Linearly ordered colourings of hypergraphs

TAMIO-VESA NAKAJIMA, Department of Computer Science, University of Oxford, UK

STANISLAV ŽIVNÝ, Department of Computer Science, University of Oxford, UK

A linearly ordered (LO) k-colouring of an r-uniform hypergraph assigns an integer from $\{1, \ldots, k\}$ to every vertex so that, in every edge, the (multi)set of colours has a unique maximum. Equivalently, for $r = 3$, if two vertices in an edge are assigned the same colour, then the third vertex is assigned a larger colour (as opposed to a different colour, as in classic non-monochromatic colouring). Barto, Battistelli, and Berg [STACS'21] studied LO colourings on 3-uniform hypergraphs in the context of promise constraint satisfaction problems (PCSPs). We show two results.

First, given a 3-uniform hypergraph that admits an LO 2-colouring, one can find in polynomial time an LO k-colouring with $k = O(\sqrt[3]{n \log \log n / \log n})$.

Second, given an r-uniform hypergraph that admits an LO 2-colouring, we establish NP-hardness of finding an LO k-colouring for every constant uniformity $r \geq k + 2$. In fact, we determine relationships between polymorphism minions for all uniformities $r \geq 3$, which reveals a key difference between $r < k + 2$ and $r \geq k + 2$ and which may be of independent interest. Using the algebraic approach to PCSPs, we actually show a more general result establishing NP-hardness of finding an LO k-colouring for LO ℓ-colourable r-uniform hypergraphs for $2 \leq \ell \leq k$ and $r \geq k - \ell + 4$.

CCS Concepts: • **Theory of computation → Problems, reductions and completeness**; **Approximation algorithms analysis**; • **Mathematics of computing → Graph theory**.

Additional Key Words and Phrases: hypegraph colourings, promise constraint satisfaction, PCSP, polymorphisms, minions, algebraic approach

1 INTRODUCTION

The computational complexity of the *approximate graph colouring* problem [23] is an outstanding open problem in theoretical computer science. Given a 3-colourable graph G on n vertices, is it possible to find a k-colouring of G? On the tractability side, the current best is a polynomial-time algorithm of Kawarabayashi and Thorup [31] that finds a k-colouring with $k = k(n) = n^{0.199}$ colours. On the intractability side, the state-of-the-art for constant k has only recently been improved from $k = 4$, due to Khanna, Linial, and Safra [32] and Guruswami and Khanna [24] to $k = 5$, due to Barto, Bulín, Krokhin, and Opršal [5]. The authors of [5] introduced a general algebraic methodology for studying the computational complexity of so-called promise constraint satisfaction problems (PCSPs). Going beyond the work in [5], for graphs with a promised higher chromatic number than three, the current best intractability results for constantly many extra colours is due to Wrochna

Authors' addresses: Tamio-Vesa Nakajima, Department of Computer Science, University of Oxford, Oxford, UK, tamio-vesa.nakajima@cs.ox.ac.uk; Stanislav Živný, Department of Computer Science, University of Oxford, Oxford, UK, standa.zivny@cs.ox.ac.uk.

and Živný [41], building on the work of Huang [29]. Hardness of approximate graph colouring has been established under stronger assumptions. In particular, Dinur, Mossel, and Regev [20] showed NP-hardness under a non-standard variant of the UGC Conjecture of Khot [34] and Guruswami and Sandeep [26] (building on [41]) showed NP-hardness under the d-to-1 conjecture [34] for any fixed d.

The situation is much better understood for the approximate *hypergraph* colouring problem with the classic notion of a colouring leaving no edge monochromatic. A celebrated result of Dinur, Regev, and Smyth established that finding a k-colouring of a 3-uniform hypergraph that is ℓ-colourable is NP-hard for every constant $2 \leq \ell \leq k$ [21] (and this also implies the same result on r-uniform hypergraphs for every constant uniformity $r \geq 3$). This was also recently proved, with different techniques, by Wrochna [40].

Different variants of approximate hypergraph colourings, such as rainbow colourings, were studied, e.g. in [2, 12, 14, 25, 27], but most complexity classifications related to these problems are open. Some intractability results are also known for colourings with a super-constant number of colours. For graphs, conditional hardness was established by Dinur and Shinkar [22]. For hypergraphs, intractability results were obtained by Bhangale [10] and by Austrin, Bhanghale, and Potukuchi [1].

Barto, Battistelli, and Berg have recently studied systematically a certain type of PCSPs on non-Boolean domains and identified a very natural variant of k-colourings of 3-uniform hypergraphs, called *linearly ordered* (LO) k-colourings [4]. A k-colouring of a 3-uniform hypergraph with colours $[k] = \{1, \ldots, k\}$ is an LO colouring if, for every edge, it holds that, if two vertices are coloured with the same colour, then the third vertex is coloured with a larger colour. (In the classic non-monochromatic colouring, the requirement is that the third vertex should be coloured with a *different* colour, but not necessarily a larger one.) An LO 2-colouring is thus a "1-in-3" colouring. Barto et al. asked whether finding an LO k-colouring of a 3-uniform hypergraph is NP-hard for a fixed $k \geq 3$ if the input hypergraph is promised to admit an LO 2-colouring.

Contributions. While we do not resolve the question raised in [4], we obtain non-trivial results, both positive (algorithmic) and negative (hardness).

First, we present an efficient algorithm for finding an LO k-colouring of a 3-uniform hypergraph that admits an LO 2-colouring with $k = O(\sqrt[3]{n \log \log n / \log n})$. As mentioned above, there are only a few results on hypergraph colourings with super-constantly many colours, e.g. [18, 35, 36] that deal with hypergraph non-monochromatic colourings.

Second, we establish NP-hardness of finding an LO k-colouring of an r-uniform hypergraph if an LO 2-colouring is promised for every constant uniformity $r \geq k + 2$. In fact, we prove a more general result that finding an LO k-colouring of an r-uniform hypergraph admitting an LO ℓ-colouring is NP-hard for every constant $2 \leq \ell \leq k$ and $r \geq k - \ell + 4$. This result is based the algebraic approach to PCSPs and in particular on minions [5]. As a matter of fact, we establish relationships of the polymorphism minions of the LO 2- vs k-colourings on r-uniform hypergraphs, which may be of independent interest. This gives the advertised intractability result but also an impossibility result on certain types of polynomial-time reductions (namely pp-constructions [5]) between LO 2- vs k-colourings on r-uniform hypergraphs with uniformity $r \geq k + 2$ and $r < k + 2$, cf. the discussion in Section 5.

Related work. A classic non-monochromatic colouring of a hypergraph requires that no edge should be monochromatic. Rainbow colourings, introduced in [3], require that every colour should occur at least once in every edge, cf. also [2, 25, 27]. LO colourings are somewhat in-between non-monochromatic and rainbow colourings, requiring that the largest colour in every edge is

used exactly once. LO colourings – as well as a related notion of *conflict-free* colourings, which require that at least one colour in each edge should occur exactly once – have been studied under the name of *unique maximum* colourings, both for graphs (where the requirement is on paths) [17] and hypergraphs [16]. As far as we know, our results are incomparable with existing results from the rich body of literature on unique maximum colourings. We refer the reader to the papers cited above and the references therein.

2 PRELIMINARIES

An *r-uniform hypergraph* H is a pair (V, E) where V is the set of *vertices* of the hypergraph, and $E \subseteq V^r$ is the set of *edges* of the hypergraph. In our context the order of the vertices in each edge is irrelevant. We will allow vertices to appear multiple times in edges; however, we exclude edges of the form $(v, \ldots, v)$ — such edges would be impossible in the problems we will consider anyway. We say that two distinct vertices u, v are *neighbours* if they both belong to some edge $e \in E$. Let $N(u)$ be the set of neighbours of u. Call a set S an *independent set* of a hypergraph H if and only if no two members of S are neighbours.

A *linearly ordered* (LO) k-*colouring* of an r-uniform hypergraph $H = (V, E)$ is an assignment $c : V \to [k]$ of colours from $[k] = \{1, \ldots, k\}$ to the vertices of H such that, for each edge $(v_1, \ldots, v_r) \in E$, the sequence $c(v_1), \ldots, c(v_r)$ has a unique maximum. We omit the "k-" if the number of colours is unimportant.

Example 2.1. Consider the hypergraph $H = (V, E)$, where $V = [4] = \{1, 2, 3, 4\}$ and $E = \{(1, 2, 3), (1, 2, 4)\}$. The assignment $c = \{1 \mapsto 1, 2 \mapsto 1, 3 \mapsto 2, 4 \mapsto 2\}$ is an LO 2-colouring, and $c'(x) = x$ is an LO 4-colouring. On the other hand, $c''(x) = 3 - c(x)$ is not an LO colouring at all, since both of the edges have two equal maximal elements when mapped through c''.

Finding an LO k-colouring, for constant $k \geq 3$, of a 3-uniform hypergraph that admits an LO 2-colouring was studied by Barto et al. [4] in the context of promise constraint satisfaction problems (PCSPs), which we define next.

Promise CSPs. Promise CSPs have been introduced in the works of Austrin, Guruswami, and Håstad [3] and Brakensiek and Guruswami [13]. We follow the notation and terminology of Barto, Bulín, Krokhin, and Opršal [5], adapted to structures consisting of a single relation.

An *r-ary structure* is a pair $\mathbf{D} = (D, R^{\mathbf{D}})$, where $R^{\mathbf{D}} \subseteq D^r$ and D is finite. We call D the *domain* of the structure, and $R^{\mathbf{D}}$ the *relation* of the structure. For two r-ary structures $\mathbf{A}, \mathbf{B}$, a *homomorphism from $\mathbf{A}$ to $\mathbf{B}$* is a function $h : A \to B$ such that, for each $(a_1, \ldots, a_r) \in R^{\mathbf{A}}$, we have $(h(a_1), \ldots, h(a_r)) \in R^{\mathbf{B}}$. This is written $h : \mathbf{A} \to \mathbf{B}$. If we wish to assert only the existence of such a homomorphism, we write $\mathbf{A} \to \mathbf{B}$.

We now define fixed-template *promise constraint satisfaction problems* (PCSPs). Let $\mathbf{A}$ and $\mathbf{B}$ be two r-ary structures with $\mathbf{A} \to \mathbf{B}$; we call $\mathbf{A}$ and $\mathbf{B}$ *templates*. In the *search* version of the PCSP with templates $\mathbf{A}$ and $\mathbf{B}$, denoted by $\text{PCSP}(\mathbf{A}, \mathbf{B})$, the task is: Given an r-ary structure $\mathbf{I}$ with the promise that $\mathbf{I} \to \mathbf{A}$, find a homomorphism from $\mathbf{I}$ to $\mathbf{B}$, which exists by the composition of promised homomorphism from $\mathbf{I}$ to $\mathbf{A}$ and the assumed homomorphism from $\mathbf{A}$ to $\mathbf{B}$. In the *decision* version of $\text{PCSP}(\mathbf{A}, \mathbf{B})$, the task is: Given an r-ary structure $\mathbf{I}$, output YES if $\mathbf{I} \to \mathbf{A}$, and output NO if $\mathbf{I} \not\to \mathbf{B}$. Observe that the decision version can be reduced to the search version: to solve the decision version, run an algorithm for the search version, then check if it gives a correct answer. However, it is not known whether the search version is polynomial-time reducible to the decision version in general. In any case, we will use $\text{PCSP}(\mathbf{A}, \mathbf{B})$ to mean the decision version when proving hardness, and the search version when showing algorithmic results.

LO colourings can be readily seen as PCSPs. First, observe that an r-uniform hypergraph can be seen as an r-ary structure. Second, define an r-ary structure $\mathbf{LO}_k^r$ with domain $[k]$, and whose relation contains a tuple $(c_1, \ldots, c_r)$ if and only if the sequence $c_1, \ldots, c_r$ has a unique maximum. Then, an LO k-colouring of an r-uniform hypergraph $\mathbf{H}$ is the same as a homomorphism from $\mathbf{H}$ (viewed as an r-ary structure) to $\mathbf{LO}_k^r$. Thus, the problem of finding an LO k-colouring of an r-uniform hypergraph that has an LO 2-colouring is the same as $\mathrm{PCSP}(\mathbf{LO}_2^r, \mathbf{LO}_k^r)$.

Results. In Section 3, we study the computational complexity of $\mathrm{PCSP}(\mathbf{LO}_2^3, \mathbf{LO}_{k(n)}^3)$, were $k(n)$ depends on the input size; here n denotes the number of vertices of the input (3-uniform) hypergraph. As in Example 2.1, this is obviously possible for $k(n) = n$. As our first contribution, we will present in Theorem 3.1 an efficient algorithm with $k(n) = O(\sqrt[3]{n \log \log n / \log n})$ colours.

In Section 5, we study the computational complexity of $\mathrm{PCSP}(\mathbf{LO}_\ell^r, \mathbf{LO}_k^r)$ for constant uniformity $r \geq 3$ and constant ℓ and k with $2 \leq \ell \leq k$. We establish NP-hardness of $\mathrm{PCSP}(\mathbf{LO}_\ell^r, \mathbf{LO}_k^r)$ for every constant $2 \leq \ell \leq k$ and constant uniformity $r \geq k - \ell + 4$, cf. Corollary 5.7. Our hardness results are based on the algebraic theory of minions [5], briefly introduced in Section 4. In fact, we establish relationships between polymorphism minions of $\mathrm{PCSP}(\mathbf{LO}_2^r, \mathbf{LO}_k^r)$ for all $r \geq 3$, cf. Theorem 5.4.

This is the full version of an ICALP 2022 paper [37], which showed weaker tractability and intractability results. Firstly, in [37] NP-hardness of only $\mathrm{PCSP}(\mathbf{LO}_k^r, \mathbf{LO}_{k+1}^r)$ was shown for $k \geq 2$ and $r \geq 5$. Secondly, in [37] we designed an algorithm that, for a given 3-uniform hypergraph that admits an LO 2-colouring, finds an LO k-colouring with $k = O(\sqrt{n \log \log n}/\log n)$.[1] Some of the techniques used to establish the tractability result in the current paper are different in character to those used for approximate graph colouring, which was the main technique used in the tractability result in [37]. In particular, the techniques in [37] (see also the full version [38]) built on [9, 39]. We note that we do not know how to apply the SDP-based methods from [30] directly as they seem specific to approximate graph colouring and less suited for LO colourings.[2]

3 ALGORITHMIC RESULTS

In this section we will prove the tractability results advertised above. In particular, we prove the following.

THEOREM 3.1. *There is a polynomial-time algorithm that, if given an LO 2-colourable 3-uniform hypergraph with n vertices, finds an LO $O(\sqrt[3]{n \log \log n / \log n})$-colouring.*

Our algorithm will use the notion of *linear hypergraphs*: we call a 3-uniform hypergraph *linear* if no two edges intersect in 2 or more vertices.[3]

PROPOSITION 3.2. *There is a polynomial-time algorithm that, if given an LO 2-colourable 3-uniform hypergraph H, constructs an LO 2-colourable linear 3-uniform hypergraph H' with no more vertices than H such that, if given an LO k-colouring of H', one can compute in polynomial time an LO k-colouring of H.*

PROOF. Suppose H is not linear already. Then it has edges (x, y, a) and (x, y, b). In all LO 2-colourings of H, a and b must be assigned the same colours. Therefore it is safe to merge a and b. Repeat this procedure until H is linear. To find an LO k-colouring of H given one of H', simply undo the merges. □

---

[1]The previous version of this paper [38] also contains a weaker result for 4-uniform hypergraphs.
[2]Interestingly, the approximability result for the independent set problem from [28], which we will use as a black-box in Section 3, does build on [30].
[3]This needs to hold even for a pair of non-distinct vertices. For instance, we forbid the edges $(1, 1, 2)$ and $(1, 1, 3)$ from existing simultaneously in the hypergraph.

In order to more elegantly express the algorithm we now propose, we will adapt the notion of "progress", introduced by Blum [11], to approximate LO colouring.

Definition 3.3. To make progress towards an LO $O(f(n))$-colouring of a 3-uniform hypergraph consists of one of the following:

Type 1 Finding an independent set of size $\Omega(n/f(n))$.
Type 2 Finding a set of $\Omega(n/f(n))$ vertices that intersects each edge in 0 or 2 vertices.

That this notion of progress is useful is shown in the following proposition.

PROPOSITION 3.4. *Suppose there is a polynomial-time algorithm that, if given a* linear *LO 2-colourable 3-uniform hypergraph, makes progress towards an LO $O(f(n))$-colouring, where $f(n) = \Theta(n^\alpha(\log n)^\beta(\log\log n)^\gamma)$ and $\alpha > 0$. Then, there is a polynomial-time algorithm that finds an LO $O(f(n))$-colouring of an LO 2-colourable 3-uniform hypergraph.*

PROOF. We will describe a recursive procedure that achieves our goal. Due to Proposition 3.2, we can assume that our input hypergraph is linear.

Now, at each recursive call, apply our progress-making algorithm. If we make type 1 progress, then colour the set of vertices with a *large colour*; if we make type 2 progress, then colour the set of vertices with a *small colour*. In any case, remove all edges incident to the coloured set.

First, note that this algorithm always produces a valid LO colouring. It suffices to note that for type 1 progress, any edge that was not coloured recursively intersects the maximum colour in one vertex; whereas for type 2 progress, any edge that was not coloured recursively intersects the minimum colour in two vertices. All such edges are thus coloured correctly.

To see why we use only $O(f(n))$ colours, consider the number of iterations needed to reduce the number of uncoloured vertices in the hypergraph by half — it is not difficult to see, for our choice of f, that it takes at most $O(n/(n/f(n))) = O(f(n))$ iterations. Applying the Master Theorem [19], noting that $f(n) = \Omega(n^{\alpha'})$ for some $\alpha' > 0$, it follows that there are $O(f(n))$ iterations overall. (We can take any α' such that $0 < \alpha' < \alpha$.) This bound on the number of iterations also immediately gives us polynomial run time.   □

With this framework in hand, we are ready to prove our result. We will essentially split into two cases: one for sparse hypergraphs, and one for dense hypergraphs.

PROPOSITION 3.5. *Fix $\Delta = \Theta(n^\alpha(\log n)^\beta(\log\log n)^\gamma)$, with $\alpha > 0$. There is a polynomial-time algorithm that makes progress towards an LO $O(\Delta\log\log n/\log n)$-colouring of an LO 2-colourable 3-uniform linear hypergraph if $|E| = O(n\Delta)$.*

Fix H to be our linear hypergraph. We will prove Proposition 3.5 with a series of lemmata.

LEMMA 3.6. *H has $\Omega(n)$ vertices with degree $O(\Delta)$.*

PROOF. Observe that the average degree of H is $O(\Delta)$. Applying Markov's inequality to a randomly chosen vertex immediately gives us our result.   □

Let V' be the set of vertices of H with degree $O(\Delta)$, and let E' be the set of edges induced by V'.

LEMMA 3.7. *An independent set of size $|E'|/\Delta$ exists within V'.*

PROOF. Each edge in E' must contain at least one vertex coloured with 2 in an LO 2-colouring; each such vertex is included in at most Δ edges. Thus there exists an independent set of size $|E'|/\Delta$.   □

We will use the following algorithm as a black box.

1:6

Tamio-Vesa Nakajima and Stanislav Živný

LEMMA 3.8 ([28]). *There is a polynomial-time algorithm that, if given a graph with an independent set of size s and average degree d, finds an independent set of size at least $\Omega(s \log d/d \log \log d)$.*

COROLLARY 3.9. *There is a polynomial-time algorithm that, if given a graph with an independent set of size s and average degree d, finds an independent set of size at least*

$$\max(\Omega(s \log d/d \log \log d), \Omega(n/d)).$$

PROOF. As established above for the more general case of hypergraphs, since the input graph has $\Theta(nd)$ edges, there must be $\Omega(n)$ vertices with degree $O(d)$. Thus, by greedily selecting an independent set on these vertices, we get the lower bound of $\Omega(n/d)$. Take the larger of this independent set and the one generated by the previous algorithm to get the desired result. □

PROOF OF PROPOSITION 3.5. Apply the algorithm from Corollary 3.9 to the primal graph of $H' = (V', E')$. (The primal graph of a hypergraph replaces each hyperedge with a clique.) This graph has an independent set of size $|E'|/\Delta$ and average degree $d = O(|E'|/|V'|) = O(|E'|/n)$. Therefore, the algorithm will find an independent set of size at least equal to the maximum of $\Omega((|E'|/\Delta) \log d/(|E'|/n) \log \log d) = \Omega(n \log d/\Delta \log \log n)$ and $\Omega(|V'|/d) = \Omega(n/d)$. If $d = o(\Delta \log \log n/\log n)$ then we get the independent set size we want from $\Omega(n/d)$. On the other hand, if $d = \Omega(\Delta \log \log n/\log n) = \Omega(n^{\alpha-\epsilon})$ for some $\epsilon < \alpha$, then we get the independent set size we want from $\Omega(n \log d/\Delta \log \log d)$ — the assumption on the size of d is necessary to make $\log d/\log \log d = \Omega(\log n/\log \log n)$. □

PROPOSITION 3.10. *There is a polynomial-time algorithm that makes progress towards an LO $O(\sqrt{n/\Delta})$-colouring of an LO 2-colourable 3-uniform linear hypergraph if $|E| = \Omega(n\Delta)$.*

Fix H to be a 3-uniform linear hypergraph that admits an LO 2-colouring. The following two lemmata prove Proposition 3.10.

LEMMA 3.11. *There exists an LO 2-colouring of H with $\Omega(\sqrt{n\Delta})$ vertices set to 1.*

PROOF. Consider any LO 2-colouring of H (one must exist by assumption). Consider the set S of vertices set to 1. Each edge intersects S in exactly two vertices; furthermore, every pair of vertices is included in at most one edge by linearity. Thus, by connecting these pairs of vertices, we find that we can construct a simple graph on S with $|E|$ edges. Thus $|S|^2 \geq |E| = \Omega(n\Delta)$, from which it follows that $|S| = \Omega(\sqrt{n\Delta})$. □

Definition 3.12. For any 3-uniform hypergraph H, let $\mathcal{E}(H)$ denote a set of linear equations mod 2. These equations have one variable x_v for each vertex v of H, and one equation for each edge of H. The equation for edge (a, b, c) is $x_a + x_b + x_c \equiv 0 \mod 2$.

LEMMA 3.13 ([33, THEOREM 2.12, (2)]). *The problem of finding a solution to a set of linear equations mod 2 that maximises the number of variables set to 1 can be approximated to within some constant factor in polynomial time.*

PROOF OF PROPOSITION 3.10. Consider $\mathcal{E}(H)$. Observe that a set of vertices that intersects each edge in an even number of vertices is equivalent to a solution to $\mathcal{E}(H)$. Furthermore, the size of the set is the same as the number of variables set to one in this solution; it is therefore sufficient to show that we can find a solution to this system of linear equations mod 2 that sets $\Omega(\sqrt{n\Delta})$ variables to one.

Now, consider the LO 2-colouring of H that sets $\Omega(\sqrt{n\Delta})$ variables to 1. By recolouring 2 into 0, this gives a solution to $\mathcal{E}(H)$ that sets at least $\Omega(\sqrt{n\Delta})$ variables to 1. Finally, using the algorithm from Lemma 3.13, we can find a solution that is at most a constant factor away from this one — i.e. with at least $\Omega(\sqrt{n\Delta})$ variables set to 1. This concludes the proof. □

ACM Trans. Comput. Theory, Vol. 1, No. 1, Article 1. Publication date: January 2022.

LEMMA 3.8 ([28]). *There is a polynomial-time algorithm that, if given a graph with an independent set of size s and average degree d, finds an independent set of size at least $\Omega(s \log d/d \log \log d)$.*

COROLLARY 3.9. *There is a polynomial-time algorithm that, if given a graph with an independent set of size s and average degree d, finds an independent set of size at least*

$$\max(\Omega(s \log d/d \log \log d), \Omega(n/d)).$$

PROOF. As established above for the more general case of hypergraphs, since the input graph has $\Theta(nd)$ edges, there must be $\Omega(n)$ vertices with degree $O(d)$. Thus, by greedily selecting an independent set on these vertices, we get the lower bound of $\Omega(n/d)$. Take the larger of this independent set and the one generated by the previous algorithm to get the desired result. □

PROOF OF PROPOSITION 3.5. Apply the algorithm from Corollary 3.9 to the primal graph of $H' = (V', E')$. (The primal graph of a hypergraph replaces each hyperedge with a clique.) This graph has an independent set of size $|E'|/\Delta$ and average degree $d = O(|E'|/|V'|) = O(|E'|/n)$. Therefore, the algorithm will find an independent set of size at least equal to the maximum of $\Omega((|E'|/\Delta) \log d/(|E'|/n) \log \log d) = \Omega(n \log d/\Delta \log \log n)$ and $\Omega(|V'|/d) = \Omega(n/d)$. If $d = o(\Delta \log \log n/\log n)$ then we get the independent set size we want from $\Omega(n/d)$. On the other hand, if $d = \Omega(\Delta \log \log n/\log n) = \Omega(n^{\alpha-\epsilon})$ for some $\epsilon < \alpha$, then we get the independent set size we want from $\Omega(n \log d/\Delta \log \log d)$ — the assumption on the size of d is necessary to make $\log d/\log \log d = \Omega(\log n/\log \log n)$. □

PROPOSITION 3.10. *There is a polynomial-time algorithm that makes progress towards an LO $O(\sqrt{n/\Delta})$-colouring of an LO 2-colourable 3-uniform linear hypergraph if $|E| = \Omega(n\Delta)$.*

Fix H to be a 3-uniform linear hypergraph that admits an LO 2-colouring. The following two lemmata prove Proposition 3.10.

LEMMA 3.11. *There exists an LO 2-colouring of H with $\Omega(\sqrt{n\Delta})$ vertices set to 1.*

PROOF. Consider any LO 2-colouring of H (one must exist by assumption). Consider the set S of vertices set to 1. Each edge intersects S in exactly two vertices; furthermore, every pair of vertices is included in at most one edge by linearity. Thus, by connecting these pairs of vertices, we find that we can construct a simple graph on S with $|E|$ edges. Thus $|S|^2 \geq |E| = \Omega(n\Delta)$, from which it follows that $|S| = \Omega(\sqrt{n\Delta})$. □

Definition 3.12. For any 3-uniform hypergraph H, let $\mathcal{E}(H)$ denote a set of linear equations mod 2. These equations have one variable x_v for each vertex v of H, and one equation for each edge of H. The equation for edge (a, b, c) is $x_a + x_b + x_c \equiv 0 \mod 2$.

LEMMA 3.13 ([33, THEOREM 2.12, (2)]). *The problem of finding a solution to a set of linear equations mod 2 that maximises the number of variables set to 1 can be approximated to within some constant factor in polynomial time.*

PROOF OF PROPOSITION 3.10. Consider $\mathcal{E}(H)$. Observe that a set of vertices that intersects each edge in an even number of vertices is equivalent to a solution to $\mathcal{E}(H)$. Furthermore, the size of the set is the same as the number of variables set to one in this solution; it is therefore sufficient to show that we can find a solution to this system of linear equations mod 2 that sets $\Omega(\sqrt{n\Delta})$ variables to one.

Now, consider the LO 2-colouring of H that sets $\Omega(\sqrt{n\Delta})$ variables to 1. By recolouring 2 into 0, this gives a solution to $\mathcal{E}(H)$ that sets at least $\Omega(\sqrt{n\Delta})$ variables to 1. Finally, using the algorithm from Lemma 3.13, we can find a solution that is at most a constant factor away from this one — i.e. with at least $\Omega(\sqrt{n\Delta})$ variables set to 1. This concludes the proof. □

PROOF OF THEOREM 3.1. Set $\Delta = n^{1/3}(\log n)^{2/3}(\log \log n)^{-2/3}$ so that $\sqrt{n/\Delta} = \Delta \log \log n/\log n$ and combine Proposition 3.5 and Proposition 3.10. Thus, we can always make progress towards an LO $O(\sqrt[3]{n \log \log n/\log n})$-colouring in polynomial time, if given an LO 2-colourable 3-uniform linear hypergraph. $\qquad\square$

4 ALGEBRAIC THEORY OF FIXED-TEMPLATE PROMISE CSPS

We recount the algebraic theory of fixed-template PCSPs developed in [5] and specialised to templates with a single relation (of arity r).

The *p-the power* of an r-ary template $\mathbf{A} = (A, R^{\mathbf{A}})$ is a template $\mathbf{A}^p = (A^p, R^{\mathbf{A}^p})$ where

$$R^{\mathbf{A}^p} = \{((a_1^1, \ldots, a_1^p), \ldots, (a_r^1, \ldots, a_r^p)) \mid (a_1^1, \ldots, a_r^1) \in R^{\mathbf{A}}, \ldots, (a_1^p, \ldots, a_r^p) \in R^{\mathbf{A}}\}.$$

In other words, a tuple of $R^{\mathbf{A}^p}$ contains r vectors of p elements of A, such that if these are written as a matrix with r rows and p columns, each column is a member of $R^{\mathbf{A}}$. For two r-ary templates $\mathbf{A}, \mathbf{B}$, a *p-ary polymorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a homomorphism $f : \mathbf{A}^p \to \mathbf{B}$. We let $\mathrm{ar}(f)$ denote the arity of a polymorphism.

Example 4.1. Consider the binary template $\mathbf{A} = ([2], R^{\mathbf{A}})$, where $R^{\mathbf{A}}$ is the binary disequality relation $\neq$ (restricted to $[2]^2$). The power $\mathbf{A}^5$ has domain $[2]^5$ and relation $\{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a}, \mathbf{b} \in [2]^5, a_i \neq b_i, i = 1, \ldots, 5\}$. This relation is constructed as follows: $(\mathbf{a}, \mathbf{b})$ belongs to the relation if and only if every column of a matrix with 5 columns and 2 rows constructed out of $\mathbf{a}, \mathbf{b}$ satisfies $\neq$. The matrix is the following one:

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \end{pmatrix}.$$

Thus, for each column to satisfy $\neq$, we must have $a_i \neq b_i$ for $i = 1, \ldots, 5$, as indicated above.

Now, consider a quinary polymorphism $f : \mathbf{A}^5 \to \mathbf{A}$. This is a function $f : [2]^5 \to [2]$ that satisfies the following property: if given a matrix with 2 rows and 5 columns, such that each column is a member of $R^{\mathbf{A}}$, then by applying f to the rows of this matrix we also get a member of $R^{\mathbf{A}}$. For instance, for the matrix

$$\begin{pmatrix} 1 & 2 & 2 & 1 & 1 \\ 2 & 1 & 1 & 2 & 2 \end{pmatrix},$$

we deduce that the pair $(f(1, 2, 2, 1, 1), f(2, 1, 1, 2, 2)) \in R^{\mathbf{A}}$ i.e. $f(1, 2, 2, 1, 1) \neq f(2, 1, 1, 2, 2)$. One such polymorphism is given by selecting the values of f from $[2]$ such that $f(x_1, \ldots, x_5) \equiv \sum_{i=1}^{5} x_i$ (mod 2).

The real power of this theory comes from *minions*.[4] A *minion* $\mathcal{M}$ is just a sequence of sets $\mathcal{M}^{(0)}, \mathcal{M}^{(1)}, \ldots$, equipped with (so-called minor) operations, one for each $\pi : [p] \to [q]$; given $f \in \mathcal{M}^{(p)}$ the minor operation yields $f_\pi \in \mathcal{M}^{(q)}$. The operation must satisfy the following conditions:

- For $f \in \mathcal{M}^{(p)}$, if $id : [p] \to [p]$ is the identity on $[p]$, then $f_{id} = f$.
- For $f \in \mathcal{M}^{(p)}$, $\pi : [p] \to [q]$ and $\sigma : [q] \to [t]$, we have $f_{\sigma \circ \pi} = (f_\pi)_\sigma$.

An important class of minions is class of *polymorphism minions*. The polymorphism minion $\mathcal{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ for two templates $\mathbf{A}, \mathbf{B}$ with the same arity is a minion where $\mathcal{M}^{(p)}$ is the set of p-ary polymorphisms from $\mathbf{A}$ to $\mathbf{B}$, and where, for $f : A^p \to B$ and $\pi : [p] \to [q]$, f_π is given by

---

[4]In category-theoretic terms, a minion is a functor from the skeleton of the category of finite sets to the category of sets. The objects of the first category are sets $[p]$ for $p \in \mathbb{N}$, and the arrows are functions between them. The functor equivalent to a minion $\mathcal{M}$ takes $[p]$ to $\mathcal{M}^{(p)}$, and $\pi : [p] \to [q]$ to $f \mapsto f_\pi$.

$f_\pi(x_1, \ldots, x_q) = f(x_{\pi(1)}, \ldots, x_{\pi(p)})$.[5] It is not difficult to check that, if $f : \mathbf{A}^p \to \mathbf{B}$ and $\pi : [p] \to [q]$, then $f_\pi : \mathbf{A}^q \to \mathbf{B}$, as required.

In order to be able to relate polymorphism minions with the complexity of PCSPs, we use *minion homomorphisms.*[6] A *minion homomorphism from $\mathcal{M}$ to $\mathcal{N}$* is a mapping ξ that takes each $\mathcal{M}^{(p)}$ to $\mathcal{N}^{(p)}$ and that satisfies the following condition: for any $\pi : [p] \to [q]$ and $f \in \mathcal{M}^{(p)}$, $\xi(f)_\pi = \xi(f_\pi)$. The following theorem links minion homomorphisms to PCSPs in the sense that minion homomorphisms capture precisely a certain type of polynomial-time reductions, know as primitive-positive constructions.[7]

Theorem 4.2 ([5, Theorems 3.1 and 4.12]). *For r-ary templates $\mathbf{A}, \mathbf{B}$ and r'-ary templates $\mathbf{A}', \mathbf{B}'$, a primitive-positive construction-based polynomial-time reduction from $\mathrm{PCSP}(\mathbf{A}', \mathbf{B}')$ to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ exists if and only if $\mathrm{Pol}(\mathbf{A}, \mathbf{B}) \to \mathrm{Pol}(\mathbf{A}', \mathbf{B}')$.*

In particular, a minion homomorphism between polymorphism minions implies a polynomial-time reduction (in the other direction). Unfortunately, it is usually a complex task to explicitly construct minion homomorphisms. An auxiliary construction called the *free structure* allows us to construct them more easily. For an arbitrary minion $\mathcal{M}$ and an r-ary template $\mathbf{A} = (A, R^\mathbf{A})$, the *free structure* $\mathbf{F} = \mathbf{F}_\mathcal{M}(\mathbf{A})$ of $\mathcal{M}$ generated by $\mathbf{A}$ is an r-ary template whose domain is $F = \mathcal{M}^{(|A|)}$. To construct its relation $R^\mathbf{F}$, first identify A with $[n]$ for $n = |A|$, and then enumerate the tuples of $R^\mathbf{A}$ as vectors $\mathbf{r}^1, \ldots, \mathbf{r}^k$, where $k = |R^\mathbf{A}|$. Construct functions $\pi_1, \ldots, \pi_r : [k] \to [n]$ where $\pi_i(j) = \mathbf{r}_i^j$. (If we were to arrange $\mathbf{r}^1, \ldots, \mathbf{r}^k$ as columns of a matrix with r rows and k columns, then $\pi_i(1), \ldots, \pi_i(k)$ is the i-th row of the matrix.) Now, the tuple $(f_1, \ldots, f_r)$, where $f_1, \ldots, f_r \in \mathcal{M}^{(n)}$, belongs to $R^\mathbf{F}$ if and only if for some $f \in \mathcal{M}^{(k)}$ we have $f_i = f_{\pi_i}$.

Example 4.3. Consider some polymorphism minion $\mathcal{M}$ and the ternary template $\mathbf{LO}_3^3$. We will construct $\mathbf{F} = \mathbf{F}_\mathcal{M}(\mathbf{LO}_3^3)$. The domain is $\mathcal{M}^{(3)}$. To construct the relation of $\mathbf{F}$, we first arrange the 15 tuples of $R^{\mathbf{LO}_3^3}$ into a matrix with 3 rows and 15 columns:

$$\begin{pmatrix} 2 & 1 & 1 & 3 & 1 & 1 & 3 & 2 & 3 & 1 & 2 & 1 & 3 & 2 & 2 \\ 1 & 2 & 1 & 1 & 3 & 1 & 2 & 3 & 1 & 3 & 1 & 2 & 2 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 & 3 & 1 & 1 & 2 & 2 & 3 & 3 & 2 & 2 & 3 \end{pmatrix}.$$

Row i of this matrix can be seen as a function $\pi_i : [15] \to [3]$. Now the relation $R^\mathbf{F}$ contains precisely the tuples $(f_{\pi_1}, f_{\pi_2}, f_{\pi_3})$ for all $f \in \mathcal{M}^{(15)}$. Substituting the definition for f_{π_i}, we find that these polymorphisms $f_{\pi_1}, f_{\pi_2}, f_{\pi_3}$ are:

$$(x, y, z) \mapsto f(y, x, x, z, x, x, z, y, z, x, y, x, z, y, y)$$
$$(x, y, z) \mapsto f(x, y, x, x, z, x, y, z, x, z, x, y, y, z, y)$$
$$(x, y, z) \mapsto f(x, x, y, x, x, z, x, x, y, y, z, z, y, y, z)$$

Observe that the matrix and the arguments of f are actually arranged in the same configuration, with 1 replaced by x, 2 by y and 3 by z.

The following theorem connects minion homomorphisms and free structures.

Theorem 4.4 ([5, Lemma 4.4]). *If $\mathcal{M}$ is a minion and $\mathbf{A}, \mathbf{B}$ are r-ary templates, the homomorphisms $h : \mathbf{F}_\mathcal{M}(\mathbf{A}) \to \mathbf{B}$ are in a (natural) 1-to-1 correspondence to the minion homomorphisms $\xi : \mathcal{M} \to \mathrm{Pol}(\mathbf{A}, \mathbf{B})$.*[8] *As a consequence, $\mathbf{F}_\mathcal{M}(\mathbf{A}) \to \mathbf{B}$ if and only if $\mathcal{M} \to \mathrm{Pol}(\mathbf{A}, \mathbf{B})$.*

---

[5]Put differently, minors in polymorphism minions permute variables, identify variables, and introduce dummy variables.
[6]In category-theoretic terms, a minion homomorphism is just a natural transformation.
[7]Primitive-positive constructions (or pp-constructions, for short) capture so-called "gadget reductions", cf. [7, Section 3].
[8]In category-theoretic terms, $\mathbf{F}_-(\mathbf{A})$ and $\mathrm{Pol}(\mathbf{A}, -)$ are functors between (in opposite directions) the category of r-ary templates and the category of minions, and $\mathbf{F}_-(\mathbf{A}) \dashv \mathrm{Pol}(\mathbf{A}, -)$.

5 HARDNESS RESULTS

In this section we will investigate the hardness of $\text{PCSP}(\textbf{LO}_\ell^r, \textbf{LO}_k^r)$.

First, we will establish that $\text{PCSP}(\textbf{LO}_2^r, \textbf{LO}_k^r)$ is NP-hard for each k, for *some* (large but constant) r (cf. Theorem 5.1). We then use minion homomorphisms to show that $\text{PCSP}(\textbf{LO}_\ell^r, \textbf{LO}_k^r)$ is NP-hard for $r \geq k - \ell + 4$ (cf. Corollary 5.7).

Second, we will show that $\text{PCSP}(\textbf{LO}_2^r, \textbf{LO}_k^r)$ with $r \geq k+2$ *cannot* be reduced to $\text{PCSP}(\textbf{LO}_2^r, \textbf{LO}_k^r)$ with $r < k+2$ using primitive-positive constructions (i.e. gadget reductions [5]). Thus in particular, it is not possible to prove hardness of $\text{PCSP}(\textbf{LO}_2^3, \textbf{LO}_3^3)$ and $\text{PCSP}(\textbf{LO}_2^4, \textbf{LO}_3^4)$ via gadget reductions from $\text{PCSP}(\textbf{LO}_2^5, \textbf{LO}_3^5)$. More generally, if $\text{PCSP}(\textbf{LO}_2^{r'}, \textbf{LO}_k^{r'})$ is proved NP-hard for *some* $r' \geq k+2$ then this will imply NP-hardness of $\text{PCSP}(\textbf{LO}_2^r, \textbf{LO}_k^r)$ for *every* $r \geq k+2$ but no gadget reductions would imply NP-hardness of $\text{PCSP}(\textbf{LO}_2^r, \textbf{LO}_k^r)$ for $2 \leq r < k+2$ (cf. Theorem 5.4).

5.1 Source of hardness

In this subsection we will show that, for each $k \geq 2$, $\text{PCSP}(\textbf{LO}_2^r, \textbf{LO}_k^r)$ is NP-hard for some large r.

THEOREM 5.1. *For any $k \geq 2$, there exists some large r such that $\text{PCSP}(\textbf{LO}_2^r, \textbf{LO}_k^r)$ is NP-hard.*

To do this, we will construct a hardness condition and apply it to this problem. The following hardness condition is almost identical to the one from [15] (later reformulated in [6]). It is somewhat more general, in that the arity of the polymorphisms are bounded — we do not need to prove something for polymorphisms of arbitrarily large arity.

For the purposes of this section, a *chain of minors* is a sequence of polymorphisms with minors between them: $f_0 \xrightarrow{\pi_{0,1}} \ldots \xrightarrow{\pi_{k-1,k}} f_k$. We let $\pi_{i,j}$ denote the composition of the minors between f_i and f_j i.e. $\pi_{i,j} = \pi_{j-1,j} \circ \ldots \circ \pi_{i,i+1}$. Thus $f_i \xrightarrow{\pi_{i,j}} f_j$.

THEOREM 5.2. *Suppose $\mathcal{M} = \text{Pol}(\textbf{A}, \textbf{B})$. Fix constants $\ell, k \in \mathbb{N}$. There exists a constant $m = m(\ell, k)$ such that the following holds. Suppose that for each polymorphism $f : \textbf{A}^n \to \textbf{B}$ where $n \leq m$ we assign a set $I(f) \subseteq [n]$ of size at most k, such that for each chain of minors $f_0 \xrightarrow{\pi_{0,1}} \ldots \xrightarrow{\pi_{\ell-1,\ell}} f_\ell$ containing polymorphisms of arity at most m, there exist i, j such that $\pi_{i,j}(I(f_i)) \cap I(f_j) \neq \emptyset$. Then $PMC_\mathcal{M}(m)$ is NP-hard, and furthermore $\text{PCSP}(\textbf{A}, \textbf{B})$ is NP-hard.*

We do not need the definition of the "promise satisfaction of a bipartite minor condition" problem from [5], denoted by $PMC_\mathcal{M}(m)$ in the statement, as we never use it; it is only included to match the result from [15].

PROOF. Identical to the proof in [15, Corollary 4.2], but noting that I (called sel in [15]) is only ever applied on polymorphisms with arity at most m, and that the selection of m depends only on ℓ and k. □

We define $\mathcal{M}_{\ell,k}^r = \text{Pol}(\textbf{LO}_\ell^r, \textbf{LO}_k^r)$. In order to apply this condition, we will have to better understand $\mathcal{M}_{\ell,k}^r$ from a combinatorial point of view — the following remark gives a useful way to see the polymorphisms in $\mathcal{M}_{\ell,k}^r$.

Remark 5.1. Observe that an (r')-ary polymorphism $f \in (\mathcal{M}_{2,k}^r)^{(r')}$ is a function from $[2]^{r'}$ to $[k]$; if it is applied to the rows of an $r \times r'$ matrix whose columns are tuples of the relation of $\textbf{LO}_2^r$ (i.e. they contain one 2 and otherwise are 1), then the resulting values contain a unique maximum. Similarly to Barto et. al. [4], we view f as a function from the powerset of $[r']$ to $[k]$ indicating the coordinates of 2s. (For example, the input tuple $(1, 2, 1, 2)$ is seen as equivalent to the input set $\{2, 4\}$.) Under this view, f is a polymorphism if and only if, for any partition $A_1, \ldots, A_r$ of $[r']$, the sequence $f(A_1), \ldots, f(A_r)$ has a unique maximum element. (Observe that each part A_i corresponds

to a row in the matrix mentioned earlier. Put differently, part A_i corresponds to columns in which the i-th row contains a (unique in its column) 2.)

LEMMA 5.3. *Suppose $f : 2^{[n]} \to [k]$ is a function such that, for each partition $A_1, \dots, A_p, p \in [n]$, of $[n]$, $f(A_1), \dots, f(A_p)$ has a unique maximum.*

Fix a partition $A_1, \dots, A_p$ of $[n]$. Suppose that the unique maximum of $f(A_1), \dots, f(A_p)$ is equal to the unique maximum of $f\{1\}, \dots, f\{n\}$. Then, if the unique maximum of the first is $f(A_i)$ and the unique maximum of the second is $f\{j\}$, it follows that $j \in A_i$.

PROOF. Suppose not. Thus suppose $j \notin A_i$. Consider the partition $A_i, \{1\}, \dots, \{j\}, \dots, \{n\}$, where all singletons included in A_i have been removed. This partition must have a unique maximum. Since $f(A_i) = f\{j\}$, it must be larger than $f\{j\}$ and $f(A_i)$. But this is impossible, since $f\{j\}$ is the maximum of $f\{1\}, \dots, f\{n\}$.   □

PROOF OF THEOREM 5.1. Fix $m = m(1, k)$ and $r = m + 2$. Consider any $f \in (\mathcal{M}^r_{2,k})^{(n)}$ for $n \leq m$. For any partition $A_1, \dots, A_p$ of $[n]$ where $p \in [n]$, observe that due to the partition $A_1, \dots, A_p, \emptyset, \dots, \emptyset$, where $\emptyset$ is added $r - p \geq m + 2 - m = 2$ times, it follows that $f(A_1), \dots, f(A_p)$ has a unique maximum. This means that we can apply Lemma 5.3 to any such f.

For any $f \in (\mathcal{M}^r_{2,k})^{(n)}$ for $n \leq m$ it follows that $f\{1\}, \dots, f\{n\}$ has a unique maximum. If it is given by $f\{i\}$, then set $I(f) = \{i\}$. We will now show that this selection I satisfies the condition of Theorem 5.2, and thus PCSP($\mathbf{LO}^r_2, \mathbf{LO}^r_k$) is NP-hard.

Consider any chain of minors $f_0 \xrightarrow{\pi_{0,1}} \dots \xrightarrow{\pi_{k-1,k}} f_k$. By the pigeonhole principle applied to $f_0(I(f_0)), \dots, f_k(I(f_k))$, for some $f = f_i \xrightarrow{\pi = \pi_{i,j}} f_j = g$ we have $f(I(f)) = g(I(g)) = c \in [k]$. Supposing that $p = \text{ar}(f), q = \text{ar}(g), p, q \leq m$, it follows that the unique maximum value in $f\{1\}, \dots, f\{p\}$ and in $g\{1\}, \dots, g\{q\}$ are both c. Suppose the first is given by $f\{i\}$ and the second is given by $g\{j\}$. Thus $I(f) = \{i\}$ and $I(g) = \{j\}$.

Observe that, by definition, $g\{x\} = f(\pi^{-1}(x))$. It follows that the unique maximum of

$$f(\pi^{-1}(1)), \dots, f(\pi^{-1}(q))$$

is given by $f(\pi^{-1}(j))$. Since additionally $f\{i\} = f(\pi^{-1}(j))$ and $f\{i\}$ is the unique maximum of $f\{1\}, \dots, f\{i\}$, by Lemma 5.3 it follows that $i \in \pi^{-1}(j)$ or equivalently $\pi(i) = j$. This implies that $\pi(I(f)) \cap I(g) = \{j\} \neq \emptyset$. Thus by Theorem 5.2 it follows that PCSP($\mathbf{LO}^r_2, \mathbf{LO}^r_k$) is NP-hard.   □

5.2   Minion homomorphisms

How can we now leverage this basic hardness result to other values of r? We will use chains of minion homomorphisms to do this. Our main result in this section is the following.

THEOREM 5.4. *For each $k \geq 3$, we have that $\mathcal{M}^{k+2}_{2,k} \leftrightarrows \mathcal{M}^{k+3}_{2,k} \leftrightarrows \dots$. Furthermore for each $2 \leq r < k + 2$, $\mathcal{M}^{k+2}_{2,k} \to \mathcal{M}^r_{2,k}$, yet $\mathcal{M}^r_{2,k} \nrightarrow \mathcal{M}^{k+2}_{2,k}$.*

Theorem 5.4 is illustrated in Figure 1. The more complicated Figure 2 illustrates minion relationships from Theorem 5.4 and those implied by Theorems 5.17, 5.19, and 5.20.

Remark 5.2. For $k = 3$, we know the precise relationship of all minions in Theorem 5.4: $\mathcal{M}^3_{2,3} \nrightarrow \mathcal{M}^4_{2,3}$ and $\mathcal{M}^4_{2,3} \nrightarrow \mathcal{M}^3_{2,3}$; i.e, the (in this case) two minions in the "left fan" in Figure 2 are incomparable. This follows from Theorem 5.19 and Theorem 5.20.

Combining Theorems 4.2, 5.1, and 5.4 gives the following.

COROLLARY 5.5. *PCSP($\mathbf{LO}^r_2, \mathbf{LO}^r_k$) is NP-hard for $r \geq k + 2$. Moreover, there is no polynomial-time reduction using pp-constructions from PCSP($\mathbf{LO}^{r'}_2, \mathbf{LO}^{r'}_k$) to PCSP($\mathbf{LO}^r_2, \mathbf{LO}^r_k$) for $r' \geq k + 2$ and $3 \leq r < k + 2$.*
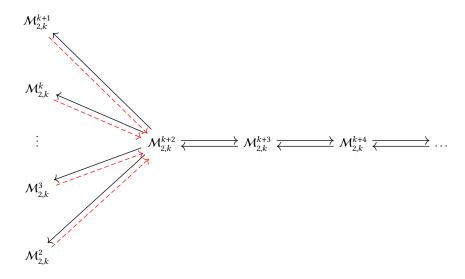
Fig. 1. Minions from Theorem 5.4. Solid black arrows indicate the existence of minion homomorphisms, whereas red dashed arrows indicate the non-existence of minion homomorphisms.

The next theorem will allow us to lift NP-hardness of PCSP($\mathbf{LO}_2^r, \mathbf{LO}_k^r$) to PCSP($\mathbf{LO}_{2+a}^r, \mathbf{LO}_{k+a}^r$) for every positive integer a.

THEOREM 5.6. *For every $r \geq 3$ and $2 \leq \ell < k$, $\mathcal{M}_{\ell+1,k+1}^r \to \mathcal{M}_{\ell,k}^r$.*

PROOF. Consider any p-ary polymorphism $f \in (\mathcal{M}_{\ell+1,k+1}^r)^{(p)}$. Consider the value of f for inputs $a_1, \ldots, a_p \in [\ell]$; due to the following matrix with $r \geq 3$ rows

$$\begin{pmatrix} a_1 + 1 & \ldots & a_p + 1 \\ a_1 & \ldots & a_p \\ \vdots & \ddots & \vdots \\ a_1 & \ldots & a_p \end{pmatrix},$$

we can deduce that $f(a_1, \ldots, a_p) < f(a_1 + 1, \ldots, a_p + 1) \in [k+1]$. This implies that $f(a_1, \ldots, a_p) \in [k]$. We claim this implies that f, restricted to $[\ell]^p$, is a polymorphism of $\mathcal{M}_{\ell,k}^r$. Consider matrix of inputs a_i^j where $i \in [p]$, $j \in [r]$, such that each column $a_i^1, \ldots, a_i^r$ is a tuple of $\mathbf{LO}_\ell^r$. Thus each column is also a tuple of $\mathbf{LO}_{\ell+1}^r$. Since f is a polymorphism of PCSP($\mathbf{LO}_{\ell+1}^r, \mathbf{LO}_{k+1}^r$), we deduce that

$$(f(a_1^1, \ldots, a_p^1), \ldots, f(a_1^r, \ldots, a_p^r))$$

is a tuple of $\mathbf{LO}_{k+1}^r$ i.e. has a unique maximum. But we already know these values belong to $[k]$. Since they have a unique maximum, they are a tuple of $\mathbf{LO}_k^r$. Thus f, restricted to $[k]^p$, is a polymorphism of $\mathcal{M}_{\ell,k}^r$.

We now claim that the map $f \mapsto f|_{[k]^p}$ taking a p-ary polymorphism to its restriction on $[k]^p$ is a minion homomorphism $\mathcal{M}_{\ell+1,k+1}^r \to \mathcal{M}_{\ell,k}^r$. To see why, consider any polymorphism $f \in (\mathcal{M}_{\ell+1,k+1}^r)^{(p)}$ and a function $\pi : [p] \to [q]$. What we need to prove is that

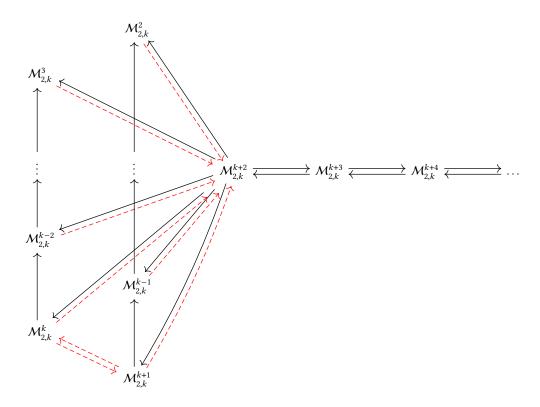$$(f_\pi)|_{[k]^p} = (f|_{[k]^p})_\pi.$$

Fig. 2. Minions from Theorems 5.4, 5.17, 5.19, and 5.20. Solid black arrows indicate the existence of minion homomorphisms, whereas red dashed arrows indicate the non-existence of minion homomorphisms. We have taken the case when k is odd; if k is even, then $\mathcal{M}_{2,k}^2$ and $\mathcal{M}_{2,k}^3$ are swapped.

But note that, for $x_1, \ldots, x_p \in [k]$,

$$((f_\pi)|_{[k]^p})(x_1, \ldots, x_p) = f_\pi(x_1, \ldots, x_p) = f(x_{\pi(1)}, \ldots, x_{\pi(p)})$$
$$= (f|_{[k]^p})(x_{\pi(1)}, \ldots, x_{\pi(p)}) = (f|_{[k]^p})_\pi(x_1, \ldots, x_p).$$

This concludes the proof.   □

Theorems 4.2, 5.6, 5.1, and 5.4 imply the following:

COROLLARY 5.7. $\mathrm{PCSP}(\mathbf{LO}_\ell^r, \mathbf{LO}_k^r)$ is NP-hard for $2 \le \ell \le k$ and $r \ge k - \ell + 4$.

The rest of this section is devoted to the proof of Theorem 5.4.

In order to construct minion homomorphisms, as the first milestone we exhibit a simple necessary and sufficient condition for the existence of a minion homomorphism to $\mathcal{M}_{2,k}^r$, and a sufficient condition for such a homomorphism to not exist.

LEMMA 5.8. *Fix $r \ge 2$ and $k \ge 3$. Consider any polymorphism minion $\mathcal{M}$. For any element $f \in \mathcal{M}^{(r)}$, let $f_1(x, y) = f(y, x, \ldots, x)$, $f_2(x, y) = f(x, y, x, \ldots, x)$, ..., $f_r(x, y) = f(x, \ldots, x, y)$. Now, $\mathcal{M} \to \mathcal{M}_{2,k}^r$ if and only if there exists some $\omega : \mathcal{M}^{(2)} \to [k]$ such that, for all $f \in \mathcal{M}^{(r)}$, there exists a unique maximum value among $\omega(f_1), \ldots, \omega(f_r)$.*

PROOF. We construct $\mathbf{F}_{\mathcal{M}}(\mathbf{LO}_2^r)$. The tuples of the relation of $\mathbf{LO}_2^r$ are all the r-dimensional vectors containing exactly one 2, with the other entries equal to 1. We can arrange these tuples into an r-by-r matrix where the diagonal contains 2 and all the other elements are 1. Replacing 1 with x and 2 with y, and applying f, we get the definitions of $f_1, \ldots, f_r$. Thus the relation of $\mathbf{F}_{\mathcal{M}}(\mathbf{LO}_2^r)$ contains precisely the tuples of the form $(f_1, \ldots, f_r)$ for $f \in \mathcal{M}^{(r)}$.

Thus our condition amounts to the existence of a homomorphism $\omega : \mathbf{F}_{\mathcal{M}}(\mathbf{LO}_2^r) \to \mathbf{LO}_k^r$. By Theorem 4.4, this is equivalent to $\mathcal{M} \to \mathcal{M}_{2,k}^r$.   □

COROLLARY 5.9. *Fix* $r \geq 2$, $r' \geq 2$ *and* $k \geq 3$. *If* $f\{1\}, \ldots, f\{r'\}$ *has a unique maximum for any function* $f : 2^{[r']} \to [k]$ *such that, for any partition* $A_1, \ldots, A_r$ *of* $[r']$, $f(A_1), \ldots, f(A_r)$ *has a unique maximum, then* $\mathcal{M}_{2,k}^r \to \mathcal{M}_{2,k}^{r'}$.

PROOF. Apply Lemma 5.8 using $\omega(f) = f(1, 2)$, and reinterpreting polymorphisms as indicated by Remark 5.1. Observe that the polymorphisms $f \in (\mathcal{M}_{2,k}^r)^{(r')}$ can then be seen as functions from $2^{[r']}$ to $[k]$ such that, for any partition $A_1, \ldots, A_r$ of $[r']$, a unique maximum exists among $f(A_1), \ldots, f(A_r)$. The precondition of our corollary then implies that $f\{1\}, \ldots, f\{r'\}$ have a unique maximum. By definition, $\omega(f_i) = f_i(1, 2) = f\{i\}$, where f_i is defined as in Lemma 5.8. Thus a unique maximum exists among $\omega(f_1), \ldots, \omega(f_r)$. It follows, by Lemma 5.8, that $\mathcal{M}_{2,k}^r \to \mathcal{M}_{2,k}^{r'}$.   □

As a second milestone, we introduce the notion of (k, p)-edge co-colouring and show in Corollary 5.14 and Lemma 5.15 that cliques have no co-colourings with certain parameters. For this, we will need three technical lemmata, namely Lemmata 5.11, 5.12, and 5.13.

Definition 5.10. A (k, p)-edge co-colouring of a graph G is an assignment of k colours to the edges of G such that any p disjoint edges of G are not assigned the same colour. Two edges are considered disjoint if the sets of their endpoints are disjoint.

The "co-" prefix is included since, in such a colouring, we colour *disjoint* edges with different colours, as opposed to incident edges as with normal edge colouring.

LEMMA 5.11. *A graph with* $k + 3$ *vertices and* $3k$ *edges has an independent set of size 2.*

PROOF. There are $(k + 3)(k + 2)/2$ possible independent sets of size 2. Each edge eliminates one of them. Since $(k + 3)(k + 2)/2 > 3k$ for $k \in \mathbb{R}$, it follows that at least one independent set of size 2 remains after adding in all the edges.   □

LEMMA 5.12. *A graph with* n *vertices where all pairs of edges intersect is either a cycle of length 3 or a star graph (ignoring vertices with no neighbours).*

PROOF. This is immediately true if there are no edges, so suppose at least one edge, say $\{x, y\}$, exists. All other edges must intersect $\{x, y\}$, so they must be of the form $\{x, a\}$ or $\{y, b\}$ for some $\{a, b\}$. If all the edges are of the form $\{x, a\}$ or $\{y, b\}$ respectively, then the graph is a star, as required. Otherwise, there exist two edges $\{x, a\}, \{y, b\}$. For these edges to intersect, it must be the case that $a = b$; thus we have found the cycle $x, y, a = b$ in our graph. To show that only this cycle exists within our graph, consider any edge $\{u, v\}$ in our graph. It must intersect $\{x, y\}$, so we can assume, without loss of generality, that $u = x$. It must also intersect $\{y, b\}$, so we deduce that $v \in \{y, b\}$. Thus our edge must already exist within the cycle.   □

LEMMA 5.13. *If* G *is a graph with* $k + 3$ *vertices with a* $(k, 2)$-*edge co-colouring, then* G *has an independent set of size 2.*

PROOF. We prove this fact inductively. The result is immediate for $k = 0$ (in which case the graph has no edges).

Suppose a colour c exists for which a vertex exists that is adjacent to all edges of colour c. Removing this vertex from the graph and applying the inductive hypothesis gives us the required independent set. Thus suppose that no vertex exists that covers all edges of a particular colour. In this case, by Lemma 5.12, it follows that, for each colour, the edges with that colour form a cycle of length 3. Thus our graph has $3k$ edges at most. It follows by Lemma 5.11 that there exists an independent set of size 2.   □

COROLLARY 5.14. *Thus, for any $k \in \mathbb{N}$, K_{k+3} has no $(k, 2)$-edge co-colouring.*

PROOF. This is the contrapositive of Lemma 5.13.   □

Remark 5.3. This corollary is tight, in the sense that K_{k+2} admits a $(k, 2)$-edge co-colouring, viz. colour edge $\{x, y\}$ with $\max\{x, y, 3\}$, assuming that the vertex-set of K_{k+2} is $[k + 2]$. This colouring will reappear implicitly in Theorem 5.19 and Theorem 5.20.

LEMMA 5.15. *For any $k \geq 3$, K_{10k} has no $(k, 3)$-edge co-colouring.*

PROOF. Suppose such a colouring exists, and let c be the most frequent colour. Since there are $10k(10k - 1)/2$ edges in total, and k colours, it follows that there are at least $5(10k - 1) \geq 45k$ edges of colour c. Select an edge e from these, removing all edges that share an endpoint with it (of which there are at most $20k$). There now remain at least $25k$ edges. Do the same thing again, selecting an edge e', and leaving at least $5k$ edges. Select e'' from among these edges. The edges e, e', e'' contradict the fact that K_{10k} is a $(k, 3)$-edge co-colouring.   □

As the third milestone, using the developed results so far, we will establish, firstly, minion homomorphisms in Theorems 5.16, 5.17, and 5.18, and, secondly, the lack of minion homomorphisms in Theorems 5.19 and 5.20.

THEOREM 5.16. *For $k \geq 3$, $r \geq 10k + 1$, $\mathcal{M}_{2,k}^r \to \mathcal{M}_{2,k}^{r+1}$.*

PROOF. Consider any function $f : 2^{[r+1]} \to [k]$ such that, for any partition $A_1, \ldots, A_r$ of $[r + 1]$, $f(A_1), \ldots, f(A_r)$ has a unique maximum. By Corollary 5.9, it is sufficient to show that $f\{1\}, \ldots, f\{r + 1\}$ has a unique maximum.

Consider all pairs $\{x, y\}$ such that the unique maximum of $f\{x, y\}, f\{1\}, \ldots, f\{r + 1\}$ is given by $f\{x, y\}$. Construct a graph G whose vertices are $[r + 1]$, and whose edges are given by all such pairs $\{x, y\}$. Observe that f is a $(k, 3)$-edge co-colouring of G. This is because, for any disjoint edges e, e', e'', due to the partition formed out of e, e', e'', the empty set twice and all remaining singletons, there must exist a unique maximum cost edge among e, e', e''.

Now suppose that no unique maximum exists among $f\{1\}, \ldots, f\{r+1\}$. Without loss of generality, let $f\{r + 1\} = f\{r\} = c$, and $f\{i\} \leq c$ for $i \in [r + 1]$. Now note that, for any distinct $x, y \in [r - 1]$, $f\{x, y\}$ must be the unique maximum of $f\{x, y\}, f\{1\}, \ldots, f\{r + 1\}$. Thus the subgraph of G given by restricting to $[r - 1]$ is a complete graph — in fact, as $r - 1 \geq 10k$, the subgraph G given by restricting to $[10k]$ is a complete graph. Due to f it is also $(k, 3)$-edge co-colourable. But this contradicts Lemma 5.15. Thus we deduce that $f\{1\}, \ldots, f\{r + 1\}$ has a unique maximum.   □

THEOREM 5.17. *For any $r \geq 2, k \geq 3$, $\mathcal{M}_{2,k}^{r+2} \to \mathcal{M}_{2,k}^r$.*

PROOF. Consider any function $f : 2^{[r]} \to [k]$ such that, for any partition $A_1, \ldots, A_{r+2}$ of $[r]$, $f(A_1), \ldots, f(A_{r+2})$ has a unique maximum. By Corollary 5.9, it is sufficient to show that $f\{1\}, \ldots, f\{r\}$ has a unique maximum. But consider the partition $\{1\}, \ldots, \{r\}, \emptyset, \emptyset$, and note that the largest value cannot be $f(\emptyset)$ (since $f(\emptyset)$ appears twice). Thus we deduce that one of $f\{1\}, \ldots, f\{r\}$ is the maximum, and furthermore that this maximum is strictly larger than all the other values in this sequence. Thus, $\mathcal{M}_{2,k}^{r+2} \to \mathcal{M}_{2,k}^r$.   □

THEOREM 5.18. *For $k \geq 3, r \geq k + 2$, $\mathcal{M}_{2,k}^r \rightarrow \mathcal{M}_{2,k}^{r+2}$.*

PROOF. Let $f : 2^{[r+2]} \rightarrow [k]$ be a function such that, for any partition $A_1, \ldots, A_r$ of $[r + 2]$, $f(A_1), \ldots, f(A_r)$ has a unique maximum. By Corollary 5.9, it is sufficient to show that $f\{1\}, \ldots, f\{r + 2\}$ has a unique maximum. Assume for contradiction that it does not. Then we have two cases depending on the maximum cost value among them.

Maximum is 1. In this case, construct a complete graph G on $[r + 2]$, and assign edge $\{x, y\}$ cost $f\{x, y\}$. Observe that this G is $(k, 2)$-edge co-colourable, since, for any disjoint edges $\{x, y\}, \{z, t\}$, due to partition $\{x, y\}, \{z, t\}, \{1\}, \ldots, \{r + 2\}$, one of the edges is assigned the unique maximum value by f, and thus the edges are assigned distinct costs. This contradicts Corollary 5.14, as $r + 2 \geq k + 3$, and thus a subgraph of G with $k + 3$ vertices is a $(k, 2)$-edge co-colourable copy of K_{k+3}.

Maximum is not 1. In this case, suppose the maximum is $c > 1$, and suppose that $f\{r + 1\} = f\{r + 2\} = c$. Construct a complete graph G on $[r]$, where edge $\{x, y\}$ is assigned cost $\max(f\{x, y\}, c)$. Observe that this G is $(k - 1, 2)$-edge co-colourable, since for any disjoint edges $\{x, y\}, \{z, t\}$, due to the partition $\{x, y\}, \{z, t\}, \{1\}, \ldots, \{r + 2\}$, these edges are assigned distinct costs, one of which is strictly greater than c (and thus these costs remain distinct after the operation $x \mapsto \max(x, c)$). This contradicts Corollary 5.14, as $r \geq k + 2$, and thus a subgraph of G with $k + 2$ vertices is a $(k - 1, 2)$-edge co-colourable copy of K_{k+2}.

Thus, since a contradiction is found in all cases, $\mathcal{M}_{2,k} \rightarrow \mathcal{M}_{2,k}^{r+2}$. $\square$

THEOREM 5.19. *For $k \geq 3$, $\mathcal{M}_{2,k}^k \nrightarrow \mathcal{M}_{2,k}^{k+1}$.*

PROOF. Let $v(x) = \max(2, x - 1)$. Let $f : 2^{[k+1]} \rightarrow [k]$ be a function that maps the empty set and singletons to 1, sets of size k or $k + 1$ to k, and any other set S to $\max_{x \in S} v(x)$. Observe that $f_1 = \ldots = f_{k+1}$ (all of them map $(1, 1)$ and $(1, 2)$ to 1, and $(2, 1)$ and $(2, 2)$ to k). Thus if f is a polymorphism, by Lemma 5.8, and using the interpretation of Remark 5.1, our conclusion follows.

To see why f is a polymorphism consider any partition $A_1, \ldots, A_k$ of $[k + 1]$. If the partition has one part of size k or $k + 1$, then it has only one such part, and the other parts are either singletons or empty sets — thus in this case $f(A_1), \ldots, f(A_k)$ contains one k and $k - 1$ ones. Otherwise, observe that the partition must contain at least one part of size 2 or higher. The cost of this part will be the maximum element in the part minus 1, or 2, whichever is higher. All other parts of size 1 or 0 are assigned 1. The only way in which this partition could lack a unique maximum element is if there are two parts of size 2 or above, both of whose maximum values through v are at most 2. This is impossible by the pigeonhole principle, since there are only 3 elements mapped by v to 1 or 2. $\square$

THEOREM 5.20. *For $k \geq 3$, $\mathcal{M}_{2,k}^{k+1} \nrightarrow \mathcal{M}_{2,k}^k$.*

PROOF. Let $v(x) = \max(3, x)$. Let $f : 2^{[k]} \rightarrow [k]$ be a function that maps singletons to 1, the empty set to 2, sets of size k or $k + 1$ to k, and any other set S to $\max_{x \in S} v(x)$. Observe that $f_1 = \ldots = f_k$ (all of them map $(1, 1)$ to 2, $(1, 2)$ to 1, and $(2, 1)$ and $(2, 2)$ to k). Thus if f is a polymorphism, by Lemma 5.8, and using the interpretation of Remark 5.1, our conclusion follows.

To see why f is a polymorphism consider any partition $A_1, \ldots, A_{k+1}$ of $[k]$. If the partition has one part of size k or $k + 1$, then it has only one such part, and the other parts are either singletons or empty sets — thus in this case $f(A_1), \ldots, f(A_{k+1})$ contains one k and k ones. Otherwise, observe that the partition must either contain at least one part of size 2 or higher, or it must contain the empty set and all singletons. In the latter case, the empty set supplies the unique maximum. In the former case, the cost of a non-singleton part will be the maximum element in the part or 3, whichever is higher. All parts of size 1 or 0 are assigned 1 or 2. The only way in which this partition

could lack a unique maximum element is if there are two parts of size 2 or above, both of whose maximum values through v are at most 3. This is impossible, by the pigeonhole principle, since there are at most 3 elements mapped by v to 1, 2 or 3. □

Finally, we will now combine Theorems 5.16–5.20 to establish Theorem 5.4.

PROOF OF THEOREM 5.4. Fix $k \geq 3$. Consider any $r \geq k + 2$. Letting r' be any number greater or equal to $10k$ with the same parity as r, note that, by applying Theorem 5.18, Theorem 5.16 and Theorem 5.17, we have

$$\mathcal{M}_{2,k}^r \to \mathcal{M}_{2,k}^{r+2} \to \dots \to \mathcal{M}_{2,k}^{r'-2} \to \mathcal{M}_{2,k}^{r'} \to \mathcal{M}_{2,k}^{r'+1} \to \mathcal{M}_{2,k}^{r'-1} \to \dots \to \mathcal{M}_{2,k}^{r+3} \to \mathcal{M}_{2,k}^{r+1}.$$

Thus, for $r \geq k + 2$, $\mathcal{M}_{2,k}^r \to \mathcal{M}_{2,k}^{r+1}$. Furthermore, by applying Theorem 5.17,

$$\mathcal{M}_{2,k}^r \to \mathcal{M}_{2,k}^{r-2} \to \dots,$$

and

$$\mathcal{M}_{2,k}^r \to \mathcal{M}_{2,k}^{r+1} \to \mathcal{M}_{2,k}^{r-1} \to \mathcal{M}_{2,k}^{r-3} \dots.$$

Thus we deduce that, for any r' such that $2 \leq r' \leq r$, $\mathcal{M}_{2,k}^r \to \mathcal{M}_{2,k}^{r'}$. This supplies all the minion homomorphisms required by Theorem 5.4.

Now, consider any $2 \leq r < k + 2$. Note that by Theorem 5.19 and Theorem 5.20, we know that $\mathcal{M}_{2,k}^{k+1} \not\to \mathcal{M}_{2,k}^k$ and $\mathcal{M}_{2,k}^k \not\to \mathcal{M}_{2,k}^{k+1}$. Note that $\mathcal{M}_{2,k}^{k+2} \to \mathcal{M}_{2,k}^k$ and $\mathcal{M}_{2,k}^{k+2} \to \mathcal{M}_{2,k}^{k+1}$. Thus by contrapositive, $\mathcal{M}_{2,k}^k \not\to \mathcal{M}_{2,k}^{k+2}$ and $\mathcal{M}_{2,k}^{k+1} \not\to \mathcal{M}_{2,k}^{k+2}$. Now, depending on the parity of r, by Theorem 5.17, $\mathcal{M}_{2,k}^k \to \mathcal{M}_{2,k}^r$ or $\mathcal{M}_{2,k}^{k+1} \to \mathcal{M}_{2,k}^r$. Thus in either case, by contrapositive, $\mathcal{M}_{2,k}^r \not\to \mathcal{M}_{2,k}^{k+1}$. This supplies all the homomorphism nonexistence proofs required by Theorem 5.4. □

6 CONCLUSIONS

The question about the complexity of $\mathrm{PCSP}(\mathbf{LO}_2^3, \mathbf{LO}_k^3)$ for constant $k \geq 3$ raised in [4] stays open. More generally, the complexity of $\mathrm{PCSP}(\mathbf{LO}_\ell^r, \mathbf{LO}_k^r)$ is open except for the hardness results obtained in this paper: We established NP-hardness for every constant $2 \leq k \leq \ell$ and every constant uniformity $r \geq \ell - k + 4$.

Minion homomorphisms (and lack thereof) between the polymorphism minions $\mathcal{M}_{2,k}^r$ for various values of r have interesting implications for the complexity of PCSPs more broadly, beyond our hardness results. In particular, if one were to prove NP-hardness of LO 2- vs. k-colourings on r-uniform hypergraphs for $r < k + 2$, then our results imply that this NP-hardness does *not* follow from NP-hardness of the same problem with uniformity at least $k + 2$ via minion homomorphisms and thus in particular cannot be obtained from the latter problem via "gadget reductions" [5]. This is in contrast to the case of (non-promise) CSPs, where it is known [8] (cf. also [7])[9] that all NP-hardness can be shown using minion homomorphisms.[10]

Going beyond the realm of fixed-template PCSPs [5] (which limits the number of colours by a constant), what is the smallest function $k(n)$ for which $\mathrm{PCSP}(\mathbf{LO}_2^3, \mathbf{LO}_{k(n)}^3)$ is solvable efficiently? There is no clear reason to believe that positive result from the present paper with $k(n) = O(\sqrt[3]{n \log \log n / \log n})$ is optimal.

---

[9][7] uses the terminology of height 1 identities.
[10]This would not be the first occurrence of this phenomenon in the context of PCSPs; a recent example of the same phenomenon comes from [41] for the problem of approximate graph colouring.

ACKNOWLEDGMENTS

REFERENCES

[1] Per Austrin, Amey Bhangale, and Aditya Potukuchi. Simplified inpproximability of hypergraph coloring via t-agreeing families, 2019.

[2] Per Austrin, Amey Bhangale, and Aditya Potukuchi. Improved inapproximability of rainbow coloring. In *Proc. 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 1479–1495, 2020.

[3] Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\epsilon)$-Sat is NP-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017.

[4] Libor Barto, Diego Battistelli, and Kevin M. Berg. Symmetric Promise Constraint Satisfaction Problems: Beyond the Boolean Case. In *Proc. 38th International Symposium on Theoretical Aspects of Computer Science (STACS'21)*, volume 187 of *LIPIcs*, pages 10:1–10:16, 2021.

[5] Libor Barto, Jakub Bulín, Andrei A. Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. *J. ACM*, 68(4):28:1–28:66, 2021.

[6] Libor Barto and Marcin Kozik. Combinatorial Gap Theorem and Reductions between Promise CSPs. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, pages 1204–1220, 2022.

[7] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and how to use them. In Andrei Krokhin and Stanislav Živný, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017.

[8] Libor Barto, Jakub Opršal, and Michael Pinsker. The wonderland of reflections. *Isr. J. Math*, 223(1):363–398, Feb 2018.

[9] Bonnie Berger and John Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 5(3):459–466, 1990.

[10] Amey Bhangale. NP-Hardness of Coloring 2-Colorable Hypergraph with Poly-Logarithmically Many Colors. In *Proc. 45th International Colloquium on Automata, Languages, and Programming (ICALP'18)*, volume 107 of *LIPIcs*, pages 15:1–15:11, 2018.

[11] Avrim Blum. New approximation algorithms for graph coloring. *J. ACM*, 41(3):470–516, 1994.

[12] Joshua Brakensiek and Venkatesan Guruswami. New hardness results for graph and hypergraph colorings. In *Proc. 31st Conference on Computational Complexity (CCC'16)*, volume 50 of *LIPIcs*, pages 14:1–14:27, 2016.

[13] Joshua Brakensiek and Venkatesan Guruswami. Promise Constraint Satisfaction: Algebraic Structure and a Symmetric Boolean Dichotomy. *SIAM J. Comput.*, 50(6):1663–1700, 2021.

[14] Joshua Brakensiek and Venkatesan Guruswami. The quest for strong inapproximability results with perfect completeness. *ACM Trans. Algorithms*, 17(3):27:1–27:35, 2021.

[15] Alex Brandts, Marcin Wrochna, and Stanislav Živný. The complexity of promise SAT on non-Boolean Domains. *ACM Trans. Comput. Theory*, 13(4), 2021.

[16] Panagiotis Cheilaris, Balázs Keszegh, and Dömötör Pálvölgyi. Unique-maximum and conflict-free coloring for hypergraphs and tree graphs. *SIAM J. Discret. Math.*, 27(4):1775–1787, 2013.

[17] Panagiotis Cheilaris and Géza Tóth. Graph unique-maximum and conflict-free colorings. *J. Discrete Algorithms*, 9(3):241–251, 2011.

[18] Eden Chlamtac and Gyanit Singh. Improved approximation guarantees through higher levels of SDP hierarchies. In *Proc. 11th International Workshiop on Approximation, Randomization and Combinatorial Optimization (APPROX'08)*, volume 5171 of *Lecture Notes in Computer Science*, pages 49–62. Springer, 2008.

[19] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.

[20] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional Hardness for Approximate Coloring. *SIAM J. Comput.*, 39(3):843–873, 2009.

[21] Irit Dinur, Oded Regev, and Clifford Smyth. The hardness of 3-uniform hypergraph coloring. *Comb.*, 25(5):519–535, September 2005.

[22] Irit Dinur and Igor Shinkar. On the Conditional Hardness of Coloring a 4-Colorable Graph with Super-Constant Number of Colors. In *Proc. 13th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'10)*, volume 6302 of *LNCS*, pages 138–151. Springer, 2010.

[23] M. R. Garey and D. S. Johnson. The complexity of near-optimal graph coloring. *J. ACM*, 23(1):43–49, 1976.

[24] Venkatesan Guruswami and Sanjeev Khanna. On the hardness of 4-coloring a 3-colorable graph. *SIAM J. Discret. Math*, 18(1):30–40, 2004.

[25] Venkatesan Guruswami and Euiwoong Lee. Strong inapproximability results on balanced rainbow-colorable hypergraphs. *Comb.*, 38(3):547–599, 2018.

[26] Venkatesan Guruswami and Sai Sandeep. d-To-1 Hardness of Coloring 3-Colorable Graphs with O(1) Colors. In *Proc. 47th International Colloquium on Automata, Languages, and Programming (ICALP'20)*, volume 168 of *LIPIcs*, pages 62:1–62:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

[27] Venkatesan Guruswami and Sai Sandeep. Rainbow coloring hardness via low sensitivity polymorphisms. *SIAM J. Discret. Math*, 34(1):520–537, 2020.

[28] Magnús M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Apppl.*, 4(1):1–16, 2000.

[29] Sangxia Huang. Improved hardness of approximating chromatic number. In *Proc. 16th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'13)*, pages 233–243. Springer, 2013.

[30] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998.

[31] Ken-ichi Kawarabayashi and Mikkel Thorup. Coloring 3-colorable graphs with less than $n^{1/5}$ colors. *J. ACM*, 64(1):4:1–4:23, 2017.

[32] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Comb.*, 20(3):393–415, Mar 2000.

[33] Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000.

[34] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC'02)*, pages 767–775. ACM, 2002.

[35] Michael Krivelevich, Ram Nathaniel, and Benny Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *J. Algorithms*, 41(1):99–113, 2001.

[36] Michael Krivelevich and Benny Sudakov. Approximate coloring of uniform hypergraphs. *Journal of Algorithms*, 49(1):2–12, 2003.

[37] Tamio-Vesa Nakajima and Stanislav Živný. Linearly Ordered Colourings of Hypergraphs. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *LIPIcs*, pages 128:1–128:18, 2022.

[38] Tamio-Vesa Nakajima and Stanislav Živný. Linearly ordered colourings of hypergraphs. Technical report, 2022. Version 1, April 2022.

[39] Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *J. ACM*, 30(4):729–735, 1983.

[40] Marcin Wrochna. A note on hardness of promise hypergraph colouring. Technical report, 2022.

[41] Marcin Wrochna and Stanislav Živný. Improved hardness for *H*-colourings of *G*-colourable graphs. In *Proc. 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 1426–1435, 2020.