Parameterized Complexity of Fair Vertex Evaluation Problems

Dušan Knop
Algorithmics and Computational Complexity, Faculty IV, TU Berlin
Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic
dusan.knop@tu-berlin.de

Tomáš Masařík
Department of Applied Mathematics, Charles University, Prague, Czech Republic
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
masarik@kam.mff.cuni.cz

Tomáš Toufar
Computer Science Institute, Charles University, Prague, Czech Republic
toufi@iuuk.mff.cuni.cz

Abstract

A prototypical graph problem is centered around a graph-theoretic property for a set of vertices and a solution to it is a set of vertices for which the desired property holds. The task is to decide whether, in the given graph, there exists a solution of a certain quality, where we use size as a quality measure. In this work, we are changing the measure to the fair measure (cf. Lin and Sahni [27]). The fair measure of a set of vertices S is (at most) k if the number of neighbors in the set S of any vertex (in the input graph) does not exceed k. One possible way to study graph problems is by defining the property in a certain logic. For a given objective, an evaluation problem is to find a set (of vertices) that simultaneously minimizes the assumed measure and satisfies an appropriate formula. More formally, we study the MSO Fair Vertex Evaluation, where the graph-theoretic property is described by an MSO formula.

In the presented paper we show that there is an FPT algorithm for the MSO Fair Vertex Evaluation problem for formulas with one free variable parameterized by the twin cover number of the input graph and the size of the formula. One may define an extended variant of MSO Fair Vertex Evaluation for formulas with ℓ free variables; here we measure a maximum number of neighbors in each of the ℓ sets. However, such variant is W[1]-hard for parameter ℓ even on graphs with twin cover one.

Furthermore, we study the Fair Vertex Cover (Fair VC) problem. Fair VC is among the simplest problems with respect to the demanded property (i.e., the rest forms an edgeless graph). On the negative side, Fair VC is W[1]-hard when parameterized by both treedepth and feedback vertex set of the input graph. On the positive side, we provide an FPT algorithm for the parameter modular width.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Logic; Theory of computation → Graph algorithms analysis

Keywords and phrases Fair objective, metatheorem, fair vertex cover, twin cover, modular width

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.33

Related Version `http://arxiv.org/abs/1803.06878`

Funding *Dušan Knop*: Supported by DFG under project “MaMu”, NI 369/19 and partly supported by the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16 019/0000765 “Research Center for Informatics”.
Tomáš Masařík : Author was supported by SVV–2017–260452 and by GAUK 1514217 of Charles University and by the CE-ITI grant project P202/12/G061 of GA ČR.

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 33; pp. 33:1–33:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A prototypical graph problem is centered around a fixed property for a set of vertices. A solution is any set of vertices for which the given property holds. In a similar manner, one can define the solution as a set of vertices such that the given property holds when we remove this set of vertices from the input graph. This leads to the introduction of deletion problems – a standard reformulation of some classical problems in combinatorial optimization introduced by Yannakakis [37]. Formally, for a graph property π we formulate a *vertex deletion problem*. That is, given a graph $G = (V, E)$, find a smallest possible set of vertices W such that $G \setminus W$ satisfies the property π. Many classical problems can be formulated in this way such as Minimum Vertex Cover (π describes an edgeless graph) or Minimum Feedback Vertex Set (π is valid for forests).

Clearly, the complexity of a graph problem is governed by the associated property π. We can either study one particular problem or a broader class of problems with related graph-theoretic properties. One such relation comes from logic, for example, two properties are related if it is possible to express both by a first order (FO) formula. Then, it is possible to design a *model checking algorithm* that for any property π expressible in the fixed logic decides whether the input graph with the vertices from W removed satisfies π or not.

Undoubtedly, Courcelle's Theorem [4] for graph properties expressible in the monadic second-order logic (MSO) on graphs of bounded treewidth plays a prime role among model checking algorithms. In particular, Courcelle's Theorem provides for an MSO sentence φ an algorithm that given an n-vertex graph G with treewidth k decides whether φ holds in G in time $f(k, |\varphi|)n$, where f is some computable function and $|\varphi|$ is the quantifier depth of φ. In terms of parameterized complexity, such an algorithm is called *fixed-parameter tractable* (the problem belongs to the class FPT for the combined parameter $k + |\varphi|$). We refer the reader to monographs [7, 8] for background on parameterized complexity theory and algorithm design. There are many more FPT model-checking algorithms, e.g., an algorithm for (existential counting) modal logic model checking on graphs of bounded treewidth [33], MSO model checking on graphs of bounded neighborhood diversity [24], or MSO model checking on graphs of bounded shrubdepth [14] (generalizing the previous result). First order logic (FO) model checking received recently quite some attention as well and algorithms for graphs with bounded degree [34], nowhere dense graphs [16], and some dense graph classes [11] were given.

Fair Objective. Fair deletion problems, introduced by Lin and Sahni [27], are such modifications of deletion problems where instead of minimizing the size of the deleted set we change the objective. The Fair Vertex Deletion problem is defined as follows. For a given graph $G = (V, E)$ and a property π, the task is to find a set $W \subseteq V$ which minimizes the maximum number of neighbors in the set W over all vertices, such that the property π holds in $G \setminus W$. Intuitively, the solution should not be concentrated in a neighborhood of any vertex. In order to classify (fair) vertex deletion problems we study the associated decision version, that is, we are interested in finding a set W of size at most k, for a given number k. Note that this can introduce only polynomial slowdown, as the value of our objective is bounded by 0 from below and by the number of vertices of the input graph from above (provided a solution exists). Since we are about to use a formula with a free variable X to express the desired property π, we naturally use X to represent the set of deleted vertices in the formula. The *fair cost of a set* $W \subseteq V$ is defined as $\max_{v \in V} |N(v) \cap W|$. We refer to the function that assigns each set $W \subseteq V$ its fair cost as to the *fair objective function*. Here, we give a formal definition of the computational problem when the property π is defined in some logic L.

> FAIR L VERTEX DELETION
> **Input:** An undirected graph $G = (V, E)$, an L sentence φ, and a positive integer k.
> **Question:** Is there a set $W \subseteq V$ of fair cost at most k such that $G \setminus W \models \varphi$?

Let $\varphi(X)$ be a formula with one free variable and let $G = (V, E)$ be a graph. For a set $W \subseteq V$ by $\varphi(W)$ we mean that we substitute W for X in φ. The definition below can be naturally generalized to contain ℓ free variables. We would like to point out one crucial difference between deletion and evaluation problems, namely that in evaluation problems we have access to the variable that represents the solution. This enables evaluation problems to impose some conditions on the solution, e.g., we can ensure that the graph induced by the solution has diameter at most 2 or is triangle-free.

> FAIR L VERTEX EVALUATION[1]
> **Input:** An undirected graph $G = (V, E)$, an L formula $\varphi(X)$ with one free variable, and a positive integer k.
> **Question:** Is there a set $W \subseteq V$ of fair cost at most k such that $G \models \varphi(W)$?

Minimizing the fair cost arises naturally for edge problems in many situations as well, e.g., in defective coloring [6], which has been substantialy studied from the practical network communication point of view [17, 19]. A graph $G = (V, E)$ is (k, d)-colorable if every vertex can be assigned a color from the set $[k]$ in such a way that every vertex has at most d neighbors of the same color. This problem can be reformulated in terms of fair deletion; we aim to find a set of edges F such that graph $G' = (V, F)$ has maximum degree d and $G \setminus F$ can be partitioned into k independent sets.

Related Results. There are several possible research directions once a model checking algorithm is known. One possibility is to broaden the result either by enlarging the class of graphs it works for or by extending the expressive power of the concerned logic, e.g., by introducing a new predicate [23]. Another obvious possibility is to find the exact complexity of the general model checking problem by providing better algorithms (e.g., for subclasses [24]) and/or lower-bounds for the problem [9, 25]. Finally, one may instead of deciding a sentence turn attention to finding a set satisfying a formula with a free variable that is optimal with respect to some objective function [1, 5, 15]. In this work, we take the last presented approach – extending a previous work on MSO model checking for the fair objective function.

When extending a model checking result to incorporate an objective function or a predicate, we face two substantial difficulties. On the one hand, we are trying to introduce as strong extension as possible, while on the other, we try not to worsen the running time too much. Of course, these two are in a clash. One evident possibility is to sacrifice the running time and obtain an XP algorithm, that is, an algorithm running in time $f(k)|G|^{g(k)}$. For example there is an XP algorithm for the FAIR MSO_2 VERTEX EVALUATION problems parameterized by treewidth (and the size of the formula) by Kolman et al. [21] running in time $f(|\varphi|, \mathrm{tw}(G))|G|^{g(\mathrm{tw}(G))}$. An orthogonal extension is due to Szeider [35] for the so-called local cardinality constraints (MSO-LCC) who gave an XP algorithm parameterized by treewidth. If we decide to keep the FPT running time, such a result is not possible for treedepth (consequently for treewidth), as we give W[1]-hardness result for a very basic

[1] This problem is called GENERALIZED FAIR L VERTEX-DELETION in [29] and in the respective conference version [28]. However, we believe that evaluation is a more suitable expression and coincides with standard terminology in logic.

Fair $\mathsf{L}_\emptyset$ Vertex Deletion problem[2] in this paper. A weaker form of this hardness was already known for FO logic [29]. Ganian and Obdržálek [15] study CardMSO and provide an FPT algorithm parameterized by neighborhood diversity. Recently, Masařík and Toufar [29] examined fair objective and provide an FPT algorithm for the Fair MSO_1 Vertex Evaluation problem parameterized by neighborhood diversity. See also [20] for a discussion of various extensions of the MSO.

We want to turn a particular attention to the Fair Vertex Cover (Fair VC) problem which, besides its natural connection with Vertex Cover, has some further interesting properties. For example, we can think about classical vertex cover as having several crossroads (vertices) and roads (edges) that we need to monitor. However, people could get uneasy if they will see too many policemen from one single crossroad. In contrast, if the vertex cover has low fair cost, it covers all roads while keeping a low number of policemen in the neighborhood of every single crossroad.

1.1 Our Results

We give a metatheorem for graphs having bounded twin cover. Twin cover (introduced by Ganian [12]; see also [13]) is one possible generalization of the vertex cover number. Here, we measure the number of vertices needed to cover all edges that are not twin-edges; an edge $\{u, v\}$ is a *twin-edge* if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. Ganian introduced twin cover in the hope that it should be possible to extend algorithms designed for parameterization by the vertex cover number to a broader class of graphs.

► **Theorem 1.** *The* Fair MSO_1 Vertex Evaluation *problem parameterized by the twin cover number and the quantifier depth of the formula admits an* FPT *algorithm.*

We want to point out here that in order to obtain this result we have to reprove the original result of Ganian [12] for MSO_1 model checking on graphs of bounded twin cover. For this, we extend arguments given by Lampis [24] in the design of an FPT algorithm for MSO_1 model checking on graphs of bounded neighborhood diversity. We do this to obtain better insight into the structure of the graph (a kernel) on which model checking is performed (its size is bounded by a function of the parameter). This, in turn, allows us to find a solution minimizing the fair cost and satisfying the MSO_1 formula. The result by Ganian in version [12] is based on the fact that graphs of bounded twin cover have bounded shrubdepth and so MSO_1 model checking algorithm on shrubdepth ([14, 10]) can be used.

When proving hardness results it is convenient to show the hardness result for a concrete problem that is expressible by an MSO_1 formula, yet as simple as possible. Therefore, we introduce a key problem for Fair Vertex Deletion – the Fair VC problem.

Fair Vertex Cover (Fair VC)	
Input:	An undirected graph $G = (V, E)$, and a positive integer k.
Question:	Is there a set $W \subseteq V$ of fair cost at most k such that $G \setminus W$ is an edgeless graph?

The Fair VC problem can be expressed in any logic that can express an edgeless graph (we denote such logic $\mathsf{L}_\emptyset$) which is way weaker than FO. Therefore, we propose a systematic study of the Fair VC problem which, up to our knowledge, have not been considered before.

► **Theorem 2.** *The* Fair VC *problem parameterized by treedepth* $\mathrm{td}(G)$ *and feedback vertex set* $\mathrm{fvs}(G)$ *combined is* W[1]*-hard.*

[2] Here, $\mathsf{L}_\emptyset$ stands for any logic that can express an edgeless graph.

Table 1 The table summarize some related (with a citation) and all the presented (with a reference) results on the studied parameters. Green cells denote FPT results, and red cells represent hardness results. Logic L in metatheorems is specified by a logic used in the respective theorem. Symbol $*$ denotes implied results from previous research and symbol ✓denotes new implied results. A question mark (?) indicates that the complexity is unknown.

	vc	fvs + td	tc	nd	cvd	mw
Fair VC	$*$	T2	✓	$*$	?	T3
FV L Del	MSO_2 $*$	$L_\emptyset$ ✓	MSO_1 ✓	MSO_1 $*$	?	?
FV L Eval	MSO_2 [29]	$L_\emptyset$ ✓	MSO_1 T1	MSO_1 $*$	?	?
ℓ-FV L Eval	MSO_1 $*$	$L_\emptyset$ ✓	MSO_1 T5	MSO_1 [20]	MSO_1 ✓	MSO_1 ✓

Note that this immediately yields W[1]-hardness and $f(w)n^{o(\sqrt{w})}$ lower bound for Fair $L_\emptyset$ Vertex Evaluation. Previously, an $f(w)n^{o(w^{1/3})}$ lower bound was given for FO logic by Masařík and Toufar [29]. Thus our result is stronger in both directions, i.e., the lower bound is stronger, and the logic is less powerful. On the other hand, we show that Fair VC can be solved efficiently in dense graph models.

▶ **Theorem 3.** *The* Fair VC *problem parameterized by modular width* $\mathrm{mw}(G)$ *admits an* FPT *algorithm with running time* $2^{\mathrm{mw}(G)} \cdot \mathrm{mw}(G) \cdot n^3$*, where* n *is the number of vertices in* G*.*

We point out that the Fair VC problem is (rather trivially) AND-compositional and thus it does not admit a polynomial kernel for parameterization by modular width.

▶ **Lemma 4.** *The* Fair VC *problem parameterized by the modular width of the input graph does not admit a polynomial kernel, unless* $\mathsf{NP} \subseteq \mathsf{coNP}/\mathrm{poly}$*.*

Moreover, an analog to Theorem 3 cannot hold for parameterization by shrubdepth of the input graph. This is a consequence of Theorem 2 and the fact that if a class of graphs has bounded treedepth, then it has bounded shrubdepth (cf. [14, Proposition 3.4]).

Another limitation in a rush for extensions of Theorem 1 is given when aiming for more free variables. More formally, the problem ℓ-Fair L Vertex Evaluation has formula $\varphi(X_1, \dots, X_\ell)$ with ℓ free variables as an input and $W_1, \dots, W_\ell$ are the corresponding sets in G of fair cost at most k. The *fair cost* of $W_1, \dots, W_\ell$ is defined as $\max_{v \in V} \max_{i \in [\ell]} |N(v) \cap W_i|$. It is very surprising that such a generalization is not possible for parameterization by twin cover, since the same extension is possible for parameterization by neighborhood diversity [20]. In fact, they prove something even stronger, i.e., an FPT algorithm parameterized by neighborhood diversity in the context of $\mathsf{MSO}^{\mathsf{L}}_{\mathsf{lin}}$ is given in [20]. In $\mathsf{MSO}^{\mathsf{L}}_{\mathsf{lin}}$ one can specify both lower- and upper-bound for each vertex and each free variable (i.e., a feasibility interval is given for every vertex).

▶ **Theorem 5** (★[3]). *The* ℓ-Fair FO Vertex Evaluation *problem is* W[1]*-hard for parameter* ℓ *even on graphs with twin cover of size one.*

The reduction is done from the Unary ℓ-Bin Packing problem; the lower-bound and W[1]-hardness of Unary ℓ-Bin Packing was given by Jansen et al. [18].

For an overview of the results, please refer to Table 1 and to Figure 1 for the hierarchy of classes.

[3] We mark a result by ★ if the proof is omitted and deferred to the full version (available on arXiv).

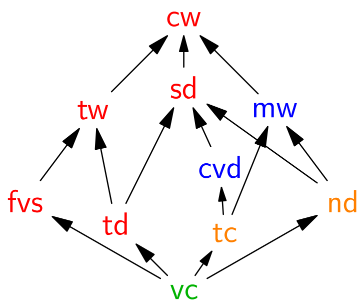

Figure 1 Hierarchy of graph parameters with depicted complexity of the FAIR L VERTEX EVALUATION problem. An arrow indicates that a graph parameter upper-bounds the other. Thus, hardness results are implied in the direction of arrows, and FPT algorithms are implied in the reverse direction. Green colors indicate FPT results for MSO_2, orange are FPT for MSO_1, blue are open, and red are hardness results. We denote treewidth by tw, shrubdepth by sd, and clique width by cw. We refer to book [7] for definitions. Other parameters and their respective abbreviations are defined in Subsection 1.2.

1.2 Preliminaries

For a positive integer n we denote $[n]$ the set $\{1, \ldots, n\}$. We deal with simple undirected graphs, for further standard notation we refer to monographs: graph theory [30] and parameterized complexity [7]. For a vertex v by $N(v)$ we denote the neighborhood of v and by $N[v]$ we denote the closed neighborhood of vertex v, i.e., $N(v) \cup \{v\}$.

A parameter closely related to twin cover is *cluster vertex deletion* (cvd(G)), that is, the smallest number of vertices one has to delete from a graph in order to get a collection of (disjoint) cliques. *Treedepth* of a graph G (td(G)) is the minimum height of a rooted forest whose transitive closure contains the graph G [31]. *Feedback vertex set* (fvs(G)) is the minimum number of vertices of a graph G whose removal leaves the graph without cycles. *Neighborhood diversity* (nd(G)) is the smallest integer r such that the the vertex set of G can be partitioned into r sets $V_1, \ldots, V_r$ in such a way that the graph $G[V_i]$ is either a clique or an edgeless graph for all $i \in [r]$ and the bipartite graph $G[V_i, V_j]$ is either a complete bipartite graph or an edgeless graph for all distinct $i, j \in [r]$. *Modular width* of a graph G (mw(G)), is the smallest positive integer r such that G can be obtained from an algebraic expression of width at most r, defined as follows. The *width of an expression* A is the maximum number of operands used by any occurrence of the substitution operation in A, where A is an algebraic expression that uses the following operations:

1. Create an isolated vertex.
2. The *substitution operation* with respect to a template graph T with vertex set $[r]$ and graphs $G_1, \ldots, G_r$ created by algebraic expression. The substitution operation, denoted by $T(G_1, \ldots, G_r)$, results in the graph on vertex set $V = V_1 \cup \cdots \cup V_r$ and edge set $E = E_1 \cup \cdots \cup E_r \cup \bigcup_{\{i,j\} \in E(T)} \big\{\{u, v\} : u \in V_i, v \in V_j\big\}$, where $G_i = (V_i, E_i)$ for all $i \in [r]$.

An algebraic expression of width mw(G) can be computed in linear time [36].

We stick with standard definitions and notation in logic. For a comprehensive summary, please consult a book by Libkin [26].

2 Metatheorems for Fair Evaluation

We show an FPT algorithm as it is stated in Theorem 1. We give a more detailed statement that implies the promised result straigthforwardly.

We split the proof into two parts. First, we show an algorithm for MSO_1 model checking parameterized by twin cover of the graph (Proposition 8). In the second part, we prove that we can even compute the optimal fair cost (Proposition 12) and so derive the promised result.

Overview of the Algorithm. For the model checking algorithm, we closely follow the approach of Lampis [24]. The idea is that if there is a large set of vertices with the same closed neighborhood, then some of them are irrelevant, i.e., we can delete them without affecting the truthfulness of the given formula φ. For graphs of bounded neighborhood diversity using this rule alone can reduce the number of vertices below a bound that depends on $\mathrm{nd}(G)$ and $|\varphi|$ only, thus providing an FPT model checking algorithm. For the graphs of bounded twin cover, this approach can be used to reduce the size of all (twin) cliques, yet their number can still be large. We take the approach one step further and describe the deletion of irrelevant cliques in a similar manner; these rules together yield a model checking algorithm for graphs of bounded twin cover.

The reduction rules also lead to a notion of *shape* of a set $W \subseteq V$. The motivation behind shapes is to partition all subsets of V such that if two sets W, W' have the same shape, then $G \models \varphi(W)$ if and only if $G \models \varphi(W')$. This allows us to consider only one set of each shape for the purposes of model checking. Since the number of all distinct shapes is bounded by some function of parameters, we can essentially brute force through all possible shapes.

A final ingredient is an algorithm that for a given shape outputs a subset of vertices with this shape that minimizes the fair cost. This algorithm uses ILP techniques, in particular minimizing quasiconvex function subject to linear constraints.

Notation. In what follows $G = (V, E)$ is a graph and K is its twin cover of size k. An MSO_1 formula φ contains q_S set quantifiers and q_v element (vertex) quantifiers. Given a twin cover K and $A \subseteq K$, we say that A is the *cover set* of a set $S \subseteq V \setminus K$ if every $v \in S$ has $N(v) \cap K = A$. Note that, by the definition of twin cover, for all $u, v \in V \setminus K$ with $\{u, v\} \in E$ we have that A is a cover set for u if and only if A is a cover set for v. We say that two cliques have the same *type* if they have the same size and the same cover set. Clearly, if the cover set is fixed, two cliques agree on type if and only if their sizes are the same. We define a *labeled graph*, that is, a graph and a collection of labels on the vertices. We say that two cliques have the *same labeled type* if all of them have the same size, the same cover set and the same labels on vertices.

2.1 Model checking

▶ **Proposition 6** ([24, Lemma 5 and Theorem 4]). *Let ϕ be an MSO_1 formula and let G be a labeled graph. If there is a set S of more than $2^{q_S} q_v$ vertices having the same closed neighborhood and the same labels, then for any $v \in S$ we have $G \models \varphi$ if and only if $G \setminus v \models \varphi$.*

In particular, if G is a graph with just one label, then for any clique C where each vertex has exactly the same closed neighborhood in G the following holds. Either there is a vertex $v \in C$ such that $G \models \phi$ if and only if $G \setminus v \models \phi$ or the size of C is bounded by $2^{q_S+1} q_v$.

Proposition 6 bounds the size of a maximum clique in $G \setminus K$ because we can apply it repeatedly for each clique that is bigger than the threshold $2^{q_S+1} q_v$. Now, we need to bound the number of cliques of each type. For this, we establish the following technical lemma.

▶ **Lemma 7** (★). *Let G be a labeled graph with twin cover K. Let φ be an MSO_1 formula with q_v element quantifiers and q_S set quantifiers. Suppose the size of a maximum clique in $G \setminus K$ is bounded by r. If there are strictly more than*

$$\alpha(q_S, q_v) = 2^{rq_S}(q_v + 1)$$

cliques of the same labeled type $\mathcal{T}$, then there exists a clique C of the labeled type $\mathcal{T}$ such that $G \models \varphi$ if and only if $G \setminus C \models \varphi$.

From this, we can derive a model checking algorithm.

▶ **Proposition 8** (★ Model checking on graphs of bounded twin cover). *Let G be a graph with twin cover K of size k and the size of the maximum clique in $G \setminus K$ bounded by $2^{q_S} q_v$ and φ is an MSO_1 sentence then either there exists a clique $C \in G \setminus K$ such that $G \models \phi$ if and only if $G \setminus C \models \varphi$ or the size of G is bounded by*

$$k + (q_v + 1)q_v^2 2^{k+2q_S+2^{q_S} q_S q_v} = 2^{\mathcal{O}(k+2^{q_S} q_S q_v)} .$$

2.2 Finding a Fair Solution

In the upcoming proof we follow the ideas of Masařík and Toufar [29]. They define, for a given formula $\varphi(X)$, a so-called shape of a set $W \subseteq V$ in G. The idea behind a shape is that in order to do the model checking we have deleted some vertices from G that cannot change the outcome of $\varphi(X)$, however, we have to derive a solution of minimal cost in the whole graph G. Thus the shape characterizes a set under which $\varphi(X)$ holds and we have to be able to find a set $W \subseteq V(G)$ for which $\varphi(W)$ holds and W minimizes the fair cost among sets having this shape.

Shape. Let $G = (V, E)$ be a graph, $\varphi(X)$ an MSO formula, $K \subseteq V$ a twin cover of G, $A \subseteq K$, and let $r = 2^{q_S+2} q_v$ and $\alpha = 2^{r(q_S+1)}(q_v + 1)$. Let $\mathcal{C}$ be the collection of all cliques in G such that A is their cover set. We define an *A-shape*. An A-shape of size r is a two dimensional table S_A of dimension $(r + 2) \times (r + 2)$ indexed by $\{0, 1, \dots, r + 1\} \times \{0, 1, \dots, r + 1\}$. Each entry $S_A(i, j) \in \{0, \dots, \alpha + 1\}$. The interpretation of $S_A(i, j)$ is the minimum of $\alpha + 1$ and of the number of cliques C with $N(C) = A$ such that

$$\min(\alpha + 1, |C \cap W|) = i \text{ and } \min(\alpha + 1, |C \setminus W|) = j.$$

Finally, the shape of X in G is a collection of A-shapes for all $A \subseteq K$.

A solution for $\mathcal{C}$ with cover set A can be formally described by a function $\mathrm{sol}\colon \mathcal{C} \to \mathbb{N} \times \mathbb{N}$. The solution sol is *valid* if for every $C \in \mathcal{C}$ with $\mathrm{sol}(C) = (i, j)$ either $i + j = |C|$ or $|C| \geq r$, $i = r + 1$ (or equivalently $j = r + 1$), and $i + j < |C|$. For an illustration of a valid assignment please refer to Figure 2. We say that a valid solution sol is *compatible* with the shape S_A if $S(i, j) = \left|\mathrm{sol}^{-1}(i, j)\right|$, whenever $S(i, j) \leq \alpha$ and $\left|\mathrm{sol}^{-1}(i, j)\right| \geq \alpha + 1$ if $S(i, j) = \alpha + 1$. The A-shape S_A is said to be valid if there exists a valid solution for S_A. Note that such a solution does not exist if the shape specifies too many (or too few) cliques of certain sizes. The shape S is *valid* if all its A-shapes are valid.

The following lemma is a key observation about shapes.

▶ **Lemma 9.** *Let φ be an MSO_1 formula with one free variable, G a graph and W, W' two subsets of vertices having the same shape. Then $G \models \varphi(W)$ if and only if $G \models \varphi(W')$.*

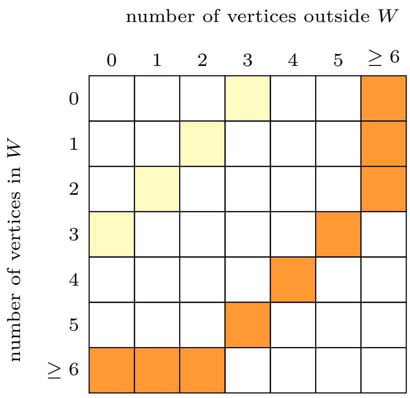

Figure 2 Example of a 7×7 A-shape. All cliques of size 3 will be assigned to yellow (light gray) fields, while cliques of size 8 will be assigned to orange (darker gray) fields.

number of vertices outside W

number of vertices in W	0	1	2	3	4	5	≥ 6
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
≥ 6	?	?	?	?	?	?	6

Figure 3 An example of uncertainty in computation of objective function. The value in the last row depends on the size of the clique we are assigning to those cells. The value in the cell is how much we pay for any compatible clique assigned to this cell.

Proof. The proof follows using Proposition 6 and Lemma 7. Indeed, if we take the graph G with one label corresponding to set W and apply the reduction rules given by Proposition 6 and Lemma 7 and repeat the same process with W', we obtain two isomorphic labeled graphs. ◀

Lemma 9 allows us to say that a formula with one free variable holds under a shape since it is irrelevant which subset of vertices of this particular shape is assigned to the free variable. Also note that deciding whether the formula holds under the shape can be done in FPT time by simply picking arbitrary assignment of the given shape and running a model checking algorithm.

Lemma 11 computes a solution of minimal cost for an A-shape. We do this by reducing the task to integer linear programming (ILP) while using non-linear objective. A fuction $f\colon \mathbb{R}^p \to \mathbb{R}$ is *separable convex* if there exist convex functions $f_i\colon \mathbb{R} \to \mathbb{R}$ for $i \in [p]$ such that $f(x_1, \ldots, x_p) = \sum_{i=1}^{p} f_i(x_i)$.

▶ **Theorem 10** ([32] – simplified)**.** *Integer linear programming with objective minimization of a separable convex function in dimension p is* FPT *with respect to p and space exponential in L the length of encoding of the ILP instance.*

▶ **Lemma 11.** *Let $G = (V, E)$ be a graph, K be a twin cover of G, and $\emptyset \neq A \subseteq K$. There is an algorithm that given an A-shape S_A of size r computes a valid solution for S_A of minimal cost in time $f(|K|, r) \cdot |G|^{O(1)}$ or reports that S_A is not valid.*

Proof. Let $\mathcal{C}$ be the collection of all cliques such that A is their cover set. We split the task of finding a minimal solution to S_A into two independent parts depending on the size of cliques assigned in the phase.

The first phase is for cliques in $\mathcal{C}$ with sizes at most r. Observe that these can be assigned deterministically in a greedy way. This is because no cell of S_A is shared by two sizes and we can see that if there are more cells with value α on the corresponding diagonal we can always prefer the top one as this minimizes the cost (see Figure 3). However, this is not possible for larger cliques as they may in general share some cells of S_A and thus we defer them to the second phase.

Now observe that the most important vertices for computing the cost are the vertices constituting the set A. To see this just note that all other vertices see only A and their neighborhood (a clique) which is at least as large as for the vertices in A. It follows that we should only care about the number of selected vertices such that A is their cover set. Thus if the size of all cliques in $\mathcal{C}$ is bounded in terms of k we are done. Alas, this is not the case.

We split the set $\mathcal{C}$ into sets $\mathcal{C}_1, \dots, \mathcal{C}_{2r}$, and $\mathcal{C}_{\max}$. A clique $C \in \mathcal{C}$ belongs to $\mathcal{C}_{|C|}$ if $1 \le |C| \le 2r$ and belongs to $\mathcal{C}_{\max}$ otherwise. Note that cliques from $\mathcal{C}_{\max}$ may be assigned only to cells having at least one index $r+1$. As mentioned we are about to design an ILP with a non-linear objective function. This ILP has variables $x^q_{i,j}$ that express the number of cliques from the set $\mathcal{C}_q$ assigned to the cell (i,j) of S_A (that is, $1 \le i,j \le r+1$ and $q \in Q = \{1,\dots,2r\} \cup \{\max\}$). The obvious conditions are the following (the $\trianglerighteq$ symbol translates to $=$ if $S(i,j) \le \alpha$ while it translates to $\ge$ if $S(i,j) = \alpha+1$).

$$\sum_{q \in Q} x^q_{i,j} \trianglerighteq S_A(i,j) \qquad 0 \le i,j \le r+1$$

$$\sum_{0 \le i,j \le r+1} x^q_{i,j} = |\mathcal{C}_q| \qquad \forall q \in Q$$

$$x^q_{i,j} \ge 0 \qquad 0 \le i,j \le r+1,\ \forall q \in Q$$

We are about to minimize the following objective

$$\sum_{0 \le i \le r+1; 0 \le j \le r} \ \sum_{1 \le q \le 2r} (q-j) x^q_{i,j} + \sum_{0 \le i \le r+1} \ \sum_{\forall q} i \cdot x^q_{i,r+1} + g\left(x^{\max}_{r+1,0}, \dots, x^{\max}_{r+1,r}\right) ,$$

where $g \colon \mathbb{N}^r \to \mathbb{N}$ is a function that has access to sizes of all cliques in $\mathcal{C}_{\max}$ and computes the minimum possible assignment. We claim that the function g is a separable convex function in variables $x^{\max}_{r+1,0}, \dots, x^{\max}_{r+1,r}$. The first summand of the objective function describes the cliques of size at most $2r$. Their price corresponds to the number of vertices in the clique q minus the number of vertices that are not selected j. The second summand corresponds to the last row, where the cheapest price is always the number of selected vertices i. The last summand, discussed in the following paragraph, describes the assignment to the last row. The result then follows from Theorem 10 as the number of integral variables is $O(r^3)$.

Observe that the value of $g\left(x^{\max}_{r+1,0}, \dots, x^{\max}_{r+1,r}\right)$ is equal to sum of sizes of cliques "assigned to the last row" minus $\sum_{j=0}^{r} j \cdot x^{\max}_{r+1,j}$. Now, $g\left(x^{\max}_{r+1,0}, \dots, x^{\max}_{r+1,r}\right) = g'\left(\sum_{j=0}^{r} x^{\max}_{r+1,j}\right) - \sum_{j=0}^{r} j \cdot x^{\max}_{r+1,j}$. Since all cliques in $\mathcal{C}_{\max}$ are eligible candidates to be assigned to the last row and since it is always cheaper to assign there those of the smallest size among them, we can define g' based only on the number of cliques that are assigned to the last row. This finishes the proof since g' is a convex function. We defer the details on polynomial space to the full version of the paper. ◀

Now we are ready to prove the main result of this section. It essentially follows by the exhaustive search among all possible shapes S such that φ is true under S and application of Lemma 11.

▶ **Proposition 12.** *Let $G = (V,E)$ be a graph with twin cover K of size k. For an* MSO_1 *formula $\varphi(X)$ with one free variable that represents the set to be deleted it is possible to find a set $W \subseteq V$ such that*

- *$\varphi(W)$ holds in G and*
- *the cost of W is minimized among all subset of V satisfying $\varphi(X)$*

in time $f(k, |\varphi|)|V|^{O(1)}$ for some computable function f.

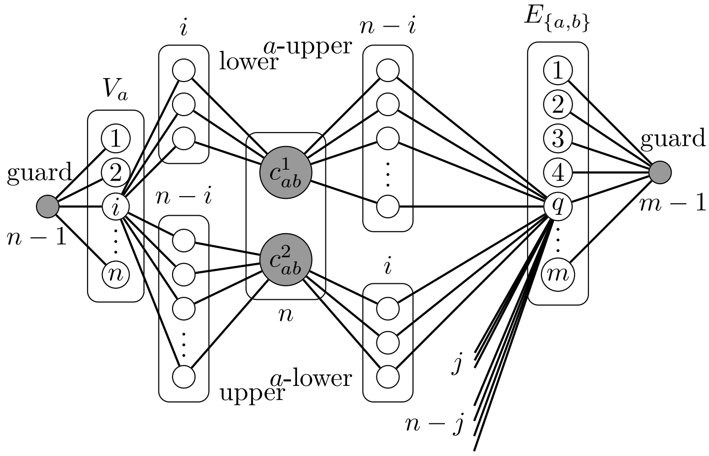

Figure 4 An overview of the reduction in the proof of Theorem 2. The gray vertices are enforced to be a part the fair vertex cover. If a vertex fair objective was lowered, then the resulting threshold is beneath the vertex (the group of vertices).

Proof sketch. We proceed as follows. For every possible selection of $K \cap W$ we generate all possible shapes and check whether $\varphi(X)$ evaluates to true under shape S and if so, we compute W for S having the minimal fair-cost. ◀

3 The Fair VC problem

3.1 Hardness for Treedepth and Feedback Vertex Set

We observe substantial connection between Fair VC and Target Set Selection (TSS). It is worth mentioning that Vertex Cover can be formulated in the language of TSS by setting the threshold to $\deg(v)$ for every vertex v. As a result, our reduction given here is, in certain sense, dual to the one given by Chopin et al. [3] for the TSS problem. However, we will show that the structure of the solution for Fair VC is, in fact, the complement of the structure of the solution for TSS given therein. The archetypal W[1]-hard problem is the ℓ-Multicolored Clique problem [7]:

ℓ-Multicolored Clique	*Parameter:* ℓ
Input:	An ℓ-partite graph $G = (V_1 \cup \cdots \cup V_\ell, E)$, where V_c is an independent set for every $c \in [\ell]$ and they are pairwise disjoint.
Question:	Is there a clique of the size ℓ in G?

Proof sketch of Theorem 2. Refer to figure 4. Observe that we can enforce a vertex v to be a part of the fair vertex cover by attaching $k+1$ degree 1 vertices to v. Notice further that we may adjust (lower) the global budget k for individual vertex v by attaching vertices to v and then attaching k new leaves to the newly added vertices.

There are three types of gadgets in our reduction, namely the vertex selection gadget, the edge selection gadget, and the incidence check gadget. We start by enumerating the vertices in each color class by numbers from $[n]$ and edges by numbers in $[m]$. Now, we construct a graph H in which we are going to look for a vertex cover of small fair cost. Throughout the proof a, b are distinct numbers from $[\ell]$.

A selection gadget consists of z *choice* vertices (representing either the color class V_a with $z = n$ or $E_{\{a,b\}}$ in which case $z = m$), a special vertex called *guard*, and a group of n^2 *enumeration* vertices. The guard vertex is connected to all choice vertices, it is enforced to be a part of the fair vertex cover, and its budget is lowered so that at most $z - 1$ choice vertices can be in any fair vertex cover. The i-th choice vertex is connected to n private enumeration vertices. We further divide these vertices into two parts – the *lower part* consists of q vertices and the *upper part* consists of $n - q$ vertices, where q refers to the vertex number. That is, $q = i$ in case of a vertex choice gadget and for an edge choice gadget, we let $q = v_i$, where v_i is the number of the vertex incident to i-th edge in the corresponding colorclass.

The ab-incidence check gadget consist of two vertices c^1_{ab} and c^2_{ab}. Both c^1_{ab} and c^2_{ab} are enforced to be a part of the solution and with a lowered budget in a way that at most n vertices in the neighborhood of each of them can be part of any fair vertex cover. The vertex c^1_{ab} is connected to every lower part vertex in the selection gadget for V_a and to every upper a-part vertex in the selection gadget for $E_{\{a,b\}}$. For c^2_{ab} we exchange the role of upper- and lower- parts.

▷ Claim 13. Suppose (G, ℓ) is a yes-instance then there is a vertex cover in H with fair cost at most $k = \max(m - 1, 2n)$.

Let $K \subseteq V_1 \times \cdots \times V_\ell$ be a clique in G. We now construct a vertex cover C_K of H having $|N(w) \cap C_K| \leq k$ for all $w \in W$. The set C_K contains the following:

1. all enforced vertices (including all guard and check vertices),
2. if $v \in V_a \cap K$ is the i-th vertex of V_a, then all selection vertices of V_a but the vertex i are in C_K and lower and upper enumeration vertices of i are in C_K, and
3. if $v \in V_a \cap K$ and $u \in V_b \cap K$ are adjacent through q-th edge of $E_{\{a,b\}}$, then all selection vertices of $E_{\{a,b\}}$ but the vertex q are in C_K and q's enumeration vertices are in C_K.

The proof of the reverse direction is deferred to the full version due to space limitations.

It remains to discuss the ETH based lower-bound. This follows immediately from our reduction and the result of Chen et al. [2] who proved that there is no $f(k)n^{o(\ell)}$ algorithm for ℓ-Multicolored Clique unless ETH fails. Since we have $\operatorname{td}(G) + \operatorname{fvs}(G) = O(\ell^2)$ in our reduction, the lower-bound follows. ◀

3.2 FPT algorithm for Modular Width

Since an algebraic expression A of width $\operatorname{mw}(G)$ can be computed in linear time [36], we can assume that we have A on the input. We construct the rooted ordered tree $\mathcal{T}$ corresponding to A. Each node $t \in \mathcal{T}$ is assigned a graph $G^t \subseteq G$, that is, the graph constructed by the subexpression of A rooted at t. Suppose we are performing substitution operation at node t with respect to template graph T and graphs $G_1, \ldots, G_r$. Denote the resulting graph G^t and denote by n_i the size of $V(G_i)$.

Proof sketch of Theorem 3. The computation will be carried out recursively from the bottom of the tree $\mathcal{T}$. We first describe the structure of all vertex covers C in G^t. Observe that if $ij \in E(T)$, then at least one of $V(G_i), V(G_j)$ must be a subset of C. Thus, the set $C_T := \{i : V(G_i) \subseteq C\}$ is a vertex cover of the template graph T. We call the C_T the *type of the vertex cover* C. Furthermore, every set $C \cap V(G_i)$ must be a vertex cover of G_i. Since there are at most 2^r vertex covers of T, we try all of these. Furthermore, every set $C \cap V(G_i)$ must be a vertex cover of G_i.

We now describe the fair cost of the cover C in terms of fair costs and sizes of the sets $C \cap V(G_i)$. Let $c_i = |C \cap V(G_i)|$ and let f_i denote the fair cost of $C \cap V(G_i)$ in G_i. The *fair cost of C in $W \subseteq V(G)$* is defined as $\max_{v \in W} |C \cap N(v)|$. For $i \in [r]$ the fair cost of C in $V(G_i)$ can be expressed as $f_i + \sum_{j : ij \in E(T)} c_j$. If we know the type C_T of the cover C, this expression can be simplified based on whether i lies in C_T or not. If $i \in C_T$, then f_i is $\Delta(G_i)$ (the maximal degree of G_i). If, on the other hand, $i \notin C_T$, then all its neighbors j are in C_T and in this case $c_j = n_j$. Combining these observations together we get

$$\text{fair cost of } C \text{ in } G_i = \begin{cases} \Delta(G_i) + \sum_{j \notin C_T : ij \in E(T)} c_j + \sum_{j \in C_T : ij \in E(T)} n_j & i \in C_T, \\ f_i + \sum_{j : ij \in E(T)} n_j & i \notin C_T. \end{cases}$$

From this we can design a dynamic program that computes the solution. ◀

4 Conclusions

Fair Edge L Deletion problems. One can define *edge deletion problems* in a similar way as vertex deletion problems.

FAIR L EDGE DELETION
Input: An undirected graph $G = (V, E)$, an L sentence φ, and a positive integer k.
Question: Is there a set $F \subseteq E$ such that $G \setminus F \models \varphi$ and for every vertex v of G, it holds that $|\{e \in F : e \ni v\}| \le k$?

Recall, in dense graph classes one cannot obtain an MSO_2 model checking algorithm running in FPT-time [25]. This is the reason why evaluation problems do not make sense in this context. In sparse graph classes, this problem was studied in [29] where W[1]-hardness was obtained for Fair FO Edge Deletion on graphs of bounded treedepth and FPT algorithm was derived for Fair MSO_2 Edge Evaluation on graphs of bounded vertex cover.

The crucial open problem is to resolve the parameterized complexity of the FAIR FO EDGE DELETION problems for parameterization by neighborhood diversity and twin cover. To motivate the study we prove the following hardness result for parameterization by the cluster vertex deletion number. Recall that for any graph its cluster vertex deletion number is upper-bounded by the size of its twin cover.

▶ **Theorem 14** (★). *The* FAIR FO EDGE DELETION *problem is* W[1]*-hard when parameterized by the cluster vertex deletion number of the input graph.*

Generalization of parameters. Another open problem is to resolve the parameterized complexity of the FAIR MSO_1 VERTEX EVALUATION problems with respect to graph parameters generalizing neighborhood diversity or twin cover (e.g., modular width or cluster vertex deletion number respectively).

MSO with Local Linear Constraints. Previously, an FPT algorithm for evaluation of a fair objective was given for parameter neighborhood diversity [29]. That algorithm was extended [20] to a so-called *local linear constraints* again for a formula $\varphi(\cdot)$ with one free variable that is defined as follows. Every vertex $v \in V(G)$ is accompanied with two positive integers $\ell(v), u(v)$, the lower and the upper bound, and the task is to find a set X that not only $G \models \varphi(X)$ but for each $v \in V(G)$ it holds that $\ell(v) \le |N(v) \cap X| \le u(v)$. Note that this is a generalization as fair objective of value t can be tested in this framework by setting $\ell(v) = 0$ and $u(v) = t$ for every $v \in V(G)$. Is this extension possible for parameterization by the twin cover number?

Towards new fair problems. As we proposed the examination of Fair VC already, we would like to turn an attention to exploring fair versions of other classical and well-studied Vertex Deletion problems. In contrast, certain Fair Edge Deletion problems have got some attention before, namely Fair Feedback Edge Set [27] or Fair Edge Odd Cycle Transversal [22]. Besides Fair VC we propose a study of Fair Dominating Set and Fair Feedback Vertex Set. In particular, it would be really interesting to know whether fair variants of Vertex Cover and Dominating Set admit a similar behavior as in the classical setting.

Furthemore, We would like to ask whether there is an NP-hard Fair Vertex Deletion problem that admits an FPT algorithm for parameterization by treedepth (and feedback vertex set) of the input graph.

References

1 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy Problems for Tree-Decomposable Graphs. *Journal of Algorithms*, 12(2):308–340, June 1991. doi:10.1016/0196-6774(91)90006-k.

2 Jianer Chen, Benny Chor, Michael R. Fellows, Xiuzhen Huang, David Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computation*, 201(2):216–231, 2005. doi:10.1016/j.ic.2005.05.001.

3 Morgan Chopin, André Nichterlein, Rolf Niedermeier, and Mathias Weller. Constant Thresholds Can Make Target Set Selection Tractable. *Theory Comput. Syst.*, 55(1):61–83, 2014. doi:10.1007/s00224-013-9499-3.

4 Bruno Courcelle. The Monadic Second-order Logic of Graphs. I. Recognizable Sets of Finite Graphs. *Information and Computation*, 85(1):12–75, March 1990. doi:10.1016/0890-5401(90)90043-h.

5 Bruno Courcelle and Mohamed Mosbah. Monadic Second-Order Evaluations on Tree-Decomposable Graphs. *Theor. Comput. Sci.*, 109(1&2):49–82, 1993. doi:10.1016/0304-3975(93)90064-z.

6 Lenore J. Cowen, Robert Cowen, and Douglas R. Woodall. Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valency. *Journal of Graph Theory*, 10(2):187–195, 1986. doi:10.1002/jgt.3190100207.

7 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

8 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

9 Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic*, 130(1):3–31, 2004. Papers presented at the 2002 IEEE Symposium on Logic in Computer Science (LICS). doi:10.1016/j.apal.2004.01.007.

10 Jakub Gajarský and Petr Hliněný. Kernelizing MSO Properties of Trees of Fixed Height, and Some Consequences. *Logical Methods in Computer Science*, Volume 11, Issue 1, April 2015. doi:10.2168/LMCS-11(1:19)2015.

11 Jakub Gajarský, Petr Hliněný, Jan Obdržálek, Daniel Lokshtanov, and M. S. Ramanujan. A New Perspective on FO Model Checking of Dense Graph Classes. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5–8, 2016*, pages 176–184. ACM, 2016. doi:10.1145/2933575.2935314.

12 Robert Ganian. Twin-Cover: Beyond Vertex Cover in Parameterized Algorithmics. In Dániel Marx and Peter Rossmanith, editors, *Parameterized and Exact Computation - 6th International Symposium IPEC 2011, Saarbrücken, Germany, September 6-8, 2011. Revised*

Selected Papers, volume 7112 of *Lecture Notes in Computer Science*, pages 259–271. Springer, 2011. doi:10.1007/978-3-642-28050-4_21.

13 Robert Ganian. Improving Vertex Cover as a Graph Parameter. *Discrete Mathematics & Theoretical Computer Science*, 17(2):77–100, 2015. URL: http://dmtcs.episciences.org/2136.

14 Robert Ganian, Petr Hliněný, Jaroslav Nešetřil, Jan Obdržálek, and Patrice Ossona de Mendez. Shrub-depth: Capturing Height of Dense Graphs. *Logical Methods in Computer Science*, 15(1), 2019. URL: https://lmcs.episciences.org/5149.

15 Robert Ganian and Jan Obdržálek. Expanding the Expressive Power of Monadic Second-Order Logic on Restricted Graph Classes. In Thierry Lecroq and Laurent Mouchard, editors, *Combinatorial Algorithms—24th International Workshop, IWOCA 2013, Revised Selected Papers*, volume 8288 of *Lecture Notes in Computer Science*, pages 164–177. Springer, 2013. doi:10.1007/978-3-642-45278-9_15.

16 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding First-Order Properties of Nowhere Dense Graphs. *J. ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095.

17 Frédéric Havet, Ross J. Kang, and Jean-Sébastien Sereni. Improper coloring of unit disk graphs. *Networks*, 54(3):150–164, 2009. doi:10.1002/net.20318.

18 Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *J. Comput. Syst. Sci.*, 79(1):39–49, 2013. doi:10.1016/j.jcss.2012.04.004.

19 Ross J. Kang, Tobias Müller, and Jean-Sébastien Sereni. Improper colouring of (random) unit disk graphs. *Discrete Mathematics*, 308(8):1438–1454, 2008. Third European Conference on Combinatorics. doi:10.1016/j.disc.2007.07.070.

20 Dušan Knop, Martin Koutecký, Tomáš Masařík, and Tomáš Toufar. Simplified Algorithmic Metatheorems Beyond MSO: Treewidth and Neighborhood Diversity. In Hans L. Bodlaender and Gerhard J. Woeginger, editors, *Graph-Theoretic Concepts in Computer Science: 43rd International Workshop, WG 2017, Eindhoven, The Netherlands, June 21-23, 2017, Revised Selected Papers*, pages 344–357, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-68705-6_26.

21 Petr Kolman, Bernard Lidický, and Jean-Sébastien Sereni. Fair Edge Deletion Problems on TreeDecomposable Graphs and Improper Colorings, 2010. URL: http://orion.math.iastate.edu/lidicky/pub/kls10.pdf.

22 Petr Kolman, Bernard Lidický, and Jean-Sébastien Sereni. On Fair Edge Deletion Problems, 2009. URL: https://kam.mff.cuni.cz/~kolman/papers/kls09.pdf.

23 Juha Kontinen and Hannu Niemistö. Extensions of MSO and the monadic counting hierarchy. *Information and Computation*, 209(1):1–19, 2011. doi:10.1016/j.ic.2010.09.002.

24 Michael Lampis. Algorithmic Meta-theorems for Restrictions of Treewidth. *Algorithmica*, 64(1):19–37, 2012. doi:10.1007/s00453-011-9554-x.

25 Michael Lampis. Model Checking Lower Bounds for Simple Graphs. *Logical Methods in Computer Science*, 10(1):1–21, 2014. doi:10.2168/LMCS-10(1:18)2014.

26 Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

27 Li-Shin Lin and Sartaj Sahni. Fair Edge Deletion Problems. *IEEE Trans. Comput.*, 38(5):756–761, 1989. doi:10.1109/12.24280.

28 Tomáš Masařík and Tomáš Toufar. Parameterized Complexity of Fair Deletion Problems. In T.V. Gopal, Gerhard Jäger, and Silvia Steila, editors, *Theory and Applications of Models of Computation: 14th Annual Conference, TAMC 2017, Bern, Switzerland, April 20-22, 2017, Proceedings*, pages 628–642, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-55911-7_45.

29 Tomáš Masařík and Tomáš Toufar. Parameterized complexity of fair deletion problems. *Discrete Applied Mathematics*, 2019. doi:10.1016/j.dam.2019.06.001.

30 Jiří Matoušek and Jaroslav Nešetřil. *Invitation to Discrete Mathematics (2. ed.)*. Oxford University Press, 2009.

31 Jaroslav Nešetřil and Patrice Ossona De Mendez. *Sparsity: graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012. doi:10.1007/978-3-642-27875-4.

32 Timm Oertel, Christian Wagner, and Robert Weismantel. Integer convex minimization by mixed integer linear optimization. *Operations Research Letters*, 42(6):424–428, 2014. doi:10.1016/j.orl.2014.07.005.

33 Michał Pilipczuk. Problems Parameterized by Treewidth Tractable in Single Exponential Time: A Logical Approach. In Filip Murlak and Piotr Sankowski, editors, *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, volume 6907 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2011. doi:10.1007/978-3-642-22993-0_47.

34 Detlef Seese. Linear Time Computable Problems and First-Order Descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.

35 Stefan Szeider. Monadic second order logic on graphs with local cardinality constraints. *ACM Trans. Comput. Log*, 12(2):1–21, 2011. doi:10.1145/1877714.1877718.

36 Marc Tedder, Dereck G. Corneil, Michel Habib, and Christophe Paul. Simpler Linear-Time Modular Decomposition Via Recursive Factorizing Permutations. In *ICALP 2008*, pages 634–645, 2008. doi:10.1007/978-3-540-70575-8_52.

37 Mihalis Yannakakis. Edge-Deletion Problems. *SIAM J. Comput.*, 10(2):297–309, 1981. doi:10.1137/0210021.