


# Faster Edge Coloring by Partition Sieving

Shyan Akmal   

INSAIT, Sofia University “St. Kliment Ohridski”, Bulgaria

Tomohiro Koana  

Utrecht University, The Netherlands

Research Institute for Mathematical Sciences, Kyoto University, Japan

---

## Abstract

---

In the EDGE COLORING problem, we are given an undirected graph  $G$  with  $n$  vertices and  $m$  edges, and are tasked with finding the smallest positive integer  $k$  so that the edges of  $G$  can be assigned  $k$  colors in such a way that no two edges incident to the same vertex are assigned the same color. EDGE COLORING is a classic NP-hard problem, and so significant research has gone into designing fast *exponential-time* algorithms for solving EDGE COLORING and its variants exactly. Prior work showed that EDGE COLORING can be solved in  $2^m \text{poly}(n)$  time and polynomial space, and in graphs with average degree  $d$  in  $2^{(1-\varepsilon_d)m} \text{poly}(n)$  time and exponential space, where  $\varepsilon_d = (1/d)^{\Theta(d^3)}$ .

We present an algorithm that solves EDGE COLORING in  $2^{m-3m/5} \text{poly}(n)$  time and polynomial space. Our result is the first algorithm for this problem which simultaneously runs in faster than  $2^m \text{poly}(m)$  time and uses only polynomial space. In graphs of average degree  $d$ , our algorithm runs in  $2^{(1-6/(5d))m} \text{poly}(n)$  time, which has far better dependence in  $d$  than previous results. We also consider a generalization of EDGE COLORING called LIST EDGE COLORING, where each edge  $e$  in the input graph comes with a list  $L_e \subseteq \{1, \dots, k\}$  of colors, and we must determine whether we can assign each edge a color from its list so that no two edges incident to the same vertex receive the same color. We show that this problem can be solved in  $2^{(1-6/(5k))m} \text{poly}(n)$  time and polynomial space. The previous best algorithm for LIST EDGE COLORING took  $2^m \text{poly}(n)$  time and space.

Our algorithms are algebraic, and work by constructing a special polynomial  $P$  based off the input graph that contains a multilinear monomial (i.e., a monomial where every variable has degree at most one) if and only if the answer to the LIST EDGE COLORING problem on the input graph is YES. We then solve the problem by detecting multilinear monomials in  $P$ . Previous work also employed such monomial detection techniques to solve EDGE COLORING. We obtain faster algorithms both by carefully constructing our polynomial  $P$ , and by improving the runtimes for certain structured monomial detection problems using a technique we call partition sieving.

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

**Keywords and phrases** Coloring, Edge coloring, Chromatic index, Matroid, Pfaffian, Algebraic algorithm

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2025.7

**Related Version** *Full Version:* <https://arxiv.org/abs/2501.05570>

**Funding** *Shyan Akmal:* Partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

*Tomohiro Koana:* Supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (project CRACKNP under grant agreement No. 853234). Also supported by JSPS KAKENHI Grant Numbers JP20H05967.

**Acknowledgements** We thank the anonymous reviewers for helpful feedback on this work.

## 1 Introduction

Coloring graphs is a rich area of research in graph theory and computer science. Given a graph  $G$ , a *proper vertex coloring* of  $G$  is an assignment of colors to its nodes such that no two adjacent vertices receive the same color. In the VERTEX COLORING problem, we are given an undirected graph  $G$  on  $n$  vertices, and are tasked with computing the smallest positive



© Shyan Akmal and Tomohiro Koana;

licensed under Creative Commons License CC-BY 4.0

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).

Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thăng;

Article No. 7; pp. 7:1–7:18



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



integer  $k$  such that  $G$  admits a proper vertex coloring using  $k$  colors. VERTEX COLORING is a classic combinatorial problem, NP-hard to solve even approximately [27]. An influential line of research has worked on designing faster and faster exponential-time algorithms for this problem [24, 17, 13, 8], culminating in an algorithm solving VERTEX COLORING in  $O^*(2^n)$  time and space [11], where throughout we write  $O^*(f(n))$  as shorthand for  $f(n) \text{ poly}(n)$ .

In this paper, we study exact algorithms for the closely related EDGE COLORING problem. Given a graph  $G$ , a *proper edge coloring* of  $G$  is an assignment of colors to its edges such that no two edges incident to a common vertex receive the same color. The smallest positive integer  $k$  such that  $G$  admits a proper edge coloring using  $k$  colors is the *chromatic index* of  $G$ , denoted by  $\chi'(G)$ . In the EDGE COLORING problem, we are given an undirected graph  $G$  on  $n$  vertices and  $m$  edges, and are tasked with computing  $\chi'(G)$ .

Let  $\Delta$  denote the maximum degree of the input graph  $G$ . For each vertex  $v$  in  $G$ , a proper edge coloring of  $G$  must assign distinct colors to the edges incident to  $v$ . Thus  $\chi'(G) \geq \Delta$ . A classic result in graph theory known as Vizing's theorem proves that in fact  $\chi'(G) \leq \Delta + 1$ , so that the simple  $\Delta$  lower bound is close to the truth. Moreover, a proper edge coloring of  $G$  using  $(\Delta + 1)$  colors can be found in near-linear time [2]. Consequently, solving EDGE COLORING amounts to distinguishing between the cases  $\chi'(G) = \Delta$  and  $\chi'(G) = \Delta + 1$ . Despite this powerful structural result, the EDGE COLORING problem is NP-hard just like VERTEX COLORING, even for graphs where all vertices have degree exactly  $\Delta = 3$  [21].

In recent years, researchers have made major strides in our knowledge of algorithms for *approximate* variants of EDGE COLORING in the distributed [19, 5, 3] and dynamic settings [15, 14, 6]. In comparison, much less is known about the exponential-time complexity for solving the EDGE COLORING problem *exactly*.

A simple way to solve EDGE COLORING is by reduction to VERTEX COLORING. Given the input graph  $G$ , we can construct in polynomial time the *line graph*  $L(G)$  whose nodes are edges of  $G$ , and whose nodes are adjacent if the corresponding edges in  $G$  are incident to a common vertex. By construction, the solution to VERTEX COLORING on  $L(G)$  gives the solution to EDGE COLORING on  $G$ . If  $G$  has  $m$  edges and maximum degree  $\Delta$ , then  $L(G)$  has  $m$  vertices and maximum degree  $(2\Delta - 1)$ . Using the aforementioned  $O^*(2^n)$  time and space algorithm for VERTEX COLORING, this immediately implies a  $O^*(2^m)$  time and space algorithm for EDGE COLORING. In graphs with maximum degree  $\Delta$ , VERTEX COLORING can be solved in  $O^*(2^{(1-\varepsilon)n})$  time and space, where  $\varepsilon = (1/2)^{\Theta(\Delta)}$  [9]. By applying the above reduction, EDGE COLORING can similarly be solved in  $O^*(2^{(1-\varepsilon)m})$  time and space, for  $\varepsilon = (1/2)^{\Theta(\Delta)}$ .

Line graphs are much more structured objects than generic undirected graphs (for example, there are small patterns that line graphs cannot contain as induced subgraphs [4, Theorem 3]), so one might hope to solve EDGE COLORING faster without relying on a black-box reduction to VERTEX COLORING. Nonetheless, there is only one result from previous work which solves EDGE COLORING on general undirected graphs without using this reduction. Specifically, [10, Section 5.6] shows that EDGE COLORING can be solved in  $O^*(2^m)$  time and polynomial space. In comparison, it remains open whether VERTEX COLORING can be solved in  $O^*(2^n)$  time using only polynomial space.

In summary, algorithms from previous work can solve EDGE COLORING in

- $O^*(2^m)$  time using polynomial space, or in
- faster than  $O^*(2^m)$  time in bounded degree graphs, using exponential space.

Given this state of affairs, it is natural to ask:

*Can EDGE COLORING be solved in faster than  $O^*(2^m)$  time, using only polynomial space?*

We answer this question affirmatively by proving the following result.

► **Theorem 1.1.** *There is a randomized algorithm which solves EDGE COLORING with high probability and one-sided error in  $O^*(2^{m-3n/5})$  time and polynomial space.*

If the input graph has average degree  $d$ , then  $m = dn/2$ , so Theorem 1.1 equivalently states that EDGE COLORING can be solved in  $O^*(2^{(1-\varepsilon)m})$  time for  $\varepsilon = \frac{6}{5d}$ . In comparison, the current fastest algorithm for VERTEX COLORING on graphs with average degree  $d$  takes  $O^*(2^{(1-\delta)n})$  time, where  $\delta = (1/d)^{\Theta(d^3)}$  [18, Lemma 4.4 and Theorem 5.4]. Prior to our work, the fastest algorithm for EDGE COLORING in the special case of regular graphs (i.e., graphs where all vertices have the same degree) took  $O^*(2^{m-n/2})$  time [10, Theorem 6]. The regularity assumption simplifies the EDGE COLORING problem significantly, as in this case the problem reduces to finding a collection of  $(\Delta - 1)$  mutually disjoint perfect matchings, which together consist of only  $(m - n/2)$  edges. Theorem 1.1 improves upon this runtime with an algorithm that succeeds on *all* undirected graphs, not just regular graphs.

In the case of regular graphs, we obtain improvements beyond Theorem 1.1 when the graphs have high degree. We say a graph is  $d$ -regular if all its nodes have degree  $d$ . Given a positive integer  $k$ , let

$$H_k = \sum_{j=1}^k \frac{1}{j} \tag{1}$$

denote the  $k^{\text{th}}$  Harmonic number. We prove the following result.

► **Theorem 1.2.** *For each integer  $d \geq 6$ , there is a randomized algorithm that solves EDGE COLORING with high probability and one-sided error on  $d$ -regular graphs in  $O^*(2^{m-\alpha_d n})$  time and polynomial space, for  $\alpha_d = 1 - H_{d+1}/(d+1)$ .*

For example for  $d = 6$ , Theorem 1.2 shows that we can solve EDGE COLORING in 6-regular graphs in  $O^*(2^{m-0.62n})$  time, slightly faster than the runtime presented in Theorem 1.1. The savings in the exponent are better for larger  $d$ , approaching a runtime of  $O^*(2^{m-n})$  as the degree  $d$  grows since we have  $\alpha_d = 1 - \Theta((\log d)/d)$ .

We also consider a generalization of EDGE COLORING called LIST EDGE COLORING. In this problem, we are given an undirected graph  $G$  with  $m$  edges as before, together with a positive integer  $k$  and a list of possible colors  $L_e \subseteq \{1, \dots, k\}$  for each edge  $e$ . We are tasked with determining whether  $G$  admits a proper edge coloring, which assigns each edge  $e$  a color from the list  $L_e$ . If we set  $k = \Delta$  to be the maximum degree of  $G$  and set  $L_e = \{1, \dots, k\}$  for each edge  $e$ , we recover the standard EDGE COLORING problem. Using more sophisticated arguments, we generalize Theorem 1.1 to the LIST EDGE COLORING problem.

► **Theorem 1.3.** *There is a randomized algorithm that solves LIST EDGE COLORING with high probability and one-sided error in  $O^*(2^{m-3n/5})$  time and polynomial space.*

Prior to our work, no algorithm was known for LIST EDGE COLORING which simultaneously ran in faster than  $O^*(2^m)$  time and used polynomial space.

## 1.1 Our Techniques

Our algorithms for EDGE COLORING and LIST EDGE COLORING are algebraic, and involve reducing these problems to certain *monomial detection* tasks, by now a common paradigm in graph algorithms. To achieve the  $O^*(2^{m-3n/5})$  runtime and polynomial space bounds for these problems, we combine and improve techniques in polynomial sieving, matroid constructions, and enumeration using matrices.

## 7:4 Faster Edge Coloring by Partition Sieving

The main technical ingredient in our work builds off the *determinantal sieving* technique introduced in [16]. Given a homogeneous multivariate polynomial  $P(X)$  of degree  $k$  and a linear matroid  $\mathcal{M}$  on  $X$  of rank  $k$ , this technique allows one to test in  $O^*(2^k)$  time whether  $P(X)$  contains a multilinear monomial (i.e., a monomial where each variable has degree at most one) that forms a basis in  $\mathcal{M}$ . We recall the formal definition of matroids in Section 2. For the purpose of this overview, a matroid  $\mathcal{M}$  is a family of subsets of variables of  $P$  satisfying certain properties, and a basis in  $\mathcal{M}$  is a set in this family whose size equals the rank  $k$ . In our applications, we construct  $P$  to enumerate certain structures in the input graph. Finding a multilinear monomial in  $P$  corresponds to identifying a particularly nice structure in the graph (e.g., a solution to the EDGE COLORING problem). Determinantal sieving is useful because one can pick  $\mathcal{M}$  in such a way that identifying a monomial in  $P$  that forms a basis in  $\mathcal{M}$  recovers extra information about the relevant structures in the graph.

We build off determinantal sieving and introduce the *partition sieving* method. This technique involves a *partition matroid*  $\mathcal{D}$ , which is a simple type of matroid defined using an underlying partition of the variable set of  $P$ . Given a partition matroid  $\mathcal{D}$  whose partition has  $p$  parts, and a polynomial  $P$  *compatible* with  $\mathcal{D}$  (“compatible” is a technical condition defined in Section 3 – intuitively, it requires that monomials in  $P$  have degrees respecting the partition defining  $\mathcal{D}$ ), we can test whether  $P$  contains a multilinear monomial forming a basis in  $\mathcal{D}$  in  $O^*(2^{k-p})$  time and polynomial space. This offers an exponential speed-up over determinantal sieving. To apply partition sieving, we need to carefully design both the polynomial  $P$  and the partition matroid  $\mathcal{D}$  to be compatible in our technical framework.

We design the polynomial  $P$  using the notion of the Pfaffian of a matrix. If  $A$  is a symbolic skew-symmetric adjacency matrix of  $G$ , then its Pfaffian  $\text{Pf } A$  is a polynomial that enumerates the perfect matchings in  $G$ . The EDGE COLORING problem may be rephrased as asking whether the edge set of  $G$  can be partitioned into a collection of  $k$  matchings. One can design a polynomial that enumerates  $k$  matchings by computing a product of  $k$  Pfaffians of adjacency matrices. However, this simple construction does not meet the technical conditions we need to employ partition sieving. Instead we utilize the Ishikawa-Wakayama formula [22], a convolution identity for Pfaffians and determinants, to design a polynomial that enumerates matchings which satisfy certain degree constraints. Specifically, the formula lets us construct a polynomial  $P$  whose monomials correspond to  $k$ -tuples of matchings in  $G$ , where each vertex in  $G$  appears as an endpoint in exactly  $\deg_G(v)$  of the matchings in the tuple.

We design the partition matroid  $\mathcal{D}$  in terms of a *dominating set* of the input graph. A subset of vertices  $D$  is a dominating set in  $G$  if every vertex in  $V \setminus D$  is adjacent to a node in  $D$ . Given a dominating set  $D$  in  $G$ , we show that it is always possible to construct a partition matroid  $\mathcal{D}$  with  $p = |V \setminus D|$  parts, compatible with our enumerating polynomial  $P$  (in our proofs,  $\mathcal{D}$  is actually only compatible with a polynomial obtained from  $P$  by the inclusion-exclusion principle, but we ignore that distinction in this overview). We achieve compatibility using the degree condition imposed on  $P$ , mentioned at the end of the previous paragraph. Partition sieving then lets us detect a multilinear monomial in  $P$ , and thus solve EDGE COLORING, in  $O^*(2^{m-|V \setminus D|})$  time.

► **Lemma 1.4.** *There is a randomized algorithm that, given a graph  $G$  on  $n$  vertices and  $m$  edges and a dominating set  $D$  of  $G$ , solves EDGE COLORING on  $G$  with high probability and one-sided error in  $O^*(2^{m-n+|D|})$  time and polynomial space.*

In other words, by finding smaller dominating sets in  $G$ , we can solve EDGE COLORING faster. In polynomial time, it is easy to construct a dominating set of size at most  $n/2$  in any connected graph  $G$  with  $n$  vertices. Consequently, the partition sieving framework lets us solve EDGE COLORING in  $O^*(2^{m-n/2})$  time. We obtain faster algorithms using the fact

that if a graph with  $n \geq 8$  nodes has minimum degree two, then it contains a dominating set of size at most  $2n/5$  [29]. We make this result effective, showing how to *find* such a dominating set in  $O^*(2^{m-3n/5})$  time and polynomial space. When solving EDGE COLORING, it turns out one can assume without loss of generality that the graph has minimum degree two, and so this framework yields a  $O^*(2^{m-3n/5})$  time algorithm for the problem. The situation is far more complex for LIST EDGE COLORING, where unit-degree vertices cannot be removed freely. For this problem, we construct a more complicated polynomial  $P$ , by enforcing additional matroid conditions in the Ishikawa-Wakayama formula. Intuitively, we identify a subgraph  $G_{\text{new}}$  of minimum degree two in  $G$ , and use  $P$  to determine whether  $G_{\text{new}}$  admits a list edge coloring which can be *extended* to all of  $G$ .

## Organization

In Section 2 we review notation, basic assumptions, and useful facts about polynomials, matrices, and matroids. In Section 3 we introduce our partition sieving method for monomial detection. In Section 4 we present a generic algorithmic template for solving coloring problems using polynomials. In Section 5 we apply this framework to design our algorithms for EDGE COLORING and prove Theorems 1.1 and 1.2. We conclude in Section 6 with a summary of our work and some open problems. Our LIST EDGE COLORING algorithm and the proof of Theorem 1.3 is deferred to the full version of this paper.

## 2 Preliminaries

### General Notation

Given a positive integer  $a$ , we let  $[a] = \{1, \dots, a\}$  denote the set of the first  $a$  consecutive positive integers. Given a set  $S$  and positive integer  $k$ , we let  $\binom{S}{k}$  denote the family of subsets of  $S$  of size  $k$ . Given a positive integer  $k$ , we let  $H_k$  denote the  $k^{\text{th}}$  Harmonic number, defined in Equation (1).

### Graph Notation and Assumptions

Throughout, we consider graphs which are undirected. We let  $n$  and  $m$  denote the number of vertices and edges in the input graphs for the problems we study. We assume that the input graphs are connected, so that  $m \geq n - 1$ . This is without loss of generality, since when solving EDGE COLORING and LIST EDGE COLORING on disconnected graphs, it suffices to solve the problems separately on each connected component. Given a graph  $G$  and vertex  $v$ , we let  $\deg_G(v)$  denote the number of neighbors of  $v$  (i.e., the degree of  $v$  in  $G$ ). For the LIST EDGE COLORING problem where each edge is assigned a list of colors from a subset of  $[k]$ , we assume that  $\deg_G(v) \leq k$  for all vertices  $v$  in  $G$ , since if some vertex  $v$  has degree greater than  $k$ , its incident edges must all be assigned distinct colors in a proper edge coloring, and the answer to the LIST EDGE COLORING problem is trivially no.

We let  $\Pi(G)$  denote the set of perfect matchings of  $G$  – partitions of the vertex set of  $G$  into parts of size two, such that each part consists of two adjacent vertices.

### Dominating Sets

A dominating set  $D$  in a graph  $G$  is a subset of vertices of  $G$  such that every vertex in  $G$  is either in  $D$  or adjacent to a node in  $D$ . We collect some useful results relating to finding small dominating sets in graphs.

## 7:6 Faster Edge Coloring by Partition Sieving

► **Lemma 2.1** (Simple Dominating Set). *There is a polynomial time algorithm which takes in a connected graph on  $n$  nodes and outputs a dominating set for the graph of size at most  $n/2$ .*

Lemma 2.1 is due to Ore [32], and we include a proof in the full version of this paper.

► **Lemma 2.2** (Dominating Sets in Dense Graphs [12, 29]). *If  $G$  has  $n \geq 8$  nodes, and every vertex in  $G$  has degree at least two, then  $G$  has a dominating set of size at most  $2n/5$ .*

► **Lemma 2.3** (Dominating Sets in Regular Graphs [1, 33]). *Any  $d$ -regular graph on  $n$  vertices has a dominating set of size at most  $(H_{d+1}/(d+1))n$ .*

► **Lemma 2.4** (Dominating Set Algorithm). *Let  $G'$  be a graph obtained by starting with  $G$ , and repeatedly deleting vertices of degree one until no vertices of degree one remain. Suppose at most  $n/5$  unit-degree vertices were deleted from  $G$  to produce  $G'$ . Then a minimum-size dominating set of  $G'$  can be found in  $O^*(2^{m-3n/5})$  time and polynomial space.*

Lemma 2.4 is proved in the full version of this paper.

### Finite Field Arithmetic

Throughout, we work with polynomials and matrices over a field  $\mathbb{F} = \mathbb{F}_{2^\ell}$  of characteristic two, where  $\ell = \text{poly}(n)$  is sufficiently large. Arithmetic operations over  $\mathbb{F}$  take  $\text{poly}(n)$  time. All elements  $a \in \mathbb{F}$  satisfy  $a^{2^\ell} = a$ . Consequently, given any element  $a \in \mathbb{F}$ , we can compute its unique square root as  $a^{2^{\ell-1}}$  by repeated squaring in  $\text{poly}(n)$  time.

### Polynomials

Let  $P(X)$  be a polynomial over a set of variables  $X = \{x_1, \dots, x_n\}$ . A monomial  $\mathbf{m}$  is a product  $\mathbf{m} = x_1^{m_1} \cdots x_n^{m_n}$  for some nonnegative integers  $m_1, \dots, m_n$ . A monomial  $\mathbf{m}$  is *multilinear* if  $m_i \leq 1$  for each  $i \in [n]$ . We say  $\mathbf{m}$  appears in the polynomial  $P$  if the coefficient of  $\mathbf{m}$  in  $P$  is nonzero. We define the *support* of  $\mathbf{m}$  to be the set of indices

$$\text{supp}(\mathbf{m}) = \{i \in [n] \mid m_i > 0\}$$

of variables which appear with positive degree in  $\mathbf{m}$ .

Similarly, we define the *odd support* of  $\mathbf{m}$  to be the set

$$\text{osupp}(\mathbf{m}) = \{i \in [n] \mid m_i \equiv 1 \pmod{2}\}$$

of indices of variables which appear with odd degree in  $\mathbf{m}$ . By definition,  $\text{osupp}(\mathbf{m}) \subseteq \text{supp}(\mathbf{m})$  for all monomials  $\mathbf{m}$ .

The *degree* of a monomial  $\mathbf{m} = x_1^{m_1} \cdots x_n^{m_n}$  is defined to be  $\text{deg}(\mathbf{m}) = m_1 + \cdots + m_n$ . Given  $S \subseteq [n]$ , we define the degree of  $\mathbf{m}$  restricted to the variables indexed by  $S$  to be

$$\sum_{i \in S} m_i.$$

The *total degree* (or just *degree*) of  $P$  is defined to be

$$\text{deg } P = \max_{\mathbf{m}} \text{deg}(\mathbf{m})$$

where the maximum is taken over all monomials  $\mathbf{m}$  with nonzero coefficient in  $P$ . We say that a polynomial  $P$  is *homogeneous* if  $\text{deg}(\mathbf{m}) = \text{deg } P$  for all monomials  $\mathbf{m}$  in  $P$ .

We recall the Lagrange Interpolation formula, which allows one to recover coefficients of a low degree univariate polynomial from a small number of evaluations of that polynomial.

► **Proposition 2.5** (Lagrange Interpolation). *Let  $P(z)$  be a univariate polynomial of degree less than  $n$ . Suppose that  $P(z_i) = p_i$  for distinct  $z_1, \dots, z_n \in \mathbb{F}$ . Then*

$$P(z) = \sum_{i \in [n]} p_i \prod_{j \in [n] \setminus \{i\}} \frac{z - z_j}{z_i - z_j}.$$

*So given  $n$  evaluations of  $P$  at distinct points, we can compute all the coefficients of  $P(z)$  in polynomial time.*

We also record the following observation, proven for example in [30, Theorem 7.2], which shows that to test whether a low degree polynomial  $P$  is nonzero, it suffices to check whether a random evaluation of  $P$  over a sufficiently large field is nonzero.

► **Proposition 2.6** (Schwartz-Zippel Lemma). *Let  $P$  be a nonzero polynomial over a finite field  $\mathbb{F}$  of degree at most  $d$ . If each variable of  $P$  is assigned an independent, uniform random value from  $\mathbb{F}$ , then the corresponding evaluation of  $P$  is nonzero with probability  $1 - d/|\mathbb{F}|$ .*

## Matrices

Given a matrix  $A$  and sets of rows  $I$  and columns  $J$ , we let  $A[I, J]$  denote the submatrix of  $A$  restricted to rows in  $I$  and columns in  $J$ . We let  $A[I, \cdot]$  denote the submatrix of  $A$  restricted to rows in  $I$  but using all columns, and  $A[\cdot, J]$  denote the submatrix of  $A$  restricted to columns in  $J$  but using all rows. If the rows and column sets of  $A$  are identical, we write  $A[S]$  as shorthand for  $A[S, S]$ . We let  $O$  denote the all-zeros matrix, whose dimensions will always be clear from context. We let  $A^\top$  denote the transpose of  $A$ . A matrix  $A$  is *symmetric* if  $A = A^\top$ , and *skew-symmetric* if it is symmetric and has zeros along its main diagonal (we employ this definition because we work over characteristic two).

If  $A$  is a square matrix, we let  $\det A$  denote the determinant of  $A$ .

Given a set  $V$ , a perfect matching on  $V$  is a partition of  $V$  into sets of two elements each. We let  $\Pi(V)$  denote the set of perfect matchings on  $V$ . Given a skew-symmetric matrix  $A$  with rows and columns indexed by a set  $V$ , we define the *Pfaffian* of  $A$  to be

$$\text{Pf } A = \sum_{M \in \Pi(V)} \prod_{\{u, v\} \in M} A[u, v]. \quad (2)$$

The standard definition for Pfaffians involves a sign term (see e.g., [31, Section 7.3.2]), which we ignore here because we work over a field of characteristic two. It is well known that for all skew-symmetric  $A$ , we have  $\det A = (\text{Pf } A)^2$  (see e.g., [31, Proposition 7.3.3] for a proof). Consequently, the Pfaffian of matrix  $A$  over the field  $\mathbb{F}_{2^\ell}$  can be computed in  $\text{poly}(n, \ell) \leq \text{poly}(n)$  time by using the equation  $\text{Pf } A = \sqrt{\det A} = (\det A)^{2^{\ell-1}}$ .

We record the following useful facts for computation with Pfaffians and determinants.

► **Proposition 2.7** (Direct Sums). *For any skew-symmetric matrices  $A_1$  and  $A_2$ , we have*

$$\text{Pf} \begin{pmatrix} A_1 & O \\ O & A_2 \end{pmatrix} = (\text{Pf } A_1) (\text{Pf } A_2)$$

*over a field of characteristic two.*

Proposition 2.7 is proved in the full version of this paper.

► **Proposition 2.8** (Ishikawa-Wakayama Formula). *For a skew-symmetric  $n \times n$ -matrix  $A$  and a  $k \times n$ -matrix  $B$  with  $k \leq n$ , we have*

$$\text{Pf } BAB^\top = \sum_{S \in \binom{[n]}{k}} \det B[\cdot, S] \text{Pf } A[S].$$

## 7:8 Faster Edge Coloring by Partition Sieving

Proposition 2.8 is due to [22] (see [23, Section 4.3] for a recent alternate proof).

### Matroids

A *matroid* is a pair  $\mathcal{M} = (X, \mathcal{I})$ , consisting of a *ground set*  $X$  and a collection  $\mathcal{I}$  of subsets of  $X$  called *independent sets*, satisfying the following axioms:

1.  $\emptyset \in \mathcal{I}$ .
2. If  $B \in \mathcal{I}$  and  $A \subset B$ , then  $A \in \mathcal{I}$ .
3. For any  $A, B \in \mathcal{I}$  with  $|A| < |B|$ , there exists an element  $b \in B \setminus A$  such that  $A \cup \{b\} \in \mathcal{I}$ .

A maximal independent set in a matroid  $\mathcal{M}$  is called a *basis*. It is well known that all bases of  $\mathcal{M}$  have the same size  $r$ , which is referred to as the *rank* of  $\mathcal{M}$ . We say  $S \subseteq V$  *spans*  $\mathcal{M}$  if  $S$  is a superset of a basis of  $\mathcal{M}$ . If there exists a matrix  $A$  with column set  $X$  such that for every  $S \subseteq X$ , the set  $S$  is independent in  $\mathcal{M}$  if and only if  $A[:, S]$  has full rank, then we say that  $\mathcal{M}$  is a *linear matroid represented* by  $A$ . Provided we work over a large enough field of size  $\text{poly}(|X|)$ , it is known that if  $\mathcal{M}$  is linear, then it can be represented by a matrix whose number of rows equals the rank of  $\mathcal{M}$  (see e.g., [26] or [28, Section 3.7]).

Given matroids  $\mathcal{M}_1, \dots, \mathcal{M}_k$  over disjoint ground sets  $X_1, \dots, X_k$  respectively, we define the direct sum

$$\mathcal{M} = \bigoplus_{i=1}^k \mathcal{M}_i$$

of these matroids to be the ground set  $X = X_1 \sqcup \dots \sqcup X_k$  together with the family  $\mathcal{I}$  of subsets  $S$  of  $X$  with the property that  $S \cap X_i$  is independent in  $\mathcal{M}_i$  for all  $i \in [k]$ . It is well known that  $\mathcal{M}$  is a matroid as well, whose rank is equal to the sum of the ranks of the  $\mathcal{M}_i$ .

The following result lets us represent direct sums of linear matroids [28, Section 3.4].

► **Proposition 2.9 (Matroid Sum).** *Given linear matroids  $\mathcal{M}_1, \dots, \mathcal{M}_k$  represented by matrices  $A_1, \dots, A_k$  respectively, their direct sum is represented by the block-diagonal matrix*

$$A = \begin{pmatrix} A_1 & O & \dots & O \\ O & A_2 & \dots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \dots & A_k \end{pmatrix}.$$

In our EDGE COLORING algorithm, we make use of two simple types of matroids: uniform and partition matroids. Given a ground set  $X$  and a nonnegative integer  $r$ , the *uniform matroid* over  $X$  of rank  $r$  has as its independent sets all subsets of  $X$  of size at most  $r$ .

► **Proposition 2.10 (Uniform Matroid Representation [28, Section 3.5]).** *Given a set  $X$ , a nonnegative integer  $r$ , and a field  $\mathbb{F}$  with  $|\mathbb{F}| > |X|$ , we can construct an  $r \times |X|$  matrix representing the uniform matroid over  $X$  of rank  $r$  in  $\text{poly}(|X|)$  time.*

Given a set  $X$  partitioned  $X = X_1 \sqcup \dots \sqcup X_k$  into parts  $X_i$ , and a list of nonnegative integer capacities  $c_1, \dots, c_k$ , the *partition matroid*  $\mathcal{M}$  over the ground set  $X$  for this partition and list of capacities has as its independent sets all subsets  $S \subseteq X$  satisfying  $|S \cap X_i| \leq c_i$  for all  $i \in [k]$ . Equivalently,  $\mathcal{M}$  is the direct sum of uniform matroids of rank  $c_i$  over  $X_i$  for all  $i \in [k]$ . By definition,  $\mathcal{M}$  has rank  $r = (c_1 + \dots + c_k)$ . By Propositions 2.9 and 2.10 we see that we can efficiently construct representations of partition matroids [28, Proposition 3.5].

► **Proposition 2.11 (Partition Matroid Representation).** *Given a partition matroid  $\mathcal{M}$  over  $X$  of rank  $r$ , and a field  $\mathbb{F}$  with  $|\mathbb{F}| > |X|$ , we can construct an  $r \times |X|$  matrix representing  $\mathcal{M}$  in  $\text{poly}(|X|)$  time.*



### 3 Polynomial Sieving

In this section, we develop faster algorithms for a certain monomial detection problem. This is our main technical result, which allows us to obtain faster algorithms for EDGE COLORING and LIST EDGE COLORING. We recall the following well-known sieving technique:

► **Proposition 3.1** (Inclusion-Exclusion). *Let  $P(X)$  be a polynomial over a field of characteristic two, with variable set  $X = \{x_1, \dots, x_n\}$ . Fix  $T \subseteq X$ . Let  $Q(X)$  be the polynomial consisting of all monomials (with corresponding coefficients) in  $P$  that are divisible by  $\prod_{x \in T} x$ . Then*

$$Q = \sum_{S \subseteq T} P_S$$

where  $P_S$  is the polynomial obtained from  $P$  by setting  $x_i = 0$  for all  $i \in S$ .

See [35, Lemma 2] for a proof of Proposition 3.1.

Combining Propositions 2.5 and 3.1, we obtain the following result.

► **Lemma 3.2** (Coefficient Extraction). *Let  $P(X)$  be a polynomial over a field of characteristic two, with variable set  $X = \{x_1, \dots, x_n\}$ . For  $T \subseteq X$ , let  $Q(X \setminus T)$  be the coefficient of  $\prod_{x \in T} x$  in  $P(X)$ , viewed as a polynomial over the variable set  $X \setminus T$ . Then we can evaluate  $Q(X \setminus T)$  at a point as a linear combination of  $2^{|T|}$  poly( $n$ ) evaluations of  $P(X)$ .*

**Proof.** Fix a subset  $T \subseteq X$ . Let  $F$  be the polynomial with variable set  $X \cup \{z\}$  obtained by substituting each variable  $x \in T$  with  $xz$  in  $P$ . Let  $G(X)$  be the coefficient of  $z^{|T|}$  in  $F$ . By definition of  $F$ ,  $G$  is the polynomial obtained by taking the monomials in  $P$  whose degree restricted to  $T$  equals  $|T|$ . By Proposition 2.5, we can compute  $G(X)$  at any point as a linear combination of poly( $n$ ) evaluations of  $P$ .

By definition,  $Q(X \setminus T)$  is the polynomial obtained by taking monomials of  $G(X)$  that are divisible by  $\prod_{x \in T} x$ , and then setting all variables in  $T$  equal to 1. So by Proposition 3.1 we can evaluate  $Q(X \setminus T)$  as the sum of  $2^{|T|}$  evaluations of  $G$ . Then by the conclusion of the previous paragraph, we get that we can evaluate  $Q(X \setminus T)$  at any point as a linear combination of  $2^{|T|}$  poly( $n$ ) evaluations of  $P$  as claimed. ◀

The inclusion-exclusion principle allows us to sieve for the multilinear term in a polynomial that is the product of all its variables. This was extended to the parameterized setting, where the goal is to sieve for multilinear terms of degree  $k$  in  $O^*(2^k)$  time [7, 10, 36], and then further extended to multilinear monomial detection in the matroid setting [16].

► **Lemma 3.3** (Basis Sieving [16, Theorem 1.1]). *Let  $P(X)$  be a homogeneous polynomial of degree  $k$  over a field  $\mathbb{F}$  of characteristic 2, and let  $\mathcal{M} = (X, \mathcal{I})$  be a matroid on  $X$  of rank  $k$ , with a representation over  $\mathbb{F}$  given. Then there is a randomized algorithm running in  $O^*(2^k)$  time and polynomial space, which determines with high probability if  $P(X)$  contains a multilinear monomial  $\mathbf{m}$  such that  $\text{supp}(\mathbf{m})$  is a basis of  $\mathcal{M}$ .*

Moreover, [16] introduced a variant of sieving that takes the odd support into account. This tool is key to our exponential speed-up for the EDGE COLORING problem.

► **Lemma 3.4** (Odd Sieving [16, Theorem 1.2]). *Let  $P(X)$  be a polynomial of degree  $d$  over a field  $\mathbb{F}$  of characteristic 2, and let  $\mathcal{M} = (X, \mathcal{I})$  be a matroid on  $X$  of rank  $k$ , represented by a  $k \times n$  matrix over  $\mathbb{F}$ . Then using  $O^*(d2^k)$  evaluations of  $P(X)$  and polynomial space, we can determine if  $P(X)$  contains a monomial  $\mathbf{m}$  such that  $\text{osupp}(\mathbf{m})$  spans  $\mathcal{M}$ .*

## 7:10 Faster Edge Coloring by Partition Sieving

Let  $P(X)$  be a homogeneous polynomial of degree  $d$  in variables  $X = \{x_1, \dots, x_n\}$ . Consider a partition  $X = X_1 \sqcup \dots \sqcup X_p$  into parts  $X_i$ . Let  $c_1, \dots, c_p \geq 0$  be integers with

$$c_1 + \dots + c_p = d.$$

Let  $\mathcal{M}$  be the partition matroid defined by the partition of  $X$  into the parts  $X_i$  and the capacities  $c_i$  for  $i \in [p]$ . We say  $P(X)$  is *compatible* with  $\mathcal{M}$  if for every  $i \in [p]$  and every monomial  $\mathbf{m}$  in  $P$ , the degree of  $\mathbf{m}$  restricted to  $X_i$  equals  $c_i \geq 1$ .

The following result gives an improvement over Lemma 3.3 for polynomials that are compatible with partition matroids.

► **Theorem 3.5 (Partition Sieving).** *Let  $P(X)$  be a polynomial of degree  $d$  over a field  $\mathbb{F}$  of characteristic 2 with  $|\mathbb{F}| > d$ , and let  $\mathcal{M} = (X, \mathcal{I})$  be a partition matroid on  $X$  of rank  $d$  with  $p$  partition classes. If  $P$  is compatible with  $\mathcal{M}$ , then there is a randomized algorithm which runs in  $O^*(2^{d-p})$  time and polynomial space that determines with high probability if  $P(X)$  contains a monomial  $\mathbf{m}$  such that  $\text{supp}(\mathbf{m})$  is a basis of  $\mathcal{M}$ .*

**Proof.** Let  $X = X_1 \sqcup \dots \sqcup X_p$  be the partition and  $c_1, \dots, c_p$  be the capacities defining the partition matroid  $\mathcal{M}$ . By assumption,  $c_i \geq 1$  for all  $i \in [p]$ . Let  $\mathcal{M}'$  be the partition matroid over  $X$  defined by the same partition as  $\mathcal{M}$ , but with capacities  $(c_i - 1)$  for  $i \in [p]$ . From this construction,  $\mathcal{M}'$  has rank  $(d - p)$ .

Our algorithm to determine whether  $P(X)$  contains a monomial whose support spans  $\mathcal{M}$  works as follows. By Proposition 2.11, a linear representation of  $\mathcal{M}$  over  $\mathbb{F}$  can be computed in polynomial time. We run the odd sieving algorithm of Lemma 3.4 with  $P$  and  $\mathcal{M}'$  to determine in  $O^*(2^{d-p})$  time whether  $P$  contains a monomial whose odd support spans  $\mathcal{M}'$ . We return YES if such a monomial exists, and return NO otherwise. We now prove that this algorithm is correct.

First, suppose that  $P(X)$  has a monomial  $\mathbf{m}$  whose support  $\text{supp}(\mathbf{m})$  is a basis of  $\mathcal{M}$ . By the compatibility condition,  $\mathbf{m}$  is multilinear, which implies that  $\text{supp}(\mathbf{m}) = \text{osupp}(\mathbf{m})$ . Since  $\text{osupp}(\mathbf{m})$  is a basis of  $\mathcal{M}$ , and thus spans  $\mathcal{M}$ , it follows that  $\text{osupp}(\mathbf{m})$  spans  $\mathcal{M}'$ .

Conversely, suppose  $P(X)$  has a monomial  $\mathbf{m}$  such that  $\text{osupp}(\mathbf{m})$  spans  $\mathcal{M}'$ . We claim that  $\text{supp}(\mathbf{m})$  spans  $\mathcal{M}$ . Indeed, since  $\text{osupp}(\mathbf{m})$  spans  $\mathcal{M}'$ , the set  $\text{osupp}(\mathbf{m})$  contains at least  $(c_i - 1)$  variables of  $X_i$  for each  $i \in [p]$ . Hence its superset  $\text{supp}(\mathbf{m})$  contains at least  $(c_i - 1)$  variables of each part  $X_i$  as well.

▷ **Claim 3.6.** The set  $\text{supp}(\mathbf{m})$  contains exactly  $c_i$  variables of each part  $X_i$ .

**Proof.** Suppose to the contrary that  $\text{supp}(\mathbf{m})$  does not contain  $c_j$  variables of  $X_j$  for some index  $j \in [p]$ . Then  $\mathbf{m}$  contains exactly  $(c_j - 1)$  variables of  $X_j$ . Since  $P$  is compatible with  $\mathcal{M}$ , the monomial  $\mathbf{m}$  has degree exactly  $c_j$  when restricted to  $X_j$ . The only way this is possible is if  $c_j \geq 2$ , and  $\mathbf{m}$  has exactly  $(c_j - 2)$  variables in  $X_j$  of degree one and a single variable in  $X_j$  with degree two. The variable of degree two does not show up in the odd support of  $\mathbf{m}$ , which implies that  $\text{osupp}(\mathbf{m})$  has at most  $(c_j - 2)$  elements, which contradicts the assumption that  $\text{osupp}(\mathbf{m})$  spans  $\mathcal{M}'$ . ◁

By Claim 3.6, the set  $\text{supp}(\mathbf{m})$  has exactly  $c_i$  variables from  $X_i$  for each  $i \in [p]$ . Since  $P$  is compatible with  $\mathcal{M}$ , this implies that  $\mathbf{m}$  has degree one in every variable in  $X$ . Hence  $\text{supp}(\mathbf{m}) = \text{osupp}(\mathbf{m})$  spans  $\mathcal{M}$ , as claimed.

This shows that the algorithm is correct, and proves the desired result. ◀

## 4 Coloring Algorithm Template

Recall that in the LIST EDGE COLORING problem we are given a graph  $G = (V, E)$ , an integer  $k \geq 1$ , and lists  $L_e \subseteq [k]$  of colors for every edge  $e \in E$ , and are tasked with determining if  $G$  contains an assignment of colors  $c(e) \in L_e$  to each edge such that no two edges incident to the same vertex are assigned the same color. We call such an assignment a proper edge coloring of  $G$  with respect to the lists  $L_e$ , or just a proper list edge coloring if the  $L_e$  are clear from context.

For each  $i \in [k]$ , let  $G_i = (V, E_i)$  be the subgraph of  $G$ , where we define  $E_i \subseteq E$  to be the set of all edges  $e$  in the graph which have  $i \in L_e$  available as a color. Let  $\mathcal{F}$  denote the collection of  $k$ -tuples  $(M_1, \dots, M_k)$  of matchings such that

1. for all  $i \in [k]$ ,  $M_i$  is a matching in  $G_i$ , and
2. for all vertices  $v \in V$ ,  $v$  appears as the endpoint of exactly  $\deg_G(v)$  of the  $M_i$  matchings.

Observe that  $G$  admits a proper list edge coloring if and only if  $\mathcal{F}$  contains a tuple of edge-disjoint matchings. Indeed, suppose  $G$  admits a proper list edge coloring. For each  $i \in [k]$ , the set of edges assigned color  $i$  must be a matching  $M_i \subseteq E_i$  in  $G$ . Moreover, every edge is assigned a color, so each vertex  $v$  appears as an endpoint in  $\deg_G(v)$  of these matchings, hence the matchings form an edge-disjoint tuple in  $\mathcal{F}$ . Conversely, given an edge-disjoint tuple  $(M_1, \dots, M_k) \in \mathcal{F}$ , assigning the edges in  $M_i$  color  $i$  gives a color to every edge in  $G$  by condition 2 above, and thus yields a proper list edge coloring by condition 1.

The basic idea of our algorithmic template is to construct a polynomial  $P$  which enumerates a certain subfamily  $\mathcal{C} \subseteq \mathcal{F}$ . We then argue that  $P$  contains a multilinear monomial satisfying certain matroid constraints if and only if  $\mathcal{C}$  contains a tuple of edge-disjoint matchings. Using similar reasoning to the previous paragraph, this then lets us solve LIST EDGE COLORING by running monomial detection algorithms on  $P$ .

### 4.1 Enumerating Matchings

In this section, we design a polynomial  $P$  whose monomials enumerate tuples of matchings satisfying special conditions. To construct  $P$ , we first define certain polynomial matrices.

For each edge  $e \in E$  we introduce a variable  $x_e$ , and for each pair  $(e, i) \in E \times [k]$  we introduce a variable  $y_{ei}$ . Let  $X$  be the set of  $x_e$  variables and  $Y$  be the set of  $y_{ei}$  variables.

For each  $i \in [k]$ , define  $A_i$  to be the matrix with rows and columns indexed by  $V$  with

$$A_i[u, w] = A_i[w, u] = \begin{cases} x_e y_{ei} & \text{if } e = \{u, w\} \in E_i, \\ 0 & \text{otherwise.} \end{cases}$$

For each  $i \in [k]$ , define  $V_i = \{v_i \mid v \in V\}$  to be a copy of  $V$ . Let

$$W = \bigsqcup_{i=1}^k V_i$$

be the union of these copies. Let  $A$  be the symmetric matrix with rows and columns indexed by  $W$ , defined by taking

$$A = \begin{pmatrix} A_1 & O & \dots & O \\ O & A_2 & \dots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \dots & A_k \end{pmatrix} \tag{3}$$

and identifying  $V_i$  as the row and column set indexing  $A_i$ .

## 7:12 Faster Edge Coloring by Partition Sieving

Note that  $A$  is  $(kn) \times (kn)$ , and over a field of characteristic two is skew-symmetric.

For each vertex  $v \in V$ , let  $S_v = \{v_i \mid i \in [k]\}$  be the set of  $k$  copies of  $v$ .

For each vertex  $v$ , let  $\mathcal{M}_v$  be a linear matroid of rank  $\deg_G(v)$  over  $S_v$ . These matroids will be specified later on, in our algorithms for EDGE COLORING and LIST EDGE COLORING. Let  $\mathcal{W}$  be the direct sum

$$\mathcal{W} = \bigoplus_{v \in V} \mathcal{M}_v \quad (4)$$

of these matroids.

By Proposition 2.7 the matroid  $\mathcal{W}$  is linear and has rank equal to the sum of the degrees of vertices in  $G$ , which is  $2m$ . Let  $B$  be a  $2m \times |W|$  matrix representing  $\mathcal{W}$ .

Let  $\mathcal{C}$  be the collection of  $k$ -tuples of matchings  $(M_1, \dots, M_k)$  in  $G$  such that

1. all edges in  $M_i$  have lists containing the color  $i$ ,
2. every vertex  $v$  is incident to exactly  $\deg_G(v)$  edges across the matchings  $M_i$ , and
3. for each vertex  $v$ , the set of copies  $v_i$  taken over all  $i$  such that  $v$  appears as an endpoint of an edge in  $M_i$  forms an independent set in  $\mathcal{M}_v$ .

We define the polynomial

$$P(X, Y) = \text{Pf } BAB^\top. \quad (5)$$

The next result shows that  $P$  enumerates tuples of matchings from  $\mathcal{C}$ .

► **Lemma 4.1** (Enumerating Matchings). *We have*

$$\text{Pf } BAB^\top = \sum_{(M_1, \dots, M_k) \in \mathcal{C}} c_{M_1, \dots, M_k} \left( \prod_{i \in [k]} \prod_{e \in M_i} y_{ei} \right) \left( \prod_{e \in M_1 \cup \dots \cup M_k} x_e \right),$$

where each  $c_{M_1, \dots, M_k} \neq 0$  is a constant depending only on  $M_1, \dots, M_k$ .

**Proof.** Let  $U \subseteq W$  be a subset of  $W$  of size  $2m$ . By Proposition 2.8, we have

$$\text{Pf } BAB^\top = \sum_{U \in \binom{W}{2m}} \det B[\cdot, U] \text{Pf } A[U]. \quad (6)$$

By Equation (3) and Proposition 2.7, for every subset  $U \subseteq W$  of size  $2m$  we have

$$\text{Pf } A[U] = \prod_{i \in [k]} \text{Pf } A_i[U \cap V_i].$$

Substituting the above equation into Equation (6) yields

$$\text{Pf } BAB^\top = \sum_{U \in \binom{W}{2m}} \det B[\cdot, U] \prod_{i \in [k]} \text{Pf } A_i[U \cap V_i].$$

Recall that given a graph  $H$ , we let  $\Pi(H)$  denote the set of perfect matchings of  $H$ . By expanding out the definition of  $\text{Pf } A_i[U \cap V_i]$  in the above equation, we see that

$$\text{Pf } BAB^\top = \sum_{U \in \binom{W}{2m}} \det B[\cdot, U] \prod_{i \in [k]} \sum_{M_i \in \Pi(G[U_i])} \prod_{e \in M_i} A_i[e],$$

where  $U_i \subseteq V$  is defined as a copy of  $U \cap V_i$ , i.e.,  $U_i = \{v \mid v_i \in U \cap V_i\}$ . By interchanging the order of the product and the summation, we see that the above is equivalent to summing over all collections  $(M_1, \dots, M_k)$ , where  $M_i$  is a matching in  $G[U_i]$  for each  $i \in [k]$ :

$$\text{Pf } BAB^\top = \sum_{U \in \binom{W}{2m}} \det B[\cdot, U] \sum_{(M_1, \dots, M_k)} \prod_{i \in [k]} \prod_{e \in M_i} A_i[e].$$

Let  $\vec{M} = (M_1, \dots, M_k)$  be a tuple such that the summand corresponding to  $\vec{M}$  in the above equation appears with nonzero coefficient in  $\text{Pf } BAB^\top$ . Since  $A_i[e]$  is nonzero if and only if  $i \in L_e$ , the tuple  $\vec{M}$  meets condition 1 in the definition of  $\mathcal{C}$ . By the definition of  $B$ , we have  $\det B[\cdot, U] \neq 0$  if and only if  $U \cap S_v$  is independent in  $\mathcal{M}_v$  for every vertex  $v \in V$ . Thus  $\vec{M}$  meets condition 3 in the definition of  $\mathcal{C}$ . Since  $\mathcal{M}_v$  has rank  $\deg_G(v)$  for each vertex  $v$ ,  $U$  has size  $2m$ , and the sum of degrees of vertices in  $G$  is  $2m$ , we see that the only way for  $U \cap S_v$  to be independent in  $\mathcal{M}_v$  for all  $v$  is if in fact  $|U \cap S_v| = \deg_G(v)$  always. So  $\vec{M}$  satisfies condition 2 from the definition of  $\mathcal{C}$ .

Thus, the nonzero terms in the above sum correspond precisely to tuples  $\vec{M} \in \mathcal{C}$ , which proves the desired result.  $\blacktriangleleft$

## 4.2 Partition Sieving With Dominating Sets

In this section, we describe a generic way of going from a dominating set in  $G$  to a certain partition matroid  $\mathcal{D}$ . After defining this  $\mathcal{D}$ , we will apply Theorem 3.5 to  $P$  and  $\mathcal{D}$  to achieve a fast algorithm for detecting a multilinear monomial in  $P$ .

Recall that a *dominating set* of a graph  $G$  is a subset of vertices  $D$  such that every vertex in  $G$  is either in  $D$  or adjacent to a vertex in  $D$ .

Fix a dominating set  $D \subseteq V$  of the input graph  $G$ . Let  $V' = V \setminus D$ . Let  $E'$  be the set of edges in  $G$  with one endpoint in  $D$  and one endpoint in  $V'$ , and  $X' = \{x_e \mid e \in E'\}$  be the corresponding set of variables. For each  $v \in V'$ , let

$$\partial(v) = \{e \in E' \mid e \ni v\}$$

be the set of edges incident to  $v$  which connect  $v$  to a vertex in  $D$ . By definition, we have a partition

$$E' = \bigsqcup_{v \in V'} \partial(v).$$

Let  $\mathcal{D}$  be the partition matroid over  $E'$  with respect to the above partition, and with capacities  $c_v = \deg_{E'}(v)$  for each  $v \in V'$ , where  $E'$  is viewed as a subgraph of  $G$ . Since  $D$  is a dominating set, every vertex  $v \in V'$  is adjacent to some node in  $D$ , and thus the capacities  $c_v \geq 1$  are all positive.

We now use this partition matroid to sieve over the polynomial  $P$ , defined in Equation (5).

**► Lemma 4.2** (Accelerated Multilinear Monomial Detection). *There is a randomized algorithm that determines with high probability whether  $P(X, Y)$  has a monomial divisible by  $\prod_{e \in E} x_e$ , running in  $O^*(2^{m-|V'|})$  time and polynomial space.*

**Proof.** Let  $Q(X', Y)$  be the coefficient of

$$\prod_{e \in E \setminus E'} x_e$$

in  $P(X, Y)$ . By definition,  $P(X, Y)$  contains a monomial divisible by  $\prod_{e \in E} x_e$  if and only if  $Q(X', Y)$  contains a monomial divisible by  $\prod_{e \in E'} x_e$ .

## 7:14 Faster Edge Coloring by Partition Sieving

Let  $R(Y)$  be the coefficient of  $\prod_{e \in E'} x_e$  in  $Q$ , viewed as a polynomial with variable set  $Y$ . Note that the total degree of  $R(Y)$  is  $\text{poly}(n)$ . Take a uniform random assignment over  $\mathbb{F}$  to the variables in  $Y$ . By Proposition 2.6, if  $R$  is nonzero, then it remains nonzero with high probability after this evaluation, provided we take  $|\mathbb{F}| = \text{poly}(n)$  to be sufficiently large. For the rest of this proof, we work with polynomials  $P, Q, R$  after this random evaluation, so that  $P$  is a polynomial in  $X$ ,  $Q$  is a polynomial in  $X'$ , and  $R$  is a field element.

By the discussion in the first paragraph of this proof, it suffices to check whether  $Q(X')$  contains the monomial  $\prod_{e \in E'} x_e$  in  $O^*(2^{m-|V'|})$  time and polynomial space.

We observe the following helpful property of  $Q$ .

▷ **Claim 4.3.** The polynomial  $Q(X')$  is compatible with the matroid  $\mathcal{D}$ .

*Proof.* Take an arbitrary monomial  $\mathbf{m}$  in  $Q$ . By definition, there is a corresponding monomial

$$\mathbf{m}' = \mathbf{m} \cdot \prod_{e \in E \setminus E'} x_e$$

in  $P$ . By Lemma 4.1, for every  $v \in V$ ,  $\mathbf{m}'$  has degree exactly  $\deg_G(v)$  when restricted to the variables  $x_e$  corresponding to edges  $e$  containing  $v$ . Consequently, for all  $v \in V'$ ,  $\mathbf{m}$  has degree exactly

$$\deg_G(v) - |\{e \notin E' \mid e \ni v\}| = \deg_{E'}(v)$$

when restricted to variables  $x_e$  with  $e \in \partial(v)$ . Since this holds for arbitrary monomials  $\mathbf{m}$  in  $Q$ , the polynomial  $Q$  is compatible with the partition matroid  $\mathcal{D}$  as claimed. ◁

By Lemma 4.1,  $P$ , viewed as a polynomial in  $X$ , is homogeneous of degree  $|E| = m$ . Hence  $Q$ , viewed as a polynomial in  $X'$ , is homogeneous of degree  $d = |E'|$ . By Claim 4.3,  $Q$  is compatible with  $\mathcal{D}$ . Since  $\mathcal{D}$  is a partition matroid with positive capacities and  $p = |V'|$  parts in its underlying partition, by Theorem 3.5 we can determine whether  $Q$  contains a monomial  $\mathbf{m}$  in the  $X'$  variables such that  $\text{supp}(\mathbf{m})$  is a basis of  $\mathcal{D}$  using

$$O^*(d2^{d-p}) \leq O^*(2^{|E'| - |V'|}) \tag{7}$$

evaluations of  $Q$ . Since  $E'$  is the unique basis of  $\mathcal{D}$ , this means we can test whether  $Q(X')$  contains the monomial  $\prod_{e \in E'} x_e$  using  $O^*(2^{|E'| - |V'|})$  evaluations of  $Q(X')$ .

By Lemma 3.2,  $Q(X')$  can be evaluated at any point by computing a linear combination of  $O^*(2^{m-|E'|})$  evaluations of  $P(X)$ . Combining this observation with Equation (7), we get that we can determine whether  $Q(X')$  contains the monomial  $\prod_{e \in E'} x_e$  by computing a linear combination of

$$O^*(2^{m-|E'|} \cdot 2^{|E'| - |V'|}) \leq O^*(2^{m-|V'|})$$

evaluations of  $P(X, Y)$ . Each evaluation of  $P(X, Y) = \text{Pf } BAB^\top$  takes polynomial time, so the desired result follows. ◀

## 5 Edge Coloring Algorithm

In this section, we present our algorithm for EDGE COLORING and prove Theorems 1.1 and 1.2. Recall that in this problem, we are given an input graph  $G$  with  $n$  vertices,  $m$  edges, and maximum degree  $\Delta$ . To solve the problem, it suffices to determine whether  $G$  admits a proper edge coloring using at most  $\Delta$  colors (i.e., whether  $\chi'(G) \leq \Delta$ ).

## 5.1 Removing Low Degree Vertices

► **Lemma 5.1** (Unit-Degree Deletion). *Let  $G$  be a graph with a vertex  $v$  of degree one. Let  $G'$  be the graph obtained by deleting  $v$  from  $G$ . Then  $\chi'(G') \leq \Delta$  if and only if  $\chi'(G) \leq \Delta$ .*

**Proof.** If  $G$  admits a proper edge coloring with  $\Delta$  colors, then this coloring is also a proper edge coloring for  $G'$  with  $\Delta$  colors.

Conversely, suppose  $G'$  admits a proper edge coloring with  $\Delta$  colors. Since  $\deg_G(v) = 1$ , there is a unique vertex  $w$  in  $G$  that  $v$  is adjacent to. Then

$$\deg_{G'}(w) \leq \deg_G(w) - 1 \leq \Delta - 1$$

because  $G$  has maximum degree  $\Delta$ . So vertex  $w$  has edges using at most  $(\Delta - 1)$  distinct colors incident to it. Taking the coloring of  $G'$  and assigning edge  $\{v, w\}$  a color not used by the edges incident to  $w$  in  $G'$  recovers a proper edge coloring of  $G$  with at most  $\Delta$  colors. ◀

## 5.2 Instantiating the Template

Recall the definitions from Section 4. We view the EDGE COLORING problem as a special case of LIST EDGE COLORING, where  $k = \Delta$  and every edge  $e$  is assigned the list  $L_e = [\Delta]$ . For each vertex  $v$ , we set  $\mathcal{M}_v$  to be the uniform matroid over  $S_v$  of rank  $\deg_G(v)$ . Then by Equation (4),  $\mathcal{W}$  is the partition matroid over  $W$  defined by the partition

$$W = \bigsqcup_{v \in V} S_v$$

and the capacities  $c_v = \deg_G(v)$  for parts  $S_v$ . By Proposition 2.11, we can construct the matrix  $B$  representing  $\mathcal{W}$  in polynomial time. In this case, condition 3 is identical to condition 2 in the definition of the collection  $\mathcal{C}$ .

► **Lemma 5.2** (Characterization by Polynomials). *We have  $\chi'(G) \leq \Delta$  if and only if the polynomial  $P(X, Y)$  has a monomial divisible by  $\prod_{e \in E} x_e$ .*

**Proof.** Suppose  $P(X, Y) = \text{Pf } BAB^\top$  contains a monomial divisible by  $\prod_{e \in E} x_e$ . Since  $P$  is homogeneous of degree  $m$  in the  $X$  variables, by Lemma 4.1 this means there exists a  $\Delta$ -tuple of *edge-disjoint* matchings  $(M_1, \dots, M_\Delta) \in \mathcal{C}$ , such that every edge of  $G$  appears in some matching  $M_i$ . Consequently, assigning the edges in  $M_i$  color  $i$  yields a proper edge coloring of  $G$  using at most  $\Delta$  colors.

Conversely, suppose  $G$  has a proper edge coloring with  $\Delta$  colors. Without loss of generality, let the set of colors used be  $[\Delta]$ . For each  $i \in [\Delta]$ , let  $M_i$  be the set of edges assigned color  $i$ . Since this is a proper edge coloring, each  $M_i$  is a matching in  $G$ . Since every edge is assigned a unique color, each vertex  $v$  has exactly  $\deg_G(v)$  edges across the matchings  $M_i$ . Hence  $(M_1, \dots, M_\Delta) \in \mathcal{C}$ . Then by Lemma 4.1,  $P$  contains a monomial divisible by  $\prod_{e \in E} x_e$ . ◀

We can now prove Lemma 1.4 as a simple application of Lemma 5.2.

► **Lemma 1.4.** *There is a randomized algorithm that, given a graph  $G$  on  $n$  vertices and  $m$  edges and a dominating set  $D$  of  $G$ , solves EDGE COLORING on  $G$  with high probability and one-sided error in  $O^*(2^{m-n+|D|})$  time and polynomial space.*

**Proof.** Let  $G$  have vertex set  $V$  and edge set  $E$ , and let  $V' = V \setminus D$ . By Lemma 5.2, we have  $\chi'(G) \leq \Delta$  if and only if  $P(X, Y)$  has a monomial divisible by  $\prod_{e \in E} x_e$ . By Lemma 4.2, we can determine whether such a monomial exists in

$$O^*(2^{m-|V'|}) \leq O^*(2^{m-n+|D|})$$

time and polynomial space, as desired. ◀

Having established Lemma 1.4, we are ready to present our EDGE COLORING algorithms.

► **Theorem 1.1.** *There is a randomized algorithm which solves EDGE COLORING with high probability and one-sided error in  $O^*(2^{m-3n/5})$  time and polynomial space.*

**Proof.** To solve EDGE COLORING, it suffices to determine whether  $\chi'(G) \leq \Delta$ .

If  $G$  has a vertex of degree one, delete it. Keep performing such deletions, until we are left with a graph containing no vertices of degree one. By repeated application of Lemma 5.1, the resulting graph admits a proper edge coloring using  $\Delta$  colors if and only if  $\chi'(G) \leq \Delta$ . If the resulting graph has constant size, we can determine this trivially. Each such deletion decreases the number of edges and vertices in the graph by one, and thus also decreases the value of  $(m - 3n/5)$ . Consequently, to show the desired result, we may assume without loss of generality that  $G$  has minimum degree two, and at least eight vertices.

Since  $G$  has minimum degree two and  $n \geq 8$  nodes, by Lemma 2.2 the graph has a dominating set of size at most  $2n/5$ . By Lemma 2.4, we can then find a dominating set  $D$  in  $G$  with  $|D| \leq 2n/5$  in  $O^*(2^{m-3n/5})$  time and polynomial space. Hence by Lemma 1.4 we can solve EDGE COLORING on  $G$  in  $O^*(2^{m-3n/5})$  time and polynomial space, as desired. ◀

► **Theorem 1.2.** *For each integer  $d \geq 6$ , there is a randomized algorithm that solves EDGE COLORING with high probability and one-sided error on  $d$ -regular graphs in  $O^*(2^{m-\alpha_d n})$  time and polynomial space, for  $\alpha_d = 1 - H_{d+1}/(d+1)$ .*

**Proof.** Let  $G$  be the input graph. By exhaustive search over all subsets of vertices, we can find a minimum-size dominating set  $D$  of  $G$  in  $O^*(2^n)$  time and polynomial space. Being  $d$ -regular,  $G$  has  $m = dn/2 \geq 3n$  edges, so we find  $D$  in  $O^*(2^{m-n}) \leq O^*(2^{m-\alpha_d n})$  time.

Since  $G$  is  $d$ -regular, by Lemma 2.3 we must have  $|D| \leq n(H_{d+1}/(d+1))$ . Then by Lemma 1.4 we can solve EDGE COLORING on  $G$  in

$$O^*(2^{m-n+|D|}) \leq O^*(2^{m-\alpha_d n})$$

time as claimed. ◀

## 6 Conclusion

In this paper, we presented the first algorithms for EDGE COLORING and LIST EDGE COLORING which run in faster than  $O^*(2^m)$  time while using only polynomial space. Our algorithm is based off a *partition sieving* procedure that works over polynomials of degree  $d$  and partition matroids with  $p$  parts. We showed how to implement this sieving routine in  $O^*(2^{d-p})$  time when the polynomial is in some sense compatible with the matroid, an improvement over the  $O^*(2^d)$  runtime that arose in previous techniques.

Overall, we solve EDGE COLORING in faster than  $O^*(2^m)$  time by

- Step 1** designing a polynomial  $P$  enumerating matchings meeting certain degree constraints,
- Step 2** finding a dominating set  $D$  of size at most  $2n/5$  in the input graph, and
- Step 3** applying partition sieving with a partition matroid on  $p = |D|$  parts.

Implementing **step 2** above was possible for EDGE COLORING because we could assume that the input graph had minimum degree two without loss of generality, and such graphs on  $n \geq 8$  nodes always have dominating sets of size at most  $2n/5$ . For LIST EDGE COLORING we cannot make this assumption, but nonetheless implement a variant of **step 2** by identifying a subgraph with minimum degree two and using extension matroids to handle unit-degree vertices for free. For all sufficiently large  $n$ , it is also known that graphs on  $n$  nodes with minimum degree three have dominating sets of size at most  $3n/8$  [34]. If degree-two vertices could somehow be “freely removed” from the input graph just like unit-degree vertices can,



one could use this bound on the dominating set to solve coloring problems faster. More general connections between the minimum degree of a graph and the minimum size of its dominating set have also been studied [20].

Understanding the precise time complexity of EDGE COLORING and its variants remains an interesting open research direction. Is EDGE COLORING solvable in  $O^*(1.9999^m)$  time, or at least in  $O^*(2^{m-n})$  time? Establishing any nontrivial lower bounds for EDGE COLORING would also be very interesting. The reductions from [21] imply that solving EDGE COLORING in graphs with maximum degree  $\Delta = 3$  requires  $2^{\Omega(n)}$  time, assuming the Exponential Time Hypothesis (ETH), a standard hypothesis from fine-grained complexity. Does ETH similarly imply a  $2^{\Omega(m)}$  or even a  $2^{\Omega(n \log n)}$  time lower bound for EDGE COLORING in dense graphs (this question was previously raised in [25, Open problem 4.25])?

---

## References

- 1 Vladimir I Arnautov. Estimation of the exterior stability number of a graph by means of the minimal degree of the vertices. *Prikl. Mat. i Programirovanie*, 11(3-8):126, 1974.
- 2 Sepehr Assadi, Soheil Behnezhad, Sayan Bhattacharya, Martín Costa, Shay Solomon, and Tianyi Zhang. Vizing’s theorem in near-linear time, 2024. [arXiv:2410.05240](https://arxiv.org/abs/2410.05240).
- 3 Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Distributed edge coloring in time polylogarithmic in  $\Delta$ . In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, PODC ’22, pages 15–25. ACM, July 2022. doi:10.1145/3519270.3538440.
- 4 Lowell W Beineke. Derived graphs and digraphs. *Beiträge zur graphentheorie*, pages 17–33, 1968.
- 5 Anton Bernshteyn. A fast distributed algorithm for  $(\Delta + 1)$ -edge-coloring. *Journal of Combinatorial Theory, Series B*, 152:319–352, January 2022. doi:10.1016/j.jctb.2021.10.004.
- 6 Sayan Bhattacharya, Martín Costa, Nadav Panski, and Shay Solomon. Arboricity-Dependent Algorithms for Edge Coloring. In *Proceedings of the 19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, volume 294 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SWAT.2024.12.
- 7 Andreas Björklund. Determinant sums for undirected Hamiltonicity. *SIAM Journal on Computing*, 43(1):280–299, 2014. doi:10.1137/110839229.
- 8 Andreas Björklund and Thore Husfeldt. Exact algorithms for exact satisfiability and number of perfect matchings. *Algorithmica*, 52(2):226–249, December 2007. doi:10.1007/s00453-007-9149-8.
- 9 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Trimmed moebius inversion and graphs of bounded degree. *Theory of Computing Systems*, 47(3):637–654, January 2009. doi:10.1007/s00224-009-9185-7.
- 10 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *Journal of Computer and System Sciences*, 87:119–139, 2017. doi:10.1016/j.jcss.2017.03.003.
- 11 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, 2009. doi:10.1137/070683933.
- 12 M Blank. An estimate of the external stability number of a graph without suspended vertices. *Prikl. Mat. i Programirovanie*, 10:3–11, 1973.
- 13 Jesper Makhholm Byskov. Enumerating maximal independent sets with applications to graph colouring. *Operations Research Letters*, 32(6):547–556, November 2004. doi:10.1016/j.orl.2004.03.002.
- 14 Aleksander B. G. Christiansen, Eva Rotenberg, and Juliette Vlieghe. Sparsity-Parameterised Dynamic Edge Colouring. In *Proceedings of the 19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, volume 294 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:18, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SWAT.2024.20.

- 15 Ran Duan, Haoqing He, and Tianyi Zhang. *Dynamic Edge Coloring with Improved Approximation*, pages 1937–1945. Society for Industrial and Applied Mathematics, January 2019. doi:10.1137/1.9781611975482.117.
- 16 Eduard Eiben, Tomohiro Koana, and Magnus Wahlström. Determinantal sieving. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA 2024)*, pages 377–423. SIAM, 2024. doi:10.1137/1.9781611977912.16.
- 17 David Eppstein. *Small Maximal Independent Sets and Faster Exact Graph Coloring*, pages 462–470. Springer Berlin Heidelberg, 2001. doi:10.1007/3-540-44634-6\_42.
- 18 Alexander Golovnev, Alexander S. Kulikov, and Ivan Mihajlin. Families with infants: Speeding up algorithms for NP-hard problems using FFT. *ACM Trans. Algorithms*, 12(3):35:1–35:17, 2016. doi:10.1145/2847419.
- 19 David G. Harris. Distributed local approximation algorithms for maximum matching in graphs and hypergraphs. In *Proceedings of the 60th Annual Symposium on Foundations of Computer Science (FOCS 2019)*, pages 700–724, 2019. doi:10.1109/FOCS.2019.00048.
- 20 Michael A. Henning. Bounds on domination parameters in graphs: a brief survey. *Discuss. Math. Graph Theory*, 42(3):665–708, 2022. doi:10.7151/DMGT.2454.
- 21 Ian Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10(4):718–720, 1981. doi:10.1137/0210055.
- 22 Masao Ishikawa and Masato Wakayama. Minor summation formula of pfaffians. *Linear and Multilinear algebra*, 39(3):285–305, 1995.
- 23 Tomohiro Koana and Magnus Wahlström. Faster algorithms on linear delta-matroids. *arXiv preprint arXiv:2402.11596*, 2024. doi:10.48550/arXiv.2402.11596.
- 24 Eugene L. Lawler. A note on the complexity of the chromatic number problem. *Inf. Process. Lett.*, 5(3):66–67, 1976. doi:10.1016/0020-0190(76)90065-X.
- 25 Moshe Lewenstein, Seth Pettie, and Virginia Vassilevska Williams. Structure and hardness in P (dagstuhl seminar 16451). *Dagstuhl Reports*, 6(11):1–34, 2016. doi:10.4230/DAGREP.6.11.1.
- 26 Daniel Lokshantov, Pranabendu Misra, Fahad Panolan, and Saket Saurabh. Deterministic truncation of linear matroids. *ACM Transactions on Algorithms*, 14(2):14:1–14:20, 2018. doi:10.1145/3170444.
- 27 Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, September 1994. doi:10.1145/185675.306789.
- 28 Dániel Marx. A parameterized view on matroid optimization problems. *Theoretical Computer Science*, 410(44):4471–4479, 2009. doi:10.1016/j.tcs.2009.07.027.
- 29 William McCuaig and Bruce Shepherd. Domination in graphs with minimum degree two. *Journal of Graph Theory*, 13(6):749–762, 1989. doi:10.1002/JGT.3190130610.
- 30 Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, August 1995. doi:10.1017/cbo9780511814075.
- 31 Kazuo Murota. *Matrices and matroids for systems analysis*, volume 20. Springer Science & Business Media, 1999.
- 32 Oystein Ore. Theory of graphs. In *Colloquium Publications*. American Mathematical Society, 1962.
- 33 Charles Payan. Sur le nombre d’absorption d’un graphe simple, 1975.
- 34 Bruce A. Reed. Paths, stars and the number three. *Comb. Probab. Comput.*, 5:277–295, 1996. doi:10.1017/S0963548300002042.
- 35 Magnus Wahlström. Abusing the Tutte matrix: An algebraic instance compression for the K-set-cycle problem. In *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, volume 20 of *LIPICs*, pages 341–352. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPICs.STACS.2013.341.
- 36 Ryan Williams. Finding paths of length  $k$  in  $O^*(2^k)$  time. *Information Processing Letter*, 109(6):315–318, 2009. doi:10.1016/J.IPL.2008.11.004.