Towards Tight Bounds for the Graph Homomorphism Problem Parameterized by Cutwidth via Asymptotic Matrix Parameters

Carla Groenland
Delft Institute of Applied Mathematics, The Netherlands

Isja Mannens
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Jesper Nederlof
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Marta Piecyk
Warsaw University of Technology, Poland

Paweł Rzążewski
Warsaw University of Technology, Poland
University of Warsaw, Poland

Abstract

A homomorphism from a graph G to a graph H is an edge-preserving mapping from $V(G)$ to $V(H)$. In the graph homomorphism problem, denoted by $\textsc{Hom}(H)$, the graph H is fixed and we need to determine if there exists a homomorphism from an instance graph G to H. We study the complexity of the problem parameterized by the *cutwidth* of G, i.e., we assume that G is given along with a linear ordering $v_1, \ldots, v_n$ of $V(G)$ such that, for each $i \in \{1, \ldots, n-1\}$, the number of edges with one endpoint in $\{v_1, \ldots, v_i\}$ and the other in $\{v_{i+1}, \ldots, v_n\}$ is at most k.

We aim, for each H, for algorithms for $\textsc{Hom}(H)$ running in time $c_H^k n^{\mathcal{O}(1)}$ and matching lower bounds that exclude $c_H^{k \cdot o(1)} n^{\mathcal{O}(1)}$ or $c_H^{k(1-\Omega(1))} n^{\mathcal{O}(1)}$ time algorithms under the (Strong) Exponential Time Hypothesis. In the paper we introduce a new parameter that we call $\operatorname{mimsup}(H)$. Our main contribution is strong evidence of a close connection between c_H and $\operatorname{mimsup}(H)$:

- an information-theoretic argument that the number of states needed in a natural dynamic programming algorithm is at most $\operatorname{mimsup}(H)^k$,
- lower bounds that show that for almost all graphs H indeed we have $c_H \geq \operatorname{mimsup}(H)$, assuming the (Strong) Exponential-Time Hypothesis, and
- an algorithm with running time $\exp(\mathcal{O}(\operatorname{mimsup}(H) \cdot k \log k)) n^{\mathcal{O}(1)}$.

In the last result we do not need to assume that H is a fixed graph. Thus, as a consequence, we obtain that the problem of deciding whether G admits a homomorphism to H is fixed-parameter tractable, when parameterized by cutwidth of G and $\operatorname{mimsup}(H)$.

The parameter $\operatorname{mimsup}(H)$ can be thought of as the p-th root of the maximum induced matching number in the graph obtained by multiplying p copies of H via a certain graph product, where p tends to infinity. It can also be defined as an *asymptotic rank parameter* of the adjacency matrix of H. Such parameters play a central role in, among others, algebraic complexity theory and additive combinatorics. Our results tightly link the parameterized complexity of a problem to such an asymptotic matrix parameter for the first time.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases graph homomorphism, cutwidth, asymptotic matrix parameters

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.77

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2312.03859` [31]

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 77; pp. 77:1–77:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 853234 for CG, IM, JN and grant agreement No. 948057 for PRz). MP was funded by Polish National Science Centre, grant no. 2022/45/N/ST6/00237.

Acknowledgements The authors are grateful to Koblich for enlightening discussions about communication complexity.

1 Introduction

The study of the fine-grained complexity of NP-hard problems parameterized by width parameters has recently received an explosive amount of attention. In this study one aims to determine, for a given computational problem, a function f such that (1) the problem can be solved in $f(k)n^{\mathcal{O}(1)}$ time on given instances formed by an n-vertex graph along with an appropriate decomposition with width k, and (2) any improvement to $f(k)^{o(1)}n^{\mathcal{O}(1)}$ or even $f(k)^{1-\Omega(1)}n^{\mathcal{O}(1)}$ time would violate a standard hypothesis, typically being respectively the Exponential Time Hypothesis (ETH) and the Strong Exponential Time Hypothesis (SETH). Characterizing this complexity tightly often gives a deep insight in the combinatorial structure of the problem at hand, in particular about the relation that indicates when two "subsolutions" (for some definition of "subsolutions") combine into a global solution. An example where such insights had major consequences is Hamiltonian Cycle and the Traveling Salesperson Problem [4, 17, 50].

In contrast to the study of such fine-grained complexity, on the other side of the spectrum, a celebrated *meta-theorem* by Courcelle [13] shows that every graph property definable in the *monadic second-order logic* can be decided in time $f(k) \cdot n$ on n-vertex graphs given with a tree decomposition of width k. While this is extremely general, it is not precise at all in the sense that the functions $f(k)$ given by Courcelle's theorem are typically doubly-exponential or more, while more tailored algorithms with single-exponential functions exist. This begs the question: Could there be such a meta-theorem that gives a more fine-grained upper bound akin to the ones sought after above? Unfortunately, such a fine-grained meta-theorem still seems out of reach, and many recent works apply some highly non-trivial problem-specific insights to actually get the combination of tight algorithms and lower bounds [4, 5, 15, 16, 37, 38, 43, 46, 57, 60].

An intermediate step towards more general results such as Courcelle's theorem is to consider general problems that capture many natural well-studied problems as special cases. Such a step was already taken for certain locally checkable vertex subset problems, which capture natural problems including Independent Set and Dominating Set [25]. A particularly rich and elegant family of such problems can be defined via *graph homomorphisms*. A homomorphism from G to H is a mapping $\varphi : V(G) \to V(H)$ such that for every $uv \in E(G)$ we have $\varphi(u)\varphi(v) \in E(H)$. If H is the complete graph on k vertices, such mappings φ are exactly proper k-colorings, and this is why these mappings are often referred to as a H*-colorings* of G. For a fixed graph H, by Hom(H) we denote the computational problem in which one needs to determine whether there is a homomorphism to H from an input graph G. The complexity dichotomy for Hom(H) was provided by Hell and Nešetřil [34]: Hom(H) is polynomial-time solvable if H is bipartite, and NP-hard otherwise. So the cases relevant to our work are when H is non-bipartite.

There has been impressive work on the complexity of Hom(H) in various settings [7, 9, 11, 12, 14, 19, 20, 29, 58]. From the fine-grained perspective, a lot of attention was put in the parameterization by treewidth of the instance graph [21, 23, 26, 53, 54]. In particular, for Hom(H) and some of its close relatives we exactly understand the fastest possible (up to the SETH) running time of algorithms parameterized by treewidth. Perhaps even more interestingly, the techniques developed in this line of research led to a deep understanding of combinatorial properties of Hom(H) and its variants, and the results obtained on the way can be used far beyond the bounded-treewidth case.

Typically, the lower bounds for (a variant of) Hom(H) are shown by a reduction from (a variant of) q-Coloring, where the choice of q depends on H. In particular, Marx, Lokshtanov, and Saurabh [44] showed that for any $q \geq 3$, the q-Coloring problem on every instance G cannot be solved in time $(q-\varepsilon)^{\mathrm{tw}(G)} \cdot |V(G)|^{\mathcal{O}(1)}$ for any $\varepsilon > 0$, unless the SETH fails (here $\mathrm{tw}(G)$ is the treewidth of G). Similar lower bounds for q-Coloring are also known for other parameters, like *cliquewidth* [40], *feedback vertex set number* [44], *vertex cover number* [35], or *component-order-connectivity* [23]. A common element in all these results is that the constant in the base of the exponential factor in the complexity bound is an increasing function of the number q of colors.

However, it appears that this is not the case for all natural width parameters. For a linear ordering $v_1, v_2, \ldots, v_n$ of vertices of a graph G, its *width* is the maximum number of edges between the sets $\{v_1, \ldots, v_i\}$ and $\{v_{i+1}, \ldots, v_n\}$, over all $i \in \{1, \ldots, n-1\}$. The *cutwidth* of G, denoted by $\mathrm{ctw}(G)$, is the minimum width of a linear ordering of $V(G)$.

In stark contrast to the results listed above, Jansen and Nederlof [36] showed that for every q, the q-Coloring problem on instances G given with a linear ordering of width k can be solved in randomized[1] time $2^k \cdot |V(G)|^{\mathcal{O}(1)}$. In particular, the base of the exponential factor does not depend on q.

This phenomenon appears to be very fragile, e.g., it no longer occurs for the *counting* variant of q-Coloring [32]. In the context of the discussion above, it is very natural to ask about the situation for Hom(H), i.e., whether the natural dynamic programming approach that works in time $|V(H)|^k \cdot |V(G)|^{\mathcal{O}(1)}$ can be improved. In particular, whether there exists an absolute constant c_H, such that for every graph H, the Hom(H) problem on n-vertex instances of cutwidth k can be solved in time $c_H^k \cdot n^{\mathcal{O}(1)}$. This question was answered in the negative by Piecyk and Rzążewski [56], who showed that the base c_H of the exponential factor in the complexity bound (seen as a function of H) grows to infinity even if we restrict ourselves to cycles. More specifically, they show that c_H is lower-bounded by the number of edges in a maximum *induced matching*[2] in H, multiplied by 2.

Note that a maximum induced matching of a clique has only one edge, so this lower bound matches the running time of the randomized algorithm for q-Coloring [36].

However, Piecyk and Rzążewski [56] failed to provide an algorithm matching this lower bound, even functionally, i.e., an algorithm with running time $f(p,k) \cdot n^{\mathcal{O}(1)}$, where f is any function of the size p of a maximum matching in H and the cutwidth k of the instance. Thus, while the size of a maximum induced matching in H certainly plays some important role in the complexity of Hom(H) parameterized by the cutwidth, it is far from clear whether it indeed determines the base of the exponential factor. The discussion above leads to the following.

▶ **Open Problem 1.** *Describe, for any fixed non-bipartite graph H, a constant c_H such that:*

1. *There is an algorithm that, for all $k, n \in \mathbb{N}$, given an n-vertex graph G with linear ordering of width k, solves* Hom(H) *in time $c_H^k \cdot n^{\mathcal{O}(1)}$, and*
2. *Assuming the SETH, for any $\varepsilon > 0$, there is no algorithm that, for all $k, n \in \mathbb{N}$, given an n-vertex graph G with linear ordering of width k, solves* Hom(H) *in time $(c_H - \varepsilon)^k \cdot n^{\mathcal{O}(1)}$.*

---

[1] A slightly slower deterministic algorithm was also given.

[2] An *induced matching* of a graph is a set M of edges such that the graph induced by the endpoints of M is a matching.

Recall that when H is bipartite, $\textsc{Hom}(H)$ is already known to be solvable in polynomial time. Therefore, we restrict ourselves to non-bipartite graphs.

Moreover, for each graph H we have $c_H \leq |V(H)|$, as a straightforward dynamic programming algorithm works in time $|V(H)|^k \cdot n^{\mathcal{O}(1)}$.

Our contribution. We make significant progress towards Open Problem 1. In particular, for each non-bipartite graph H we define a constant c_H which we conjecture to have the desired properties. We prove, for almost all graphs, that $\textsc{Hom}(H)$ in n-vertex instances given with a linear ordering of width k cannot be solved in time $(c_H - \varepsilon)^k \cdot n^{\mathcal{O}(1)}$ for any $\varepsilon > 0$, assuming the SETH. Moreover, we give a dynamic programming approach of which we show the table sizes can be compressed to $c_H^k \cdot n^{\mathcal{O}(1)}$ (see the paragraph on *representative sets* below for more details). This can be interpreted as an upper bound, for each $i \in \{1, \dots, n-1\}$ on the amount of *information* of the graph $G[\{v_1, \dots, v_i\}]$ needed to decide $\textsc{Hom}(H)$ based on *only* $G - G[\{v_1, \dots, v_i\}]$. Unfortunately, this is an existential result and we do not yet know how to efficiently perform this compression. We give partial progress towards such a computation, yielding an algorithm with running time $\exp(2c_H k \log k) n^{\mathcal{O}(1)}$.

For a 0/1-matrix A, we define $\operatorname{mim}(A)$ as the largest r for which A has an $r \times r$ permutation submatrix.[3] The aforementioned work [56] shows that c_H needs to be at least $\operatorname{mim}(A_H)$, if A_H is the adjacency matrix of H. However, one of our main insights is that, as the cutwidth k increases, the accurate parameter for measuring the aforementioned amount of needed information on $G[\{v_1, \dots, v_i\}]$ is actually $\operatorname{mim}(A_H^{\otimes k})$, where $A_H^{\otimes k}$ denotes the result of taking the Kronecker product of k copies of A_H. Specifically, we introduce a new asymptotic rank parameter, called *mimsup*, defined by

$$\operatorname{mimsup}(A) = \limsup_{k \to \infty} \operatorname{mim}(A^{\otimes k})^{1/k}.$$

For a graph H, we define $\operatorname{mimsup}(H)$ to be $\operatorname{mimsup}(A_H)$, where A_H is the adjacency matrix of H. We remark that $\operatorname{mimsup}(H)$ can be also defined in a purely graph-theoretic way, in terms of the size of a maximum matching in a certain graph product. See Section 2 for more thorough definitions and details. We prove the results above for c_H equal to $\operatorname{mimsup}(H)$ (modulo some standard preprocessing of H).

The parameter becomes especially clean and elegant if H is a *projective core*; such graphs play a prominent role in the study of graph homomorphisms [29, 42, 54]. Their definition is somewhat complicated, we refer the interested reader to the full version of the paper (in Appendix). Let us just mention that this class captures *almost all graphs* [33, 45]. Formally, let p_n denote the probability that an n-vertex graph, chosen uniformly at random, is a non-bipartite projective core. Then p_n tends to 1 as n tends to infinity. Furthermore, up to some conjectures from algebraic graph theory from the early 2000s [41, 42], every graph H that cannot be simplified by the above-mentioned preprocessing is actually a projective core. We refer the interested reader to [54] for more information.

Going back to our setting, if H is a non-bipartite projective core, then we simply have $c_H = \operatorname{mimsup}(H)$. The first evidence that c_H is indeed the "right" choice of the parameter is the following lower bound[4].

---

[3] It is easily seen that $\operatorname{mim}(A)$ equals the maximum size of an induced matching in the bipartite graph that has A as biadjacency matrix. If A is symmetric, it is the adjacency matrix of a graph H and $\operatorname{mim}(A)$ equals twice the maximum size of an induced matching in H.

[4] Full proofs of statements marked with (♠) can be found in the full version of the paper [31]

▶ **Theorem 1** (♠).

1. *There exists* $\delta > 0$*, such that for every non-bipartite projective core* H*, there is no algorithm solving every instance* G *of* Hom(H) *in time* $\mathrm{mimsup}(H)^{\delta\cdot\mathrm{ctw}(G)} \cdot n^{\mathcal{O}(1)}$*, unless the ETH fails.*
2. *Let* H *be a connected non-bipartite projective core. There is no algorithm solving every instance* G *of* Hom(H) *in time* $(\mathrm{mimsup}(H) - \varepsilon)^{\mathrm{ctw}(G)} \cdot |V(G)|^{\mathcal{O}(1)}$ *for any* $\varepsilon > 0$*, unless the SETH fails.*

We next elaborate on the mentioned dynamic programming approach along with the table size compression via which we aim to match these lower bounds.

Representative Sets. A crucial technique in our arguments is that of *representative sets*. This is a method that allows us to considerably speed up dynamic programming algorithms by sparsifying the associated tables. Specifically, dynamic programming algorithms generally define a space of possible partial solutions $\mathcal{S}$, and a dynamic programming table stores a subset $\mathcal{A}$ of partial solutions that are valid in the given instance. A binary *compatibility matrix* M with rows and columns indexed by $\mathcal{S}$ indicates whether two partial solutions combine into a global solution. Generally speaking, a *representative set* of a set $\mathcal{A} \subseteq \mathcal{S}$ is a subset $\mathcal{A}'$ such that for each $j \in \mathcal{S}$ we have that there exists $i \in \mathcal{A}$ with $M[i, j] = 1$ if and only if there exists $i' \in \mathcal{A}'$ with $M[i', j] = 1$; see Section 3.1 for details more specific to the setting of our paper.

The power of representative sets lies in that (i) by definition, in any dynamic programming algorithm we can replace the set $\mathcal{A}$ with the smaller set $\mathcal{A}'$ without missing solutions, and (ii) for many matrices M, surprisingly small representative sets are guaranteed to exist. This underlies, for example, fast algorithms for the k-Path problem [48] or connectivity problems parameterized by treewidth [4, 27]. However, a serious bottleneck in these algorithm is the *computation* of such representative sets: It withholds us, for example, for getting faster algorithms for connectivity problems such as Traveling Salesperson (both parameterized by treewidth [4, 17] and the classic parameterization by the number of cities [50, Theorem 3]), and polynomial kernelization algorithms for Odd Cycle Transversal [39].

This already led some researchers to design faster algorithms for finding representative sets in special settings. A natural setting that comes up, for example for connectivity problems parameterized by treewidth, is to find representative sets for sets of partial solutions with a certain *product structure*. In [28], the authors show that representative sets for such families can be found faster than known for general families.

In this paper, the computation of representative sets is also a major bottleneck; in fact, modulo the standard conjectures discussed above, it is the only issue that withholds us from solving Open Problem 1 completely. Specifically, we show:

▶ **Theorem 2** (Informal statement of Theorem 6). *In the context of the natural dynamic programming algorithm for* Hom(H) *parameterized by cutwidth* k*, there exist representative sets of size at most* $\mathrm{mimsup}(H)^k$*.*

Thus, by the definition of representative sets, any algorithm that computes these representative sets fast enough would imply a $\mathrm{mimsup}(H)^{\mathrm{ctw}(G)} n^{\mathcal{O}(1)}$ time algorithm for Hom(H) and thus solve Open Problem 1. We view this as strong evidence that our lower bounds cannot be improved. Indeed, state-of-the-art hardness reduction techniques (like [44]) for problems parameterized by width parameters encode assignments to decision variables as states of dynamic programming tables and gradually check constraints on global consistency of these assignments throughout the graph. Our proof of existence of small representative

$$\begin{array}{|ccc|ccc|ccc|}
\hline
\color{red}{\mathbf{0}} & \color{red}{\mathbf{0}} & 0 & \color{red}{\mathbf{0}} & \color{red}{\mathbf{1}} & 1 & 0 & 1 & 1 \\
\color{red}{\mathbf{0}} & \color{red}{\mathbf{0}} & 0 & \color{red}{\mathbf{1}} & \color{red}{\mathbf{0}} & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
\hline
\color{red}{\mathbf{0}} & \color{red}{\mathbf{1}} & 1 & \color{red}{\mathbf{0}} & \color{red}{\mathbf{0}} & 0 & 0 & 1 & 1 \\
\color{red}{\mathbf{1}} & \color{red}{\mathbf{0}} & 1 & \color{red}{\mathbf{0}} & \color{red}{\mathbf{0}} & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
\hline
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
\hline
\end{array}$$

Figure 1 Illustration of a maximum induced matching (or equivalently, induced permutation submatrix) of size $\mathrm{mim}(A_H^{\otimes 2})$ shown in red, where $A_H = \left(\begin{smallmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{smallmatrix}\right)$ and $H = K_3$. More generally, the proof from [36] for determining the chromatic number of a graph shows that whenever H is a complete graph, $\mathrm{mim}(A_H^{\otimes k}) = 2^k$ and thus $\mathrm{mimsup}(H) = 2$.

sets means the number of assignments that need to be considered in order to find a global solution is also small, which means that this kind of approach to design lower bounds hits a natural barrier at our lower bound.

Coping Algorithmically with the Mimsup Parameter. Matrix or graph parameters that are defined in terms of large powers are sometimes called *asymptotic rank parameters*, and they are notoriously hard to compute. For example, the value of the asymptotic rank of the matrix multiplication tensor [8] (also known as ω) or the Shannon capacity of the cycle on 7 vertices [30] remain elusive. Unfortunately, $\mathrm{mimsup}(H)$ seems no exception. Similarly to the Shannon capacity [1, Question 6], it is even not clear whether its computation is decidable. For $\mathrm{mimsup}(H)$, even for simple graphs such as $H = K_q$, determining its value is non-trivial as well. As an illustration we depict the maximum induced matching in the second Kronecker power of the adjacency matrix of K_3 in Figure 1. One of the main insights of the $2^{\mathrm{ctw}(G)} n^{\mathcal{O}(1)}$ time algorithm for the chromatic number from [36] shows that $\mathrm{mimsup}(K_q)$ in fact equals 2.

Even when the existence of small representative sets is guaranteed because $\mathrm{mimsup}(H)$ is small, it is still challenging to find them quickly. Since mimsup is about products of graphs, one may expect that this product structure can be used algorithmically. Indeed, product structure has been exploited to compute representative sets in previous work [28], but there the family that needs to be represented has some (Cartesian) product structure. In our setting, this is not guaranteed and it is much less obvious how to proceed.

Nevertheless we show that, with some loss in precision (and hence, running time), we can work with graphs with small mimsup, by approximating it with another (easier to compute) value that we call the maximum *half induced matching* number $\mathrm{him}(H)$. For a matrix A, we define $\mathrm{him}(A)$ as the largest r for which A has an $r \times r$ triangular submatrix with ones on its diagonal.[5] For a graph H, we define $\mathrm{him}(H)$ to be $\mathrm{him}(A_H)$, where A_H is the adjacency matrix of H. We show that $\mathrm{him}(A_H)$ approximates $\mathrm{mimsup}(A_H)$ in the following sense:

▶ **Theorem 3.** *For every non-bipartite graph H with adjacency matrix A_H and $k \in \mathbb{N}$,*

$$\mathrm{him}(A_H) \leq \mathrm{mimsup}(A_H) = \limsup_{k \to \infty} \mathrm{mim}(A_H^{\otimes k})^{1/k} \quad \text{and} \quad \mathrm{mim}(A_H^{\otimes k}) \leq k^{(\mathrm{him}(A_H)+1)k}.$$

[5] A submatrix of a matrix A is any matrix that can be obtained from A by removing and reordering any of its rows and columns.

The parameter $\mathrm{him}(A_H)$ is easily computable in time $2^{\mathcal{O}(|V(H)|)}$. While the lower bound on $\mathrm{mimsup}(A_H^{\otimes k})$ is relatively easy, the upper bound uses an argument similar to the "neighborhood chasing" argument for the upper bounds on multi-colored Ramsey numbers [22]. This argument can in fact be made algorithmic in the sense that it can be used to compute representative sets for $\textsc{Hom}(H)$ of size at most $\mathcal{O}(k^{\mathrm{him}(H)k})$ in time $\mathcal{O}(k^{2\mathrm{him}(H)k})$. Combining this result with the dynamic programming algorithm for $\textsc{Hom}(H)$ for graphs of small cutwidth yields the following.

▶ **Theorem 4.** *For any graphs G and H, where G is given with a linear ordering of width k, in time $\mathcal{O}(k^{2k\cdot \mathrm{mimsup}(H)} \cdot |V(H)|^4 |V(G)|)$ one can decide whether G admits a homomorphism to H.*

Let us compare the running time in Theorem 4 with the naive approach; recall that its complexity is $|V(H)|^k \cdot |V(G)|^{\mathcal{O}(1)}$. If we treat H as a constant and k as a parameter, then the latter one is faster. However, we emphasize here that in Theorem 4 we do not assume that H is a constant, so these two algorithms are incomparable. In particular, Theorem 4 shows that the homomorphism problem, where the input consists of both G and H, is *fixed-parameter-tractable* when parameterized by the cutwidth of G and $\mathrm{mimsup}(H)$.

It should be noted that a similar notion of half-induced matching of a compatibility matrix was already introduced in previous work in the context of representative sets of the AntiFactor problem [47] parameterized by treewidth and list size. However, in that setting, the authors were only able to provide a lower bound for their problem, and they did not manage to make the connection with half-induced matchings algorithmic. Additionally, their compatibility matrix has a very specific structure: it is indexed with integers and the value of an entry only depends on the sum of the values associated with the row and column.

Comparison of Mimsup With Other Rank Parameters. One of our main conceptual contributions is the introduction of the mimsup parameter in the context of parameterized algorithms. It is actually the first asymptotic rank parameter shown to be relevant in this context. Our mimsup parameter is very similar to the *asymptotic induced matching number* studied by Arunachalam et al. [2] which was introduced for k-partite, k-uniform hypergraphs (and so, in the graph setting, only for bipartite graphs). Various asymptotic variants of rank parameters have been studied for tensors. For example, this has been done for rank parameters such as subrank, tensor rank and slice rank. However, for matrices (2-tensors) these are equal to the "standard" rank for matrices and so have no interesting asymptotic aspects.

That being said, it is only natural to compare the mimsup parameter with different related rank parameters from the literature. We will discuss this now, and provide proofs that formally support this discussion in the Appendix. The approach from [36] naturally extends to solve $\textsc{Hom}(H)$ quickly for all graphs H where the so-called *support-rank* [18, 49] of the adjacency matrix of H is small. The following sequence of inequalities holds for every matrix A:

$$\mathrm{mim}(A) \leq \mathrm{him}(A) \leq \mathrm{mimsup}(A) \leq \text{support-rank}(A) \leq \mathrm{rank}(A).$$

We believe all of the inequalities can be strict. When A is the $(r \times r)$-matrix with ones on and above the diagonal and zeros below the diagonal, then $\mathrm{mim}(A) = 1 < r = \mathrm{him}(A)$ for $r \geq 2$. This means that mim is not functionally equivalent to him nor mimsup. We use random matrices to show that him and mimsup may take very different values (see Theorem 13 for a formal statement). In the Appendix we find a connection of support-rank to a parameter called *graph dimension* (or *Prague dimension*) [51, 52]. We also show there that him is not functionally equivalent to support-rank. This shows that our algorithm from Theorem 4 can be significantly faster than the discussed natural generalization of [36]. We leave it as an interesting open problem if mimsup is functionally equivalent to him or support-rank.

The aforementioned (well studied) Shannon capacity has a definition that is very similar to the mimsup parameter: It is defined in terms of the maximum size of an independent set (also called the independence number) in an appropriate graph product, and the size of a maximum induced matching of a graph equals the independence number of the square of its line graph. Unfortunately, because the definitions of mimsup and Shannon capacity use different graph products, the relation between the two is somewhat loose; see Appendix for details. Nevertheless, based on their similarity, one may expect that Shannon capacity shares some of its peculiarities with mimsup, such as an unpredictable behaviour of the value in graph powers [1].

2 Preliminaries

For an integer n, by $[n]$ we denote the set $\{1, 2, \dots, n\}$ and for integers a, b we write $[a, b] = \{a, a + 1, \dots, b\}$. For a set X, by 2^X we denote the family of all subsets of X. For $i, s \in \mathbb{N}$, the multinomial coefficient

$$\binom{is}{s, \dots, s} = \frac{(is)!}{s!, \dots, s!} = i^{(1+o(1))is}$$

is the number of partitions of $[as]$ into a parts of size s.

For a graph G and $V' \subseteq V(G)$ (resp. $E' \subseteq E(G)$), by $G[V']$ (resp. $G[E']$) we denote the subgraph of G induced by V' (resp. E'). We say two graph parameters p and q are functionally equivalent if there are functions $f, g : \mathbb{R} \to \mathbb{R}$ such that $p(G) \leq f(q(G))$ and $q(G) \leq g(p(G))$ for all graphs G.

Homomorphisms. For graphs G and H, a *homomorphism* from G to H is a mapping $\varphi : V(G) \to V(H)$ such that for every $uv \in E(G)$ we have $\varphi(u)\varphi(v) \in E(H)$. If φ is a homomorphism from G to H, we denote it by writing $\varphi : G \to H$. If G admits a homomorphism to H, we denote is shortly by $G \to H$.

In the Hom problem we are given a pair (G, H) of graphs, and we ask whether $G \to H$. In the Hom(H) the graph H is considered to be fixed and we ask whether a graph G given as an input admits a homomorpshism to H.

We will always assume that H is a connected graph. Indeed, each component of G must map to a single component of H, so the problem can be solved component-wise.

Cutwidth. Let G be a graph and consider a linear ordering $\sigma = (v_1, \dots, v_n)$ of its vertices. For $i \in [n-1]$, the *i-th cut* is the partition of $V(G)$ into sets $\{v_1, \dots, v_i\}$ and $\{v_{i+1}, \dots, v_n\}$. The *width* of such a cut is the number of edges with one endvertex in $\{v_1, \dots, v_i\}$ and the other in $\{v_{i+1}, \dots, v_n\}$. The *width* of σ is the maximum width of a cut of σ. Finally, the *cutwidth* of G, denoted by $\mathrm{ctw}(G)$, is the minimum width of a linear ordering of the vertices of G.

Associated bipartite graphs. In order to define the main parameters of our paper, we will use a notion of *associated bipartite graphs*. For a graph G, the graph G^* is defined as follows.

$$V(G^*) = \{u', u'' \mid u \in V(G)\},$$
$$E(G^*) = \{u'w'', u''w' \mid uw \in E(G)\}.$$

Induced matchings and half-induced matchings. A set $M \subseteq E$ of edges of a graph $H = (V, E)$ forms an *induced matching* if the edges in M are vertex disjoint and no edge in E is incident with two edges from M. We may also view this as two sequences of distinct vertices $v_1, \ldots, v_m$ and $u_1, \ldots, u_m$ where $v_i u_j \in E$ if and only if $i = j$. For a bipartite graph H, by $\mathrm{mim}(H)$ we denote the size of a maximum induced matching in H. For non-bipartite H, we define $\mathrm{mim}(H) := \mathrm{mim}(H^*)$.

A *half-induced matching* of a graph H consists of two sequences $v_1, \ldots, v_m$ and $u_1, \ldots, u_m$ of distinct vertices where $v_i u_i \in E$ for $i \in [m]$ and $u_i v_j \notin E$ if $1 \leq i < j \leq m$. For a bipartite graph H, we denote the size of the largest half-induced matching in H by $\mathrm{him}(H)$. We extend the definition to graphs H that are non-bipartite via $\mathrm{him}(H) = \mathrm{him}(H^*)$. This notion has been studied under the name *constrained matching* (a subset with a unique matching, see e.g. [10, 55, 59]), but we decided to use the name which appeared more recently in a similar setting to ours [47], since the word "constrained matching" has also been used for various other purposes in the algorithmic community.

Mim and him for matrices. Let $A \in \{0,1\}^{n \times n}$ be a matrix. Given a sequence $r \in [n]^a$ of distinct row indices and $c \in [n]^b$ of distinct columns indices, for some integers $a, b \in [n]$, we write $A[r, c]$ for the $a \times b$ matrix with entries $A[r, c]_{i,j} = A_{r_i, c_j}$ for $i \in [a]$ and $j \in [b]$. We refer to any matrix which arises in such a manner as a *submatrix after permutation* of A.

We write $\mathrm{mim}(A)$ for the maximum r for which A has the $r \times r$ identity matrix as submatrix after permutation (equivalently, the largest permutation submatrix). We write $\mathrm{him}(A)$ for the largest r for which A has an $r \times r$ triangular matrix with 1's on the diagonal as submatrix after permutation. We will also refer to such a submatrix as *half induced matching*. (A matrix is called *triangular* if either all entries below the diagonal, or all entries above the diagonal are 0.)

For bipartite graphs $H = (U, V, E)$, the *bi-adjacency matrix* B is indexed by rows from U and columns from V where $B[u, v] = 1$ if $uv \in E$ and $B[u, v] = 0$ otherwise. For a bipartite graph H, there is a one-to-one correspondence between induced matchings of H of size m and $m \times m$ permutation submatrices of the bi-adjancency matrix of H. In particular, for a bi-adjacency matrix B of H, $\mathrm{mim}(B) = \mathrm{mim}(H)$. Similarly, $\mathrm{him}(B) = \mathrm{him}(H)$.

For a non-bipartite graph G, if A_G is its adjacency matrix, then A_G is also the bi-adjacency matrix of G^*. This means that for non-bipartite H with adjacency matrix A_H, $\mathrm{mim}(H) = \mathrm{mim}(A_H)$ and $\mathrm{him}(H) = \mathrm{him}(A_H)$.

Mimsup. For a matrix A, we define

$$\mathrm{mimsup}(A) = \limsup_{k \to \infty} \mathrm{mim}(A^{\otimes k})^{1/k}.$$

Here $\otimes$ denotes the Kronecker product of the matrix. Given an $n \times m$ matrix $A = (a_{i,j})_{i \in [n], j \in [m]}$ and a matrix B, the Kronecker product is given by

$$A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \ldots & a_{1,m}B \\ a_{2,1}B & a_{2,2}B & \ldots & a_{2,m}B \\ & \ldots & & \\ a_{n,1}B & a_{n,2}B & \ldots & a_{n,m}B \end{pmatrix}.$$

Since $\mathrm{mim}(A \otimes B) \geq \mathrm{mim}(A)\,\mathrm{mim}(B)$, Fekete's lemma [24] applies to show that (♠)

$$\limsup_{k\to\infty} \mathrm{mim}(A^{\otimes k})^{1/k} = \lim_{k\to\infty} \mathrm{mim}(A^{\otimes k})^{1/k} = \sup_{k\in\mathbb{N}} \mathrm{mim}(A^{\otimes k})^{1/k}.$$

For a non-bipartite graph H, with adjacency matrix A, we set

$$\mathrm{mimsup}(H) = \mathrm{mimsup}(A).$$

When H is bipartite with bi-adjancency matrix[6] B, $\mathrm{mimsup}(H) = \mathrm{mimsup}(B)$. The parameters can also be defined in purely graph theoretical terms, as we now explain.

For a bipartite graph H with bipartition classes X, Y, and for $k \in \mathbb{N}$, we define $H^{\otimes k}$ to be the graph on vertex set $X^k \cup Y^k$ where there is an edge $(x_1, \dots, x_k)(y_1, \dots, y_k)$ in $H^{\otimes k}$ if and only if $x_i y_i \in E(H)$ for every $i \in [k]$. With this definition of graph product $\otimes$, we define

$$\mathrm{mimsup}(H) = \begin{cases} \limsup_{k\to\infty} \mathrm{mim}(H^{\otimes k})^{1/k} & \text{if } H \text{ is bipartite,} \\ \mathrm{mimsup}(H^*) & \text{otherwise.} \end{cases}$$

The following property of mimsup is straightforward.

▶ **Observation 5.** *If H is an induced subgraph of G, then* $\mathrm{mimsup}(H) \leq \mathrm{mimsup}(G)$.

For bipartite graphs, mimsup coincides with the parameter asymptotic induced matching number studied by [2]. Although asymptotic rank parameters (e.g. asymptotic subrank, asymptotic tensor rank and asymptotic slice rank) have been widely studied for tensors, the "non-asymptotic" parameters are usually equal to the matrix rank for matrices, which has no interesting asymptotic behaviour since $\mathrm{rank}(A^{\otimes n}) = \mathrm{rank}(A)^n$. In particular, the *subrank* in some sense looks for the largest "identity subtensor", similar to our mim, but since it allows row operations to be applied (instead of merely permutations), this notion is the same as the usual rank for matrices and the same holds for the asymptotic subrank.

3 Solving Hom with representative sets

In this section we discuss how we can use representative sets to create fast algorithms for Hom. We start by giving a definition of a representative set in our setting. Intuitively we want a representative set $\mathcal{A}'$ of $\mathcal{A}$ to carry all the important information from $\mathcal{A}$, while being smaller in size. In practice, being able to find small representative sets corresponds to having to compute less entries in a dynamic programming algorithm. So this gives the following natural extremal problem: how small of a representative set are we always guaranteed to find, that is, *what is the largest size of a set which has no smaller representative set?* After giving the definition, we explain why mimsup exactly determines the answer to this question in our setting.

Finally, we give a general framework for solving Hom instances; it consists of an algorithm that takes as input some reduction algorithm R that produces small representative sets and uses it to solve Hom on input graphs G and H. In Section 4 we give examples of such reduction algorithms.

[6] Note that mimsup is invariant under row and column permutations. This means that the choice of bi-adjacency matrix does not affect the mimsup and thus mimsup on bipartite graphs is well-defined.

3.1 Definition of Representative Set

Given a 0/1 matrix M, with rows indexed by a set $\mathcal{R}$ and $\mathcal{A} \subseteq \mathcal{R}$, we are interested in knowing whether for a column c, there is a row $r \in \mathcal{A}$ with $M[r,c] = 1$. In our case,

- each row represents a coloring of the left-hand side of the cut;
- all the colorings that can be extended to the left-hand side of the (input) graph are contained in $\mathcal{A}$;
- each column represents a coloring of the right-hand side of the cut;
- $M[r,c] = 1$ if and only if the colorings represented by row r and column c are compatible.

This makes the following definition very natural. We say that a subset $\mathcal{A}' \subseteq \mathcal{A}$ *M-represents* $\mathcal{A}$, if for any column j we have that if there is a row index $i \in \mathcal{A}$ such that $M[i,j] = 1$, then there is also $i' \in \mathcal{A}'$ such that $M[i',j] = 1$. Intuitively, this means that we do not "lose any solutions" by restricting to $\mathcal{A}'$.

We will also refer to $\mathcal{A}'$ as an *M-representative set* of $\mathcal{A}$. We may omit M if it is clear from context.

We remark that representing is transitive: if $\mathcal{A}''$ represents $\mathcal{A}'$ and $\mathcal{A}'$ represents $\mathcal{A}$, then $\mathcal{A}''$ represents $\mathcal{A}$.

Suppose we aim to solve Hom for input graphs G and H, where H is non-bipartite. We will be interested in representative sets with respect to $M = A_H^{\otimes k}$ for integers k, where A_H is the adjacency matrix of H. We assume that G is given with a linear order $v_1, \dots, v_n$ of width at most w. For an integer $i \in [n]$, then $G[\{v_1, \dots, v_i\}]$ is the "left-hand side of the graph" with respect to the ith cut and

$$X_i := \{v \in \{v_1, \dots, v_i\} \mid \exists v' \in \{v_{i+1}, \dots, v_n\}, vv' \in E(G)\}.$$

is "the left-hand side of the i-th cut." Suppose there are k edges crossing the ith cut: $\{a_1, b_1\}, \dots, \{a_k, b_k\} \in E(G)$ with $a_1, \dots, a_k \in \{v_1, \dots, v_i\}$ and $b_1, \dots, b_k \in \{v_{i+1}, \dots, v_n\}$. Let $L_i = (a_1, \dots, a_k)$ and $R_i = (b_1, \dots, b_k)$. Note that $\{a_1, \dots, a_k\} = X_i$ but some elements may be repeated. A row r (seen as "index") of the matrix $M = A_H^{\otimes k}$ is a k-tuple $(r_1, \dots, r_k) \in V(H)^k$, which corresponds to a coloring $X_i \to V(H)$ if $r_j = r_{j'}$ whenever $a_j = a_{j'}$. If similarly $c \in V(H)^k$ represents a coloring of the "right-hand side of the cut", then $M[r,c] = 1$ if and only if $r_j c_j \in E(H)$ for all $j \in [k]$, i.e. the colorings are compatible. So indeed we capture the properties informally claimed above.

Since in our setting M will be the adjacency matrix of some graph H, we may refer to H-representative sets rather than A_H-representative sets.

3.2 Connection to Mimsup

When applied to the adjacency matrix A_H of a non-bipartite graph H, the following result shows that $\mathrm{mimsup}(H)^k$ approximates how large a set $\mathcal{A} \subseteq V(H)^k$ without smaller $A_H^{\otimes k}$-representative set can be. This easily follows from the definitions but is still an important conceptual contribution.

▶ **Theorem 6.** *Let $M \in \{0,1\}^{h \times h}$ be a matrix with rows indexed by $\mathcal{R}$.*

- *For each integer $k \in \mathbb{N}$, for any $\mathcal{A} \subseteq \mathcal{R}^k$, there is a subset $\mathcal{A}' \subseteq \mathcal{A}$ of size $\mathrm{mimsup}(M)^k$ that $M^{\otimes k}$-represents $\mathcal{A}$.*
- *Conversely, for each $\varepsilon > 0$, for each sufficiently large k, there is a $\mathcal{A} \subseteq \mathcal{R}^k$, for which no $\mathcal{A}' \subseteq \mathcal{A}$ of size at most $(\mathrm{mimsup}(M) - \varepsilon)^k$ can $M^{\otimes k}$-represent $\mathcal{A}$.*

Proof. Let $\mathcal{A}' \subseteq \mathcal{A}$ be of minimal size among the subsets that M-represent $\mathcal{A}$. Then no proper subset of it M-represents $\mathcal{A}$. This means that for each $a \in \mathcal{A}'$ it cannot be removed from $\mathcal{A}'$ to get a set that M-represents $\mathcal{A}$. Thus, for each $a \in \mathcal{A}'$ there is some $\mu(a) \in V(H)^k$

such that $M[a, \mu(a)] = 1$, but for every $a' \in \mathcal{A}' \setminus \{a\}$ we have that $M[a', \mu(a)] = 0$. This gives a permutation $(|\mathcal{A}'| \times |\mathcal{A}'|)$-submatrix of M. This shows that $|\mathcal{A}'| \leq \mathrm{mim}(M^{\otimes k})$. By definition of mimsup, $\mathrm{mim}(M^{\otimes k}) \leq \mathrm{mimsup}(M)^k$.

Conversely, by definition of limit, for each $\varepsilon > 0$ there is a k_0 such that $\mathrm{mim}(M^{\otimes k}) \geq (\mathrm{mimsup}(M) - \varepsilon)^k$ for all $k \geq k_0$. Let $k \geq k_0$. Let $\mathcal{A} \subseteq \mathcal{R}$ be the rows of a largest induced permutation submatrix of $M^{\otimes k}$. Then $|\mathcal{A}| = \mathrm{mim}(M^{\otimes k}) \geq (\mathrm{mimsup}(M) - \varepsilon)^k$ and none of the strict subsets of $\mathcal{A}$ can $M^{\otimes k}$-represent it. ◀

3.3 Exploiting Representative Sets in Dynamic Programming

The main idea behind the use of representative sets in an algorithmic setting is as follows. We solve the problem with a standard dynamic programming approach, where the cells are indexed by the elements of the set $\mathcal{A}$. A representative set then forms a small subset of these indices, which still carries enough information to solve the problem. By regularly applying the reduction algorithm, we can effectively run our dynamic programming algorithm on only a small subset of the cells in the table. We formalize this in the following theorem. Let us emphasize that H is not assumed to be fixed here but rather given as an input.

▶ **Theorem 7.** *Let H be a non-bipartite graph on h vertices. Let R be a reduction algorithm that, given an integer $k \geq 2$ and a subset $\mathcal{A} \subseteq V(H)^k$, outputs a set $\mathcal{A}'$ of size $\mathsf{size}(H,k)$ that $A_H^{\otimes k}$-represents $\mathcal{A}$, running in time $\mathsf{time}(|\mathcal{A}|, H, k)$. Then there exists an algorithm that, given a linear ordering of an n-vertex graph G of width w, decides whether $G \to H$ in time*

$$\mathcal{O}\Big(\big(\mathsf{size}(H,w) \cdot h + \mathsf{time}\,(\mathsf{size}(H,w) \cdot h, H, w)\,\big)n\Big).$$

Proof. Let $v_1, \ldots, v_n$ be a linear ordering of G of width k. For $i \in [n]$, by E_i we denote the set of edges that cross the i-th cut, i.e., those with one endpoint in $\{v_1, \ldots, v_i\}$ and the other in $\{v_{i+1}, \ldots, v_n\}$. For $i \in [n]$, let X_i be the set that contains all vertices from $\{v_1, \ldots, v_i\}$ incident to an edge from E_i, i.e.,

$$X_i := \{v \in \{v_1, \ldots, v_i\} \mid \exists v' \in \{v_{i+1}, \ldots, v_n\}, vv' \in E(G)\}\,.$$

Note that we have $|X_i| \leq |E_i| \leq w$ and $X_1 = \{v_1\}$ (since G is connected). For a mapping $c : X_i \to V(H)$, we define the table entry $T_i[c]$ as true if there exists a homomorphism $\varphi : G[\{v_1, \ldots, v_i\}] \to V(H)$, such that for all $v \in X_i$ we have $\varphi(v) = c(v)$. (In other words, the keys are given by the H-colorings of X_i and the table entry is true if there is an extension of the coloring of X_i to the left-hand side of the graph.)

This table can be easily computed in time $h^{w+1} \cdot n^{\mathcal{O}(1)}$ by the following naive dynamic programming procedure. We initiate every entry $T_i[c]$ to be false and every entry $T_1[c]$ to be true. Then, for every $i \in [2, n]$, every mapping $c' : X_{i-1} \to V(H)$, such that $T_{i-1}[c']$ is true, and every $u \in V(H)$, we check whether $c : X_{i-1} \cup \{v_i\} \to V(H)$ defined as

$$c(v) = \begin{cases} u & \text{if } v = v_i, \\ c'(v) & v \in X_{i-1}. \end{cases} \tag{1}$$

is a homomorphism from $G[X_{i-1} \cup \{v_i\}]$ to H. If so, we set $T_i[c|_{X_i}]$ to true.

We first outline why this correctly computes the table entries (that is, that at the end $T_i[c]$ is true if and only if c extends to a coloring of $G[\{v_1, \ldots, v_i\}]$) and then explain how to improve on this naive algorithm. We prove the claim by induction on i. For $i = 1$, the coloring only assigns a color to v_1 and does not need to be extended (and is automatically proper). Now suppose that the claim has been shown for $i = 1, \ldots j$ and let $\alpha : X_{j+1} \to V(H)$

be a coloring. If this extends to a coloring ϕ of $G[\{v_1,\dots,v_{j+1}\}]$, then $T_j[c']$ is true for $c' = \phi|_{X_j}$ (by the induction hypothesis) and we could obtain α as the restriction from c from (1) with $u = \phi(v_{j+1})$ and $i = j+1$. So $T_{j+1}[\alpha]$ is true. Vice versa, if $T_{j+1}[\alpha]$ has been set to true, then there is a $c' : X_j \to V(H)$ and $u \in V(H)$ such that c (again defined as in (1)) is a homomorphism $G[X_j \cup \{v_{j+1}\}] \to H$ which restricts to α on X_{j+1}. By the induction hypothesis, there exists a proper coloring ϕ' that extends c' to $G[\{v_1,\dots,v_j\}]$ and we extend this to a coloring ϕ of $G[\{v_1,\dots,v_{j+1}\}]$ by setting $\phi(v_{j+1}) = u$. Then ϕ still restricts to α and all of the edge constraints have been verified by c and/or ϕ'. In particular, $G \to H$ if and only if $T_n[\emptyset]$ is true, where $\emptyset$ denotes the empty mapping ($X_n = \emptyset$).

We will speed up this naive version of the dynamic program by computing a representative table T' as follows. We first set $T'_1 = T_1$. For $i = 1, 2, \dots, n-1$ we proceed as follows. Let $k = |E_i| \le w$ and $M = A_H^{\otimes k}$. Let $\{a_1, b_1\}, \dots, \{a_k, b_k\} \in E_i$ be an enumeration of the edges, with $a_j \in \{v_1,\dots,v_i\}$ for all $j \in [k]$. For each $c : X_i \to V(H)$ such that $T'_i[c]$ is set to true, we put the k-tuple $(c(a_1),\dots,c(a_k))$ in $\mathcal{A}_i$. When $k \ge 2$, we apply the reduction algorithm R to $\mathcal{A}_i$, resulting in a set $\mathcal{A}'_i$ of size at most $\mathsf{size}(H,k)$ that $A_H^{\otimes k}$-represents $\mathcal{A}_i$. When $k = 1$, we set $\mathcal{A}'_i = \mathcal{A}_i$. We then compute the next table entries similarly as in the previous approach. Each element of $\mathcal{A}'_i$ corresponds to a coloring $c' : X_i \to V(H)$. For $u \in V(H)$, we check whether $c : X_i \cup \{v_{i+1}\} \to V(H)$ with $c(v_{i+1}) = u$ and $c|_{X_i} = c'$ is a homomorphism from $G[X_i \cup \{v_{i+1}\}]$ to H. If so, we set $T'_{i+1}[c|_{X_{i+1}}]$ to true. We repeat this for all pairs (c', u).

The procedure above is repeated for $i = 1, \dots, n-1$, after which we return $T'_n[\emptyset]$.

When $|\mathcal{A}'_i| \le \mathsf{size}(H,k)$, we find that $|\mathcal{A}_{i+1}| \le \mathsf{size}(H,k)h$ (for $k = |E_i| \le w$ and $\mathsf{size}(H,k) = h$ for $k = 1$). We may assume size is a non-decreasing function on each coordinate. So the total running time is as claimed:

$$\mathcal{O}\Big(\big(\mathsf{size}(H,w)\cdot h + \mathsf{time}\left(\mathsf{size}(H,w)\cdot h, H, w\right)\big)n\Big).$$

The fact that the dynamic programming steps preserve representation follows from transitivity of representation, but let us spell out the details.

Let Y_{i+1} be the set of endpoints on the right-hand side of the $(i+1)$th cut and enumerate the edges in E_{i+1} as $\{x_1, y_1\}, \dots, \{x_k, y_k\}$, with $x_j \in X_{i+1}$ and $y_j \in Y_{i+1}$. We will show that for every $i \in [n-1]$, if $\mathcal{A}'_i$ represents the set $\mathrm{True}_i := \{(c(x_1),\dots,c(x_k)) \mid T_i[c] = \mathrm{True}\}$, then $\mathcal{A}_{i+1}$ represents the set True_{i+1}. The same then holds for $\mathcal{A}'_{i+1}$ since the reduction algorithm is assumed to work correctly.

We started with setting $T'_1 = T_1$, so $\mathcal{A}'_1$ indeed represents True_1.

Let us first unravel the definitions to see what we need to show. Let $i \in [n-1]$ and suppose that $c : G[X_{i+1}] \to H$ extends to a homomorphism $\phi : G[\{v_1,\dots,v_{i+1}\}] \to H$ (i.e. $T_i[c] = \mathrm{True}$). For the definition of represents, we will then assume there is a homomorphism $d : G[Y_{i+1}] \to H$ for which $c \cup d$ respects all edges from the $(i+1)$th cut (those in E_{i+1}), i.e. this corresponds to a “one-entry in the compatibility matrix”. What needs to be shown is that this “one-entry” can also be generated via a coloring coming from $\mathcal{A}_{i+1}$, that is, there is $\alpha : G[X_{i+1}] \to H$, such that $(\alpha(x_1),\dots,\alpha(x_k)) \in \mathcal{A}_{i+1}$ and $(\alpha \cup d)|_{G[E_{i+1}]}$ is a homomorphism.

By assumption, $\phi \cup d$ respects all the edges with at least one endpoint in $\{v_1,\dots,v_{i+1}\}$, and in particular those with one endpoint in $\{v_1,\dots,v_i\}$. Since $\mathcal{A}'_i$ is a representative set of True_i, there must be $c' : G[X_i] \to H$ such that $(c'(x'_1),\dots,c'(x'_{k'})) \in \mathcal{A}'_i$, for $\{x'_1,\dots,x'_{k'}\} = X_i$, and where $c' \cup \phi|_{\{v_{i+1}\}} \cup d$ respects all the edges with at least one endpoint in $\{v_1,\dots,v_i\}$. We set $\alpha = (c' \cup \phi|_{\{v_{i+1}\}})|_{X_{i+1}}$. Then $(\alpha(x_1),\dots,\alpha(x_k)) \in \mathcal{A}_{i+1}$, by definition of how we obtain $\mathcal{A}_{i+1}$ from $\mathcal{A}'_i$. Moreover, $\alpha \cup d$ is a homomorphism $G[E_{i+1}] \to H$, as desired. ◀

4 Computing representative sets via half-induced matchings

In this section we present one of our main technical contributions, i.e., an algorithm to compute H-representative sets whose size is bounded in terms of $\mathrm{him}(H)$. Actually, our approach is rather general since it finds representative sets non-trivially fast for any large Kronecker power of a matrix with small him parameter.

We will show how to find a representative set that has one fewer element, by finding some element that can be safely removed. We then use this intermediate result to find our final reduction algorithm, which will result in the following lemma.

▶ **Lemma 8.** *Let $\ell \geq 1$ and $k \geq 2$ be integers. Let $A \in \{0,1\}^{h\times h}$ be a matrix with $\mathrm{him}(A) < \ell$, and let $\mathcal{A} \subseteq [h]^k$. Then we can compute $\mathcal{A}' \subseteq \mathcal{A}$ that $A^{\otimes k}$-represents $\mathcal{A}$ with $|\mathcal{A}'| \leq k^{k\ell}$ in time $\mathcal{O}(|\mathcal{A}|^2h^2k^2)$.*

We will combine the reduction algorithm from Lemma 8 with Theorem 7 to find the following result.

▶ **Theorem 9.** *The Hom problem on an instance (G, H), where G is given with a linear ordering of width k, can be solved in time $\mathcal{O}(k^{2k\cdot(\mathrm{him}(H)+1)} \cdot |V(H)|^4|V(G)|)$.*

We emphasize that the algorithm does not need to know the value of $\mathrm{him}(H)$.

Proof. Let $h = |V(H)|$ and let A_H be the adjacency matrix of H. Recall that $\mathrm{him}(A_H) = \mathrm{him}(H)$ is always an integer. By Lemma 8 we have a reduction algorithm R that returns a representative set of size $\mathsf{size}(H,k) \leq k^{k\cdot(\mathrm{him}(H)+1)}$ in time $\mathsf{time}(|\mathcal{A}|, H, k) = \mathcal{O}(|\mathcal{A}|^2h^2k^2)$. Then

$$\mathsf{time}(\mathsf{size}(H,k)\cdot h, H, k) = \mathcal{O}\left(k^2 \cdot k^{2k\cdot(\mathrm{him}(H)+1)} \cdot h^4\right).$$

By Theorem 7 we find an algorithm that decides Hom(H) in time

$$\mathcal{O}\left((\mathsf{size}(H,k)\cdot h + \mathsf{time}(\mathsf{size}(H,k)\cdot h, H, k))|V(G)|\right) = \mathcal{O}(k^{2k\cdot(\mathrm{him}(H)+1)}h^4 \cdot |V(G)|).$$

This completes the proof. ◀

Since $\mathrm{him}(H) \leq \mathrm{mimsup}(H)$ (see Lemma 11), we obtain Theorem 4 as a corollary from Theorem 9. Lemma 8 and Lemma 11 also imply Theorem 3.

In order to prove the lemma, we will perform a recursive algorithm for which we want to no longer treat all the coordinates symmetrically. We therefore define

$$g_k(\ell_1, \dots, \ell_k) = \binom{\sum_i \ell_i}{\ell_1, \dots, \ell_k}.$$

When $\ell_1 = \cdots = \ell_k = \ell$, we have $g_k(\ell, \dots, \ell) = \binom{k\ell}{\ell,\dots,\ell} \leq k^{k\ell}$ (= the number of partitions of $k\ell$ into k parts, which no longer need to have the same size). The lemma will follow easily from the following more complicated statement.

▶ **Lemma 10.** *Let $k \geq 2, \ell_1, \dots, \ell_k \geq 1$ be integers. Let $A \in \{0,1\}^{h\times h}$ be a matrix and let $\mathcal{A} \subseteq [h]^k$ with $|\mathcal{A}| \geq g_k(\ell_1, \dots, \ell_k)$. Suppose that for every $i \in [k]$, for the set of rows $\mathcal{R}_i = \{r_i \mid r \in \mathcal{A}\}$, we have $\mathrm{him}(A[\mathcal{R}_i, \cdot]) < \ell_i$. Then there exists $v \in \mathcal{A}$ such that $\mathcal{A} \setminus \{v\}$ $A^{\otimes k}$-represents $\mathcal{A}$. Moreover, v can be found in time $\mathcal{O}(\sum_{i=1}^k \ell_i \cdot |\mathcal{A}|hk)$.*

Proof. Note that $|\mathcal{A}| \geq g_k(\ell_1, \ldots, \ell_k) \geq 1$ for $\ell_1, \ldots, \ell_k, k \geq 1$ so $\mathcal{A}$ is non-empty.

For $i \in [k]$ and $u \in [h]$, let

$$\mathcal{A}_u^i = \{v = (v_1, \ldots, v_k) \in \mathcal{A} \mid A[v_i, u] = 0\}$$

be the set of rows which cannot "represent" u in the ith coordinate. We choose $v \in \mathcal{A}$ (arbitrarily). We then iterate over $u \in [h]$ and $i \in [k]$ to find if there is (u, i) for which

- $A[v_i, u] = 1$, and
- $|\mathcal{A}_u^i| \geq g_k(\ell_1, \ldots, \ell_i - 1, \ldots, \ell_k)$.

This step can be performed in time $\mathcal{O}(|\mathcal{A}|hk)$.

If we cannot find such (u, i) pair for v, then we return v as the row to be removed from $\mathcal{A}$ (and the algorithm terminates).

Otherwise, we did find (u, i). If $\ell_i = 1$, then since $\operatorname{him}(A[\mathcal{R}_i, \cdot]) < \ell_i$, we know $A[\mathcal{R}_i, \cdot]$ has all zero-entries and so $A[v_i, u] = 1$ would not have been possible. This means that $\ell_i \geq 2$. We apply the same process after updating $\ell_i \leftarrow \ell_i - 1$ and $\mathcal{A} \leftarrow \mathcal{A}_u^i$. Note that $v \notin \mathcal{A}_u^i$ and $\ell_i - 1 \geq 1$. We will show that

- when v is returned, indeed $\mathcal{A} \setminus \{v\}$ $A^{\otimes k}$-represents $\mathcal{A}$, and
- when we recursively apply the algorithm, the conditions of the lemma are again satisfied, for which it remains to show that $\operatorname{him}(A[\mathcal{R}_i', \cdot]) < \ell_i - 1$ for $\mathcal{R}_i' = \{r_i \mid r \in \mathcal{A}_u^i\}$.

Since we reduce $\sum_{i=1}^k \ell_i$ by one in each recursive call, the algorithm will terminate. Moreover, the number of recursive calls is at most $\sum_{i=1}^k \ell_i$. This shows that assuming the claims above, the time complexity is as stated.

Correctness. We first show the first claim: if the algorithm outputs v, indeed it can be removed. Note that if for some subset $\mathcal{A}' \subseteq \mathcal{A}$, it is the case that $\mathcal{A}' \setminus \{v\}$ represents $\mathcal{A}'$, then

$$\mathcal{A}' \setminus \{v\} \cup (\mathcal{A} \setminus \mathcal{A}') = \mathcal{A} \setminus \{v\}$$

will also represent $\mathcal{A}$. This means we only have to check the claims in the "base case". Suppose towards a contradiction that we wrongly outputted $v \in \mathcal{A}$, so

- there exists $u \in [h]^k$ such that $A^{\otimes k}[v, u] = 1$ yet $A^{\otimes k}[v', u] = 0$ for all $v' \in \mathcal{A} \setminus \{v\}$ (since we "wrongly" outputted v, there needs to be a reason why we could not remove it),
- for this u, for all $i \in [k]$, $|\mathcal{A}_{u_i}^i| < g_k(\ell_1, \ldots, \ell_i - 1, \ldots, \ell_k)$ (else the algorithm would have "recursed" instead of outputting v).

The fact that $A^{\otimes k}[v', u] = 0$ in the first condition, means that each $v' \in \mathcal{A} \setminus \{v\}$ is an element of $\mathcal{A}_{u_i}^i$ for some $i \in [k]$. In particular,

$$|\mathcal{A} \setminus \{v\}| \leq \sum_{i=1}^k |\mathcal{A}_{u_i}^i| \leq \sum_{i=1}^k g_k(\ell_1, \ldots, \ell_i - 1, \ldots, \ell_k) - k = g_k(\ell_1, \ldots, \ell_k) - k,$$

which contradicts the assumptions of the lemma since $k \geq 2$.

We now prove the second claim: the conditions of the lemma are satisfied when we "recurse". By assumption, $\ell_i \geq 1$ for all i and the new $\mathcal{A}$ is sufficiently large. Moreover, him can also decrease when taking submatrices, so indeed we only need to show that $\operatorname{him}(A[\mathcal{R}_i', \cdot]) < \ell_i - 1$ for $\mathcal{R}_i' = \{r_i \mid r \in \mathcal{A}_{u_i}^i\}$. If there is a half-induced matching of size $\ell_i - 1$, induced on rows $w_1, \ldots, w_{\ell_i - 1} \in \mathcal{R}_i'$ and columns $z_1, \ldots, z_{\ell_i - 1}$, then there is a half-induced matching of size ℓ_i in $A[\mathcal{R}_i, \cdot]$ by considering rows $w_1, \ldots, w_{\ell_i - 1}, v_i \in \mathcal{R}_i$ and columns $z_1, \ldots, z_{\ell - 1}, u$. But by assumption this does not exist, so indeed $\operatorname{him}(A[\mathcal{R}_i', \cdot]) < \ell_i - 1$. ◀

Proof of Lemma 8. Suppose that $|\mathcal{A}| \geq g_k(\ell, \ldots, \ell)$. For $i \in [k]$, set $\mathcal{R}_i = \{r_i \mid r \in \mathcal{A}\}$. Then $\mathrm{him}(A[\mathcal{R}_i, \cdot]) < \ell$ for each i. By Lemma 10 we can find a row v in $\mathcal{A}$ such that $\mathcal{A} \setminus \{v\}$ $A^{\otimes k}$-represents $\mathcal{A}$ in time $\mathcal{O}(\ell k \cdot |\mathcal{A}| hk) = \mathcal{O}(|\mathcal{A}| h^2 k^2)$, where we use that $\ell \leq h$.

We repeat this at most $|\mathcal{A}| - g_k(\ell, \ldots, \ell)$ times until we find the desired representative set in time $\mathcal{O}(|\mathcal{A}|^2 h^2 k^2)$. ◀

5 Comparing him and mimsup

In this section, we discuss the relation between the considered parameters.

▶ **Lemma 11.** *Let A be a matrix. Then* $\mathrm{mimsup}(A) \geq \mathrm{him}(A) \geq \mathrm{mim}(A)$.

Proof. The second inequality follows directly since each induced matching is a half-induced matching. We prove the first inequality.

Let $R = \{a_1, \ldots, a_i\}$ and $C = \{b_1, \ldots, b_i\}$ be the rows and columns respectively of a maximum half-induced matching in A. We may assume that these are ordered such that $A[a_j, b_j] = 1$ for all $j \in [i]$ and $A[a_k, b_j] = 0$ for all $k < j$. For integers $s \geq 1$, we consider the submatrix of $A^{\otimes is}$ induced on the rows consisting of "balanced" sequences, and similarly for the columns

$$\{(r_1, \ldots, r_{is}) \in R^{is} \mid |\{\ell : r_\ell = a_j\}| = s, \text{ for every } j \in [i]\},$$
$$\{(c_1, \ldots, c_{is}) \in C^{is} \mid |\{\ell : c_\ell = b_j\}| = s, \text{ for every } j \in [i]\}.$$

We claim this forms an induced matching of size $\binom{is}{s,\ldots,s}$. By Stirling approximation, $\binom{is}{s,\ldots,s} = i^{(1+o(1))is}$ and so the claim implies that $\mathrm{mimsup}(A) \geq i = \mathrm{him}(A)$. Since the size is clear from the definition, it only remains to check that it indeed forms an induced matching. To show this, we explain why the row of

$$r = (a_1, \ldots, a_1, a_2, \ldots, a_2, \ldots, a_i, \ldots, a_i)$$

has a single one entry in column

$$c = (b_1, \ldots, b_1, b_2, \ldots, b_2, \ldots, b_i, \ldots, b_i).$$

The other cases follow by symmetry. It is clear that $A^{\otimes is}[r, c] = 1$. Any column c' with $A^{\otimes is}[r, c'] = 1$ must have $A[a_1, c'_j] = 1$ for all $j \in [s]$. But $A[a_1, b_j] = 0$ when $j > 1$, so this implies that $\{j : c'_j = b_1\} = [s]$. Similarly, we require $A[a_2, c'_j] = 1$ for all $j \in [s+1, 2s]$, but c' has already "used" all its b_1's and so $\{j \mid c'_j = b_2\}$ needs to be $[s+1, 2s]$. Continuing inductively, we find that $\{j \mid c'_j = b_k\} = [(k-1)s + 1, ks]$ for all $k \in [i]$, that is, $c' = c$. ◀

We are now ready to prove previously claimed bounds.

▶ **Theorem 3.** *For every non-bipartite graph H with adjacency matrix A_H and $k \in \mathbb{N}$,*

$$\mathrm{him}(A_H) \leq \mathrm{mimsup}(A_H) = \limsup_{k\to\infty} \mathrm{mim}(A_H^{\otimes k})^{1/k} \quad \textit{and} \quad \mathrm{mim}(A_H^{\otimes k}) \leq k^{(\mathrm{him}(A_H)+1)k}.$$

Proof. Lemma 11 shows the first inequality. The proof of Lemma 10 implies that for any matrix A, $\mathrm{mim}(A^{\otimes k}) \leq k^{\mathrm{him}(A)k}$, since no smaller representative set can be found when the rows induce an induced matching (with some set of columns). ◀

The following observation implies that sequences such as $\mathrm{mim}(A^{\otimes 2^k})^{1/2^k}$ are non-decreasing.

▶ **Lemma 12 (♠).** *Given two matrices A and B,* $\mathrm{mim}(A \otimes B) \geq \mathrm{mim}(A)\,\mathrm{mim}(B)$*. In particular,* $\mathrm{mimsup}(A^{\otimes k}) = \mathrm{mimsup}(A)^k$*.*

The results above also imply that

$$\mathrm{mimsup}(A) = \limsup_{k\to\infty} \mathrm{him}(A^{\otimes k})^{1/k}.$$

At first glance, it may be natural to conjecture that in fact $\mathrm{mimsup}(A) = \mathrm{him}(A)$ for all matrices A. This is however not true, as the following result shows.

▶ **Theorem 13 (♠).** *For all sufficiently large integers h, there is a symmetric $(2h \times 2h)$ matrix A with* $\mathrm{him}(A) \leq 10 \log_2 h$ *and* $\mathrm{mimsup}(A) \geq \sqrt{h}$*.*

6 Conclusion

An obvious open problem is to fully resolve Open Problem 1. As discussed, to achieve this goal we only lack a fast algorithm that computes representative sets for partial solutions to Hom(H). A far more ambitious (and probably currently out of reach) goal is to provide a (more) fine-grained version of the Courcelle's theorem for deciding any graph property definable in the monadic second-order logic. While being homomorphic to a given graph H is of course only a very special sort of such a property, we find our progress on Open Problem 1 encouraging in this respect and hope that eventually similar connections between the complexity of more general computational problems and asymptotic rank parameters can be made as well. In particular, we believe that mimsup (or a similar parameter that tracks the asymptotic behavior under appropriate products) is likely to determine the limit of dynamic programming approaches in other settings as well, especially those determined by various graph width parameters.

Another suggested direction of research is purely combinatorial/algebraic: we expect that mimsup is an interesting parameter for further study in its own right. We suggest following questions. (i) What type of values can $\mathrm{mimsup}(H)$ take given a matrix H? Can it take non-integer values? Similar questions have recently been investigated for asymptotic tensor parameters, see e.g. [3,6]. (ii) What is the value of mimsup for a $n \times n$ random matrix, where each entry of the matrix gets sampled independently to be 1 with probability p and to be 0 with probability $1 - p$? (iii) We showed that him and the support rank are not functionally equivalent. Is mimsup functionally equivalent to either him or the support rank?

References

1 Noga Alon and Eyal Lubetzky. The shannon capacity of a graph and the independence numbers of its powers. *IEEE Trans. Inf. Theory*, 52(5):2172–2176, 2006. doi:10.1109/TIT.2006.872856.

2 Srinivasan Arunachalam, Péter Vrana, and Jeroen Zuiddam. The asymptotic induced matching number of hypergraphs: Balanced binary strings. *The Electronic Journal of Combinatorics*, 27(3), 2020. doi:10.37236/9019.

3 Andreas Blatter, Jan Draisma, and Filip Rupniewski. Countably many asymptotic tensor ranks. *arXiv*, 2022. arXiv:2212.12219.

4 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.

5 Glencora Borradaile and Hung Le. Optimal dynamic program for r-domination problems over tree decompositions. In Jiong Guo and Danny Hermelin, editors, *IPEC 2016*, volume 63 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.8.

6 Jop Briët, Matthias Christandl, Itai Leigh, Amir Shpilka, and Jeroen Zuiddam. Discreteness of asymptotic tensor ranks. *arXiv*, 2023. arXiv:2306.01718.

7 Andrei A. Bulatov and Amirhossein Kazeminia. Complexity classification of counting graph homomorphisms modulo a prime number. In Stefano Leonardi and Anupam Gupta, editors, *STOC 2022*, pages 1024–1037. ACM, 2022. doi:10.1145/3519935.3520075.

8 Peter Bürgisser, Michael Clausen, and Amin Shokrollahi. Algebraic complexity theory. In *Grundlehren der mathematischen Wissenschaften*, 1997.

9 Jin-Yi Cai and Ashwin Maran. The complexity of counting planar graph homomorphisms of domain size 3. In Barna Saha and Rocco A. Servedio, editors, *STOC 2023*, pages 1285–1297. ACM, 2023. doi:10.1145/3564246.3585173.

10 Airlie Chapman and Mehran Mesbahi. On strong structural controllability of networked systems: A constrained matching approach. In *2013 American Control Conference*, pages 6126–6131, 2013. doi:10.1109/ACC.2013.6580798.

11 Prasad Chaugule, Nutan Limaye, and Aditya Varre. Variants of homomorphism polynomials complete for algebraic complexity classes. *ACM Trans. Comput. Theory*, 13(4):21:1–21:26, 2021. doi:10.1145/3470858.

12 Rajesh Chitnis, László Egri, and Dániel Marx. List h-coloring a graph by removing few vertices. *Algorithmica*, 78(1):110–146, 2017. doi:10.1007/s00453-016-0139-6.

13 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.

14 Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *STOC 2017*, pages 210–223. ACM, 2017. doi:10.1145/3055399.3055502.

15 Radu Curticapean, Nathan Lindzey, and Jesper Nederlof. A tight lower bound for counting Hamiltonian cycles via matrix rank. In Artur Czumaj, editor, *SODA 2018*, pages 1080–1099. SIAM, 2018. doi:10.1137/1.9781611975031.70.

16 Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In Robert Krauthgamer, editor, *SODA 2016*, pages 1650–1669. SIAM, 2016. doi:10.1137/1.9781611974331.ch113.

17 Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast hamiltonicity checking via bases of perfect matchings. *J. ACM*, 65(3):12:1–12:46, 2018. doi:10.1145/3148227.

18 Ronald de Wolf. Nondeterministic quantum query and communication complexities. *SIAM Journal on Computing*, 32(3):681–699, 2003. doi:10.1137/S0097539702407345.

19 MARTIN E. DYER and CATHERINE S. GREENHILL. The complexity of counting graph homomorphisms. *RANDOM STRUCT. ALGORITHMS*, 17(3-4):260–289, 2000.

20 László Egri, Andrei A. Krokhin, Benoît Larose, and Pascal Tesson. The complexity of the list homomorphism problem for graphs. In Jean-Yves Marion and Thomas Schwentick, editors, *STACS 2010*, volume 5 of *LIPIcs*, pages 335–346. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPIcs.STACS.2010.2467.

21 László Egri, Dániel Marx, and Paweł Rzążewski. Finding list homomorphisms from bounded-treewidth graphs to reflexive graphs: a complete complexity characterization. In Rolf Niedermeier and Brigitte Vallée, editors, *STACS 2018*, volume 96 of *LIPIcs*, pages 27:1–27:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.STACS.2018.27.

22 Paul Erdős and George Szekeres. A combinatorial problem in geometry. *Compositio mathematica*, 2:463–470, 1935.

23 Baris Can Esmer, Jacob Focke, Dániel Marx, and Paweł Rzążewski. List homomorphisms by deleting edges and vertices: tight complexity bounds for bounded-treewidth graphs. *CoRR*, abs/2210.10677, 2022. doi:10.48550/arXiv.2210.10677.

24 M. Fekete. Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten. *Mathematische Zeitschrift*, 17(1):228–249, December 1923. `doi:10.1007/bf01504345`.

25 Jacob Focke, Dániel Marx, Fionn Mc Inerney, Daniel Neuen, Govind S. Sankar, Philipp Schepper, and Philip Wellnitz. Tight complexity bounds for counting generalized dominating sets in bounded-treewidth graphs. In Nikhil Bansal and Viswanath Nagarajan, editors, *SODA 2023*, pages 3664–3683. SIAM, 2023. `doi:10.1137/1.9781611977554.ch140`.

26 Jacob Focke, Dániel Marx, and Paweł Rzążewski. Counting list homomorphisms from graphs of bounded treewidth: tight complexity bounds. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *SODA 2022*, pages 431–458. SIAM, 2022. `doi:10.1137/1.9781611977073.22`.

27 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. `doi:10.1145/2886094`.

28 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Representative families of product families. *ACM Trans. Algorithms*, 13(3):36:1–36:29, 2017. `doi:10.1145/3039243`.

29 Robert Ganian, Thekla Hamm, Viktoriia Korchemna, Karolina Okrasa, and Kirill Simonov. The fine-grained complexity of graph homomorphism parameterized by clique-width. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *ICALP 2022*, volume 229 of *LIPIcs*, pages 66:1–66:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.ICALP.2022.66`.

30 Chris Godsil. Problems in algebraic combinatorics. *Electr. J. Comb.*, 2, January 1995. `doi:10.37236/1224`.

31 Carla Groenland, Isja Mannens, Jesper Nederlof, Marta Piecyk, and Pawel Rzazewski. Towards tight bounds for the graph homomorphism problem parameterized by cutwidth via asymptotic rank parameters. *CoRR*, abs/2312.03859, 2023. `doi:10.48550/arXiv.2312.03859`.

32 Carla Groenland, Isja Mannens, Jesper Nederlof, and Krisztina Szilágyi. Tight bounds for counting colorings and connected edge sets parameterized by cutwidth. In Petra Berenbrink and Benjamin Monmege, editors, *STACS 2022*, volume 219 of *LIPIcs*, pages 36:1–36:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.STACS.2022.36`.

33 Pavol Hell and Jaroslav Nešetřil. The core of a graph. *Discrete Mathematics*, 109(1-3):117–126, 1992. `doi:10.1016/0012-365X(92)90282-K`.

34 Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990. `doi:10.1016/0095-8956(90)90132-J`.

35 Lars Jaffke and Bart M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. *Discret. Appl. Math.*, 327:33–46, 2023. `doi:10.1016/j.dam.2022.11.011`.

36 Bart M. P. Jansen and Jesper Nederlof. Computing the chromatic number using graph decompositions via matrix rank. *Theor. Comput. Sci.*, 795:520–539, 2019. `doi:10.1016/j.tcs.2019.08.006`.

37 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *Discret. Appl. Math.*, 264:90–117, 2019. `doi:10.1016/j.dam.2018.11.002`.

38 Tomasz Kociumaka and Marcin Pilipczuk. Deleting vertices to graphs of bounded genus. *Algorithmica*, 81(9):3655–3691, 2019. `doi:10.1007/s00453-019-00592-7`.

39 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. `doi:10.1145/3390887`.

40 Michael Lampis. Finer tight bounds for coloring on clique-width. *SIAM J. Discret. Math.*, 34(3):1538–1558, 2020. `doi:10.1137/19M1280326`.

41 Benoit Larose. Families of strongly projective graphs. *Discussiones Mathematicae Graph Theory*, 22:271–292, 2002.

42 Benoit Larose and Claude Tardif. Strongly rigid graphs and projectivity. *Multiple-Valued Logic*, 7:339–361, 2001.

43 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/92.

44 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018. doi:10.1145/3170442.

45 Tomasz Łuczak and Jaroslav Nešetřil. Note on projective graphs. *Journal of Graph Theory*, 47(2):81–86, 2004.

46 Dániel Marx, Govind S. Sankar, and Philipp Schepper. Degrees and gaps: Tight complexity results of general factor problems parameterized by treewidth and cutwidth. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *ICALP 2021*, volume 198 of *LIPIcs*, pages 95:1–95:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.95.

47 Dániel Marx, Govind S. Sankar, and Philipp Schepper. Anti-Factor Is FPT Parameterized by Treewidth and List Size (But Counting Is Hard). In Holger Dell and Jesper Nederlof, editors, *IPEC 2022*, volume 249 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:23, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.IPEC.2022.22.

48 Burkhard Monien. The complexity of determining paths of length k. In Manfred Nagl and Jürgen Perl, editors, *WG '83*, pages 241–251. Universitätsverlag Rudolf Trauner, Linz, 1983.

49 Jesper Nederlof. Algorithms for np-hard problems via rank-related parameters of matrices. In Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors, *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 145–164. Springer, 2020. doi:10.1007/978-3-030-42071-0_11.

50 Jesper Nederlof. Bipartite TSP in $O(1.9999^n)$ time, assuming quadratic time matrix multiplication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *STOC 2020*, pages 40–53. ACM, 2020. doi:10.1145/3357713.3384264.

51 Jaroslav Nešetřil and Aleš Pultr. A Dushnik - Miller type dimension of graphs and its complexity. In Marek Karpiński, editor, *Fundamentals of Computation Theory*, pages 482–493, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg.

52 Jaroslav Nešetřil and Vojtéch Rödl. A simple proof of the Galvin-Ramsey property of the class of all finite graphs and a dimension of a graph. *Discrete Mathematics*, 23(1):49–55, 1978. doi:10.1016/0012-365X(78)90186-3.

53 Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *ESA 2020*, volume 173 of *LIPIcs*, pages 74:1–74:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.74.

54 Karolina Okrasa and Paweł Rzążewski. Fine-grained complexity of the graph homomorphism problem for bounded-treewidth graphs. *SIAM J. Comput.*, 50(2):487–508, 2021. doi:10.1137/20M1320146.

55 D.D. Olesky, Michael Tsatsomeros, and P. van den Driessche. Qualitative controllability and uncontrollability by a single entry. *Linear Algebra and its Applications*, 187:183–194, 1993. doi:10.1016/0024-3795(93)90134-A.

56 Marta Piecyk and Paweł Rzążewski. Fine-grained complexity of the list homomorphism problem: Feedback vertex set and cutwidth. In Markus Bläser and Benjamin Monmege, editors, *STACS 2021*, volume 187 of *LIPIcs*, pages 56:1–56:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.56.

57 Michał Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In *MFCS 2011*, volume 6907, pages 520–531. Springer, 2011.

58 Marc Roth and Philip Wellnitz. Counting and finding homomorphisms is universal for parameterized complexity theory. In Shuchi Chawla, editor, *SODA 2020*, pages 2161–2180. SIAM, 2020. doi:10.1137/1.9781611975994.133.

59 Maguy Trefois and Jean-Charles Delvenne. Zero forcing number, constrained matchings and strong structural controllability. *Linear Algebra and its Applications*, 484:199–218, 2015.

60 Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In Amos Fiat and Peter Sanders, editors, *ESA 2009*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009. doi:10.1007/978-3-642-04128-0_51.