \times \mathbb{N}$ such that (i) $(x, k) \in P$ if and only if $(x', k') \in Q$ and (ii) $|x'| + k' \leq g(k)$, where g is an arbitrary computable function. The reduced instance (x', k') is the *bikernel*. If $P = Q$, the reduction is called a *kernelization* and (x', k') a *kernel*. The function g is called the *size* of the (bi)kernel, and if g is a polynomial then we say that P admits a *polynomial (bi)kernel*.

It is well known that every fixed-parameter tractable problem admits a generic kernel, but the size of this kernel can have an exponential or even non-elementary dependence on the parameter [13]. Since recently there have been workable tools available for providing strong theoretical evidence that certain parameterized problems do not admit a polynomial kernel [3,25].

Rank-width. The graph invariant rank-width was introduced by Oum and Seymour [26] with the original intent of investigating the graph invariant clique-width. It later turned out that rank-width itself is a useful parameter, with several advantages over clique-width.

For a graph G and $U, W \subseteq V(G)$, let $\boldsymbol{A}_G[U, W]$ denote the $U \times W$-submatrix of the adjacency matrix over the two-element field $\mathrm{GF}(2)$, i.e., the entry $a_{u,w}$, $u \in U$ and $w \in W$, of $\boldsymbol{A}_G[U, W]$ is 1 if and only if $\{u, w\}$ is an edge of G. The *cut-rank* function ρ_G of a graph G is defined as follows: For a bipartition (U, W) of the vertex set $V(G)$, $\rho_G(U) = \rho_G(W)$ equals the rank of $\boldsymbol{A}_G[U, W]$ over $\mathrm{GF}(2)$.

A *rank-decomposition* of a graph G is a pair (T, μ) where T is a tree of maximum degree 3 and $\mu : V(G) \to \{t : t \text{ is a leaf of } T\}$ is a bijective function. For an edge e of T, the connected components of $T - e$ induce a bipartition (X, Y) of the set of leaves of T. The *width* of an edge e of a rank-decomposition (T, μ) is $\rho_G(\mu^{-1}(X))$. The *width* of (T, μ) is the maximum width over all edges of T. The *rank-width* of G is the minimum width over all rank-decompositions of G.

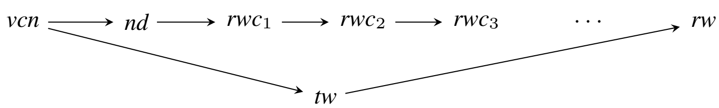

Fig. 1. Relationship between graph invariants: the vertex cover number (*vcn*), the neighborhood diversity (*nd*), the rank-width-d cover number (*rwc*$_d$), the rank-width (*rw*), and the treewidth (*tw*). An arrow from A to B indicates that for any graph class for which B is bounded also A is bounded. See Proposition 8 and [26] for references.

Theorem 6 ([22]). *Let $k \in \mathbb{N}$ be a constant and $n \geq 2$. For an n-vertex graph G, we can output a rank-decomposition of width at most k or confirm that the rank-width of G is larger than k in time $O(n^3)$.*

Theorem 7 ([20]). *Let $d \in \mathbb{N}$ be a constant and let φ and $\psi = \psi(X)$ be fixed MSO formulas. Given a graph G with $rw(G) \leq d$, we can decide whether $G \models \varphi$ in polynomial time. Moreover, a set $S \subseteq V(G)$ of minimum (maximum) cardinality such that $G \models \psi(S)$ can be found in polynomial time, if one exists.*

3 Rank-Width Covers

Let $\mathcal{C}$ be a graph class containing all trivial graphs, i.e., all graphs consisting of only a single vertex. We define a *$\mathcal{C}$-cover of G* as a modular partition $\{U_1, \ldots, U_k\}$ of $V(G)$ such that the induced subgraph $G[U_i]$ belongs to the class $\mathcal{C}$ for each $i \in [k]$. Accordingly, the *$\mathcal{C}$-cover number* of G is the size of a smallest $\mathcal{C}$-cover of G.

Of special interest to us are the classes $\mathcal{R}_d$ of graphs of *rank-width* at most d. We call the $\mathcal{R}_d$-cover number also the *rank-width-d cover number*. If $\mathcal{C}$ is the class of all complete graphs and all edgeless graphs, then the $\mathcal{C}$-cover number equals the neighborhood diversity [23], and clearly $\mathcal{C} \subsetneq \mathcal{R}_1$. Figure 1 shows the relationship between the rank-width-d cover number and some other graph invariants.

We state some further properties of rank-width-d covers.

Proposition 8. *Let vcn, nd, and rw denote the vertex cover number, the neighborhood diversity, and the rank-width of a graph G, respectively. Then the following (in)equalities hold for any $d \in \mathbb{N}$:*

1. *$rwc_d(G) \leq nd(G) \leq 2^{vcn(G)}$,*
2. *if $d \geq rw(G)$, then $|rwc_d(G)| = 1$.*

Proof. (1) The neighborhood diversity of a graph is also a rank-width-1 cover. The neighborhood diversity is known to be upper-bounded by $2^{vcn(G)}$ [23].

(2) This follows immediately from the definition of rank-width-d covers. □

3.1 Finding the Cover

Next we state several properties of modules of graphs. These will be used to obtain a polynomial algorithm for finding smallest rank-width-d covers.

The *symmetric difference* of sets A, B is $A \triangle B = (A \setminus B) \cup (B \setminus A)$. Sets A and B *overlap* if $A \cap B \neq \emptyset$ but neither $A \subseteq B$ nor $B \subseteq A$.

Definition 9. *Let $\mathcal{S} \subseteq 2^S$ be a family of subsets of a set S. We call $\mathcal{S}$* partitive *if it satisfies the following properties:*

1. *$S \in \mathcal{S}$, $\emptyset \notin \mathcal{S}$, and $\{x\} \in \mathcal{S}$ for each $x \in S$.*
2. *For every pair of overlapping subsets $A, B \in \mathcal{S}$, the sets $A \cup B$, $A \cap B$, $A \triangle B$, $A \setminus B$, and $B \setminus A$ are contained in $\mathcal{S}$.*

Theorem 10 ([9]). *The family of modules of a graph G is partitive.*

Lemma 11 ([6]). *Let G be a graph and $x, y \in V(G)$. There is a unique minimal (with respect to set inclusion) module M of G such that $x, y \in M$, and M can be computed in time $O(|V(G)|^2)$.*

Definition 12. *Let G be a graph and $d \in \mathbb{N}$. We define a relation $\sim_d^G$ on $V(G)$ by letting $v \sim_d^G w$ if and only if there is a module M of G with $v, w \in M$ and $rw(G[M]) \leq d$. We drop the superscript from $\sim_d^G$ if the graph G is clear from context.*

Proposition 13. *For every graph G and $d \in \mathbb{N}$ the relation $\sim_d$ is an equivalence relation, and each equivalence class U of $\sim_d$ is a module of G with $rw(G[U]) \leq d$.*

Corollary 14. *Let G be a graph and $d \in \mathbb{N}$. The equivalence classes of $\sim_d$ form a smallest rank-width-d cover of G.*

Proposition 15. *Let $d \in \mathbb{N}$ be a constant. Given a graph G and two vertices $v, w \in V(G)$, we can decide whether $v \sim_d w$ in polynomial time.*

Proof (of Theorem 1). Let $d \in \mathbb{N}$ be a constant. Given a graph G, we can compute the set of equivalence classes of $\sim_d$ by testing whether $v \sim_d w$ for each pair of vertices $v, w \in V(G)$. By Proposition 15, this can be done in polynomial time, and by Corollary 14, $V(G)/\sim_d$ is a smallest rank-width-d cover of G. □

4 Kernels for MSO Model Checking

In this section, we show that every MSO model checking problem admits a polynomial kernel when parameterized by the $\mathcal{C}$-cover number of the input graph, where $\mathcal{C}$ is some recursively enumerable class of graphs satisfying the following properties:

(I) $\mathcal{C}$ contains all trivial graphs, and a $\mathcal{C}$-cover of a graph G with minimum cardinality can be computed in polynomial time.
(II) There is an algorithm $\mathbb{A}$ that decides whether $G \models \varphi$ in time polynomial in $|V(G)|$ for any fixed MSO sentence φ and any graph $G \in \mathcal{C}$.

For obtaining the kernel for MSO model checking problems, we proceed as follows. First, we compute a smallest rank-width-d cover of the input graph G in polynomial time. Second, we compute for each module a small representative of constant size. Third, we replace each module with a constant size module, which results in the kernel. We show how to carry out the second and third steps below.

Let G be a graph and $U \subseteq V(G)$. Let $\boldsymbol{V}$ be an l-tuple of sets of vertices of G. We write $\boldsymbol{V}|_U = (V_1 \cap U, \dots, V_l \cap U)$ to refer to the elementwise intersection of $\boldsymbol{V}$ with U. If $\{U_1, \dots, U_k\}$ is a modular partition of G and $i \in [k]$ we will abuse notation and write $\boldsymbol{V}|_i = \boldsymbol{V}_{U_i}$ if there is no ambiguity about what partition the index refers to.

Definition 16 (Congruent). *Let q and l be non-negative integers and let G and G' be graphs with modular partitions $\boldsymbol{M} = \{M_1, \dots, M_k\}$ and $\boldsymbol{M'} = \{M'_1, \dots, M'_k\}$, respectively. Let $\boldsymbol{V_0}$ be an l-tuple of subsets of $V(G)$ and let $\boldsymbol{U_0}$ be an l-tuple of subsets of $V(G')$. We say $(G, \boldsymbol{M}, \boldsymbol{V_0})$ and $(G', \boldsymbol{M}', \boldsymbol{U_0})$ are q-*congruent *if the following conditions are met:*

1. *For every $i, j \in [k]$ with $i \neq j$, M_i and M_j are adjacent in G if and only if M'_i and M'_j are adjacent in G'.*
2. *For each $i \in [k]$, $type_q(G[M_i], \boldsymbol{V_0}|_i) = type_q(G'[M'_i], \boldsymbol{U_0}|_i)$.*

We begin by showing how congruents are related to the previously introduced notion of types.

Lemma 17. *Let q and l be non-negative integers and let G and G' be graphs with modular partitions $\boldsymbol{M} = \{M_1, \dots, M_k\}$ and $\boldsymbol{M'} = \{M'_1, \dots, M'_k\}$. Let $\boldsymbol{V_0}$ be an l-tuple of subsets of $V(G)$ and let $\boldsymbol{U_0}$ be an l-tuple of subsets of $V(G')$. If $(G, \boldsymbol{M}, \boldsymbol{V_0})$ and $(G', \boldsymbol{M}', \boldsymbol{U_0})$ are q-congruent, then $type_q(G, \boldsymbol{V_0}) = type_q(G', \boldsymbol{U_0})$.*

Next, we showcase the tool we use to replace a graph G by a small representative.

Lemma 18. *Let $\mathcal{C}$ be a recursively enumerable graph class and let q be a non-negative integer constant. Let $G \in \mathcal{C}$ be a graph. If $G \models \varphi$ can be decided in time polynomial in $|V(G)|$ for any fixed $\varphi \in MSO_{q,0}$ then one can in polynomial time compute a graph $G' \in \mathcal{C}$ such that $|V(G')|$ is bounded by a constant and $type_q(G) = type_q(G')$.*

Finally, in Lemma 19 below we use Lemma 18 to obtain our polynomial kernels.

Lemma 19. *Let q be a non-negative integer constant, and let $\mathcal{C}$ be a recursively enumerable graph class satisfying (II). Then given a graph G and a $\mathcal{C}$-cover $\{U_1, \dots, U_k\}$, one can in polynomial time compute a graph G' with modular partition $\{U'_1, \dots, U'_k\}$ such that $(G, \boldsymbol{U})$ and $(G', \boldsymbol{U}')$ are q-congruent and for each $i \in [k]$, $G'[U'_i] \in \mathcal{C}$ and the number of vertices in U'_i is bounded by a constant.*

Proposition 20. *Let φ be a fixed MSO sentence. Let $\mathcal{C}$ be a recursively enumerable graph class satisfiying (I) and (II). Then* MSO-MC$_\varphi$ *has a polynomial kernel parameterized by the $\mathcal{C}$-cover number of the input graph.*

Proof (of Theorem 2). Immediate from Theorems 1, 6, and 7 in combination with Proposition 20. □

Corollary 21. *The following problems have polynomial kernels when parameterized by the rank-width-d cover number of the input graph: c-*Coloring*, c-*Domatic Number*, c-*Partition into Trees*, c-*Clique Cover*, c-*Partition into Perfect Matchings*, c-*Covering by Complete Bipartite Subgraphs*.*

5 Kernels for MSO Optimization

By definition, MSO formulas can only directly capture decision problems such as 3-coloring, but many problems of interest are formulated as optimization problems. The usual way of transforming decision problems into optimization problems does not work here, since the MSO language cannot handle arbitrary numbers.

Nevertheless, there is a known solution. Arnborg, Lagergren, and Seese [2] (while studying graphs of bounded tree-width), and later Courcelle, Makowsky, and Rotics [10] (for graphs of bounded clique-width), specifically extended the expressive power of MSO logic to define so-called LINEMS optimization problems, and consequently also showed the existence of efficient (parameterized) algorithms for such problems in the respective cases.

The class of so-called MSO optimization problems (problems which may be stated as MSO-OPT$_\varphi^\diamondsuit$) considered here are a streamlined and simplified version of the formalism introduced in [10]. Specifically, we consider only a single free variable X, and ask for a satisfying assignment of X with minimum or maximum cardinality. To achieve our results, we need a recursively enumerable graph class $\mathcal{C}$ that satisfies (I) and (II) along with the following property:

(III)Let $\varphi = \varphi(X)$ be a fixed MSO formula. Given a graph $G \in \mathcal{C}$, a set $S \subseteq V(G)$ of minimum (maximum) cardinality such that $G \models \varphi(S)$ can be found in polynomial time, if one exists.

Our approach will be similar to the MSO kernelization algorithm, with one key difference: when replacing the subgraph induced by a module, the cardinalities of subsets of a given q-type may change, so we need to keep track of their cardinalities in the original subgraph.

To do this, we introduce an annotated version of MSO-OPT$_\varphi^\diamondsuit$. Given a graph $G = (V, E)$, an *annotation* $\mathcal{W}$ is a set of triples (X, Y, w) with $X \subseteq V, Y \subseteq V, w \in \mathbb{N}$. For every set $Z \subseteq V$ we define

$$\mathcal{W}(Z) = \sum_{(X,Y,w)\in\mathcal{W}, X\subseteq Z, Y\cap Z=\emptyset} w.$$

We call the pair $(G, \mathcal{W})$ an *annotated graph*. If the integer w is represented in binary, we can represent a triple (X, Y, w) in space $|X| + |Y| + \log_2(w)$. Consequently, we may assume that the size of the encoding of an annotated graph $(G, \mathcal{W})$ is polynomial in $|V(G)| + |\mathcal{W}| + \max_{(X,Y,w)\in\mathcal{W}} \log_2 w$.

Each MSO formula $\varphi(X)$ and $\diamondsuit \in \{\leq, \geq\}$ gives rise to an *annotated MSO-optimization problem.*

aMSO-OPT$_\varphi^\diamondsuit$
Instance: A graph G with an annotation $\mathcal{W}$ and an integer $r \in \mathbb{N}$.
Question: Is there a set $Z \subseteq V(G)$ such that $G \models \varphi(Z)$ and $\mathcal{W}(Z) \diamondsuit r$?

Notice that any instance of MSO-OPT$_\varphi^\diamondsuit$ is also an instance of aMSO-OPT$_\varphi^\diamondsuit$ with the trivial annotation $\mathcal{W} = \{ (\{v\}, \emptyset, 1) : v \in V(G) \}$. The main result of this section is a bikernelization algorithm which transforms any instance of MSO-$\text{OPT}_{\varphi}^{\diamondsuit}$ into an instance of aMSO-$\text{OPT}_{\varphi}^{\diamondsuit}$; this kind of bikernel is called an *annotated kernel* [1].

The results below are stated and proved for minimization problems aMSO-$\text{OPT}_{\varphi}^{\leq}$ only. This is without loss of generality—the proofs for maximization problems are symmetric.

Lemma 22. *Let q and l be non-negative integers and let G and G' be a graphs such that G and G' have the same $q + l$ MSO type. Then for any l-tuple $\boldsymbol{V}$ of sets of vertices of G, there exists an l-tuple $\boldsymbol{U}$ of sets of vertices of G' such that $type_q(G, \boldsymbol{V}) = type_q(G', \boldsymbol{U})$.*

Proof. Suppose there exists an l-tuple $\boldsymbol{V}$ of sets of vertices of G, and a formula $\varphi = \varphi(X_1, \ldots, X_l) \in \text{MSO}_{q,l}$ such that $G \models \varphi(V_1, \ldots, V_l)$ but for every l-tuple $\boldsymbol{U}$ of sets of vertices of G' we have $G' \not\models \varphi(U_1, \ldots, U_l)$. Let $\psi = \exists X_1 \ldots \exists X_l\ \varphi$. Clearly, $\psi \in \text{MSO}_{q+l,0}$ and $G \models \psi$ but $G' \not\models \psi$, a contradiction. □

Using Lemma 22 and the results of Section 4, we may proceed directly to the construction of our annotated kernel.

Lemma 23. *Let $\varphi = \varphi(X)$ be a fixed MSO formula and $\mathcal{C}$ be a recursively enumerable graph class satisfiying (II) and (III). Then given an instance (G, r) of* MSO-$\text{OPT}_{\varphi}^{\leq}$ *and a $\mathcal{C}$-cover $\{U_1, \ldots, U_k\}$ of G, an annotated graph $(G', \mathcal{W})$ satisfying the following properties can be computed in polynomial time.*

1. *$(G, r) \in$ MSO-$\text{OPT}_{\varphi}^{\leq}$ if and only if $(G', \mathcal{W}, r) \in a$MSO-$\text{OPT}_{\varphi}^{\leq}$.*
2. *$|V(G')| \in O(k)$.*
3. *The encoding size of $(G', \mathcal{W})$ is $O(k \log(|V(G)|))$.*

The last obstacle we face is that the annotation itself may be "too large" for the kernel. Here we use the following simple folklore result, which allows us to prove that either our annotated kernel is "small enough", or we can solve our problem in polynomial time (and subsequently output a trivial yes/no instance).

Fact 2 (Folklore). *Given an MSO sentence φ and a graph G, one can decide whether $G \models \varphi$ in time $O(2^{nl})$, where $n = |V(G)|$ and $l = |\varphi|$.*

Proposition 24. *Let $\varphi = \varphi(X)$ be a fixed MSO formula, and let $\mathcal{C}$ be a recursively enumerable graph class satisfying (I), (II), and (III). Then* MSO-$\text{OPT}_{\varphi}^{\leq}$ *has a polynomial bikernel parameterized by the $\mathcal{C}$-cover number of the input graph.*

Proof (of Theorem 3). Immediate from Theorems 1, 6, and 7 when combined with Proposition 24. □

Corollary 25. *The following problems have polynomial bikernels when parameterized by the rank-width-d cover number of the input graph:* MINIMUM DOMINATING SET, MINIMUM VERTEX COVER, MINIMUM FEEDBACK VERTEX SET, MAXIMUM INDEPENDENT SET, MAXIMUM CLIQUE, LONGEST INDUCED PATH, MAXIMUM BIPARTITE SUBGRAPH, MINIMUM CONNECTED DOMINATING SET.

6 Conclusion

Recently Bodlaender et al. [4] and Fomin et al. [18] established *meta-kernelization theorems* that provide polynomial kernels for large classes of parameterized problems. The known meta-kernelization theorems apply to optimization problems parameterized by *solution size*. Our results are, along with very recent results parameterized by the modulator to constant-treedepth [19], the first meta-kernelization theorems that use a *structural parameter* of the input and not the solution size. In particular, we would like to emphasize that our Theorem 3 applies to a large class of optimization problems where the solution size can be arbitrarily large.

It is also worth noting that our structural parameter, the rank-width-d cover number, provides a trade-off between the maximum rank-width of modules (the constant d) and the maximum number of modules (the parameter k). Different problem inputs might be better suited for smaller d and larger k, others for larger d and smaller k. This two-dimensional setting could be seen as a contribution to *multivariate complexity analysis* as advocated by Fellows et al. [15].

We conclude by mentioning possible directions for future research. We believe that some of our results can be extended from modular partitions to partitions into splits [8,11][1]. This would indeed result in more general parameters, however the precise details require further work (one problem is that while all modules are partitive, only strong splits have this property). Another direction is to focus on polynomial kernels for problems which cannot be described by MSO logic, such as HAMILTONIAN PATH or CHROMATIC NUMBER.

References

1. Abu-Khzam, F.N., Fernau, H.: Kernels: Annotated, proper and induced. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 264–275. Springer, Heidelberg (2006)
2. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. J. Algorithms 12(2), 308–340 (1991)
3. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. of Computer and System Sciences 75(8), 423–434 (2009)
4. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (meta) kernelization. In: FOCS 2009, pp. 629–638. IEEE Computer Society (2009)
5. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Preprocessing for treewidth: A combinatorial analysis through kernelization. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 437–448. Springer, Heidelberg (2011)
6. Bui-Xuan, B.-M., Habib, M., Limouzy, V., de Montgolfier, F.: Algorithmic aspects of a general modular decomposition theory. Discr. Appl. Math. 157(9), 1993–2009 (2009)
7. Bui-Xuan, B.-M., Telle, J.A., Vatshelle, M.: Boolean-width of graphs. Theoretical Computer Science 412(39), 5187–5204 (2011)
8. Charbit, P., de Montgolfier, F., Raffinot, M.: Linear time split decomposition revisited. SIAM J. Discrete Math. 26(2), 499–514 (2012)
9. Chein, M., Habib, M., Maurer, M.: Partitive hypergraphs. Discrete Math. 37(1), 35–50 (1981)

[1] We thank Sang-il Oum for pointing this out to us.

10. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. 33(2), 125–150 (2000)
11. Cunningham, W.H.: Decomposition of directed graphs. SIAM J. Algebraic Discrete Methods 3(2), 214–228 (1982)
12. Diestel, R.: Graph Theory, 2nd edn. Graduate Texts in Mathematics, vol. 173. Springer, New York (2000)
13. Downey, R., Fellows, M.R., Stege, U.: Parameterized complexity: A framework for systematically confronting computational intractability. In: Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future. AMS-DIMACS, vol. 49, pp. 49–99. American Mathematical Society (1999)
14. Fellows, M.R.: The lost continent of polynomial time: Preprocessing and kernelization. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 276–277. Springer, Heidelberg (2006)
15. Fellows, M.R., Jansen, B.M., Rosamond, F.: Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. European J. Combin. 34(3), 541–566 (2013)
16. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series, vol. XIV. Springer, Berlin (2006)
17. Fomin, F.V.: Kernelization. In: Ablayev, F., Mayr, E.W. (eds.) CSR 2010. LNCS, vol. 6072, pp. 107–108. Springer, Heidelberg (2010)
18. Fomin, F.V., Lokshtanov, D., Misra, N., Saurabh, S.: Planar f-deletion: Approximation, kernelization and optimal fpt algorithms. In: FOCS 2012, pp. 470–479. IEEE Computer Society (2012)
19. Gajarský, J., Hliněný, P., Obdrzálek, J., Ordyniak, S., Reidl, F., Rossmanith, P., Villaamil, F.S., Sikdar, S.: Kernelization using structural parameters on sparse graph classes. CoRR, abs/1302.6863 (2013)
20. Ganian, R., Hliněný, P.: On parse trees and Myhill-Nerode-type tools for handling graphs of bounded rank-width. Discr. Appl. Math. 158(7), 851–867 (2010)
21. Guo, J., Niedermeier, R.: Invitation to data reduction and problem kernelization. ACM SIGACT News 38(2), 31–45 (2007)
22. Hliněný, P., Oum, S.: Finding branch-decompositions and rank-decompositions. SIAM J. Comput. 38(3), 1012–1032 (2008)
23. Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. Algorithmica 64(1), 19–37 (2012)
24. Libkin, L.: Elements of Finite Model Theory. Springer (2004)
25. Misra, N., Raman, V., Saurabh, S.: Lower bounds on kernelization. Discrete Optimization 8(1), 110–128 (2011)
26. Oum, S., Seymour, P.: Approximating clique-width and branch-width. J. Combin. Theory Ser. B 96(4), 514–528 (2006)
27. Rosamond, F.: Table of races. In: Parameterized Complexity Newsletter, pp. 4–5 (2010), `http://fpt.wikidot.com/`