Extended MSO Model Checking via Small Vertex Integrity

Tatsuya Gima
Nagoya University, Japan

Yota Otachi
Nagoya University, Japan

Abstract

We study the model checking problem of an extended MSO with local and global cardinality constraints, called $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$, introduced recently by Knop, Koutecký, Masařík, and Toufar [*Log. Methods Comput. Sci.*, 15(4), 2019]. We show that the problem is fixed-parameter tractable parameterized by vertex integrity, where vertex integrity is a graph parameter standing between vertex cover number and treedepth. Our result thus narrows the gap between the fixed-parameter tractability parameterized by vertex cover number and the W[1]-hardness parameterized by treedepth.

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

Keywords and phrases vertex integrity, monadic second-order logic, cardinality constraint, fixed-parameter tractability

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.20

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2207.01024`

Funding *Yota Otachi*: JSPS KAKENHI Grant Numbers JP18H04091, JP18K11168, JP18K11169, JP20H05793, JP21K11752, JP22H00513.

Acknowledgements The authors thank Michael Lampis and Valia Mitsou for fruitful discussions and sharing a preliminary version of [37].

1 Introduction

One of the most successful goals in algorithm theory is to have a meta-theorem that constructs an efficient algorithm from a *description* of a target problem in a certain format (see e.g., [29, 30, 35]). Courcelle's theorem [6, 8, 9, 11] is arguably the most successful example of such an algorithmic meta-theorem, which says (with Bodlaender's algorithm [4]) that: if a problem on graphs can be expressed in monadic second-order logic (MSO), then the problem can be solved in linear time on graphs of bounded treewidth. Many natural problems that are NP-hard on general graphs are shown to have expressions in MSO and thus have linear-time algorithms on graphs of bounded treewidth [1].

Although the expressive power of MSO captures many problems, it is known that MSO cannot represent some kinds of cardinality constraints [10]. For example, it is easy to express the problem of finding a proper vertex coloring with r colors in MSO as the existence of a partition of the vertex set into r independent sets, where the length of the corresponding MSO formula depends on r. However, the variant of the problem that additionally requires the r independent sets to be of the same size cannot be expressed in MSO even if $r = 2$ (see [10]). Indeed, this problem is known to be W[1]-hard parameterized by r and treewidth [16].[1] See [3, 28, 41] for many other examples of such problems.

[1] We assume that the readers are familiar with the concept of parameterized complexity. For standard definitions, see e.g., [12].

33rd International Symposium on Algorithms and Computation (ISAAC 2022).
Editors: Sang Won Bae and Heejin Park; Article No. 20; pp. 20:1–20:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

For those problems that do not admit MSO expressions and are hard on graphs of bounded treewidth, there is a successful line of studies on smaller graph classes with more restricted structures. For example, by techniques tailored for individual problems, several problems are shown to be tractable on graphs of bounded vertex cover number (see e.g., [15, 17, 18]). Such results are known also for more general parameters such as twin-cover [24], neighborhood diversity [36], and vertex integrity [28]. Then the natural challenge would be finding a meta-theorem covering (at least some of) such results. Recently, such meta-theorems are intensively studied for extended MSO logics with "cardinality constraints." In this paper, we follow this line of research and focus on vertex integrity as the structural parameter of input graphs. The *vertex integrity* of a graph is the smallest number $k = s + t$ such that by removing s vertices of the graph, every component can be made to have at most t vertices. The concept of vertex integrity was introduced first in the context of network vulnerability [2]. It basically measures how difficult it is to break a graph into small components by removing a small number of vertices. This can be seen as a generalization of vertex cover number, which asks to remove vertices to make the graph edge-less (corresponding to the case $t = 1$ of the definition of vertex integrity). On the other hand, the concept of treedepth can be seen as a recursive generalization of vertex integrity. This is because we can define the treedepth of a graph recursively as follows: the treedepth of a graph is 1 if it has no edges, and otherwise it is the minimum number $k = s + t$ such that s vertices can be removed from the graph to make the treedepth of every component of the remaining graph at most t. Actually, their definitions give us the inequality treedepth $\leq$ vertex integrity $\leq$ vertex cover number $-$ 1 for every graph (see [28]).

There is another issue about Courcelle's theorem that the dependency of the running time on the parameters (the treewidth of the input graph and the length of formula) is quite high [20]. To cope with this issue, faster algorithms are proposed for special cases such as vertex cover number, neighborhood diversity, and max-leaf number [36], twin-cover [24], shrubdepth [25], treedepth [23], and vertex integrity [37]. The methods in these results are similar in the sense that they find a smaller part of the input graph that is equivalent to the original graph under the given MSO formula. Interestingly, these techniques are used also in studies of extended MSO logics in these special cases. Our study is no exception, and we use a result in [37] as a key lemma.

Meta-theorems on extended MSO with cardinality constraints

In this direction, there are two different lines of research, which have been merged recently. One line considers "global" cardinality constraints and the other considers "local" cardinality constraints.

Recall that the property of having a partition into r independent sets of equal size cannot be expressed in MSO. A remedy for this would be to allow a predicate like $|X| = |Y|$. The concept of global cardinality constraints basically implements this but in a more general way (see Section 2 for formal definitions). It is known that the model checking for the extended MSO logic with global cardinality constraints is fixed-parameter tractable parameterized by neighborhood diversity [27].

The concept of local cardinality constraints was originally introduced as the *fairness* of a solution [39]. The fairness of a solution (a vertex set or an edge set) upper-bounds the number of neighbors each vertex can have in the solution. It is known that finding a vertex cover with an upper bound on the fairness is W[1]-hard parameterized by treedepth and feedback vertex set number [33]. On the other hand, the problem of finding a vertex set satisfying an MSO formula and fairness constraints is fixed-parameter tractable parameterized by neighborhood diversity [40] and by twin-cover [33]. The general concept of local cardinality constraint extends the concept of fairness by having for each vertex, an individual set of the allowed numbers of neighbors in the solution. It is known that the extension of MSO with local cardinality constraints admits an XP algorithm (i.e., a slicewise-polynomial time algorithm) parameterized by treewidth [42].

Knop, Koutecký, Masařík, and Toufar [34] recently converged two lines and studied the model checking of extended MSO with both local and global cardinality constraints. It is shown that the problem admits an XP algorithm parameterized by treewidth. Furthermore, they showed that the problem is fixed-parameter tractable parameterized by neighborhood diversity if the cardinality constraints are "linear," where each local cardinality constraint is a set of consecutive integers and each global cardinality constraint is a linear inequality.

Our results

We study the linear version of the problem in [34] mentioned above; that is, the model checking of the extended MSO logic with linear local and global cardinality constraints. We show that this problem, called $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking, is fixed-parameter tractable parameterized by vertex integrity. This result fills a missing part in the map on the complexity of $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking as vertex integrity fits between vertex cover number and treedepth: for the former the problem is fixed-parameter tractable [34], and for the latter it is W[1]-hard [33]. Note that by MSO, we mean MSO_1, which does not allow edge and edge-set variables. After proving the main result, we show that the same result holds even for the same extension of MSO_2. We apply the results to several problems and show some new examples that are fixed-parameter tractable parameterized by vertex integrity. We also show that some known results can be obtained as applications of our results.[2]

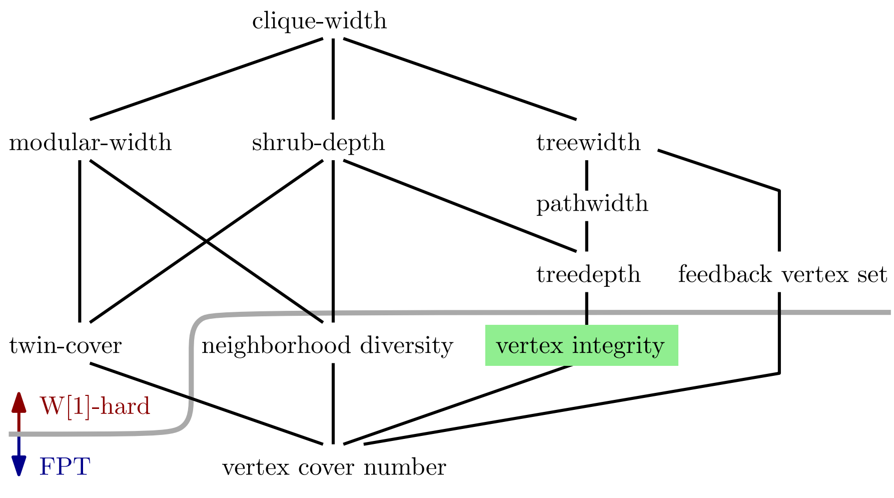

Figure 1 Some of the major graph parameters and the complexity of $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking. If one parameter is an ancestor of another, then the ancestor is upper-bounded by a function of the descendant. The fixed-parameter tractability parameterized by neighborhood diversity is shown in [34]. The W[1]-hardness parameterized by twin-cover and by treedepth and feedback vertex set number are shown in [33].

[2] Omitted from the conference version. See the full version.

2 Preliminaries

For two integers a and b, we define $[a, b] = \{x \in \mathbb{Z} \mid a \leq x \leq b\}$. We write $[b]$ for the set $[1, b]$. For two tuples $\boldsymbol{A} = (A_1, \ldots, A_p)$ and $\boldsymbol{B} = (B_1, \ldots, B_q)$ of vertex sets, the concatenation $(A_1, \ldots, A_p, B_1, \ldots, B_q)$ is denoted by $\boldsymbol{A} \dotplus \boldsymbol{B}$. For a function $f : X \to Y$ and a set $A \subseteq X$, the restriction of f to A is denoted by $f|_A$.

2.1 Graphs and colored graphs

We consider undirected graphs without self-loops or multiple edges. Let $G = (V, E)$ be a graph. The vertex set and the edge set of G are denoted by $V(G)$ and $E(G)$, respectively. A *component* of G is a maximal connected induced subgraph of G. For a vertex set S of a graph G, the subgraph of G induced by $V \setminus S$ is denoted by $G - S$.

A *p-color list* $\boldsymbol{C}$ of G is a tuple $\boldsymbol{C} = (C_1, \ldots, C_p)$ of p vertex sets $C_i \subseteq V$. Denote the set of colors assigned by $\boldsymbol{C}$ to $v \in V$ by $\mathrm{col}_{\boldsymbol{C}}(v)$. Note that each vertex can have multiple colors. That is, $\mathrm{col}_{\boldsymbol{C}}(v) = \{i \in [p] \mid v \in C_i\}$. Note that $\mathrm{col}_{\boldsymbol{C}}(v)$ can be computed in time polynomial in $|V|$ and p. We call a tuple $(G, \boldsymbol{C})$ a *p-colored graph*. If the context is clear, we simply call it a graph.

Let $\mathcal{G}_1 = (G_1, \boldsymbol{C}_1)$ and $\mathcal{G}_2 = (G_2, \boldsymbol{C}_2)$ be p-colored graphs. A bijection $\psi : V(G_1) \to V(G_2)$ is an *isomorphism* from $\mathcal{G}_1$ to $\mathcal{G}_2$ if ψ satisfies the following conditions:

- $\{u, v\} \in E(G_1)$ if and only if $\{\psi(u), \psi(v)\} \in E(G_2)$ for all $u, v \in V(G_1)$;
- $\mathrm{col}_{\boldsymbol{C}_1}(v) = \mathrm{col}_{\boldsymbol{C}_2}(\psi(v))$ for all $v \in V(G_1)$.

We say that $\mathcal{G}_1$ and $\mathcal{G}_2$ are *isomorphic* if such ψ exists.

2.2 Vertex integrity

A $\mathsf{vi}(k)$*-set* S of a graph G is a set of vertices such that the number of vertices of every component of $G - S$ is at most $k - |S|$. The *vertex integrity* of a graph G, denoted by $\mathsf{vi}(G)$, is the minimum integer k such that there is a $\mathsf{vi}(k)$-set of G. In other words, it can be defined as follows:

$$\mathsf{vi}(G) = \min_{S \subseteq V(G)} \left\{ |S| + \max_{C \in \mathrm{cc}(G-S)} |V(C)| \right\},$$

where $\mathrm{cc}(G - S)$ is the set of connected components of $G - S$. A $\mathsf{vi}(k)$-set of G, if any exists, can be found in $O(k^{k+1}n)$ time [13], where n is the number of vertices in G.

As mentioned above, the concept of vertex integrity was originally introduced in the context of network vulnerability [2], but recently it and its close relatives are used as structural parameters in algorithmic studies. The *safe number* was introduced with a similar motivation [22] and later shown to be (non-trivially) equivalent to the vertex integrity in the sense that the safe number is bounded if and only if so is the vertex integrity for every graph [21]. The definition of *fracture number* is almost the same as the one for vertex integrity, where the only difference is that it asks the maximum (instead of the sum) of the orders of S and a maximum component of $G - S$ to be bounded by k. The *ℓ-component order connectivity* [13] measures the size of S and the maximum order of a component of $G - S$ separately, and defined to be the minimum size k of a set S such that each component of $G - S$ has order at most ℓ. For example, 1-component order connectivity is exactly the vertex cover number. Also, the 2-component order connectivity is studied as the matching-splittability [31]. A graph has vertex integrity at most k if and only if the graph has ℓ-component order connectivity at most $k - \ell$ for some ℓ.

The fracture number was used to design efficient algorithms for Integer Linear Programming [14], Bounded-Degree Vertex Deletion [26], and Locally Constrained Homomorphism [7]. The vertex integrity was used in the context of subgraph isomorphism on minor-closed graph classes [5], and then used to design algorithms for several problems that are hard on graphs of bounded treedepth such as Capacitated Dominating Set, Capacitated Vertex Cover, Equitable Coloring, Equitable Connected Partition[3], Imbalance, Maximum Common (Induced) Subgraph, and Precoloring Extension [28]. A faster algorithm for MSO Model Checking parameterized by vertex integrity is also known [37].

2.3 Monadic second-order logic

A *monadic second-order logic formula* (an MSO *formula*, for short) over p-colored graphs is a formula that matches one of the following, where x and y denote vertex variables, X denotes a vertex-set variable, C_i denotes a vertex-set constant (color): $E(x, y)$; $x = y$; $x \in X$; $x \in C_i$; $\exists x.\varphi$, $\forall x.\varphi$, $\exists X.\varphi$, $\forall X.\varphi$, $\varphi \land \psi$, $\varphi \lor \psi$, and $\neg\varphi$, where φ and ψ are MSO formulas. These symbols have the following semantic meaning: $E(x, y)$ means that x and y are adjacent; and the others are the usual ones. Additionally, for convenience, we introduce MSO symbols `true` and `false` that are always interpreted as true and false, respectively. Note that this version of MSO is often called MSO_1. In Section 4, we consider a variant called MSO_2, which has stronger expression power.

A variable is *bound* if it is quantified and *free* otherwise. An MSO formula is *closed* if it has no free variables and *open* otherwise. We assume that every free variable is a set variable, because a free vertex variable can be simulated by a free vertex-set variable with an MSO formula expressing that the set is of size 1. An *assignment* of an open MSO formula φ with s free set variables over G is a tuple $\boldsymbol{X}^G = (X_1^G, \dots, X_s^G)$ of s vertex sets $X_i^G \subseteq V(G)$. Let $\mathcal{G}$ be a p-colored graph, and φ be an MSO formula. If φ is closed, we write $\mathcal{G} \models \varphi$ if $\mathcal{G}$ satisfies the property expressed by φ. Otherwise, we write $(\mathcal{G}, \boldsymbol{X}^G) \models \varphi$ where $\boldsymbol{X}^G$ is an assignment of φ if $\mathcal{G}$ and $\boldsymbol{X}^G$ satisfies the property expressed by φ.

From the definition of MSO, one can see that no MSO formula can distinguish isomorphic p-colored graphs. See e.g., [37] for a detailed proof.

▶ **Lemma 2.1** (Folklore). *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be isomorphic p-colored graphs. For every* MSO *formula φ, we have $\mathcal{G}_1 \models \varphi$ if and only if $\mathcal{G}_2 \models \varphi$.*

2.4 Extensions of MSO

We introduce an extension of MSO proposed by Knop, Koutecký, Masařík, and Toufar [34]. Let φ be an MSO formula with s free set variables $X_1, \dots, X_s$, and G be a graph with n vertices.

We introduce a linear constraint on the cardinalities of vertex sets $|X_i|$. A *global linear cardinality constraint* is an s-ary relation R expressed by a linear inequality $a_1|X_1| + \cdots + a_s|X_s| \leq b$, where a_i and b are integers and the arguments X_i are the free variables of φ. In the extension of MSO introduced later, global cardinality constraints are used as atomic formulas.

[3] In [15], Equitable Connected Partition was shown to be W[1]-hard parameterized simultaneously by pathwidth, feedback vertex set number, and the number of parts. In the full version, we strengthen the W[1]-hardness by replacing pathwidth in the parameter with treedepth.

A *local linear cardinality constraint* of G on φ is a mapping $\alpha_i^G\colon V(G) \to 2^{[n]}$, where $\alpha_i^G(v) = [l_i^v, u_i^v]$ with some integers l_i^v and u_i^v. Each α_i^G is a constraint on the number of neighbors of each vertex that are in X_i. We say that an assignment $\boldsymbol{X}^G = (X_1^G, \dots X_s^G)$ *obeys* a tuple $\boldsymbol{\alpha}^G = (\alpha_1^G, \dots, \alpha_s^G)$ of local linear cardinality constraints if $|X_i \cap N(v)| \in \alpha_i^G(v)$ for all $v \in V(G)$ and $i \in [s]$.

An $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ *formula* on a p-colored graph $\mathcal{G} = (G, \boldsymbol{C})$ is a tuple $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$ where φ, $\boldsymbol{R}$, and $\boldsymbol{\alpha}^G$ are defined as follows. The tuple $\boldsymbol{R} = (R_1, \dots, R_g)$ is a tuple of global linear cardinality constraints, and $\boldsymbol{\alpha}^G = (\alpha_1^G, \dots, \alpha_s^G)$ is a tuple of local linear cardinality constraints. The formula φ is an MSO formula with s free set variables that additionally has the g global linear cardinality constraints R_i as symbols. Now we write $(\mathcal{G}, \boldsymbol{R}, \boldsymbol{X}^G) \models \varphi$ if $(\mathcal{G}, \boldsymbol{X}^G) \models \varphi'$ where φ' is an ordinary MSO formula obtained from φ by replacing every symbol R_i with the symbol `true` or `false` representing the truth value of the formula $(|X_1^G|, \dots, |X_s^G|) \in R_i$.

Our problem, $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking, is defined as follows.

$\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking
Input: A p-colored graph $\mathcal{G} = (G, \boldsymbol{C})$ and an $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$.
Question: Is there an assignment $\boldsymbol{X}^G = (X_1^G, \dots, X_s^G)$ of φ such that $(\mathcal{G}, \boldsymbol{R}, \boldsymbol{X}^G) \models \varphi$ and $\boldsymbol{X}^G$ obeys $\boldsymbol{\alpha}^G$?

It is known that $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking is fixed-parameter tractable parameterized by neighborhood diversity [34], W[1]-hard parameterized by treedepth and feedback vertex set [34], and W[1]-hard parameterized by twin-cover [33].

3 Model checking algorithm

In this section, we present our main result, the fixed-parameter algorithm for $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking parameterized by vertex integrity. Before going into the details, let us sketch the rough and intuitive ideas of the algorithm. Recall that our goal is to find a tuple of vertex sets in a graph of bounded vertex integrity that satisfies

- an MSO formula φ equipped with global linear cardinality constraints, and
- local linear cardinality constraints.

We first show that for the ordinary MSO Model Checking with a fixed formula on graphs of bounded vertex integrity, there is a small number of equivalence classes, called *shapes*, of tuples of vertex sets such that two tuples of the same shape are equivalent under the formula. To use the concept of shapes, we remove the global constraints from φ by replacing each of them with a guessed truth value and we find a solution that meets the guesses. Let φ' be the resultant (ordinary) MSO formula. We guess the shape of the solution and check whether a tuple with the guessed shape satisfies φ' using known efficient algorithms. If the guessed shape passed this test, then we check whether there is a tuple with the shape satisfying the global and local cardinality constraints. We can do this by expressing the rest of the problem as an integer linear programming (ILP) formula as often done for similar problems (see e.g., [34]). The ILP formula we construct has constraints for forcing a solution to be found

- to have the guessed shape,
- to satisfy the guessed global cardinality constraints, and
- to satisfy the local cardinality constraints.

The first two will be straightforward from the definitions given below. For the local cardinality constraints, we observe that after guessing the intersections of a $\mathsf{vi}(k)$-set S and each set in the solution, we know whether all vertices in $V(G) - S$ obeys the local cardinality constraints.

Thus we only need to express the local cardinality constraints in ILP for the vertices in S. Finally, we will observe that the number of variables and constraints in the constructed ILP formula depends only on k and $|\varphi|$. This will give us the desired result.

In the next subsections, we formally describe and prove the ideas explained above.

3.1 MSO model checking

Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph and S be a subset of $V(G)$. We define an equivalence relation of the components of $G - S$ as follows. Two components A_1 and A_2 of $G - S$ have the same $(\mathcal{G}, S)$-*type* if there is an isomorphism ψ from $(G[S \cup A_1], \boldsymbol{C}|_{S \cup A_1})$ to $(G[S \cup A_2], \boldsymbol{C}|_{S \cup A_2})$ such that the restriction $\psi|_S$ is the identity function. We call such an isomorphism a $(\mathcal{G}, S)$-*type isomorphism*. Clearly, having the same type is an equivalence relation. We say that a component A of $G - S$ is of $(\mathcal{G}, S)$-type t (or just type t) by using a canonical form t of the members of the $(\mathcal{G}, S)$-type equivalence class of A. Denote by $\tau_{(\mathcal{G},S)}(A)$ the type of a component A of $G - S$. We will omit the index $(\mathcal{G}, S)$ if it is clear from the context.

We define the canonical form of a $(\mathcal{G}, S)$-type as the "lexicographically" smallest one in the equivalence class in some sense (see [28] for such canonical forms of uncolored graphs). If S is a $\mathsf{vi}(k)$-set, then in time depending only on $p + k$ we can compute the canonical form of the equivalence class that a component A of $G - S$ belongs to. Thus we can compute (the canonical forms of) all $(\mathcal{G}, S)$-types in time $f(p + k)|G|^{O(1)}$ for some computable function f. Furthermore, in time $f'(p + s + k)|G|^{O(1)}$ for some computable function f', we can compute all $(\mathcal{G}', S)$-types for all $\mathcal{G}'$ obtained from $\mathcal{G}$ by adding s new colors; that is, $\mathcal{G}' = (G, \boldsymbol{C} \dotplus \boldsymbol{X})$ for some $\boldsymbol{X} \in (V(G))^s$.

The next lemma, due to Lampis and Mitsou [37], is one of the main ingredients of our algorithm. It basically says that in the ordinary MSO MODEL CHECKING we can ignore some part of a graph if it has too many parts that have the same type.

▶ **Lemma 3.1** ([37]). *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph G, $S \subseteq V$, A be a component of $G - S$, $|A| \le k$, and φ be a closed* MSO *formula with q quantifiers. If there are at least $2^{kq} + 1$ type $\tau(A)$ components in $G - S$, then $(G, \boldsymbol{C}) \models \varphi$ if and only if $(G - V(A), \boldsymbol{C}') \models \varphi$, where $\boldsymbol{C}'$ is the restriction of $\boldsymbol{C}$ to $V(G) \setminus V(A)$.*

Lemma 3.1 leads to the following concept "shape", which can be seen as equivalence classes of assignments.

▶ **Definition 3.2** (Shape). *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph, S be a* $\mathsf{vi}(k)$*-set of G, and φ be an* MSO *formula with s free set variables $(X_1, \ldots, X_s)$ and q quantifiers. Let $\mathcal{T}$ be the set of all $(\mathcal{G}, S)$-types, and $\mathcal{T}'$ be the set of all possible $(\mathcal{G}', S)$-types in $(p + s)$-colored graphs $\mathcal{G}'$ obtained from $\mathcal{G}$ by adding s new colors.*

*An S-*shape *is the pair (σ_S, σ) of a function $\sigma_S \colon S \to 2^{\{X_1,\ldots,X_s\}}$ and a function $\sigma \colon \mathcal{T}' \to [0, 2^{kq}] \cup \{\top\}$.*

*Let $\boldsymbol{X}^G = (X_1^G, \ldots, X_s^G)$ be an assignment of φ, and $\mathcal{G}' = (G, \boldsymbol{C} \dotplus \boldsymbol{X}^G)$. The S-*shape *of $\boldsymbol{X}^G$ is (σ_S, σ) if the following conditions are satisfied:*

- *for each $i \in [s]$ and $v \in S$, $X_i \in \sigma_S(v)$ if and only if $v \in X_i^G$;*
- *for each $t' \in \mathcal{T}'$,*

$$\sigma(t') = \begin{cases} c(t') & c(t') \le 2^{kq}, \\ \top & \textit{otherwise}, \end{cases}$$

where $c(t')$ is the number of $(\mathcal{G}', S)$-type t' components of $G - S$.

Let (σ_S, σ) be an S-shape. If there is an assignment $\boldsymbol{X}^G$ of φ such that the S-shape of $\boldsymbol{X}^G$ is (σ_S, σ), we say that the S-shape (σ_S, σ) is *valid*.

The following lemma indicates that S-shapes act as a sort of equivalence classes.

▶ **Lemma 3.3.** *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph, S be a* vi(k)*-set of G, and φ be an* MSO *formula with s free set variables $(X_1, \dots, X_s)$. Let $\boldsymbol{X}^G$ and $\boldsymbol{Y}^G$ be assignments of φ such that their shapes are equal. Then, $(\mathcal{G}, \boldsymbol{X}^G) \models \varphi$ if and only if $(\mathcal{G}, \boldsymbol{Y}^G) \models \varphi$.*

Proof. Let $\mathcal{G}_X = (G, \boldsymbol{C} \dotplus \boldsymbol{X}^G)$, $\mathcal{G}_Y = (G, \boldsymbol{C} \dotplus \boldsymbol{Y}^G)$, and (σ_S, σ) be the S-shape of $\boldsymbol{X}^G$ (and of $\boldsymbol{Y}^G$). Then φ can be seen as a closed MSO formula for the $(p+s)$-colored graphs $\mathcal{G}_X$ and $\mathcal{G}_Y$. Thus we can apply Lemma 3.1 to $\mathcal{G}_X$, S, and φ, and obtain a graph $\mathcal{G}'_X = (G'_X, \boldsymbol{C}'_X)$, such that $\mathcal{G}_X \models \varphi$ if and only if $\mathcal{G}'_X \models \varphi$ and the number of each type t components of $G'_X - S$ is at most 2^{kq}, where q is the number of quantifiers in φ. We also obtain a graph $\mathcal{G}'_Y$ in the same way as for $\mathcal{G}'_X$. This reduction does not delete any vertex of S. The number of components for each type t of $G'_X - S$ or $G'_Y - S$ is $\sigma(t)$ if $\sigma(t) \neq \top$ and 2^{kq} if $\sigma(t) = \top$. Therefore, there is an isomorphism from $\mathcal{G}'_X$ to $\mathcal{G}'_Y$, and thus $\mathcal{G}'_X \models \varphi$ if and only if $\mathcal{G}'_Y \models \varphi$ by Lemma 2.1. ◀

Now, we estimate the number of candidates for S-shapes. Observe that in Definition 3.2, the number of candidates for σ_S depends only on k and s. The size of $\mathcal{T}$ depends only on k and p because it is at most the product of the number of $k \times k$ adjacency matrices and the number of p-color lists for graphs of at most k vertices. Similarly, the size of $\mathcal{T}'$ depends only on k, p and s. Since σ is a function from $\mathcal{T}'$ to $[0, 2^{kq}] \cup \{\top\}$, the number of candidates for σ depends only on k, p, s, and q. Thus the number of S-shapes depends only on k, p, s, and q.

▶ **Observation 3.4.** *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph with n vertices, S be a* vi(k)*-set of G, and φ be an* MSO *formula with s free set variables and q quantifiers. The number of S-shapes depends only on k, p, s and q.*

3.2 Pre-evaluating the global constraints

Recall that in an $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula, the global cardinality constraints are used as atomic formulas. Namely, each of them takes the value true or false depending on the cardinalities of the free variables. To separate these constraints from the model checking process, the approach of pre-evaluation was used in the previous studies [27, 34].

▶ **Definition 3.5** (Pre-evaluation)**.** *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph, and $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$ be an $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula where $\boldsymbol{R} = (R_1, \dots, R_g)$. We call a function $\gamma\colon \{R_1, \dots, R_g\} \to \{\texttt{true}, \texttt{false}\}$ a* pre-evaluation*. Denote by $\gamma(\varphi)$ the* MSO *formula that obtained by mapping each global linear cardinality constraints R_i by γ.*

Since each global linear cardinality constraint R_i can be represented by a linear inequality, so is its complement $\bar{R}_i = [0, n]^s \setminus R_i$. Thus, for a pre-evaluation γ, the integers $x_1, \dots, x_s \in [0, n]$ that satisfies the following conditions can be represented by a system of linear inequalities:

- If $\gamma(R_i) = \texttt{true}$, then $(x_1, \dots, x_s) \in R_i$.
- Otherwise, $(x_1, \dots, x_s) \notin R_i$.

Denote by $\boldsymbol{R}_\gamma(x_1, \dots, x_s)$ this system of linear inequalities. If an assignment $\boldsymbol{X}^G = (X_1^G, \dots, X_s^G)$ of φ satisfies the system of inequalities $\boldsymbol{R}_\gamma(|X_1^G|, \dots, |X_s^G|)$, we say that $\boldsymbol{X}^G$ *meets* the pre-evaluation γ.

3.3 Making the local constraints uniform

Observe that for a $\mathsf{vi}(k)$-set S of a graph G, a vertex v of a component of $G - S$ has at most $k-1$ neighbors. In other words, $|N(v)| \in [0, k-1]$ for each vertex $v \in V(G-S)$. Therefore, $|N(v) \cap X| \in \alpha(v)$ if and only if $|N(v) \cap X| \in \alpha(v) \cap [0, k-1]$ for every combination of $X \subseteq V(G)$, $\alpha \colon V(G) \to [0, n]$, and $v \in V(G-S)$. Thus, we can reduce the range of local constraints as follows.

▶ **Observation 3.6.** *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph, S be a $\mathsf{vi}(k)$-set of G, $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$ be an $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula where $\boldsymbol{\alpha}^G = (\alpha_1, \dots, \alpha_s)$, and $\boldsymbol{X}^G$ be an assignment of φ. Denote by $\boldsymbol{\beta}^G$ the local constraints obtained from $\boldsymbol{\alpha}^G$ by restricting to $\alpha_i^G(v) \cap [0, k-1]$ for each $v \in V(G) \setminus S$ and $i \in [s]$. Then, $\boldsymbol{X}^G$ obeys $\boldsymbol{\alpha}^G$ if and only if $\boldsymbol{X}^G$ obeys $\boldsymbol{\beta}^G$.*

▶ **Definition 3.7** (Uniform). *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph with n vertices, S be a $\mathsf{vi}(k)$-set of G, and $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$ be an $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula where $\boldsymbol{\alpha}^G = (\alpha_1, \dots, \alpha_s)$. We say that the graph $\mathcal{G}$ is* uniform *on the local constraints $\boldsymbol{\alpha}^G$ if for every pair of components A_1 and A_2 of $G - S$ with the same $(\mathcal{G}, S)$-type, there is a $(\mathcal{G}, S)$-type isomorphism ψ from A_1 to A_2 such that $\alpha_i^G(v) = \alpha_i^G(\psi(v))$ for all $i \in [s]$ and $v \in V(A_1)$.*

We can obtain a uniform graph $\mathcal{G}'$ on $\boldsymbol{\alpha}^G$ from nonuniform graph $\mathcal{G} = (G, \boldsymbol{C})$ on $\boldsymbol{\alpha}^G$ as follows. For each $i \in [s]$, assign to every vertex $v \in V(G) \setminus S$ a new color $C^i_{\alpha_i(v)}$ corresponding to the local constraints $\alpha_i(v)$. Then we obtain a uniform graph $\mathcal{G}' = (G, \boldsymbol{C} \dotplus (C^i_B)_{i \in [s], B \subseteq [0, k-1]})$. The number of new colors added to $\mathcal{G}'$ is at most sk^2.

▶ **Lemma 3.8.** *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph with n vertices, S be a $\mathsf{vi}(k)$-set of G, and $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$ be an $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula where $\boldsymbol{\alpha}^G = (\alpha_1, \dots, \alpha_s)$. Assume that the graph $\mathcal{G}$ is* uniform *on the local constraints $\boldsymbol{\alpha}^G$. Let $\boldsymbol{X}^G = (X_1^G, \dots, X_s^G)$ and $\boldsymbol{Y}^G = (Y_1^G, \dots, Y_s^G)$ be assignments of φ with the same S-shape (σ_S, σ). Then for each $i \in [s]$ and $v \in V(G) \setminus S$, $|X_i^G \cap N(v)| \in \alpha_i^G(v)$ if and only if $|Y_i^G \cap N(v)| \in \alpha_i^G(v)$.*

Proof. By symmetry, it suffices to prove the only-if direction. Assume that $|X_i^G \cap N(v)| \in \alpha_i^G(v)$ for each $i \in [s]$ and $v \in V(G) \setminus S$. Let $\mathcal{G}_X = (G, \boldsymbol{C} \dotplus \boldsymbol{X}^G)$, $\mathcal{G}_Y = (G, \boldsymbol{C} \dotplus \boldsymbol{Y}^G)$, and A_Y be a component of $G - S$. Since the S-shape of $\boldsymbol{X}^G$ and $\boldsymbol{Y}^G$ are the same, there is a component A_X of $G - S$ such that the $(\mathcal{G}_X, S)$-type of A_X is equal to the $(\mathcal{G}_Y, S)$-type of A_Y. Then, there is an isomorphism ψ from A_Y to A_X such that $|Y_i^G \cap N(v)| = |X_i^G \cap N(\psi(v))|$ and $\alpha_i^G(v) = \alpha_i^G(\psi(v))$ for each $v \in A_Y$ and $i \in [s]$, because $\mathcal{G}$ is uniform on $\boldsymbol{\alpha}$. Therefore, $|Y_i^G \cap N(v)| = |X_i^G \cap N(\psi(v))| \in \alpha_i^G(\psi(v)) = \alpha_i^G(v)$ for each $v \in V(G) \setminus S$ and $i \in [s]$. ◀

3.4 The whole algorithm

We reduce the feasibility test of global and local constraints to the feasibility test of an ILP formula with a small number of variables. The variant of ILP we consider is formalized as follows.

p-Variable Integer Linear Programming Feasibility (p-ILP)
Input: A matrix $A \in \mathbb{Z}^{m \times p}$ and a vector $\boldsymbol{b} \in \mathbb{Z}^m$.
Question: Is there a vector $\boldsymbol{x} \in \mathbb{Z}^p$ such that $A\boldsymbol{x} \le \boldsymbol{b}$?

Lenstra [38] showed that p-ILP is fixed-parameter tractable parameterized by the number of variables p, and this algorithmic result was later improved by Kannan [32] and by Frank and Tardos [19].

▶ **Theorem 3.9** ([19,32,38]). *p-ILP can be solved using $O(p^{2.5p+o(p)} \cdot L)$ arithmetic operations and space polynomial in L, where L is the number of the bits in the input.*

The next technical lemma is the main tool for our algorithm.

▶ **Lemma 3.10.** *Let $\mathcal{G} = (G, \boldsymbol{C})$ be a p-colored graph, S be a* vi(k)*-set of G, and $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$ be an $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula where φ has s free set variables $X_1, \ldots X_s$, $\boldsymbol{R} = (R_1, \ldots, R_g)$, and $\boldsymbol{\alpha}^G = (\alpha_1, \ldots, \alpha_s)$. Assume that $\mathcal{G}$ is uniform on $\boldsymbol{\alpha}^G$. Then, there is an algorithm that given a valid S-shape (σ_S, σ), decides whether there exists an assignment $\boldsymbol{X}^G = (X^G_1, \ldots, X^G_s)$ such that its S-shape is (σ_S, σ), $(\mathcal{G}, \boldsymbol{X}^G, \boldsymbol{R}) \models \varphi$, and $\boldsymbol{X}^G$ obeys $\boldsymbol{\alpha}^G$ in time $f(k, |\varphi|)n^{O(1)}$ for some computable function f.*

Proof. Our task is to find an assignment $\boldsymbol{X}^G$ such that
1. the S-shape of $\boldsymbol{X}^G$ is (σ_S, σ),
2. $(\mathcal{G}, \boldsymbol{X}^G, \boldsymbol{R}) \models \varphi$, and
3. $|X^G_i \cap N(v)| \in \alpha^G_i(v)$ for all $v \in V(G)$ and $i \in [s]$.

Condition 1 can be handled easily by linear inequalities in our ILP formulation. Condition 2 is equivalent to the condition that there exists a pre-evaluation γ such that $(\mathcal{G}, \boldsymbol{X}^G) \models \gamma(\varphi)$, and $\boldsymbol{X}^G$ meets γ. We check whether $(\mathcal{G}, \boldsymbol{X}^G) \models \gamma(\varphi)$ and whether $\boldsymbol{X}^G$ meets γ separately. Furthermore, Condition 3 is checked separately for vertices in S and for vertices in $V(G) \setminus S$.

Step 1. Guessing and evaluating a pre-evaluation for the global constraints. We guess a pre-evaluation γ from $2^g \leq 2^{|\varphi|}$ candidates. We check whether each assignment $\boldsymbol{X}^G$ with S-shape (σ_S, σ) satisfies the MSO formula $\gamma(\varphi)$, i.e., $(\mathcal{G}, \boldsymbol{X}^G) \models \gamma(\varphi)$. By Lemma 3.3, we only need to check whether $(\mathcal{G}, \boldsymbol{X}^G) \models \gamma(\varphi)$ for an arbitrary assignment $\boldsymbol{X}^G$ with S-shape (σ_S, σ). This can be done in $f(k, |\varphi|)n^{O(1)}$ time [8, 37]. Note that even if $(\mathcal{G}, \boldsymbol{X}^G) \models \gamma(\varphi)$ is true, this arbitrarily chosen $\boldsymbol{X}^G$ may not meet γ. In Step 3, we find an assignment with S-shape (σ_S, σ) that meets γ.

Step 2. Checking the local constraints for the vertices in $V(G) \setminus S$. By Lemma 3.8, we can check whether all shape-(σ_S, σ) assignments satisfy the local constraints for the vertices in $V(G) \setminus S$ by constructing an arbitrary assignment $\boldsymbol{Y}^G$ of S-shape (σ_S, σ) and testing whether $|Y^G_i \cap N(v)| \in \alpha_i$ for all $v \in V(G) \setminus S$ and $i \in [s]$. Since constructing an assignment $\boldsymbol{Y}^G$ can be done in $f(k, |\varphi|)n^{O(1)}$ time, this test can be done in $f(k, |\varphi|)n^{O(1)}$ time.

Step 3. Constructing a system of linear inequalities for the remaining constraints. By Steps 1 and 2, it suffices to check whether there exists an assignment $\boldsymbol{X}^G = (X^G_1, \ldots, X^G_s)$ that satisfies the following conditions:
1. the S-shape of $\boldsymbol{X}^G$ is (σ_S, σ),
2. $\boldsymbol{X}^G$ meets the pre-evaluation γ, and
3. $\boldsymbol{X}^G$ obeys the local constraints $\boldsymbol{\alpha}^G$ for the vertices in S.

To this end, we construct a system of linear inequalities as follows.

In the following, we denote by $\mathcal{G}'$ the $(p+s)$-colored graph $(G, \boldsymbol{C} \dot{+} \boldsymbol{X}^G)$, where $\boldsymbol{X}^G$ is a hypothetical solution we are searching for.

Let $\mathcal{T}$ be the set of all $(\mathcal{G}, S)$-types. For every $t \in \mathcal{T}$, the number of type-t components of $G - S$ is denoted by n_t. Let $\mathcal{T}'$ be the set of all possible $(\mathcal{H}, S)$-types in $(p+s)$-colored graphs $\mathcal{H}$ obtained from $\mathcal{G}$ by adding s new colors. Observe that $\mathcal{T}'$ is a superset of the set of all $(\mathcal{G}', S)$-types, no matter how $\boldsymbol{X}^G$ is chosen. For every $t' \in \mathcal{T}'$, the $(\mathcal{G}, S)$-type of a type-t' component is uniquely determined and is denoted by $t'|_p$. (This notation comes from the fact that the $(\mathcal{G}, S)$-type of a type-t' component can be determined by considering the first p-colors.) For every $t' \in \mathcal{T}'$, we introduce the variable $x_{t'}$ that represents the number of $(\mathcal{G}', S)$-type t' components. The condition that the variables $x_{t'}$ agree with σ can be expressed as follows:

$$\begin{aligned}
\sum_{t' \in \mathcal{T}',\ t'|_p = t} x_{t'} &= n_t && \text{for every } t \in \mathcal{T}, \\
x_{t'} &= \sigma(t') && \text{for every } t' \in \mathcal{T}' \text{ such that } \sigma(t') \neq \top, \\
x_{t'} &\geq 2^{kq} + 1 && \text{for every } t' \in \mathcal{T}' \text{ such that } \sigma(t') = \top.
\end{aligned}$$

For every $i \in [s]$, we introduce an auxiliary variable y_i that represents the size of the set X_i^G, which is determined by the variables $x_{t'}$. The variables y_i can be expressed as follows:

$$y_i = |\{v \in S \mid X_i \in \sigma_S(v)\}| + \sum_{t' \in \mathcal{T}'} x_{t'} \cdot \#_i(x_{t'}) \qquad \text{for every } i \in [s],$$

where $\#_i(x_{t'})$ is the number of vertices with color $p + i$ in a type-t' component, i.e., the number of vertices assigned to the variable X_i in a type-t' component. Then, as mentioned in Section 3.2, the global constraints that match the pre-evaluation γ can be represented by the system of inequalities $R_\gamma(y_1, \dots, y_s)$.

Finally, we formulate the local constraints for the vertices in S into a system of inequalities. For every $v \in S$, $i \in [s]$, and $t' \in \mathcal{T}'$, the number of neighbors of v with color $p+i$ (i.e., in the set variable X_i) in a type-t' component is denoted by $d_{i,t'}(v)$ (i.e., $d_{i,t'}(u) = |N(u) \cap X_i^G \cap V(A)|$ where A is a type-t' component). All constants $d_{i,t'}(v)$ can be computed in $f(k, |\varphi|)n^{O(1)}$ time. For every $i \in [s]$ and $v \in S$, we introduce an auxiliary variable $z_{v,i}$ that represents the number of neighbors of v in the set X_i, which is determined by the variables $x_{t'}$. The variables $z_{v,i}$ can be expressed as follows:

$$z_{v,i} = |\{u \in N(v) \cap S \mid X_i \in \sigma_S(u)\}| + \sum_{t' \in \mathcal{T}'} d_{i,t'}(v) x_{t'} \quad \text{for every } v \in S,\ i \in [s].$$

Since the local constraints α_i^G can be expressed by $\alpha_i^G(v) = [l_i^v, u_i^v]$ with some integers l_i^v and u_i^v for every vertex v, the local constraints for vertices in S can be expressed as follows:

$$l_i^v \leq z_{v,i} \leq u_i^v \qquad \text{for every } v \in S, i \in [s].$$

By finding a feasible solution to the ILP formula constructed above, we can find a desired assignment $\boldsymbol{X}^G$. Since the number of the variables in the ILP formula depends only on k and $|\varphi|$, the lemma follows by Theorem 3.9. ◀

▶ **Theorem 3.11.** $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ *Model Checking is fixed-parameter tractable parameterized by* $\mathsf{vi}(G)$ *and* $|\varphi|$.

Proof. Let $k = \mathsf{vi}(G)$. Let S be a $\mathsf{vi}(k)$-set. Such a set can be found in $O(k^{k+1}n)$ time [13]. We construct a uniform graph $\mathcal{H} = (G, \boldsymbol{C}')$ on $\boldsymbol{\alpha}^G$ from the input graph $\mathcal{G} = (G, \boldsymbol{C})$ as described in Section 3.3. Here, the number of colors of $\mathcal{H}$ depends only on k, p, and s. We compute the $(\mathcal{H}, S)$-types of the components of $G - S$ and count the number of $(\mathcal{H}, S)$-type t components for each t. This can be done in $f(k, |\varphi|)n$ time with some computable function f.

We guess an S-shape (σ_S, σ) of an assignment of the input formula φ. By Observation 3.4, the number of candidates for (σ_S, σ) depends only on k, p, and s. We check whether the guess shape (σ_S, σ) is valid. This can be done by checking whether (σ_S, σ) is consistent with the number of components of all $(\mathcal{H}, S)$-types. Hence, this can be done in $f(k, |\varphi|)n$ time with some computable function f.

By Lemma 3.10 with the graph $\mathcal{H}$, $\mathsf{vi}(k)$-set S, and the input $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula $(\varphi, \boldsymbol{R}, \alpha^G)$, the theorem follows. ◀

4 Extension to MSO_2

The MSO_2 (or GSO) logic on graphs is a generalization of MSO_1 that additionally allows edge variables, edge-set variables, and an atomic formula $I(x, y)$ meaning that the edge assigned to y is incident to the vertex assigned to x. Although MSO_2 is strictly stronger than MSO_1 in general, for graphs of bounded treewidth, the model checking problem for MSO_2 can be reduced to the one for MSO_1 in polynomial time [10]. Using a similar reduction, we show that the same holds for $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ on graphs of bounded vertex integrity.

Now we define the extension of $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ with MSO_2, which we call $\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$. In $\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$, the local cardinality constraints for vertex-set variables and the global cardinality constraints work in the same way as in $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$. The local cardinality constraints for an edge-set variable X at a vertex v restricts the number of edges in X incident to v. We also generalize the concept of p-colored graphs such that each color can contain edges as well. A $\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ *formula* on a p-colored graph $\mathcal{G} = (G, \boldsymbol{C})$ is a tuple $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$, where $\boldsymbol{R} = (R_1, \dots, R_g)$ and $\boldsymbol{\alpha}^G = (\alpha^G_1, \dots, \alpha^G_s)$ are the global and local cardinality constraints, and φ is an MSO_2 formula with s free set variables that additionally equipped with symbols $R_1, \dots, R_g$. The problem $\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking is formalized as follows.

$\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking
Input: A p-colored graph $\mathcal{G} = (G, \boldsymbol{C})$, and a $\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ formula $(\varphi, \boldsymbol{R}, \boldsymbol{\alpha}^G)$.
Question: Is there an assignment $\boldsymbol{X}^G = (X^G_1, \dots, X^G_s)$ of φ such that $(\mathcal{G}, \boldsymbol{R}, \boldsymbol{X}^G) \models \varphi$ and $\boldsymbol{X}^G$ obeys $\boldsymbol{\alpha}^G$?

We can show the following theorem by presenting a reduction from from $\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking to $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking (see the full version for the proof).

▶ **Theorem 4.1.** $\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ *Model Checking is fixed-parameter tractable parameterized by* $\mathsf{vi}(G)$ *and* $|\varphi|$.

5 Concluding remarks

In this paper, we obtained an algorithmic meta-theorem for graphs of bounded vertex integrity in a framework introduced as an extension of MSO by Knop, Koutecký, Masařík, and Toufar [34]. Namely, we showed that $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking (or more generally, $\mathsf{GSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ Model Checking) is fixed-parameter tractable parameterized by vertex integrity. This result partially covers the results of the previous study [28]: some problems admit direct translations from their definitions to expressions in $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ (e.g., Equitable r-Coloring) and some need non-trivial modifications to make them expressible in $\mathsf{MSO}^{\mathsf{GL}}_{\mathsf{Lin}}$ (e.g., Capacitated Vertex Cover). For some other problems (e.g., Imbalance and Max Common Subgraph), we were not able to determine that they can be captured by our framework or not. Also, the result newly gives algorithms for Fair Evaluation Problems [33]. It would be interesting to ask whether there is a meta-theorem that can be applied to a larger class of problems parameterized by vertex integrity. (See the full version.)

We may also consider the fine-grained complexity of our problem. We did not explicitly state the time complexity of our fixed-parameter algorithms. If we carefully analyze the running time using the algorithm by Lampis and Mitsou [37], then we can show that the algorithms run in time triple exponential in a polynomial function of the parameter. For the ordinary MSO Model Checking, it is known that under ETH, there is no $2^{2^{o(k^2)}} n^{O(1)}$-time algorithm, where k is the vertex integrity of the input graph G and n is the number of vertices of G [37]. This double-exponential lower bound applies also to our generalized problem. Filling this gap would be an interesting challenge.

References

1 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991. doi:10.1016/0196-6774(91)90006-K.

2 Curtis A. Barefoot, Roger C. Entringer, and Henda C. Swart. Vulnerability in graphs – A comparative survey. *J. Combin. Math. Combin. Comput.*, 1:13–22, 1987.

3 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy distinguishes treewidth from pathwidth. In *ESA 2020*, volume 173 of *LIPIcs*, pages 14:1–14:19, 2020. doi:10.4230/LIPIcs.ESA.2020.14.

4 Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.

5 Hans L. Bodlaender, Tesshu Hanaka, Yasuaki Kobayashi, Yusuke Kobayashi, Yoshio Okamoto, Yota Otachi, and Tom C. van der Zanden. Subgraph isomorphism on graph classes that exclude a substructure. *Algorithmica*, 82(12):3566–3587, 2020. doi:10.1007/s00453-020-00737-z.

6 Richard B. Borie, R. Gary Parker, and Craig A. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7(5&6):555–581, 1992. doi:10.1007/BF01758777.

7 Laurent Bulteau, Konrad K. Dabrowski, Noleen Köhler, Sebastian Ordyniak, and Daniël Paulusma. An algorithmic framework for locally constrained homomorphisms. *CoRR*, abs/2201.11731, 2022. arXiv:2201.11731.

8 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

9 Bruno Courcelle. The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues. *RAIRO Theor. Informatics Appl.*, 26:257–286, 1992. doi:10.1051/ita/1992260302571.

10 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic – A Language-Theoretic Approach*. Cambridge University Press, 2012. URL: https://www.cambridge.org/knowledge/isbn/item5758776/.

11 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000. doi:10.1007/s002249910009.

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

13 Pål Grønås Drange, Markus S. Dregi, and Pim van 't Hof. On the computational complexity of vertex integrity and component order connectivity. *Algorithmica*, 76(4):1181–1202, 2016. doi:10.1007/s00453-016-0127-x.

14 Pavel Dvořák, Eduard Eiben, Robert Ganian, Dušan Knop, and Sebastian Ordyniak. Solving integer linear programs with a small number of global variables and constraints. In *IJCAI 2017*, pages 607–613, 2017. doi:10.24963/ijcai.2017/85.

15 Rosa Enciso, Michael R. Fellows, Jiong Guo, Iyad A. Kanj, Frances A. Rosamond, and Ondřej Suchý. What makes equitable connected partition easy. In *IWPEC 2009*, volume 5917 of *Lecture Notes in Computer Science*, pages 122–133, 2009. doi:10.1007/978-3-642-11269-0_10.

16 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances A. Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Inf. Comput.*, 209(2):143–153, 2011. doi:10.1016/j.ic.2010.11.026.

17 Michael R. Fellows, Daniel Lokshtanov, Neeldhara Misra, Frances A. Rosamond, and Saket Saurabh. Graph layout problems parameterized by vertex cover. In *ISAAC 2008*, volume 5369 of *Lecture Notes in Computer Science*, pages 294–305, 2008. doi:10.1007/978-3-540-92182-0_28.

18 Jirí Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theor. Comput. Sci.*, 412(23):2513–2523, 2011. doi:10.1016/j.tcs.2010.10.043.

19 András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7:49–65, 1987. doi:10.1007/BF02579200.

20 Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Ann. Pure Appl. Log.*, 130(1-3):3–31, 2004. doi:10.1016/j.apal.2004.01.007.

21 Shinya Fujita and Michitaka Furuya. Safe number and integrity of graphs. *Discret. Appl. Math.*, 247:398–406, 2018. doi:10.1016/j.dam.2018.03.074.

22 Shinya Fujita, Gary MacGillivray, and Tadashi Sakuma. Safe set problem on graphs. *Discret. Appl. Math.*, 215:106–111, 2016. doi:10.1016/j.dam.2016.07.020.

23 Jakub Gajarský and Petr Hliněný. Kernelizing MSO properties of trees of fixed height, and some consequences. *Log. Methods Comput. Sci.*, 11(1), 2015. doi:10.2168/LMCS-11(1:19)2015.

24 Robert Ganian. Twin-cover: Beyond vertex cover in parameterized algorithmics. In *IPEC 2011*, volume 7112 of *Lecture Notes in Computer Science*, pages 259–271, 2011. doi:10.1007/978-3-642-28050-4_21.

25 Robert Ganian, Petr Hlinený, Jaroslav Nesetril, Jan Obdrzálek, Patrice Ossona de Mendez, and Reshma Ramadurai. When trees grow low: Shrubs and fast MSO1. In *MFCS 2012*, volume 7464 of *Lecture Notes in Computer Science*, pages 419–430, 2012. doi:10.1007/978-3-642-32589-2_38.

26 Robert Ganian, Fabian Klute, and Sebastian Ordyniak. On structural parameterizations of the bounded-degree vertex deletion problem. *Algorithmica*, 83(1):297–336, 2021. doi:10.1007/s00453-020-00758-8.

27 Robert Ganian and Jan Obdržálek. Expanding the expressive power of monadic second-order logic on restricted graph classes. In *IWOCA 2013*, volume 8288 of *Lecture Notes in Computer Science*, pages 164–177, 2013. doi:10.1007/978-3-642-45278-9_15.

28 Tatsuya Gima, Tesshu Hanaka, Masashi Kiyomi, Yasuaki Kobayashi, and Yota Otachi. Exploring the gap between treedepth and vertex cover through vertex integrity. In *CIAC 2021*, volume 12701 of *Lecture Notes in Computer Science*, pages 271–285, 2021. doi:10.1007/978-3-030-75242-2_19.

29 Martin Grohe and Stephan Kreutzer. Methods for algorithmic meta theorems. In *Model Theoretic Methods in Finite Combinatorics*, volume 558 of *Contemporary Mathematics*, pages 181–206, 2009.

30 Petr Hliněný, Sang-il Oum, Detlef Seese, and Georg Gottlob. Width parameters beyond tree-width and their applications. *Comput. J.*, 51(3):326–362, 2008. doi:10.1093/comjnl/bxm052.

31 Bart M. P. Jansen and Dániel Marx. Characterizing the easy-to-find subgraphs from the viewpoint of polynomial-time algorithms, kernels, and turing kernels. In *SODA 2015*, pages 616–629, 2015. doi:10.1137/1.9781611973730.42.

32 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12:415–440, 1987. doi:10.1287/moor.12.3.415.

33 Dušan Knop, Tomás Masarík, and Tomás Toufar. Parameterized complexity of fair vertex evaluation problems. In *MFCS 2019*, volume 138 of *LIPIcs*, pages 33:1–33:16, 2019. doi:10.4230/LIPIcs.MFCS.2019.33.

34 Dušan Knop, Martin Koutecký, Tomáš Masařík, and Tomáš Toufar. Simplified algorithmic metatheorems beyond MSO: Treewidth and neighborhood diversity. *Log. Methods Comput. Sci.*, 15(4), 2019. doi:10.23638/LMCS-15(4:12)2019.

35 Stephan Kreutzer. Algorithmic meta-theorems. In Javier Esparza, Christian Michaux, and Charles Steinhorn, editors, *Finite and Algorithmic Model Theory*, volume 379 of *London Mathematical Society Lecture Note Series*, pages 177–270. Cambridge University Press, 2011. doi:10.1017/cbo9780511974960.006.

36 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012. doi:10.1007/s00453-011-9554-x.

37 Michael Lampis and Valia Mitsou. Fine-grained meta-theorems for vertex integrity. In *ISAAC 2021*, volume 212 of *LIPIcs*, pages 34:1–34:15, 2021. doi:10.4230/LIPIcs.ISAAC.2021.34.

38 Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983. doi:10.1287/moor.8.4.538.

39 Li-Shin Lin and Sartaj Sahni. Fair edge deletion problems. *IEEE Trans. Computers*, 38(5):756–761, 1989. doi:10.1109/12.24280.

40 Tomáš Masařík and Tomáš Toufar. Parameterized complexity of fair deletion problems. *Discret. Appl. Math.*, 278:51–61, 2020. doi:10.1016/j.dam.2019.06.001.

41 Stefan Szeider. Not so easy problems for tree decomposable graphs. *Ramanujan Mathematical Society, Lecture Notes Series*, No. 13:179–190, 2010. arXiv:1107.1177.

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