A logarithmic approximation of linearly-ordered colourings

Johan Håstad
KTH Royal Institute of Technology, Stockholm, Sweden

Björn Martinsson
KTH Royal Institute of Technology, Stockholm, Sweden

Tamio-Vesa Nakajima
Department of Computer Science, University of Oxford, UK

Stanislav Živný
Department of Computer Science, University of Oxford, UK

Abstract

A linearly ordered (LO) k-colouring of a hypergraph assigns to each vertex a colour from the set $\{0, 1, \ldots, k-1\}$ in such a way that each hyperedge has a unique maximum element. Barto, Batistelli, and Berg conjectured that it is NP-hard to find an LO k-colouring of an LO 2-colourable 3-uniform hypergraph for any constant $k \geq 2$ [STACS'21] but even the case $k = 3$ is still open. Nakajima and Živný gave polynomial-time algorithms for finding, given an LO 2-colourable 3-uniform hypergraph, an LO colouring with $O^*(\sqrt{n})$ colours [ICALP'22] and an LO colouring with $O^*(\sqrt[3]{n})$ colours [ACM ToCT'23]. Very recently, Louis, Newman, and Ray gave an SDP-based algorithm with $O^*(\sqrt[5]{n})$ colours. We present two simple polynomial-time algorithms that find an LO colouring with $O(\log_2(n))$ colours, which is an exponential improvement.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Constraint and logic programming

Keywords and phrases Linear ordered colouring, Hypergraph, Approximation, Promise Constraint Satisfaction Problems

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2024.7

Category APPROX

Related Version *Full version*: `https://arxiv.org/abs/2404.19556`

Funding This work was supported by UKRI EP/X024431/1, by a Clarendon Fund Scholarship and by the Knut and Alice Wallenberg Foundation. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper.

Acknowledgements This paper is a merger of independent work by Håstad and Martinsson, and by Nakajima and Živný respectively. We are grateful to Venkat Guruswami for noting and informing the authors of the fact that we independently had found the same algorithm.

1 Introduction

Given a graph G, the *graph k-colouring* problem asks to find a colouring of the vertices of G by colours from the set $\{0, 1, \ldots, k-1\}$ in such a way that no edge is monochromatic. The *approximate graph colouring problem* asks, given a k-colourable graph G, to find an ℓ-colouring of G, where $\ell \geq k$. For $k = 3$, the state-of-the-art results are NP-hardness of the case $\ell = 5$ [2] and a polynomial-time algorithm for finding a colouring with $\ell = O(n^{0.19747})$ colours, where n is the number of vertices of the input graph G [9]. For non-monochromatic colourings of hypergraphs, it is known that finding an ℓ-colouring of a k-colourable r-uniform hypergraph is NP-hard for any constant $\ell \geq k \geq 2$ and $r \geq 3$ [7], and also some positive results are known for colourings with super-constantly many colours, e.g. [11, 10, 6].

A new variant of hypergraph colourings was identified in [1]. Given a 3-uniform hypergraph H, a colouring of the vertices of H with colours from the set $\{0, 1, \ldots, k-1\}$ is called a *linearly ordered* (LO) k-colouring if every edge e of H satisfies the following: if two vertices of e have the same colour then the third colour is larger. More generally, a colouring of a hypergraph H is an LO colouring if every edge of H has a unique maximum colour. (Note that the two definitions coincide for 3-uniform hypergraphs.) Barto et al. conjectured that finding an LO ℓ-colouring of a 3-uniform hypergraph that admits an LO k-colouring is NP-hard for every constant $\ell \geq k \geq 2$ [1] but even the case $k = 2$ and $\ell = 3$ is open. Nakajima and Živný established NP-hardness for some regimes of the parameters k, ℓ, r [13, 14] and, very recently, Filakovký et al. [8] showed NP-hardness of the case $k = 3$, $\ell = 4$, $r = 3$. More importantly for this paper, Nakajima and Živný also considered finding an LO $f(n)$-colouring of an LO 2-colourable 3-uniform hypergraph with n vertices and presented polynomial-time algorithms with $f(n) = O(\sqrt{n \log\log n / \log n})$ [13] and $f(n) = O(\sqrt[3]{n \log\log n / \log n})$ [14]. Very recently, Louis, Newman, and Ray [12] have given a polynomial-time SDP-based algorithm with $f(n) = O^*(\sqrt[5]{n})$ colours.

As our main result, we improve their results by an exponential factor.

▶ **Theorem 1.** *There is an algorithm which, if given a 3-uniform hypergraph H with $n \geq 4$ vertices and m edges that admits an LO 2-colouring, finds an LO $\log_2(n)$-colouring of H in time $O(n^3 + nm)$.*

In fact we present two different algorithms that return colourings using $O(\log n)$ colours. Both are based on solving the natural system of linear equations implied by the existence of an LO 2-colouring. In one case, the system is solved modulo 2, and in the other case, the system is solved over the rationals.

While the H which we are given as input is 3-uniform, we will need the notion that follows in greater generality; hence we define it for general hypergraphs. For each edge $\{x_1, \ldots, x_r\}$ of H, we write an equation $v_{x_1} + \cdots + v_{x_r} = 1$ where we initially use equality modulo 2 but as stated above we later use the same system over the rational numbers. Let A be this set of equations, written as a matrix with m rows and n columns. (Note that A is the *incidence matrix* of H.) Thus v is a solution if and only if $Av = 1^m$. Clearly a valid LO 2-colouring gives one solution but in the general case, the system has a large dimensional affine space as its set of solutions and the desired solution is hard to find.

2 Algorithm based on equations modulo 2

In this section all linear equations are taken modulo 2. For the following, given a set S of positive integers, an S-uniform hypergraph is a hypergraph where all edges have sizes taken from S. We first prove the following subprocedure of the main algorithm.

▶ **Lemma 2.** *There is an algorithm which, if given a $\{2, 3\}$-uniform hypergraph H with n vertices and m edges that admits an LO 2-colouring and such that the implied linear system of equations $Av = 1^m$ does not fix the value of any variable, outputs a subset T of vertices that intersects edges of size three in zero or two vertices and edges of size two in exactly one vertex. Moreover, we have $|T| \geq n/2$. The algorithm runs in $O(n^3 + nm)$ time.*

Proof. We first describe a randomised version of our algorithm, and then derandomise it. The set of solutions to $Av = 1^m$ is an affine space and hence a generic solution can be written as $v = v^0 + \sum_{i=1}^{r} a_i v^i$ for a basic solution v^0, linearly independent solutions to the homogeneous system v^i, and field elements (in this case bits) a_i. The fact that no variable is fixed implies that for each vertex x there is some positive i such that $v^i_x = 1$.

For the randomised algorithm choose $a_1, \ldots, a_r$ to be independent identically distributed uniformly random bits, and set T to be the set of variables x, such that $v_x = 0$. Clearly T satisfies the conclusion of the lemma as in each edge we have an odd number of ones. Since for every vertex x there exists positive i such that v^i_x, due to the influence of $a_i v^i_x$ we see that x is included in T with probability $1/2$. Thus, on average T contains half the vertices.

Now, we derandomise this algorithm using the method of conditional expectations. Go through the variables a_i in increasing order and fix its value once and for all. Fixing the value of a_i determines the value of some v_x while other values remain undetermined. For each value being determined $v^i_x = 1$ and hence one value of a_i gives the final value 0 and the other gives final value 1. Set a_i such that at least half the determined values are 0. After we have fixed all a_i this way, we have a final solution with at least $n/2$ zeroes.

The bottleneck of the running time of this algorithm is solving the linear system of equations. This can be done in the advertised running time since every equation has $O(1)$ entries. ◀

Proof of Theorem 1. As a preliminary step, we eliminate any variable determined by the system $Av = 1^m$. Note that if the colour of a vertex is determined by the system $Av = 1^m$, then this vertex must have that same colour in *all* LO 2-colourings. Fix these variables once and for all and eliminate them from the equation system. For all vertices that have been given the colour 1, we set the colours of the two other vertices in all of its edges to be 0. This process of identifying fixed variables and eliminating them is then repeated until the system $Av = 1^m$ contains no variables fixed to a constant. At any fixed point of this process, for every edge, either all vertices in that edge are fixed (and the edge has a unique maximum as required), or exactly one vertex in it is fixed to 0.

Now, remove all coloured vertices from the hypergraph H, shrinking the edges they belonged to. The remaining hypergraph will no longer be 3-uniform, but importantly it will still be LO 2-colourable. Our goal is still to LO colour the remaining hypergraph, since any edge partially coloured by the preliminary step above must have had exactly one vertex v fixed to 0; and hence, if we LO colour the edge that resulted from removing v, this leads to an LO colouring of the original hypergraph when v is assigned 0.

Consider the following algorithm, where i starts at 0.

1. If the hypergraph H has at most, say, 20 vertices, find an LO-colouring of H by brute force using colours i and $i + 1$. (It exists since H is LO 2-colourable.)
2. Otherwise, find the subset T guaranteed by Lemma 2.
3. Colour the vertices in T by colour i. Remove the vertices in T from H. Remove all edges that intersect T from H. Increment i by 1.
4. Repeat.

Note that $|T| \geq n/2$ and thus within $-4 + \log_2 n$ repetitions we reach the first case. Each step adds one colour and we get two additional colours from the final brute-force colouring for a total of at most $\log_2 n$ colours. The output is correct as the first time some vertex in an edge is coloured, for edges with three vertices exactly one more vertex in the same edge is coloured at the same time, and for edges with two vertices only that vertex is coloured at that time. The remaining vertex is given a higher colour and hence the edge is correctly coloured. For the time complexity, we again note that it is dominated by the time needed to solve the linear system of equations. ◀

Note that the number 20 selected above can be increased to any number that is $O(\log n)$ and the algorithm remains polynomial time (since we must compute the colouring for a subgraph of this size by brute force). If we stop the algorithm at B vertices, then we save $\log B + \Theta(1)$ colours, since this is how many colours the algorithm would have used to colour the last B vertices. By setting $B = \Theta(\log n)$, we can thus save $\Theta(1) + \log\log n$ colours while keeping run time of the algorithm polynomial in n.

A slight variant can be obtained by instead counting the number of remaining edges with no coloured vertex. Once we have no more such vertices, we colour all remaining vertices with the next colour. For such edge, a random solution v gives the four sets of values $(0,0,1)$, $(0,1,0)$, $(1,0,0)$ and $(1,1,1)$ with equal probabilities. Thus the number of edges decreases, on average, by a factor 4 for each iteration. (Note that all the edges of size 2 are solved in the first iteration, so there is no need to count them.) It is easy to achieve this deterministically by conditional expectations. We state the conclusion as a theorem.

▶ **Theorem 3.** *There is an algorithm which, if given a 3-uniform hypergraph H with n vertices and $m \geq 1$ edges that admits an LO 2-colouring, finds an LO $(2 + \frac{1}{2}\log_2(m))$-colouring of H in time $O(n^3 + nm)$.*

▶ Remark 4. Our algorithm has some similarity with algorithms for *temporal* CSPs [3]. Note that an LO ω-colouring (which means an LO-colouring, but with no restriction on the number of colours) is a temporal CSP; to solve it, one finds a subset that could be the smallest colour (by solving mod-2 equations as above), sets that colour, then continues recursively. The difference is that for an LO ω-colouring one does not care about the number of colours, so one can find any nonempty set of vertices to set the lowest colour to, whereas in our problem we are trying to find a large set of this kind. We note that the algorithm of [14] also uses this approach when setting "small colours".

▶ Remark 5. We remark that the subprocedure of our algorithm computes the exclusive or of two vectors of bits. Thus the algorithm runs very fast in practice — on most architectures hundreds of operations of this kind are done at one time by (i) packing the bits within a larger word and (ii) using SIMD instructions.

3 Algorithm using $\mathbb{Q}$

In this section we present a more complicated algorithm which uses more colours. This might seem pointless, and indeed it might be. On the other hand the ideas used are slightly different and hence there might be situations where the ideas of this section can turn out to be useful. It is also curious to see that we can use the same system of linear equations, now over the rationals, in a rather different way. The algorithm here is in fact essentially saying that we can always use the unbalanced case of [12]. As this eliminates many complications and in particular makes it possible to completely avoid any semi-definite programming, we state all facts needed in the current section rather than refer to the very similar statements in [12]. As already stated, all arithmetic in this section is over the rational numbers. In this situation, no variables can be determined as we can set v^0 to have all coordinates equal to 1/3.

We study the homogeneous system $Av = 0$ and by the assumption of LO 2-colourability it has a solution, w, with coordinates either $-\frac{1}{3}$ or $\frac{2}{3}$. Let us first show how solutions over the rational numbers can be used to find LO-colourings. This is the same lemma used in the unbalanced case of [12].

▶ **Lemma 6.** *Suppose we have a solution, u, to the homogeneous system where M is the maximal value of the absolute value of a coordinate and $m > 0$ is the minimal absolute value. Then we can LO-colour with $2 + \log_2(M/m)$ colours.*

Proof. For notational convenience let us instead require that the minimal colour in each edge should be unique. We can simply reverse the order of the colours at the end. By scaling we can assume $M = 1$. We use even colours for positive coordinates and we give x the colour 2ℓ if v_x is at most $2^{-(2\ell-1)}$ and strictly larger than $2^{-(2\ell+1)}$. For negative coordinates we use $2\ell + 1$ as the colour if v_x is between $-2^{-2\ell}$ (inclusive) and $-2^{-(2\ell+2)}$ (non-inclusive). Let us verify that this gives a correct colouring.

Take an edge (x, y, z) and suppose both x and y get the same colour 2ℓ. Then by the linear equation of the edge $v_z < -2^{-2\ell}$ and thus z has a colour below 2ℓ. The case of two vertices of odd colour is similar and as the bound on the number of colours is immediate, the lemma follows. ◀

Let us choose the vectors $(v^i)_{i=1}^r$ giving the solutions to the homogeneous system to be of unit length and orthogonal. Define $u = \sum_{i=1}^r y_i v^i$ where y_i are independent normal variables with mean 0 and standard deviation 1. The length of u is very close to $\sqrt{r}$ but to be crude we use that $E[\|u\|^2] = r$ and hence with probability $\frac{3}{4}$ the length of u is at most $2\sqrt{r}$.

Let $c^j = (v_j^1, v_j^2, \ldots v_j^r)$ be the vector of dimension r given by the jth coordinates of each vector v^i. Using this notation we see that the jth coordinate of u is a normal variable with standard deviation $\|c^j\|$. Recall that w is our assumed solution to $Av = 0$ with all coordinates either $-\frac{1}{3}$ or $\frac{2}{3}$. We can write $w = \sum_{i=1}^r a_i v^i$ for some numbers a_i, and by orthonormality $\|a\| = \|w\|$ which is at most $\frac{2}{3}\sqrt{n}$. As

$$|w_j| = |(c^j, a)| \leq \|c^j\|\|a\|,$$

using that $|w_j|$ is at least $\frac{1}{3}$ we conclude that

$$\|c^j\| \geq 1/(2\sqrt{n}).$$

The probability that a normal variable with standard deviation σ is of absolute value at most δ is at most $2\delta/(\sqrt{2\pi}\sigma)$. We conclude that for a suitable constant d the probability that $|u_j|$ is below $dn^{-3/2}$ is at most $1/(2n)$. Thus with probability at least $1/2$ the absolute value of any coordinate of u is at least $dn^{-3/2}$. Thus with probability at least $\frac{1}{4}$, we can apply Lemma 6 with $M = 2n^{1/2}$ and $m = dn^{-3/2}$ and we conclude.

▶ **Theorem 7.** *Using the system of linear equations over the rational numbers we can find, with probability at least $\frac{1}{4}$ and in polynomial time, an LO-colouring with $O(1) + 2\log_2 n$ colours.*

This algorithm is less efficient compared to the algorithm of the previous section. The main computational cost is still solving a linear system but this is more complicated over the rational numbers as coefficients are likely to grow. Our bound for the number of colours is also worse. Heuristically one could hope to have the ratio of the smallest and largest coordinate of u to be $\Theta(n)$ but not better. Thus it is possible that we could eliminate the multiplicative constant 2 in the theorem but to get substantially fewer than $\log n$ colours by this method sounds unlikely.

As a final observation in this section let us note that defining a colouring by the sign of the vector u we get a standard (non-monochromatic) 2-colouring of the hypergraph. This gives an alternative algorithm to that of [5, 4].

4 Concluding remarks

Our algorithms indicate that LO 2-colouring is quite different from many other colouring problems. The key property that we use in our algorithm is that the constraint implies a linear constraint. The analysis of the algorithms also heavily relies on the fact that we study 3-uniform hypergraphs.

It is tempting to think that the proposed methods would extend to other constraint satisfaction problems where we are guaranteed that a solution must satisfy a linear constraint. We have so far been unable to find an interesting such example.

References

1 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. `arXiv:2010.04623`, `doi:10.4230/LIPIcs.STACS.2021.10`.

2 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. `arXiv:1811.00970`, `doi:10.1145/3457606`.

3 Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57(2):9:1–9:41, 2010. `doi:10.1145/1667053.1667058`.

4 Joshua Brakensiek and Venkatesan Guruswami. An algorithmic blend of LPs and ring equations for promise CSPs. In *Proc. 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*, pages 436–455, 2019. `arXiv:1807.05194`, `doi:10.1137/1.9781611975482.28`.

5 Joshua Brakensiek and Venkatesan Guruswami. Promise Constraint Satisfaction: Algebraic Structure and a Symmetric Boolean Dichotomy. *SIAM J. Comput.*, 50(6):1663–1700, 2021. `arXiv:1704.01937`, `doi:10.1137/19M128212X`.

6 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. `doi:10.1007/978-3-540-85363-3\_5`.

7 Irit Dinur, Oded Regev, and Clifford Smyth. The hardness of 3-uniform hypergraph coloring. *Comb.*, 25(5):519–535, September 2005. `doi:10.1007/s00493-005-0032-4`.

8 Marek Filakovský, Tamio-Vesa Nakajima, Jakub Opršal, Gianluca Tasinato, and Uli Wagner. Hardness of Linearly Ordered 4-Colouring of 3-Colourable 3-Uniform Hypergraphs. In *Proc. 41st International Symposium on Theoretical Aspects of Computer Science (STACS'24)*, volume 289 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:19, 2024. `arXiv:2312.12981`, `doi:10.4230/LIPIcs.STACS.2024.34`.

9 Ken-ichi Kawarabayashi, Mikkel Thorup, and Hirotaka Yoneda. Better coloring of 3-Colorable graphs. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 331–339, New York, NY, USA, 2024. Association for Computing Machinery. `doi:10.1145/3618260.3649768`.

10 Michael Krivelevich, Ram Nathaniel, and Benny Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *J. Algorithms*, 41(1):99–113, 2001. `doi:10.1006/jagm.2001.1173`.

11 Michael Krivelevich and Benny Sudakov. Approximate coloring of uniform hypergraphs. *J. Algorithms*, 49(1):2–12, 2003. `doi:10.1016/S0196-6774(03)00077-4`.

12 Anand Louis, Alantha Newman, and Arka Ray. Improved linearly ordered colorings of hypergraphs via SDP rounding, 2024. `arXiv:2405.00427`, `doi:10.48550/arXiv.2405.00427`.

13 Tamio-Vesa Nakajima and Stanislav Živný. Linearly Ordered Colourings of Hypergraphs. In *Proc. 49th International Colloquium on Automata, Languages, and Programming (ICALP'22)*, volume 229 of *LIPIcs*, pages 128:1–128:18, 2022. `doi:10.4230/LIPIcs.ICALP.2022.128`.

14 Tamio-Vesa Nakajima and Stanislav Živný. Linearly Ordered Colourings of Hypergraphs. *ACM Trans. Comput. Theory*, 13(3–4), 2023. arXiv:2204.05628, doi:10.1145/3570909.