Algorithmica (2012) 63:532–550
DOI 10.1007/s00453-011-9559-5

Polynomial Kernelizations for MIN $F^+\Pi_1$ and MAX NP

Stefan Kratsch

Received: 13 October 2009 / Accepted: 28 July 2011 / Published online: 31 August 2011

Abstract It has been observed in many places that constant-factor approximable problems often admit polynomial or even linear problem kernels for their decision versions, e.g., VERTEX COVER, FEEDBACK VERTEX SET, and TRIANGLE PACKING. While there exist examples like BIN PACKING, which does not admit any kernel unless P = NP, there apparently is a strong relation between these two polynomial-time techniques. We add to this picture by showing that the natural decision versions of all problems in two prominent classes of constant-factor approximable problems, namely MIN $F^+\Pi_1$ and MAX NP, admit polynomial problem kernels. Problems in MAX SNP, a subclass of MAX NP, are shown to admit kernels with a linear base set, e.g., the set of vertices of a graph. This extends results of Cai and Chen (J. Comput. Syst. Sci. 54(3): 465–474, 1997), stating that the standard parameterizations of problems in MAX SNP and MIN $F^+\Pi_1$ are fixed-parameter tractable, and complements recent research on problems that do not admit polynomial kernelizations (Bodlaender et al. in J. Comput. Syst. Sci. 75(8): 423–434, 2009).

Keywords Combinatorial optimization · Kernelization · Parameterized complexity

1 Introduction

Approximation and kernelization are two major ways of coping with NP-hardness in polynomial time. The former relaxes the exactness requirement to that of finding good approximate solutions. The latter, as a formulation of preprocessing, shrinks the instance to a guaranteed size in terms of some difficulty parameter. For approximate solutions to a problem it is quite desirable to get solutions within a constant-factor of the optimum, or even arbitrarily good approximations in polynomial time through

S. Kratsch (✉)
Utrecht University, Utrecht, The Netherlands
e-mail: skratsch@mpi-inf.mpg.de

Table 1 Approximation ratio and size of problem kernels for some optimization problems

	Approximation ratio	Kernel size
VERTEX COVER	2 [24]	$\mathcal{O}(k)$ [11]
CONNECTED VERTEX COVER	2 [38]	not polynomial [13]
FEEDBACK VERTEX SET	2 [3]	$\mathcal{O}(k^2)$ [40]
BIN PACKING	1.5 [39]	none unless P = NP [22]
MINIMUM FILL-IN	$\mathcal{O}(\text{opt})$ [34]	$\mathcal{O}(k^2)$ [34]
TREEWIDTH	$\mathcal{O}(\sqrt{\log \text{opt}})$ [17]	not polynomial[(a)] [7]

[(a)]Treewidth does not admit a polynomial kernelization unless there is an and-distillation algorithm for all NP complete problems [7]. Though unlikely, this is not known to imply a collapse of the polynomial hierarchy

polynomial-time approximation schemes. In the world of preprocessing, *polynomial kernelizations* with a guaranteed size polynomial in the parameter are often the first goal, later aiming for stronger and stronger bounds down to linear kernels. Considering these two polynomial-time techniques it is only natural to study the relation between them.

This paper seeks to further the understanding of the relation between constant-factor approximation and polynomial kernelizations. This is motivated by the large number of problems that both techniques were successfully applied to so far, e.g., VERTEX COVER, MAX SAT, FEEDBACK VERTEX SET, and TRIANGLE PACKING; see Table 1 for approximability and kernelization results for some well-known problems. Let us point out that there do exist examples that rule out a general equivalence of these two notions, e.g., CONNECTED VERTEX COVER or BIN PACKING. Both problems have constant-factor approximation algorithms but none of them admits a polynomial kernel: The former admits a $\mathcal{O}(2.761^k n^c)$-time algorithm [33], and hence a kernel of size $\mathcal{O}(2.761^k)$,[1] but it has no kernel of polynomial size unless the polynomial hierarchy collapses [13]. The latter does not even admit a polynomial-time algorithm for $k = 2$ unless P = NP, by an immediate reduction from PARTITION [22]. Consider also the MINIMUM FILL-IN problem, which has a polynomial kernel but the best known ratio is $\mathcal{O}(\text{opt})$ [34]. Since a general result is ruled out we take the natural approach of considering subclasses of the class of all constant-factor approximable problems (APX), namely MIN $F^+\Pi_1$ and MAX NP.

Our Work We prove that the standard parameterizations of problems in MIN $F^+\Pi_1$ and MAX NP admit polynomial kernelizations. This extends results of Cai and Chen [8] who showed that the standard parameterizations of all problems in MIN $F^+\Pi_1$ and MAX SNP (a subclass of MAX NP) are fixed-parameter tractable; or equivalently admit some (possibly exponential) kernelization. Interestingly perhaps, both our results rely on the Sunflower Lemma due to Erdős and Rado [15].

[1]By a folklore result: run the algorithm for $\mathcal{O}(n^{c+1})$ steps, it will either provide the correct answer (and we return a yes- or no-instance of constant size) or if it does not finish then it follows that $n < 2.761^k$ and we have the kernel.

Related Work Kernelization has received significant interest over the last fifteen years, maturing from a technique to prove fixed-parameter tractability into its own field of research. In the literature there exist a significant number of positive results; we will only highlight a few from recent years, namely a kernel with $\mathcal{O}(k)$ vertices for VERTEX COVER by Chen et al. [11], a $\mathcal{O}(k^2)$ vertices kernel for FEEDBACK VERTEX SET by Thomassé, and a $\mathcal{O}(k^{d-1})$ vertices kernel for d-HITTING SET by Abu-Khzam [1]. Recently Bodlaender et al. [7] presented the first negative results concerning the existence of polynomial kernelizations for some natural fixed-parameter tractable problems. Using the notion of a *distillation algorithm* and results due to Fortnow and Santhanam [21], they were able to show that the existence of polynomial kernelizations for so-called *compositional* parameterized problems implies a collapse of the polynomial hierarchy to the third level. These seminal results led to an increased interest in polynomial lower bounds for kernelization as well as in polynomial kernelizations as a good way of understanding efficient preprocessing (and possibly ruling it out by means of polynomial lower bounds). A follow-up paper by Bodlaender et al. [5] proposed the application of polynomial-time transformations, that allow only a polynomial increase in the parameter, to transfer lower and upper bounds between problems. A number of papers already apply the framework of Bodlaender et al. [5, 7] to obtain polynomial lower bounds for a variety of problems. e.g., [13, 18, 28]. An important contribution to kernelization lower bounds was made by Dell and van Melkebeek [12], who showed, amongst others, that FEEDBACK VERTEX SET does not admit a kernelization to size $\mathcal{O}(k^{2-\epsilon})$ unless the polynomial hierarchy collapses, i.e., there may be a kernelization with $\mathcal{O}(k)$ vertices but the number of $\mathcal{O}(k^2)$ edges is essentially optimal. Another interesting recent development are meta results for kernelization due to Bodlaender et al. [6] and Fomin et al. [20]. They obtain linear and polynomial kernels for graph problems definable in extensions of monadic second order logic when restricted to planar, bounded genus, or H-minor-free graphs, given certain additional properties like finite integer index or quasi-compactness. Furthermore, two recent papers obtain complete classifications of three parameterized constraint satisfaction problems into admitting or not admitting polynomial kernels depending on the language of permitted constraints [29, 30]. For more background on kernelization we refer to the recent surveys on kernelization given by Guo and Niedermeier [23] as well as by Bodlaender [4]. In an earlier paper, Mahajan et al. [31, 32] studied MAX SNP problems and observe that kernelizations follow from the fact that NP-hard problems in MAX SNP have guaranteed lower bounds for the optimum value, motivating them to study these problems parameterized above such lower bounds. Cai and Huang [9] showed that all problems in MAX SNP admit fixed-parameter approximation schemes.

MIN $F^+\Pi_1$ and MAX NP Two decades ago Papadimitriou and Yannakakis [37] initiated the syntactic study of optimization problems to extend the understanding of approximability. They introduced the classes MAX NP and MAX SNP as natural variants of NP based on Fagin's [16] syntactic characterization of NP. Essentially problems are in MAX NP or MAX SNP if their optimum value can be expressed as the maximum number of tuples for which some existential, respectively quantifier-free, first-order formula holds. They showed that every problem in these two classes is approximable to within a constant factor of the optimum. Arora et al. complemented this by proving that no MAX SNP-complete problem has a polynomial-time approximation scheme, unless P = NP [2]. Contained in MAX SNP there are some well-known maximization problems, such as MAX CUT, MAX q-SAT, and INDEPENDENT SET on graphs of bounded degree. Its superclass MAX NP also contains MAX SAT amongst others.

Kolaitis and Thakur generalized the approach of examining the logical definability of optimization problems and defined further classes of minimization and maximization problems [26, 27]. Amongst others they introduced the class MIN $\mathrm{F}^+\Pi_1$ of problems whose optimum can be expressed as the minimum weight of an assignment (i.e., number of ones) that satisfies a certain universal first-order formula. They proved that every problem in MIN $\mathrm{F}^+\Pi_1$ is approximable to within a constant factor of the optimum. In MIN $\mathrm{F}^+\Pi_1$ there are problems like VERTEX COVER, d-HITTING SET, and TRIANGLE EDGE DELETION.

Organization of the Paper Section 2 covers the definitions of the classes MIN $\mathrm{F}^+\Pi_1$ and MAX NP, as well as the necessary details from parameterized complexity. In Sects. 3 and 4 we present polynomial kernelizations for the standard parameterizations of problems in MIN $\mathrm{F}^+\Pi_1$ and MAX NP respectively. Section 5 summarizes our results and poses some open problems.

2 Preliminaries

Logic and Complexity Classes A (relational) vocabulary is a set σ of relation symbols, each having some fixed integer as its arity. Atomic formulas over σ are of the form $R(z_1, \dots, z_t)$ where R is a t-ary relation symbol from σ and the z_i are variables. A *literal* is an atomic formula or the negation of an atomic formula. The set of quantifier-free (relational) formulas over σ is the closure of the set of all atomic formulas under negation, conjunction, and disjunction. A formula in *conjunctive normal form* is a conjunction of disjunctions of literals, called *clauses*. A formula in *disjunctive normal form* is a disjunction of conjunctions of literals, called *disjuncts*.

Definition 1 (MIN $\mathrm{F}^+\Pi_1$, MAX NP) A tuple $\mathcal{A} = (A, R_1, \dots, R_t)$ where A is a finite set and each R_i is an r_i-ary relation over A is called a *finite structure of type* $(r_1, \dots, r_t)$.

Let $\mathcal{Q}$ be an optimization problem on finite structures of type $(r_1, \dots, r_t)$.

(a) The problem $\mathcal{Q}$ is contained in MIN $\mathrm{F}^+\Pi_1$ if its optimum on finite structures $\mathcal{A} = (A, R_1, \dots, R_t)$ of type $(r_1, \dots, r_t)$ can be expressed as

$$\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) = \min_S\{|S| : (\mathcal{A}, S) \models (\forall \mathbf{x} \in A^{c_x}) : \psi(\mathbf{x}, S)\},$$

where S is a single relation symbol and $\psi(\mathbf{x}, S)$ is a quantifier-free formula in conjunctive normal form over the vocabulary $\{R_1, \dots, R_t, S\}$ on variables $\{x_1, \dots, x_{c_x}\}$. Furthermore, $\psi(\mathbf{x}, S)$ is positive in S, i.e., S does not occur negated in $\psi(\mathbf{x}, S)$.

(b) The problem $\mathcal{Q}$ is contained in MAX NP if its optimum on finite structures $\mathcal{A} = (A, R_1, \dots, R_t)$ of type $(r_1, \dots, r_t)$ can be expressed as

$$\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) = \max_{\mathcal{S}} \left| \left\{ \mathbf{x} \in A^{c_x} : (\mathcal{A}, \mathcal{S}) \models (\exists \mathbf{y} \in A^{c_y}) : \psi(\mathbf{x}, \mathbf{y}, \mathcal{S}) \right\} \right|,$$

where $\mathcal{S} = (S_1, \dots, S_u)$ is a tuple of s_i-ary relation symbols S_i and $\psi(\mathbf{x}, \mathbf{y}, \mathcal{S})$ is a quantifier-free formula in disjunctive normal form over the vocabulary $\{R_1, \dots, R_t, S_1, \dots, S_u\}$ on variables $\{x_1, \dots, x_{c_x}, y_1, \dots, y_{c_y}\}$.

The definition of MAX SNP is similar to that of MAX NP but without the existential quantification of $\mathbf{y}$, i.e., $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) = \max_{\mathcal{S}} |\{\mathbf{x} : (\mathcal{A}, \mathcal{S}) \models \psi(\mathbf{x}, \mathcal{S})\}|$.

Remark 1 Since the formulas ψ depend only on the problem $\mathcal{Q}$ they are of constant length with respect to inputs $\mathcal{A}$. Thus there is no strict need to require normal forms, but the chosen ones fit the quantification nicely, e.g., we can view $(\forall \mathbf{x}) : \psi(\mathbf{x}, S)$ as a large conjunctive normal form.

Example 1 (MINIMUM VERTEX COVER) Let $G = (V, E)$ be a finite structure of type (2) that represents a graph by a set V of vertices and a binary relation E over V as its edges. The optimum of MINIMUM VERTEX COVER on structures G can be expressed as:

$$\mathrm{opt}_{VC}(G) = \min_{S \subseteq V} \{|S| : (G, S) \models (\forall (u, v) \in V^2) : (\neg E(u, v) \vee S(u) \vee S(v))\}.$$

This implies that MINIMUM VERTEX COVER is contained in MIN $\mathrm{F}^+\Pi_1$.

Example 2 (MAXIMUM SATISFIABILITY) Formulas in conjunctive normal form can be represented by finite structures $\mathcal{F} = (F, P, N)$ of type (2, 2): Let F be the set of all clauses and variables, and let P and N be binary relations over F. Let $P(x, c)$ be true if and only if x is a literal of the clause c and let $N(x, c)$ be true if and only if $\neg x$ is a literal of the clause c. The optimum of MAX SAT on structures $\mathcal{F}$ can be expressed as:

$$\mathrm{opt}_{MS}(\mathcal{F}) = \max_{T \subseteq F} |\{c \in F : (\mathcal{F}, T) \models (\exists x \in F) : (P(x, c) \wedge T(x)) \vee (N(x, c) \wedge \neg T(x))\}|.$$

Thus MAX SAT is contained in MAX NP.

For a detailed introduction to MIN $\mathrm{F}^+\Pi_1$, MAX NP, and MAX SNP we refer the reader to [26, 27, 37]. An introduction to logic and complexity can be found in [36].

Parameterized Complexity Parameterized complexity provides a multivariate analysis of combinatorially hard problems, considering at least one additional *parameter* of input instances apart from their size. This allows a more fine-grained analysis of the required runtimes than the mere statement of NP-hardness could provide. In the following we give the necessary formal definitions, namely fixed-parameter tractability, standard parameterizations, and kernelization.

Definition 2 (Parameterized problem, Fixed-parameter tractability) A *parameterized problem* $p\text{-}\mathcal{Q}$ over the alphabet Σ is a subset of $\Sigma^* \times \mathbb{N}$; the second component is called the *parameter*.

A parameterized problem $p\text{-}\mathcal{Q}$ is *fixed-parameter tractable* if there exists an algorithm $\mathbb{A}$, a polynomial p, and a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathbb{A}$ decides $(x, k) \in p\text{-}\mathcal{Q}$ in time $f(k) \cdot p(|x|)$. FPT is the class of all fixed-parameter tractable problems.

Definition 3 (Standard parameterization) Let $\mathcal{Q}$ be a maximization (minimization) problem. Its standard parameterization is defined as $p\text{-}\mathcal{Q} := \{(\mathcal{A}, k) \mid \text{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k\}$ (respectively $p\text{-}\mathcal{Q} := \{(\mathcal{A}, k) \mid \text{opt}_{\mathcal{Q}}(\mathcal{A}) \leq k\}$ for minimization problems).

Basically, the standard parameterization of an optimization problem is its decision version, asking whether the optimum is at least k (respectively at most k), parameterized by k.

Definition 4 (Kernelization) Let $p\text{-}\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem over Σ. A polynomial-time computable function $K : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ is a *kernelization* of $p\text{-}\mathcal{Q}$ if there is a computable function $h : \mathbb{N} \to \mathbb{N}$ such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$, and letting $(x', k') := K((x, k))$, we have

1. $(x, k) \in p\text{-}\mathcal{Q} \Leftrightarrow (x', k') \in p\text{-}\mathcal{Q}$ as well as
2. $|x'| \leq h(k)$ and $k' \leq h(k)$.

We call h the *size* of the *problem kernel* (x', k'). The kernelization K is *polynomial* if h is a polynomial. We say that $p\text{-}\mathcal{Q}$ *admits a (polynomial) kernelization* if there exists a (polynomial) kernelization of $p\text{-}\mathcal{Q}$.

Essentially, a kernelization is a polynomial-time data reduction that comes with a guaranteed upper bound on the size of the resulting instance in terms of the parameter.

For an introduction to parameterized complexity we refer the reader to [14, 19, 35].

Hypergraphs and Sunflowers A *hypergraph* is a tuple $\mathcal{H} = (V, E)$ consisting of a finite set V, its vertices, and a family E of subsets of V, its edges. A hypergraph has *dimension* d if each edge has cardinality at most d. A hypergraph is d*-uniform* if each edge has cardinality exactly d.

Definition 5 (Sunflower) Let $\mathcal{H}$ be a hypergraph. A *sunflower* of cardinality r is a set $F = \{f_1, \dots, f_r\}$ of edges of $\mathcal{H}$ such that every pair has the same intersection C, i.e., for all $1 \leq i < j \leq r$: $f_i \cap f_j = C$. The set C is called the *core* of the sunflower, the disjoint sets $f_i \setminus C$ are called *petals*.

The following lemma is the beautiful Sunflower Lemma due to Erdős and Rado [15].

Lemma 1 (Sunflower Lemma) *Let $k, d \in \mathbb{N}$ and let $\mathcal{H}$ be a d-uniform hypergraph with more than $k^d \cdot d!$ edges. Then there is a sunflower of cardinality $k+1$ in $\mathcal{H}$. For every fixed d there is an algorithm that computes such a sunflower in time polynomial in $|E(\mathcal{H})|$.*

We give a short sketch of its algorithmic proof. The idea is to greedily select disjoint sets. If at least $k+1$ sets are found then they form a sunflower with core $C = \emptyset$. Otherwise all other sets must intersect the at most dk elements of the selected sets. Then the search continues among those sets that contain the most frequent element, i.e., occurring in at least $|E(\mathcal{H})|/dk$ sets. This terminates after $d-1$ rounds since each time an element is selected for the core, which contains at most $d-1$ elements.

The following corollary is an immediate extension to d-dimensional hypergraphs.

Corollary 1 *The same holds for d-dimensional hypergraphs with more than $k^d \cdot d! \cdot d$ edges.*

Proof For some $d' \in \{1, \dots, d\}$, $\mathcal{H}$ has more than $k^d \cdot d! \geq k^{d'} \cdot d'!$ edges of cardinality d'. Let $\mathcal{H}_{d'}$ be the d'-uniform subgraph induced by the edges of cardinality d'. We apply the Sunflower Lemma on $\mathcal{H}_{d'}$ and obtain a sunflower F of cardinality $k+1$ in time polynomial in $|E(\mathcal{H}_{d'})| \leq |E(\mathcal{H})|$. Clearly F is also a sunflower of $\mathcal{H}$. □

3 Polynomial Kernelization for MIN $F^+\Pi_1$

The class MIN $F^+\Pi_1$ was introduced by Kolaitis and Thakur in a framework of syntactically defined classes of optimization problems [26]. In a follow-up paper they showed that every problem in MIN $F^+\Pi_1$ is constant-factor approximable [27]. We will prove that the standard parameterization of any problem in MIN $F^+\Pi_1$ admits a polynomial kernelization.

Let us fix some optimization problem $\mathcal{Q}$ from MIN $F^+\Pi_1$ that takes as input finite structures of type $(r_1, \dots, r_t)$. Accordingly let $R_1, \dots, R_t$ be relation symbols of arity $r_1, \dots, r_t$. Since $\mathcal{Q} \in$ MIN $F^+\Pi_1$ there is a c_S-ary relation symbol S and a quantifier-free formula $\psi(\mathbf{x}, S)$ in conjunctive normal form such that:

1. the formula $\psi(\mathbf{x}, S)$ is positive in S, i.e., there are no literals $\neg S(x_1, \dots, x_{c_S})$ and
2. the optimum value of $\mathcal{Q}$ on input $\mathcal{A}$ of type $(r_1, \dots, r_t)$ can be expressed as

$$\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) = \min_{S \subseteq A^{c_S}} \{|S| : (\mathcal{A}, S) \models (\forall \mathbf{x} \in A^{c_x}) : \psi(\mathbf{x}, S)\}.$$

We denote by s the maximum number of occurrences of S in any clause of $\psi(\mathbf{x}, S)$. This value plays a crucial role in our kernelization bound. For the polynomial kernelization we consider the standard parameterization of $\mathcal{Q}$, denoted by p-$\mathcal{Q}$:

Input: A finite structure $\mathcal{A}$ of type $(r_1, \dots, r_t)$ and an integer k.
Parameter: k.
Task: Decide whether $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) \leq k$.

We will see that, given an instance $(\mathcal{A}, k)$, deciding whether $\text{opt}_{\mathcal{Q}}(\mathcal{A}) \leq k$ is equivalent to deciding an instance of s-HITTING SET, defined as follows:

Input: A hypergraph $\mathcal{H} = (V, E)$ of dimension s and an integer k.
Parameter: k.
Task: Decide whether $\mathcal{H}$ has a hitting set of size at most k, i.e., $S \subseteq V$, $|S| \leq k$, such that S has a nonempty intersection with every edge of $\mathcal{H}$.

The following definition formalizes the procedure of plugging in a specific tuple $\mathbf{x} \in A^{c_x}$ into $\psi(\mathbf{x}, S)$. That way all occurrences of relation symbols R_i can be evaluated, as they are part of the input $(\mathcal{A}, k)$, leaving only literals $S(\cdot)$.

Definition 6 Let $\mathcal{A} = (A, R_1, \ldots, R_t)$ be a finite structure of type $(r_1, \ldots, r_t)$ and let $\mathbf{x} \in A^{c_x}$. We define $\psi_{\mathbf{x}}(S)$ to be the formula obtained in the following way:

1. Replace all variables $x_1, \ldots, x_{c_x}$ by the chosen elements of A.
2. Replace all literals $R_i(\mathbf{z})$ and $\neg R_i(\mathbf{z})$ by 1 (true) or 0 (false) depending on whether $\mathbf{z}$ is contained in R_i (note that $\mathbf{z}$ is a concrete tuple from A^{r_i} by Step 1).
3. Delete all clauses that contain a 1 and delete all occurrences of 0.

Observe that application of Definition 6 yields an equivalent formula in the sense that $(\mathcal{A}, S) \models \psi(\mathbf{x}, S)$ if and only if $(\mathcal{A}, S) \models \psi_{\mathbf{x}}(S)$, since we only replace literals according to the input. It is easy to see that $\psi_{\mathbf{x}}(S)$ is a formula in conjunctive normal form on literals $S(\mathbf{z})$ for some $\mathbf{z} \in A^{c_S}$; there are at most s literals per clause. A formula $\psi_{\mathbf{x}}(S)$ can have empty clauses when all literals $R_i(\cdot)$, $\neg R_i(\cdot)$ in a clause are evaluated to 0 and there are no literals $S(\cdot)$. In that case, no assignment to S can satisfy the formula $\psi_{\mathbf{x}}(S)$, or equivalently $\psi(\mathbf{x}, S)$. Thus $(\mathcal{A}, k)$ is a no-instance and we may reject it or return a dummy no-instance of constant size. Note that clauses of $\psi_{\mathbf{x}}(S)$ cannot contain contradicting literals since $\psi(\mathbf{x}, S)$ is positive in S. Henceforth we assume all clauses of formulas $\psi_{\mathbf{x}}(S)$ to be nonempty.

We continue by defining a mapping Φ from finite structures $\mathcal{A}$ to hypergraphs $\mathcal{H}$. Then we show that $(\mathcal{A}, k)$ is a yes-instance for p-$\mathcal{Q}$ if and only if $(\Phi(\mathcal{A}), k)$ is a yes-instance for s-HITTING SET.

Definition 7 Let $\mathcal{A}$ be an instance of $\mathcal{Q}$. We define $\Phi(\mathcal{A}) := \mathcal{H}$ with $\mathcal{H} = (V, E)$. We let E be the family of all sets $e = \{\mathbf{z}_1, \ldots, \mathbf{z}_p\}$ such that $(S(\mathbf{z}_1) \vee \cdots \vee S(\mathbf{z}_p))$ is a clause of a $\psi_{\mathbf{x}}(S)$ for some $\mathbf{x} \in A^{c_x}$. We let V be the union of all sets $e \in E$.

The hypergraphs $\mathcal{H}$ obtained from the mapping Φ have dimension s since each $\psi_{\mathbf{x}}(S)$ has at most s literals per clause. The following lemma establishes the equivalence of $(\mathcal{A}, k)$ and $(\mathcal{H}, k) = (\Phi(\mathcal{A}), k)$.

Lemma 2 *Let $\mathcal{A} = (A, R_1, \ldots, R_t)$ be a finite structure of type $(r_1, \ldots, r_t)$ and let k be an integer. Then $(\mathcal{A}, k)$ is a yes-instance of p-$\mathcal{Q}$ if and only if $(\Phi(\mathcal{A}), k)$ is a yes-instance of s-HITTING SET.*

Proof It suffices to show that for all $S \subseteq A^{c_S}$:

$$(\mathcal{A}, S) \models (\forall \mathbf{x} \in A^{c_x}) : \psi(\mathbf{x}, S) \quad \text{if and only if } S \text{ is a hitting set for } \Phi(\mathcal{A}).$$

Let $\mathcal{H} = \Phi(\mathcal{A}) = (V, E)$ and let $S \subseteq A^{c_S}$:

$$(\mathcal{A}, S) \models (\forall \mathbf{x} \in A^{c_x}) : \psi(\mathbf{x}, S)$$

$$\iff \quad (\mathcal{A}, S) \models (\forall \mathbf{x} \in A^{c_x}) : \psi_{\mathbf{x}}(S)$$

$$\iff \quad (\forall \mathbf{x} \in A^{c_x}) : \text{each clause of } \psi_{\mathbf{x}}(S) \text{ has a literal } S(\mathbf{z}) \text{ for which } \mathbf{z} \in S$$

$$\iff \quad S \text{ has a nonempty intersection with every set } e \in E$$

$$\iff \quad S \text{ is a hitting set for } \mathcal{H} = (V, E).$$

Since the number of ones in the assignment to S, i.e., the number of tuples $\mathbf{z} \in A^{c_S}$ with $S(\mathbf{z}) = 1$, translates directly to the cardinality of the hitting set and vice versa, the lemma follows. □

Our kernelization will consist of the following three steps:

1. Map the given instance $(\mathcal{A}, k)$ for p-$\mathcal{Q}$ to an equivalent instance $(\mathcal{H}, k) = (\Phi(\mathcal{A}), k)$ for s-HITTING SET according to Definition 7 and Lemma 2.
2. Use a polynomial kernelization for s-HITTING SET on $(\mathcal{H}, k)$ to obtain an equivalent instance $(\mathcal{H}', k)$ with size polynomial in k.
3. Use $(\mathcal{H}', k)$ to derive an equivalent instance $(\mathcal{A}', k)$ of p-$\mathcal{Q}$. That way we will be able to conclude that $(\mathcal{A}', k)$ is equivalent to $(\mathcal{H}, k)$ and hence also to $(\mathcal{A}, k)$.

There are two kernelizations for s-HITTING SET: one by Flum and Grohe [19] based on the Sunflower Lemma due to Erdős and Rado [15] and a recent one by Abu-Khzam [1] based on crown decompositions. For our purpose of deriving an equivalent instance for p-$\mathcal{Q}$, these kernelizations have the drawback of shrinking sets during the reduction, since we need to find an equivalent instance of p-$\mathcal{Q}$ afterwards. To shrink edges we would need to shrink clauses of the formula $\psi(\mathbf{x}, S)$, but we may only change the instance $(\mathcal{A}, k)$. Fortunately we are able to modify Flum and Grohe's kernelization to use only edge deletions.

Remark 2 Crown decompositions frequently produce the strongest kernelization results by virtue of proving certain decisions to be optimal, usually independent of the solution size k. Kernelization based on sunflowers makes use of the solution size, showing that certain decisions are forced.

The sunflower-based kernelization for s-HITTING SET uses the fact that a sunflower of cardinality greater than k forces an element of its core to be selected; recall that the petals are pairwise disjoint. Thus such a sunflower may be replaced by its core. In our case the idea is to shrink sunflowers from size at least $k + 2$ down to size $k + 1$. This way the selection of an element from the core is still forced, but we are able to reduce the size of our instance without shrinking of edges.

Theorem 1 *There exists a polynomial kernelization of s-*HITTING SET *that, given an instance $(\mathcal{H}, k)$, computes an instance $(\mathcal{H}^*, k)$ such that $E(\mathcal{H}^*) \subseteq E(\mathcal{H})$, $\mathcal{H}^*$ has $\mathcal{O}(k^s)$ edges, and the size of $(\mathcal{H}^*, k)$ is $\mathcal{O}(k^s)$ as well.*

Proof Let $(\mathcal{H}, k)$ be an instance of s-HITTING SET, with $\mathcal{H} = (V, E)$. If $\mathcal{H}$ contains a sunflower $F = \{f_1, \ldots, f_{k+1}\}$ of cardinality $k + 1$ then every hitting set of $\mathcal{H}$ must have a nonempty intersection with the core C of F or with the $k + 1$ disjoint sets $f_1 \setminus C, \ldots, f_{k+1} \setminus C$. Thus every hitting set of at most k elements must have a nonempty intersection with C.

Now consider a sunflower $F = \{f_1, \ldots, f_{k+1}, f_{k+2}\}$ of cardinality $k + 2$ in $\mathcal{H}$ and let $\mathcal{H}' = (V, E \setminus \{f_{k+2}\})$. We show that the instances $(\mathcal{H}, k)$ and $(\mathcal{H}', k)$ are equivalent. Clearly every hitting set for $\mathcal{H}$ is also a hitting set for $\mathcal{H}'$ since $E(\mathcal{H}') \subseteq E(\mathcal{H})$. Let $S \subseteq V$ be a hitting set of size at most k for $\mathcal{H}'$. Since $F \setminus \{f_{k+2}\}$ is a sunflower of cardinality $k + 1$ in $\mathcal{H}'$, it follows that S has a nonempty intersection with its core C. Hence S has a nonempty intersection with $f_{k+2} \supseteq C$ too. Thus S is a hitting set of size at most k for $\mathcal{H}$, implying that $(\mathcal{H}, k)$ and $(\mathcal{H}', k)$ are equivalent.

We turn this fact into a kernelization, by starting with $\mathcal{H}^* = \mathcal{H}$ and by repeating the following step while $\mathcal{H}^*$ has more than $(k + 1)^s \cdot s! \cdot s$ edges. By Corollary 1 we obtain a sunflower of cardinality $k + 2$ in $\mathcal{H}^*$ in time polynomial in $|E(\mathcal{H}^*)|$. We delete an edge of the detected sunflower from the edge set of $\mathcal{H}^*$, thereby reducing the cardinality of the sunflower to $k + 1$. Thus, by the argument from the previous paragraph, we maintain that $(\mathcal{H}, k)$ and $(\mathcal{H}^*, k)$ are equivalent. Furthermore $E(\mathcal{H}^*) \subseteq E(\mathcal{H})$ and $\mathcal{H}^*$ has no more than $(k + 1)^s \cdot s! \cdot s \in \mathcal{O}(k^s)$ edges. Since we delete an edge of $\mathcal{H}^*$ in each step, there are $\mathcal{O}(|E(\mathcal{H})|)$ steps, and the total time is polynomial in $|E(\mathcal{H})|$. Deleting all isolated vertices from $\mathcal{H}^*$ yields a size of $\mathcal{O}(s \cdot k^s) = \mathcal{O}(k^s)$ since each edge contains at most s vertices. □

The following lemma proves that every s-HITTING SET instance that is "sandwiched" between two equivalent instances must be equivalent to both.

Lemma 3 *Let $(\mathcal{H}, k)$ be an instance of s-HITTING SET and let $(\mathcal{H}^*, k)$ be an equivalent instance with $E(\mathcal{H}^*) \subseteq E(\mathcal{H})$. Then for any $\mathcal{H}'$ with $E(\mathcal{H}^*) \subseteq E(\mathcal{H}') \subseteq E(\mathcal{H})$ the instance $(\mathcal{H}', k)$ is equivalent to $(\mathcal{H}, k)$ and $(\mathcal{H}^*, k)$.*

Proof Observe that hitting sets for $\mathcal{H}$ can be projected to hitting sets for $\mathcal{H}'$ (i.e., restricted to the vertex set of $\mathcal{H}'$) since $E(\mathcal{H}') \subseteq E(\mathcal{H})$. Thus if $(\mathcal{H}, k)$ is a yes-instance then $(\mathcal{H}', k)$ is a yes-instance too. The same argument holds for $(\mathcal{H}', k)$ and $(\mathcal{H}^*, k)$. Together with the fact that $(\mathcal{H}, k)$ and $(\mathcal{H}^*, k)$ are equivalent, this proves the lemma. □

Now we are well equipped to prove that p-$\mathcal{Q}$ admits a polynomial kernelization. The main remaining difficulty lies in finding an instance of p-$\mathcal{Q}$ that is equivalent to the kernelized s-HITTING SET instance that we already know how to obtain. It is in fact easier to find an instance of p-$\mathcal{Q}$ that is equivalent to a sandwiched instance.

Theorem 2 *Let $\mathcal{Q} \in \text{MIN F}^+\Pi_1$. The standard parameterization p-$\mathcal{Q}$ of $\mathcal{Q}$ admits a polynomial kernelization.*

Proof Let $(\mathcal{A}, k)$ be an instance of p-$\mathcal{Q}$. By Lemma 2 we have that $(\mathcal{A}, k)$ is a yes-instance of p-$\mathcal{Q}$ if and only if $(\mathcal{H}, k) = (\Phi(\mathcal{A}), k))$ is a yes-instance of s-HITTING

SET. We apply the kernelization from Theorem 1 to $(\mathcal{H}, k)$ and obtain an equivalent s-HITTING SET instance $(\mathcal{H}^*, k)$ such that $E(\mathcal{H}^*) \subseteq E(\mathcal{H})$ and $\mathcal{H}^*$ has $\mathcal{O}(k^s)$ edges.

Recall that every edge of $\mathcal{H}$, say $\{\mathbf{z}_1, \dots, \mathbf{z}_p\}$, corresponds to a clause $(S(\mathbf{z}_1) \vee \dots \vee S(\mathbf{z}_p))$ of $\psi_{\mathbf{x}}(S)$ for some $\mathbf{x} \in A^{c_x}$. Thus for each edge $e \in E(\mathcal{H}^*) \subseteq E(\mathcal{H})$ we can select a tuple $\mathbf{x}_e$ such that e corresponds to a clause of $\psi_{\mathbf{x}_e}(S)$. Let X be the set of the selected tuples $\mathbf{x}_e$ for all edges $e \in E(\mathcal{H}^*)$. Let $A' \subseteq A$ be the set of all components of tuples $\mathbf{x}_e \in X$, ensuring that $X \subseteq A'^{c_x}$. Let R'_i be the restriction of R_i to A' and let $\mathcal{A}' = (A', R'_1, \dots, R'_t)$.

Let $(\mathcal{H}', k) = (\Phi(\mathcal{A}'), k)$. By definition of Φ and by construction of $\mathcal{H}'$ we know that $E(\mathcal{H}^*) \subseteq E(\mathcal{H}') \subseteq E(\mathcal{H})$ since $X \subseteq A'^{c_x} \subseteq A^{c_x}$. Thus, by Lemma 3, we have that $(\mathcal{H}', k)$ is equivalent to $(\mathcal{H}, k)$. Furthermore, by Lemma 2, $(\mathcal{H}', k)$ is a yes-instance of s-HITTING SET if and only if $(\mathcal{A}', k)$ is a yes-instance of p-$\mathcal{Q}$. Thus $(\mathcal{A}', k)$ and $(\mathcal{A}, k)$ are equivalent instances of p-$\mathcal{Q}$.

We conclude the proof by giving an upper bound on the size of $(\mathcal{A}', k)$ that is polynomial in k. The set X contains at most $|E(\mathcal{H}^*)| \in \mathcal{O}(k^s)$ tuples. These tuples have no more than $c_x \cdot |E(\mathcal{H}^*)|$ different components. Hence the size of A' is $\mathcal{O}(c_x \cdot k^s) = \mathcal{O}(k^s)$. Thus the size of $(\mathcal{A}', k)$ is $\mathcal{O}(k^{sm})$, where m is the largest arity of a relation R_i, i.e., $m = \max\{r_1, \dots, r_t\}$. Thus $(\mathcal{A}', k)$ is an instance equivalent to $(\mathcal{A}, k)$ with size polynomial in k, since c_x, s, and m are constants independent of the input. □

4 Polynomial Kernelization for MAX NP

Papadimitriou and Yannakakis introduced MAX SNP as well as its superclass MAX NP and showed that every problem from these classes is constant-factor approximable [37]. We show that the standard parameterization of any MAX NP problem admits a polynomial kernelization.

Again let us fix some problem $\mathcal{Q} \in$ MAX NP. Let $(r_1, \dots, r_t)$ be the type of input structures for $\mathcal{Q}$ and let $R_1, \dots, R_t$ be matching relation symbols. By definition of MAX NP there is a tuple of relation symbols $\mathcal{S} = (S_1, \dots, S_u)$ of arity $s_1, \dots, s_u$ and a formula $\psi(\mathbf{x}, \mathbf{y}, \mathcal{S})$ in disjunctive normal form over the vocabulary $\{R_1, \dots, R_t, S_1, \dots, S_u\}$ such that for all finite structures $\mathcal{A}$ of type $(r_1, \dots, r_t)$ the optimum value of $\mathcal{Q}$ on input $\mathcal{A}$ can be expressed as

$$\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) = \max_{\mathcal{S}} |\{\mathbf{x} \in A^{c_x} : (\mathcal{A}, \mathcal{S}) \models (\exists \mathbf{y} \in A^{c_y}) : \psi(\mathbf{x}, \mathbf{y}, \mathcal{S})\}|.$$

Let s be the maximum number of occurrences of relations $S_1, \dots, S_u$ in any disjunct of $\psi(\mathbf{x}, \mathbf{y}, \mathcal{S})$. The standard parameterization p-$\mathcal{Q}$ of $\mathcal{Q}$ is the following problem:

Input: A finite structure $\mathcal{A}$ of type $(r_1, \dots, r_t)$ and an integer k.
Parameter: k.
Task: Decide whether $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k$.

We define formulas $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ similarly to Definition 6 in Sect. 3.

Definition 8 Let $\mathcal{A} = (A, R_1, \dots, R_t)$ be a finite structure of type $(r_1, \dots, r_t)$, let $\mathbf{x} \in A^{c_x}$, and let $\mathbf{y} \in A^{c_y}$. We define $\psi_{\mathbf{x},\mathbf{y}}(S)$ to be the formula obtained by the following steps:

1. Replace all variables $x_1, \dots, x_{c_x}, y_1, \dots, y_{c_y}$ by the chosen elements of A.
2. Replace all literals $R_i(\mathbf{z})$ and $\neg R_i(\mathbf{z})$, for some $\mathbf{z} \in A^{r_i}$, by 1 (true) or 0 (false) depending on whether $\mathbf{z}$ is contained in R_i.
3. Delete all disjuncts that contain a 0 and delete all occurrences of 1; note the difference to Definition 6 through using a different normal form.
4. Delete all disjuncts that contain contradicting literals $S_j(\mathbf{z}), \neg S_j(\mathbf{z})$ since they cannot be satisfied.

We explicitly allow empty disjuncts that are satisfied by definition for the sake of simplicity (they occur when all literals in a disjunct are evaluated to 1).

It is easy to see that $\psi(\mathbf{x}, \mathbf{y}, \mathcal{S})$ and $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ are equivalent for any choice of $\mathbf{x}$, $\mathbf{y}$, and $\mathcal{S}$, i.e., $(\mathcal{A}, \mathcal{S}) \models \psi(\mathbf{x}, \mathbf{y}, \mathcal{S})$ iff $(\mathcal{A}, \mathcal{S}) \models \psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$. Moreover, we can compute all formulas $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ for $\mathbf{x} \in A^{c_x}$, $\mathbf{y} \in A^{c_y}$ in polynomial time, since c_x, c_y, and the length of $\psi(\mathbf{x}, \mathbf{y}, \mathcal{S})$ are constants independent of $\mathcal{A}$.

Definition 9 Let $\mathcal{A} = (A, R_1, \dots, R_t)$ be a finite structure of type $(r_1, \dots, r_t)$.

(a) We define $X_{\mathcal{A}} \subseteq A^{c_x}$ as the set of all tuples $\mathbf{x}$ such that $(\exists \mathbf{y}) : \psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ holds for some $\mathcal{S}$:

$$X_{\mathcal{A}} = \{\mathbf{x} : (\exists \mathcal{S}) : (\mathcal{A}, \mathcal{S}) \models (\exists \mathbf{y}) : \psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})\}.$$

(b) For $\mathbf{x} \in A^{c_x}$ we define $Y_{\mathcal{A}}(\mathbf{x})$ as the set of all tuples $\mathbf{y}$ such that $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ holds for some $\mathcal{S}$:

$$Y_{\mathcal{A}}(\mathbf{x}) = \{\mathbf{y} : (\exists \mathcal{S}) : (\mathcal{A}, \mathcal{S}) \models \psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})\}.$$

The sets $X_{\mathcal{A}}$ and $Y_{\mathcal{A}}(\mathbf{x})$ can be computed in polynomial time because the number of tuples $\mathbf{x} \in A^{c_x}$ respectively $\mathbf{y} \in A^{c_y}$ is polynomial in the size of $\mathcal{A}$ and the formula $\psi(\mathbf{x}, \mathbf{y}, \mathcal{S})$ is of constant length independent of $\mathcal{A}$.

Lemma 4 *Let $(\mathcal{A}, k)$ be an instance of p-$\mathcal{Q}$. If $|X_{\mathcal{A}}| \geq k \cdot 2^s$ then* $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k$, *i.e., $(\mathcal{A}, k)$ is a yes-instance.*

Remark 3 In the following proof we consider assignments to variables of the formulas $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$. We point out that assigning true or false to some variable $S_i(\mathbf{z})$ corresponds to including or excluding, respectively, the tuple $\mathbf{z}$ in S_i. Note that there are $\sum_{i=1}^{u} |A|^{s_i}$ variables, one for each possible tuple of a relation S_i of arity s_i.

Proof of Lemma 4 We follow Papadimitriou and Yannakakis' [37] proof for the fact that all problems in MAX NP are constant-factor approximable. For each $\mathbf{x} \in X_{\mathcal{A}}$ we fix a tuple $\mathbf{y} \in Y_{\mathcal{A}}(\mathbf{x})$ such that $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ is satisfiable. This yields $m = |X_{\mathcal{A}}|$ formulas, say $\psi_1, \dots, \psi_m$. Now, for each formula ψ_i let f_i denote the fraction of all assignments to $\mathcal{S}$ (i.e., inclusion or exclusion of tuples $\mathbf{z}$ in the relations S_j) that satisfies ψ_i.

We will create an assignment that satisfies at least $\sum f_i$ formulas ψ_i. Let y be a variable that has not been assigned yet. We assume that ℓ variables are unassigned at that point and that $\sum f_i' \geq \sum f_i$, where the fractions f_i' are with respect to assignments to these ℓ remaining variables. For $i \in \{1, \dots, m\}$, let p_i and n_i denote the fraction of assignments to the remaining variables that satisfies ψ_i in which y is set to true or false, respectively. Thus there are $2^\ell(p_i + n_i)$ assignments which satisfy ψ_i. Assign true to y if $\sum p_i \geq \sum n_i$; else, assign false. We show that the sum of fractions f_i' never decreases (always taking f_i' to be with respect to the remaining unassigned variables): If y is set to true, then $2^\ell p_i$ assignments to the other $\ell - 1$ variables satisfy ψ_i, which corresponds to a fraction of $2^\ell p_i / 2^{\ell-1} = 2p_i$. Thus if $\sum p_i \geq \sum n_i$ then

$$\sum_{i=1}^{m} 2p_i \geq \sum_{i=1}^{m} p_i + \sum_{i=1}^{m} n_i \geq \sum_{i=1}^{m} f_i' \geq \sum_{i=1}^{m} f_i.$$

Note that $\sum_{i=1}^{m} 2p_i$ is the sum of fractions of satisfying assignments taken with respect to the remaining $\ell - 1$ variables. Similarly for the case that $\sum p_i < \sum n_i$ and y is assigned false. Thus the sum of fractions never decreases.

When all variables are assigned a value, f_i' is equal to 1 if ψ_i is satisfied and 0 else. Thus, this assignment satisfies at least $\sum f_i' \geq \sum f_i$ formulas ψ_i (recall that each satisfied formula contributes a tuple to the solution).

It is easy to see that $f_i \geq 2^{-s}$ for each formula ψ_i. Since ψ_i is satisfiable there exists a satisfiable disjunct. To satisfy a disjunct of at most s literals, at most s variables need to be assigned accordingly. Since the assignment to all other variables can be arbitrary this implies that $f_i \geq 2^{-s}$. Thus we have that $\sum f_i \geq m \cdot 2^{-s}$. Therefore $|X_{\mathcal{A}}| = m \geq k \cdot 2^s$ implies that the assignment satisfies at least k formulas, i.e., that $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k$. □

Henceforth we assume that $|X_{\mathcal{A}}| < k \cdot 2^s$. The remaining and more involved part is to bound and reduce the size of the sets $Y_{\mathcal{A}}(\cdot)$. Note the difference between $X_{\mathcal{A}}$ and sets $Y_{\mathcal{A}}(\cdot)$: every tuple $\mathbf{x} \in X_{\mathcal{A}}$ can add to the solution value, whereas tuples $\mathbf{y} \in Y_{\mathcal{A}}(\mathbf{x})$ only provide different ways of satisfying $(\exists \mathbf{y} \in A^{c_y}) : \psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$. Hence our goal is to shrink the sets $Y_{\mathcal{A}}(\mathbf{x})$ without harming satisfiability. We consider $(\exists \mathbf{y} \in A^{c_y}) : \psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ on the level of single disjuncts.

Definition 10 Let $(\mathcal{A}, k)$ be an instance of p-$\mathcal{Q}$ with $\mathcal{A} = (A, R_1, \dots, R_t)$. For $\mathbf{x} \in A^{c_x}$ we define $D_{\mathcal{A}}(\mathbf{x})$ as the set of all disjuncts of $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ over all $\mathbf{y} \in Y_{\mathcal{A}}(\mathbf{x})$.

To reduce the size of sets $D_{\mathcal{A}}(\mathbf{x})$, which will lead to a decreased number of tuples in $Y_{\mathcal{A}}(\mathbf{x})$, we again make use of the Sunflower Lemma. We will see that large sunflowers among disjuncts in $D_{\mathcal{A}}(\mathbf{x})$ represent redundant ways of satisfying $(\exists \mathbf{y} \in A^{c_y}) : \psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$. The size of each $D_{\mathcal{A}}(\mathbf{x})$ is bounded by the size of $Y_{\mathcal{A}}(\mathbf{x}) \subseteq A^{c_y}$ times the number of disjuncts of $\psi(\mathbf{x}, \mathbf{y}, \mathcal{S})$ which is a constant independent of $\mathcal{A}$. Thus the size of each $D_{\mathcal{A}}(\mathbf{x})$ is bounded by a polynomial in the input size.

The following definition of intersection and sunflowers among disjuncts treats disjuncts like sets of literals.

Definition 11 We define the *intersection of two disjuncts* as the conjunction of all literals that occur in both disjuncts. A *sunflower of a set of disjuncts* is a subset such that each pair of disjuncts in the subset has the same intersection (modulo permutation of the literals).

Definition 12 A *partial assignment* is a set L of literals such that no literal is the negation of another literal in L. A formula is *satisfiable under* L if there exists an assignment that satisfies the formula and each literal in L, i.e., there is an extension of the partial assignment L that satisfies $\mathcal{F}$ (as well as, naturally, all literals in L).

The following lemma is the basis of our data reduction. It shows that satisfiability under small partial assignments can be maintained in a reduced set of disjuncts.

Lemma 5 *Let* $(\mathcal{A}, k)$ *be an instance of p-*$\mathcal{Q}$. *For each* $\mathbf{x} \in A^{c_x}$ *there exists a set* $D^*_{\mathcal{A}}(\mathbf{x}) \subseteq D_{\mathcal{A}}(\mathbf{x})$ *of cardinality* $\mathcal{O}(k^s)$ *such that*:

1. *For every partial assignment* L *of at most* sk *literals,* $D^*_{\mathcal{A}}(\mathbf{x})$ *contains a disjunct satisfiable under* L, *if and only if* $D_{\mathcal{A}}(\mathbf{x})$ *contains a disjunct satisfiable under* L.
2. $D^*_{\mathcal{A}}(\mathbf{x})$ *can be computed in time polynomial in* $|\mathcal{A}|$.

Proof Let $\mathcal{A} = (A, R_1, \ldots, R_t)$ be a finite structure of type $(r_1, \ldots, r_t)$, let $\mathbf{x} \in A^{c_x}$, and let $D_{\mathcal{A}}(\mathbf{x})$ be a set of disjuncts according to Definition 10. We compute the set $D^*_{\mathcal{A}}(\mathbf{x})$ starting from $D^*_{\mathcal{A}}(\mathbf{x}) = D_{\mathcal{A}}(\mathbf{x})$ and successively shrinking sunflowers while the cardinality of $D^*_{\mathcal{A}}(\mathbf{x})$ is greater than $(sk + 1)^s \cdot s! \cdot s$.

We compute a sunflower of cardinality $sk + 2$, say $F = \{f_1, \ldots, f_{sk+2}\}$, in time polynomial in $|D^*_{\mathcal{A}}(\mathbf{x})|$ using Corollary 1. We delete a disjunct of F, say f_{sk+2}, from $D^*_{\mathcal{A}}(\mathbf{x})$. Let O and P be copies of $D^*_{\mathcal{A}}$ before respectively after deleting f_{sk+2}. Observe that $F' = F \setminus \{f_{sk+2}\}$ is a sunflower of cardinality $sk + 1$ in P. Let L be a partial assignment of at most sk literals and assume that no disjunct in P is satisfiable under L. This means that for each disjunct of P there is a literal in L that contradicts it, i.e., a literal that is the negation of a literal in the disjunct. We focus on the sunflower F' in P. There must be a literal in L, say ℓ, that contradicts the intersection of at least two disjuncts of F', say f and f', since $|F'| = sk + 1$ and $|L| \le sk$. Therefore ℓ is the negation of a literal in the intersection of f and f', i.e., the core of F'. Thus ℓ contradicts also f_{sk+2} and we conclude that no disjunct in $O = P \cup \{f_{sk+2}\}$ is satisfiable under the partial assignment L. The reverse argument holds since all disjuncts of P are contained in O. Thus each step maintains the desired property (1).

At the end $D^*_{\mathcal{A}}(x)$ contains no more than $(sk + 1)^s \cdot s! \cdot s \in \mathcal{O}(k^s)$ disjuncts. The computation takes time polynomial in the size of $\mathcal{A}$ since the cardinality of $D_{\mathcal{A}}(\mathbf{x})$ is bounded by a polynomial in the size of $\mathcal{A}$ and a disjunct is deleted in each step. □

As in the previous section we are able to generate a kernelized instance of another problem, that is easier to handle. The sets $D^*_{\mathcal{A}}(\mathbf{x})$ describe a possibly different formula for each $\mathbf{x}$, however, it is more convenient to view them as an image of the original instance on which it is easier to draw conclusions. Again, we will use the "sandwiching" trick.

Lemma 6 *Let* $(\mathcal{A}, k)$ *be an instance of p-*$\mathcal{Q}$ *with* $\mathcal{A} = (A, R_1, \ldots, R_t)$ *and let* $\mathbf{x} \in A^{c_x}$. *Let* $D'_{\mathcal{A}}(\mathbf{x})$ *be a subset of* $D_{\mathcal{A}}(\mathbf{x})$ *such that* $D^*_{\mathcal{A}}(\mathbf{x}) \subseteq D'_{\mathcal{A}}(\mathbf{x}) \subseteq D_{\mathcal{A}}(\mathbf{x})$. *For any* *partial assignment L of at most sk literals it holds that* $D_{\mathcal{A}}(\mathbf{x})$ *contains a disjunct satisfiable under L if and only if* $D'_{\mathcal{A}}(\mathbf{x})$ *contains a disjunct satisfiable under L.*

Proof Let L be a partial assignment of at most sk literals. If $D_{\mathcal{A}}(\mathbf{x})$ contains a disjunct satisfiable under L, then, by Lemma 5, this holds also for $D^*_{\mathcal{A}}(\mathbf{x})$. For $D^*_{\mathcal{A}}(\mathbf{x})$ and $D'_{\mathcal{A}}(\mathbf{x})$ this holds since $D^*_{\mathcal{A}}(\mathbf{x}) \subseteq D'_{\mathcal{A}}(\mathbf{x})$. The same is true for $D'_{\mathcal{A}}(\mathbf{x})$ and $D_{\mathcal{A}}(\mathbf{x})$. □

Theorem 3 *Let* $\mathcal{Q} \in$ MAX NP. *The standard parameterization p-*$\mathcal{Q}$ *of* $\mathcal{Q}$ *admits a polynomial kernelization.*

Proof The proof is organized in three parts. First, given an instance $(\mathcal{A}, k)$ of p-$\mathcal{Q}$, we construct an instance $(\mathcal{A}', k)$ of p-$\mathcal{Q}$ in time polynomial in the size of $(\mathcal{A}, k)$. In the second part, we prove that $(\mathcal{A}, k)$ and $(\mathcal{A}', k)$ are equivalent. In the third part, we conclude the proof by showing that the size of $(\mathcal{A}', k)$ is bounded by a polynomial in k. We recall the assumption that $|X_{\mathcal{A}}| < k \cdot 2^s$, based on Lemma 4.

(I) Let $(\mathcal{A}, k)$ be an instance of p-$\mathcal{Q}$. We use the sets $D_{\mathcal{A}}(\mathbf{x})$ and $D^*_{\mathcal{A}}(\mathbf{x})$ according to Definition 10 and Lemma 5. Recall that $D_{\mathcal{A}}(\mathbf{x})$ is the set of all disjuncts of $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ for every $\mathbf{y} \in Y_{\mathcal{A}}(\mathbf{x})$. Thus, for each disjunct $d \in D^*_{\mathcal{A}}(\mathbf{x}) \subseteq D_{\mathcal{A}}(\mathbf{x})$, we can select a $\mathbf{y}_d \in Y_{\mathcal{A}}(\mathbf{x})$ such that d is a disjunct of $\psi_{\mathbf{x},\mathbf{y}_d}(\mathcal{S})$. Let $Y'_{\mathcal{A}}(\mathbf{x}) \subseteq Y_{\mathcal{A}}(\mathbf{x})$ be the set of these selected tuples $\mathbf{y}_d$. Let $D'_{\mathcal{A}}(\mathbf{x})$ be the set of all disjuncts of $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ for $\mathbf{y} \in Y'_{\mathcal{A}}(\mathbf{x})$. Since $D^*_{\mathcal{A}}(\mathbf{x})$ contains some disjuncts of $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ for $\mathbf{y} \in Y'_{\mathcal{A}}(\mathbf{x})$ and $D_{\mathcal{A}}(\mathbf{x})$ contains all disjuncts of $\psi_{\mathbf{x},\mathbf{y}}(\mathcal{S})$ for $\mathbf{y} \in Y_{\mathcal{A}}(\mathbf{x}) \supseteq Y'_{\mathcal{A}}(\mathbf{x})$, we have that $D^*_{\mathcal{A}}(\mathbf{x}) \subseteq D'_{\mathcal{A}}(\mathbf{x}) \subseteq D_{\mathcal{A}}(\mathbf{x})$.

For each $\mathbf{x}$ this takes time $\mathcal{O}(|D^*_{\mathcal{A}}(\mathbf{x})| \cdot |Y^*_{\mathcal{A}}(\mathbf{x})|) \subseteq \mathcal{O}(k^s \cdot |A|^{c_y})$. Computing $Y'_{\mathcal{A}}(\mathbf{x})$ for all $\mathbf{x} \in A^{c_x}$ takes time $\mathcal{O}(|A|^{c_x} \cdot k^s \cdot |A|^{c_y})$, i.e., time polynomial in the size of $(\mathcal{A}, k)$ since k is never larger than $|A|^{c_x}$.[2]

Let $A' \subseteq A$ be the set of all components of $\mathbf{x} \in X_{\mathcal{A}}$ and $\mathbf{y} \in Y'_{\mathcal{A}}(\mathbf{x})$ for all $\mathbf{x} \in X_{\mathcal{A}}$. This ensures that $X_{\mathcal{A}} \subseteq (A')^{c_x}$ and $Y'_{\mathcal{A}}(\mathbf{x}) \subseteq (A')^{c_y}$ for all $\mathbf{x} \in X_{\mathcal{A}}$. Let R'_i be the restriction of R_i to A' and let $\mathcal{A}' = (A', R'_1, \dots, R'_t)$.

(II) We will now prove that $\text{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k$ if and only if $\text{opt}_{\mathcal{Q}}(\mathcal{A}') \geq k$, i.e., that $(\mathcal{A}, k)$ and $(\mathcal{A}', k)$ are equivalent. Assume that $\text{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k$ and let $\mathcal{S} = (S_1, \dots, S_u)$ such that $|\{\mathbf{x} : (\mathcal{A}, \mathcal{S}) \models (\exists \mathbf{y}) : \psi(\mathbf{x}, \mathbf{y}, \mathcal{S})\}| \geq k$. This implies that there must exist tuples $\mathbf{x}_1, \dots, \mathbf{x}_k \in A^{c_x}$ and $\mathbf{y}_1, \dots, \mathbf{y}_k \in A^{c_y}$ such that $\mathcal{S}$ satisfies $\psi_{\mathbf{x}_i,\mathbf{y}_i}(\mathcal{S})$ for $i = 1, \dots, k$. Thus $\mathcal{S}$ must satisfy at least one disjunct in each $\psi_{\mathbf{x}_i,\mathbf{y}_i}(\mathcal{S})$ since these formulas are in disjunctive normal form. Accordingly let $d_1, \dots, d_k$ be disjuncts such that $\mathcal{S}$ satisfies the disjunct d_i in $\psi_{\mathbf{x}_i,\mathbf{y}_i}(\mathcal{S})$ for $i = 1, \dots, k$. We show that there exists $\mathcal{S}'$ such that:

$$|\{\mathbf{x} : (\mathcal{A}', \mathcal{S}') \models (\exists \mathbf{y}) : \psi(\mathbf{x}, \mathbf{y}, \mathcal{S}')\}| \geq k.$$

For $p = 1, \dots, k$ we apply the following step: If $\mathbf{y}_p \in Y'_{\mathcal{A}}(\mathbf{x}_p)$ then do nothing. Otherwise consider the partial assignment L consisting of the at most sk literals of the disjuncts $d_1, \dots, d_k$. The set $D_{\mathcal{A}}(\mathbf{x}_p)$ contains a disjunct that is satisfiable under L, namely d_p. By Lemma 6, it follows that $D'_{\mathcal{A}}(\mathbf{x}_p)$ also contains a disjunct satisfiable

[2] That is, $(\mathcal{A}, k)$ is a no-instance if $k > |A|^{c_x}$ since k exceeds the number of tuples $\mathbf{x} \in A^{c_x}$.

under L, say d'_p. Let $\mathbf{y}'_p \in Y'_{\mathcal{A}}(\mathbf{x}_p)$ such that d'_p is a disjunct of $\psi_{\mathbf{x}_p,\mathbf{y}'_p}(\mathcal{S})$. Such a $\mathbf{y}'_p$ can be found by selection of $D'_{\mathcal{A}}(\mathbf{x}_p)$. Change $\mathcal{S}$ in the following way to satisfy the disjunct d'_p. For each literal of d'_p of the form $S_i(\mathbf{z})$ add $\mathbf{z}$ to the relation S_i. Similarly for each literal of the form $\neg S_i(\mathbf{z})$ remove $\mathbf{z}$ from S_i. This does not change the fact that $\mathcal{S}$ satisfies the disjunct d_i in $\psi_{\mathbf{x}_i,\mathbf{y}_i}(\mathcal{S})$ for $i = 1, \ldots, k$ since, by selection, d'_p is satisfiable under L. Then we replace $\mathbf{y}_p$ by $\mathbf{y}'_p$ and d_p by d'_p. Thus we maintain that $\mathcal{S}$ satisfies d_i in $\psi_{\mathbf{x}_i,\mathbf{y}_i}(\mathcal{S})$ for $i = 1, \ldots, k$.

After these steps we obtain $\mathcal{S}$ as well as tuples $\mathbf{x}_1, \ldots, \mathbf{x}_k$, $\mathbf{y}_1, \ldots, \mathbf{y}_k$ with $\mathbf{y}_i \in Y'_{\mathcal{A}}(\mathbf{x}_i)$, and disjuncts $d_1, \ldots, d_k$ such that $\mathcal{S}$ satisfies d_i in $\psi_{\mathbf{x}_i,\mathbf{y}_i}(\mathcal{S})$ for $i = 1, \ldots, k$. Let $\mathcal{S}'$ be the restriction of $\mathcal{S}$ to A'. Then we have that $(\mathcal{A}', \mathcal{S}') \models \psi_{\mathbf{x}_i,\mathbf{y}_i}(\mathcal{S}')$ for $i = 1, \ldots, k$ since A' is defined to contain the components of tuples $\mathbf{x} \in X_{\mathcal{A}}$ and of all tuples $\mathbf{y} \in Y'_{\mathcal{A}}(\mathbf{x})$ for $\mathbf{x} \in X_{\mathcal{A}}$. Hence $\mathbf{x}_i \in \{\mathbf{x} : (\mathcal{A}', \mathcal{S}') \models (\exists \mathbf{y}) : \psi(\mathbf{x}, \mathbf{y}, \mathcal{S}')\}$ for $i = 1, \ldots, k$. Thus $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}') \geq k$.

For the reverse direction assume that $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}') \geq k$. Since $A' \subseteq A$ it follows that

$$\{\mathbf{x} : (\mathcal{A}', \mathcal{S}') \models (\exists \mathbf{y}) : \psi(\mathbf{x}, \mathbf{y}, \mathcal{S}')\} \subseteq \{\mathbf{x} : (\mathcal{A}, \mathcal{S}') \models (\exists \mathbf{y}) : \psi(\mathbf{x}, \mathbf{y}, \mathcal{S}')\}.$$

Thus $|\{\mathbf{x} : (\mathcal{A}, \mathcal{S}') \models (\exists \mathbf{y}) : \psi(\mathbf{x}, \mathbf{y}, \mathcal{S}')\}| \geq k$, implying that $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k$. Therefore $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k$ if and only if $\mathrm{opt}_{\mathcal{Q}}(\mathcal{A}') \geq k$. Hence $(\mathcal{A}, k)$ and $(\mathcal{A}', k)$ are equivalent instances of p-$\mathcal{Q}$.

(III) We conclude the proof by providing an upper bound on the size of $(\mathcal{A}', k)$ that is polynomial in k. For the sets $Y'_{\mathcal{A}}(\mathbf{x})$ we selected one tuple $\mathbf{y}$ for each disjunct in $D^*_{\mathcal{A}}(\mathbf{x})$. Thus $|Y'_{\mathcal{A}}(\mathbf{x})| \leq |D^*(\mathbf{x})| \in \mathcal{O}(k^s)$ for all $\mathbf{x} \in X_{\mathcal{A}}$. The set A' contains the components of tuples $\mathbf{x} \in X_{\mathcal{A}}$ and of all tuples $\mathbf{y} \in Y'_{\mathcal{A}}(\mathbf{x})$ for $\mathbf{x} \in X_{\mathcal{A}}$. Thus

$$\begin{aligned} |A'| &\leq c_x \cdot |X_{\mathcal{A}}| + c_y \cdot \sum_{\mathbf{x} \in X_{\mathcal{A}}} |Y'_{\mathcal{A}}(\mathbf{x})| \\ &\leq c_x \cdot |X_{\mathcal{A}}| + c_y \cdot |X_{\mathcal{A}}| \cdot \mathcal{O}(k^s) \\ &< c_x \cdot k \cdot 2^s + c_y \cdot k \cdot 2^s \cdot \mathcal{O}(k^s) = \mathcal{O}(k^{s+1}). \end{aligned}$$

For each relation R'_i we have $|R'_i| \leq |A'|^{r_i} \in \mathcal{O}(k^{(s+1)r_i})$. Thus the size of $(\mathcal{A}', k)$ is bounded by $\mathcal{O}(k^{(s+1)m})$, where m is the largest arity of a relation R_i. □

For MAX SNP there is a fairly immediate stronger kernelization that relies on Lemma 4.

Corollary 2 *Let $\mathcal{Q} \in$ MAX SNP. The standard parameterization p-$\mathcal{Q}$ of $\mathcal{Q}$ admits a polynomial kernelization with a linear bound on the size of the base set of the obtained finite structure.*

Proof Let $\mathcal{Q} \in$ MAX SNP be an optimization problem on finite structures of type $(r_1, \ldots, r_t)$. Let $\mathcal{S} = (S_1, \ldots, S_u)$ be a tuple of relation symbols of arity $s_1, \ldots, s_u$. Finally let $\psi(\mathbf{x}, \mathcal{S})$ be a formula in disjunctive normal form such that the optimum value of $\mathcal{Q}$ on a finite structure $\mathcal{A}$ of type $(r_1, \ldots, r_t)$ can be expressed as

$$\text{opt}_{\mathcal{Q}}(\mathcal{A}) = \max_{\mathcal{S}} |\{\mathbf{x} : (\mathcal{A}, \mathcal{S}) \models \psi(\mathbf{x}, \mathcal{S})\}|.$$

Now, let $(\mathcal{A}, k)$ be an instance of p-$\mathcal{Q}$, with $\mathcal{A} = (A, R_1, \ldots, R_t)$. Similarly to Definition 9, we consider the set $X_{\mathcal{A}}$ of all tuples $\mathbf{x}$ such that $\psi_{\mathbf{x}}(\mathcal{S})$ holds for some $\mathcal{S}$:

$$X_{\mathcal{A}} = \{\mathbf{x} : (\exists \mathcal{S}) : (\mathcal{A}, \mathcal{S}) \models \psi_{\mathbf{x}}(\mathcal{S})\}.$$

By Lemma 4, if $|X_{\mathcal{A}}| \geq k \cdot 2^s$ then $\text{opt}_{\mathcal{Q}}(\mathcal{A}) \geq k$ and we may accept $\mathcal{A}$ as a yes-instance. Otherwise $|X_{\mathcal{A}}| \in \mathcal{O}(k)$ and by restricting A to those elements that occur in elements of $X_{\mathcal{A}}$ we obtain A' with $|A'| \in \mathcal{O}(k)$. Also restricting the relations R_i to A' we obtain an equivalent instance $\mathcal{A}' = (A', R'_1, \ldots, R'_t)$ of total size $\mathcal{O}(k^m)$ where $k = \max\{r_1, \ldots, r_t\}$. □

5 Conclusion

We have constructively established that the standard parameterizations of problems in MIN $F^+\Pi_1$ and MAX NP admit polynomial kernelizations. Thus a strong relation between constant-factor approximability and polynomial kernelizability has been showed for two large classes of problems. It remains an open problem to give a more general result that covers all known examples (e.g., FEEDBACK VERTEX SET). It might be profitable to consider closures of MAX SNP under reductions that preserve constant-factor approximability. Khanna et al. [25] proved that APX and APX-PB are the closures of MAX SNP under PTAS-preserving reductions and E-reductions, respectively. Since both classes contain BIN PACKING which does not admit a polynomial kernelization, this leads to the question whether polynomial kernelizability or fixed-parameter tractability are maintained under restricted versions of these reductions.

Furthermore, it would be interesting to see whether polynomial lower bounds similar to the results of Dell and van Melkebeek [12] can be proven. It is easy, however, to construct artificial examples with almost redundant relations of high arity being part of the finite structures, so the focus may have to be on exhibiting meaningful families of problems in MIN $F^+\Pi_1$ and MAX NP and showing lower bounds for them.

References

1. Abu-Khzam, F.N.: A kernelization algorithm for d-hitting set. J. Comput. Syst. Sci. **76**(7), 524–531 (2010)
2. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. J. ACM **45**(3), 501–555 (1998)

3. Bafna, V., Berman, P., Fujito, T.: A 2-approximation algorithm for the undirected feedback vertex set problem. SIAM J. Discrete Math. **12**(3), 289–297 (1999)
4. Bodlaender, H.L.: Kernelization: new upper and lower bound techniques. In: Chen, J., Fomin, F.V. (eds.) Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Revised Selected Papers, Copenhagen, Denmark, 10–11 September 2009. Lecture Notes in Computer Science, vol. 5917, pp. 17–37. Springer, Berlin (2009)
5. Bodlaender, H.L., Thomassé, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. In: Fiat, A., Sanders, P. (eds.) ESA. Lecture Notes in Computer Science, vol. 5757, pp. 635–646. Springer, Berlin (2009)
6. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) Kernelization. In: FOCS, pp. 629–638. IEEE Computer Society, Los Alamitos (2009)
7. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comput. Syst. Sci. **75**(8), 423–434 (2009)
8. Cai, L., Chen, J.: On fixed-parameter tractability and approximability of NP optimization problems. J. Comput. Syst. Sci. **54**(3), 465–474 (1997)
9. Cai, L., Huang, X.: Fixed-parameter approximation: conceptual framework and approximability results. Algorithmica **57**(2), 398–412 (2010)
10. Chen, J., Fomin, F.V. (eds.): Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Revised Selected Papers, Copenhagen, Denmark, 10–11 September 2009. Lecture Notes in Computer Science, vol. 5917. Springer, Berlin (2009)
11. Chen, J., Kanj, I.A., Jia, W.: Vertex cover: further observations and further improvements. J. Algorithms **41**(2), 280–301 (2001)
12. Dell, H., van Melkebeek, D.: Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In: Schulman, L.J. (ed.) STOC, pp. 251–260. ACM, New York (2010)
13. Dom, M., Lokshtanov, D., Saurabh, S.: Incompressibility through colors and ids. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S.E., Thomas, W. (eds.) ICALP (1). Lecture Notes in Computer Science, vol. 5555, pp. 378–389. Springer, Berlin (2009)
14. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, Berlin (1998)
15. Erdős, P., Rado, R.: Intersection theorems for systems of sets. J. Lond. Math. Soc. **35**, 85–90 (1960)
16. Fagin, R.: Generalized first-order spectra and polynomial-time recognizable sets. In: Complexity of Computation. SIAM-AMS Proceedings, vol. 7, pp. 43–73. SIAM, Philadelphia (1974)
17. Feige, U., Hajiaghayi, M., Lee, J.R.: Improved approximation algorithms for minimum weight vertex separators. SIAM J. Comput. **38**(2), 629–657 (2008)
18. Fernau, H., Fomin, F.V., Lokshtanov, D., Raible, D., Saurabh, S., Villanger, Y.: Kernel(s) for problems with no kernel: on out-trees with many leaves. In: Albers, S., Marion, J.-Y. (eds.) STACS. Dagstuhl Seminar Proceedings, vol. 09001, pp. 421–432. Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Germany (2009)
19. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An Eatcs Series. Springer, Berlin (2006)
20. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Bidimensionality and kernels. In: Charikar, M. (ed.) SODA, pp. 503–510. SIAM, Philadelphia (2010)
21. Fortnow, L., Santhanam, R.: Infeasibility of instance compression and succinct PCPs for NP. J. Comput. Syst. Sci. **77**(1), 91–106 (2011)
22. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of Np-completeness. Freeman, New York (1979)
23. Guo, J., Niedermeier, R.: Invitation to data reduction and problem kernelization. SIGACT News **38**(1), 31–45 (2007)
24. Halperin, E.: Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. SIAM J. Comput. **31**(5), 1608–1623 (2002)
25. Khanna, S., Motwani, R., Sudan, M., Vazirani, U.V.: On syntactic versus computational views of approximability. SIAM J. Comput. **28**(1), 164–191 (1998)
26. Kolaitis, P.G., Thakur, M.N.: Logical definability of NP optimization problems. Inf. Comput. **115**(2), 321–353 (1994)
27. Kolaitis, P.G., Thakur, M.N.: Approximation properties of NP minimization classes. J. Comput. Syst. Sci. **50**(3), 391–411 (1995)
28. Kratsch, S., Wahlström, M.: Two edge modification problems without polynomial kernels. In: Chen, J., Fomin, F.V. (eds.) Parameterized and Exact Computation, 4th International Workshop, IWPEC

2009, Revised Selected Papers, Copenhagen, Denmark, 10–11 September 2009. Lecture Notes in Computer Science, vol. 5917, pp. 264–275. Springer, Berlin (2009)
29. Kratsch, S., Wahlström, M.: Preprocessing of min ones problems: a dichotomy. In: Abramsky, S., Gavoille, C., Kirchner, C., auf der Heide, F.M., Spirakis, P.G. (eds.) ICALP (1). Lecture Notes in Computer Science, vol. 6198, pp. 653–665. Springer, Berlin (2010)
30. Kratsch, S., Marx, D., Wahlström, M.: Parameterized complexity and kernelizability of max ones and exact ones problems. In: Hlinený, P., Kucera, A. (eds.) MFCS. Lecture Notes in Computer Science, vol. 6281, pp. 489–500. Springer, Berlin (2010)
31. Mahajan, M., Raman, V., Sikdar, S.: Parameterizing MAX SNP problems above guaranteed values. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC. Lecture Notes in Computer Science, vol. 4169, pp. 38–49. Springer, Berlin (2006)
32. Mahajan, M., Raman, V., Sikdar, S.: Parameterizing above or below guaranteed values. J. Comput. Syst. Sci. **75**(2), 137–153 (2009)
33. Mölle, D., Richter, S., Rossmanith, P.: Enumerate and expand: improved algorithms for connected vertex cover and tree cover. Theory Comput. Syst. **43**(2), 234–253 (2008)
34. Natanzon, A., Shamir, R., Sharan, R.: A polynomial approximation algorithm for the minimum fill-in problem. SIAM J. Comput. **30**(4), 1067–1079 (2000)
35. Niedermeier, R.: Invitation to Fixed Parameter Algorithms. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, London (2006)
36. Papadimitriou, C.H.: Computational Complexity. Addison Wesley, Reading (1993)
37. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. **43**(3), 425–440 (1991)
38. Savage, C.D.: Depth-first search and the vertex cover problem. Inf. Process. Lett. **14**(5), 233–237 (1982)
39. Simchi-Levi, D.: New worst-case results for the bin-packing problem. Nav. Res. Logist. **41**, 579–585 (1994)
40. Thomassé, S.: A quadratic kernel for feedback vertex set. In: Mathieu, C. (ed.) SODA, pp. 115–119. SIAM, Philadelphia (2009)