Electronic Colloquium on Computational Complexity, Report No. 128 (2022)

PPP-Completeness and Extremal Combinatorics*

Romain Bourneuf[1], Lukáš Folwarczný[†2,3], Pavel Hubáček[‡2], Alon Rosen[§4], and Nikolaj I. Schwartzbach[5]

[1]ENS de Lyon
[2]Charles University, Faculty of Mathematics and Physics
[3]Institute of Mathematics, Czech Academy of Sciences
[4]Bocconi University and Reichman University
[5]Aarhus University

Abstract

Many classical theorems in combinatorics establish the emergence of substructures within sufficiently large collections of objects. Well-known examples are Ramsey's theorem on monochromatic subgraphs and the Erdős-Rado sunflower lemma. Implicit versions of the corresponding total search problems are known to be PWPP-hard; here "implicit" means that the collection is represented by a poly-sized circuit inducing an exponentially large number of objects.

We show that several other well-known theorems from extremal combinatorics – including Erdős-Ko-Rado, Sperner, and Cayley's formula – give rise to *complete* problems for PWPP and PPP. This is in contrast to the Ramsey and Erdős-Rado problems, for which establishing inclusion in PWPP has remained elusive. Besides significantly expanding the set of problems that are complete for PWPP and PPP, our work identifies some key properties of combinatorial proofs of existence that can give rise to completeness for these classes.

Our completeness results rely on efficient encodings for which finding collisions allows extracting the desired substructure. These encodings are made possible by the tightness of the bounds for the problems at hand (tighter than what is known for Ramsey's theorem and the sunflower lemma). Previous techniques for proving bounds in TFNP invariably made use of structured algorithms. Such algorithms are not known to exist for the theorems considered in this work, as their proofs "from the book" are non-constructive.

[*] Part of this work done while visiting R.B., L.F., P.H., and N.I.S. were visiting Bocconi University.

[†] Supported by the Grant Agency of the Czech Republic under the grant agreement no. 19-27871X and by the Charles University grant SVV–2020–260578.

[‡] Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019547), by the Cariplo CRYPTONOMEX grant, by the Grant Agency of the Czech Republic under the grant agreement no. 19-27871X, and by the Charles University project UNCE/SCI/004.

[§] Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019547) and Cariplo CRYPTONOMEX grant.

ISSN 1433-8092

Contents

1 Introduction

A well-known theorem by Ramsey gives a lower bound on the size of the largest monochromatic clique in any edge-coloring of the complete graph using two colors.

Ramsey [Ram30] *Any edge-coloring of the complete graph on n vertices with two colors contains a monochromatic clique of size at least $\frac{1}{2}\log n$.*

Ramsey's theorem gives rise to a natural computational search problem Ramsey [Kra05, KNY19]: given a description of an edge-coloring, output the vertices of a monochromatic clique of size $\frac{1}{2}\log n$. Since the theorem guarantees the existence of a monochromatic clique of this size, Ramsey belongs to the complexity class TFNP consisting of efficiently verifiable search problems to which a solution is guaranteed to exist [MP91].

The computational complexity of Ramsey very much depends on its representation. One the one hand, it is efficiently solvable when the graph is given *explicitly*; a folklore proof of Ramsey's theorem gives an efficient algorithm to find such a subgraph – see Appendix A. On the other hand, the situation is less clear when the graph is represented *implicitly*, e.g., via a Boolean circuit that, for any pair of vertices, outputs the corresponding color of the edge-coloring of the graph.[1]

Another TFNP problem considered in the literature that is motivated by a result in extremal combinatorics arises from the well-known Erdős-Rado sunflower lemma.

Erdős-Rado [ER60] *Any family of n-sets of cardinality greater than $n^n n!$ contains an n-sunflower of size $n+1$, i.e., subsets $A_1, A_2, \ldots, A_{n+1} \in \mathcal{F}$ such that, for some Δ, $A_i \cap A_j = \Delta$ for every distinct A_i, A_j.*

An instance of the total search problem Sunflower [KNY19] can be implicitly represented, e.g., via a Boolean circuit that, given an index of a set in the family, outputs its characteristic vector.

In general, little is known of the complexity of the implicit variants of Ramsey or Sunflower – the proofs of the corresponding theorems are either non-constructive or result in inefficient (i.e., superpolynomial-time) algorithms. Both problems are known to be PWPP-hard, as shown by Krajíček [Kra05] and Komargodski, Naor, and Yogev [KNY19]. This means that finding the desired substructure is at least as hard as finding collisions in an arbitrary poly-sized shrinking circuit and, hence, hard in the worst-case if collision-resistant hash functions exist. However, they are not known to be complete for the class PWPP and the intriguing question of whether they give rise to a complexity class distinct from PWPP has remained open for years.

1.1 Our Results

We explore new connections between classical theorems in extremal combinatorics and the complexity classes PPP [Pap94] and PWPP [Jeř16], i.e., the classes of search problems with totality guaranteed by the (weak) pigeonhole principle. We show that PPP and PWPP can be characterized via a number of new TFNP problems based on the following theorems.

Erdős-Ko-Rado [EKR61]. *Any family of distinct pairwise-intersecting k-sets on a universe of size m has size at most $\binom{m-1}{k-1}$.*

Sperner [Spe28]. *The largest antichain, i.e., a family of subsets such that no member is contained in any other, on a universe with $2n$ elements is unique and consists of all subsets of size n.*

Cayley [Cay89]. *There are exactly n^{n-2} spanning trees of the complete graph on n vertices.*

Just as for Ramsey and Sunflower, the corresponding search problems are efficiently solvable when given explicit access to the family of objects and, again, their computational complexity is open when we consider implicit access to the structure, e.g., where the instance is given by a circuit that on input i returns

[1] Given such a representation, it might be even hard to compute the degree of a node with respect to one of the two colors.

an encoding of the i^{th} object in the collection.[2] The totality of the problems we define follows from a common principle – the instances are given via an implicit representation of a sufficiently large collection of objects (e.g., subsets for Erdős-Ko-Rado) such that, by the corresponding theorem, there exists a small subset of these objects satisfying some efficiently verifiable property (e.g, a pair of disjoint subsets for Erdős-Ko-Rado).

In addition to the above completeness results, we define TFNP problems arising from the following two results in extremal combinatorics.

Mantel [Man07]. *Any triangle-free graph on n vertices has at most $n^2/4$ edges.*

Ward-Szabo [WS95]. *Any edge-coloring of the complete graph on n vertices with $2 \leq r \leq \sqrt{n}$ colors must contain a bichromatic triangle.*

We show that variants of the corresponding problems are hard for PWPP and PPP. However, proving their inclusion in PWPP or PPP remains open and they join Ramsey and Sunflower as candidate problems that might define a new class above PWPP or PPP (see Section 1.5). An overview of our results in terms of weak and strong problems (see Section 1.3) is given in Table 1.

Problem	**Hardness**	**Containment**
Ramsey	PWPP [Kra05, KNY19]	TFNP
Sunflower		
Ward-Szabo	PWPP [Theorems 7.4, 8.3 and 8.12]	
weak-Mantel		PPP [Theorems 7.7, 8.4 and 8.13]
weak-Turán$_r$		
Ward-Szabo-Colorful-Collisions		
Ward-Szabo-Collisions	PWPP [Theorems 4.4, 4.21, 5.2, 6.2, 7.4 and 7.5]	
weak-Erdős-Ko-Rado		
weak-general-Erdős-Ko-Rado$_k$		
weak-Sperner-Antichain		
weak-Cayley		
Erdős-Ko-Rado	PPP [Theorems 4.9, 4.26, 5.7 and 6.7]	
general-Erdős-Ko-Rado$_k$		
Sperner-Antichain		
Cayley		
Mantel	PPP [Theorem 8.8]	TFNP

Table 1: Summary of the complexity of problems we consider. Except for Ramsey and Sunflower, all problems were introduced in this work. The containment results for weak-general-Erdős-Ko-Rado$_k$ and general-Erdős-Ko-Rado$_k$ rely on the efficient Baranyai assumption (Assumption 4.18).

[2] Note that an implicit representation of the collection might not necessarily satisfy the assumptions of the underlying theorem. For instance, representing sets via characteristic vectors for Erdős-Ko-Rado does not ensure that they are actually k-sets or that they are distinct. Importantly, such a violation could allow evading the totality of the search problem. Nevertheless, we can ensure totality by allowing locally verifiable evidence of a malformed representation as a solution, e.g., an index not corresponding to a k-set or two indices corresponding to the same set.

1.2 Techniques and Ideas

A long-standing open problem regarding Ramsey and Sunflower has been to determine their status with respect to the classes PWPP and PPP. For the most part, the most challenging part in establishing completeness for some syntactic subclass of TFNP lies in proving hardness (see, e.g., [DGP09, Meh18, FG18]). For subclasses of TFNP such as PPAD, PPA, and PLS, the inclusion in a subclass mostly follows from the existence of an inefficient yet structured algorithm for the problem at hand; for example, the chessplayer algorithm for PPA [Pap94] or the steepest descent algorithm for PLS [JPY88]. However, this methodology seems inapplicable for proving inclusion in PWPP or PPP as these classes do not exhibit any characterizing graph-theoretic structure that could capture some class of natural algorithms.

In contrast to many existing bounds in TFNP, our work does not make use of structured algorithms but instead makes use of encodings that translate between substructures and collisions in circuits. In order to establish inclusion in PWPP, we encode the objects of the collection using a "property-preserving encoding" that encodes the objects in a way that translates some specific relation into collisions. More precisely, we want an encoding function that is efficiently computable and (nearly) optimal, such that whenever two elements have the same encoding, these two elements give a solution to the original problem. While this technique is quite general, it is not always clear how to instantiate the encoding to get the desired collisions.

Consider, for example, the total search problem corresponding to the Erdős-Ko-Rado theorem for intersecting families of n-sets on a universe of size $2n$. An instance can be given by a Boolean circuit $C\colon \{0,1\}^{\lceil \log(\binom{2n-1}{n-1}) \rceil + 1} \to \{0,1\}^{2n}$ representing a family of subsets of $[2n]$, i.e., $C(i)$ is the characteristic vector of the i-th n-set in the family. Suppose the outputs of C define distinct n-sets. Since there are more than $\binom{2n-1}{n-1}$ of them, then, by the Erdős-Ko-Rado theorem, there must exist a pair of inputs mapped to disjoint n-sets by C. We define any such pair of inputs to be a solution.[3]

When proving that the above total search problem is contained in the complexity class PWPP, at a high level, we want to encode the n-sets of the family using a shrinking circuit, in such a way that collisions correspond to disjoint sets. Observe that for n-sets in a universe of size $2n$, the only disjoint sets are complements and, hence, we get an equivalent instance of the problem if we map each set to either itself or its complement, arbitrarily. In our construction, we map each set S to the representative not containing the element 1. That is, if $1 \notin S$, the set is left unchanged and, otherwise, it is mapped to its complement $\overline{S}$. Note that, by the pigeonhole principle, two sets that do not contain 1 must have a non-empty intersection since we work with n-subsets of $[2n]$. To obtain a shrinking circuit, we make use of Cover encodings (Section 3.1) that give an optimal encoding of all n-sets by considering their lexicographic order. Notice that if the input S is not an n-set, we may map it arbitrarily to any n-set, as a collision, in this case, yields a solution to the instance of the above problem motivated by the Erdős-Ko-Rado theorem.

In contrast, the PWPP-hardness results for Ramsey and Sunflower follow an extremely elegant but rather direct (compared to other hardness results for subclasses of TFNP) technique of graph-hash product [Kra05, KNY19], which we illustrate on Ramsey. Recall that there are known randomized constructions of edge-colorings of the complete graph $K_{2^{n/4}}$ on $2^{n/4}$ vertices that do not contain a monochromatic clique of size $n/2$ [Erd47]. Given such an underlying edge-coloring of $K_{2^{n/4}}$ and a hash function h mapping n-bit strings to $n/4$-bit strings, one can construct an edge-coloring of the complete graph on 2^n vertices by assigning to every edge $(u, v) \in \{0,1\}^n \times \{0,1\}^n$ the color of the edge $(h(u), h(v)) \in \{0,1\}^{n/4} \times \{0,1\}^{n/4}$ from the underlying coloring. Since the underlying edge-coloring of $K_{2^{n/4}}$ does not contain a monochromatic clique of size $n/2$, it is easy to see that any monochromatic clique of size $n/2$ in the resulting edge-coloring of K_{2^n} (guaranteed to exist by Ramsey's theorem) must have been introduced via a collision in the hash h.

As noted by [KNY19], the structure of a PWPP-hardness proof using the graph-hash product is not restricted to total search problems corresponding to graph-theoretic theorems of existence; indeed, [KNY19] used the graph-hash product to prove also PWPP-hardness of Sunflower. On a high level, for a problem to be amenable to the graph-hash product technique, it is sufficient to be able to construct a collection of objects such that 1) it does not contain the desired substructure, 2) its size is at least a constant fraction of

[3] To ensure the totality of the problem, we introduce additional solutions corresponding to succinct certificates that C does not define a family of distinct n-sets, i.e., either an i such that $C(i)$ is not of Hamming weight n or a pair $i \neq j$ such that $C(i) = C(j)$.

the threshold necessary for the existential theorem to apply,[4] and 3) it can be efficiently indexed. Then, we can interpret the output of an appropriately shrinking hash h as an index into the small collection of objects, and, for each index, we can efficiently compute and output the corresponding element in the collection. Again, since the small collection does not contain the desired substructure, all solutions of the instance constructed via graph-hash product must in some way result from a collision in the hash h.

For example, consider the total search problem arising from Sperner's theorem on antichains – here, the threshold size is $\binom{2n}{n}$, meaning that if we have a family with strictly more than $\binom{2n}{n}$ distinct subsets of $[2n]$ then one subset from the family must be contained in another member of the family. It is straightforward to construct a family of subsets that does not contain the specific substructure (i.e., a subset that is included in another one) with size equal to the threshold size $\binom{2n}{n}$. It suffices to consider the family of all the n-subsets of $[2n]$. Similarly, for many other combinatorial problems we study, an adequate collection of objects can be found by looking at a collection of maximum size that does not contain the substructure.

We also show natural reductions between some of the problems we define (from Erdős-Ko-Rado to Sperner-Antichain for instance), which, in our opinion, highlights the relevance of these new problems and the fact that their definition is the correct one.

1.3 PPP-Completeness From Extremal Combinatorics

Up to this point, our discussion did not explicitly distinguish between the classes PWPP and PPP. However, our work highlights important structural differences between the two complexity classes. Recall that the class PWPP contains the search problems in TFNP whose totality can be proved using the *weak* pigeonhole principle: "In any assignment of $2n$ pigeons to n holes there must be two pigeons sharing the same hole."

This statement can be seen as a result in extremal combinatorics bounding the maximum number of pigeons that can be assigned to n holes without two pigeons being sent to the same hole. More generally, we say that a theorem from extremal combinatorics is "weak" if it gives an upper bound (which may or may not be tight) on the maximum size of a collection of objects that does not contain some substructure (above, two pigeons sharing the same hole). On the contrary, we say that a theorem from extremal combinatorics is "strong" if it gives a tight upper bound on the maximum size of a collection of objects that does not contain some substructure, as well as some structural property about the maximum families without the substructure. For instance, the strong pigeonhole principle can be stated as: "In any assignment of n pigeons to n holes there is either a pigeon in the first hole or two pigeons sharing the same hole." Note that it is exactly this formulation of the strong pigeonhole principle that defines the class PPP.

Many results in extremal combinatorics have a weak statement and a strong statement. For such results, we can define a problem corresponding to the weak statement, which often is related to PWPP, and a problem corresponding to the strong statement, which often is related to PPP. In this paper, all PWPP-hard problems correspond to a weak theorem in extremal combinatorics, while PPP-hard problems correspond to a strong theorems in extremal combinatorics. As an example, consider Cayley's formula and note that the bound n^{n-2} is tight. Hence, if we are given a collection of exactly n^{n-2} distinct graphs on n vertices, then either one of the graphs is not a spanning tree, or every spanning tree is in the collection. This observation induces a TFNP problem that we show to be PPP-complete.

1.4 Related Work

Compared to the majority of subclasses of TFNP that have been extensively studied and are known to capture various total search problems from diverse domains of mathematics, PPP and PWPP might seem less expressive and the first non-trivial completeness results appeared only recently.

Sotiraki, Zampetakis, and Zirdelis [SZZ18] and Ban, Jain, Papadimitriou, Psomas, and Rubinstein [BJP+19] demonstrated that PPP contains computational problems from number theory and the theory of integral

[4] This is a technical condition ensuring that we can reduce from a PWPP-complete variant of the problem of finding collisions in a shrinking hash. Note that it is easy to find collisions in functions that exhibit extreme shrinking.

lattices. In particular, Sotiraki et al. showed PPP-completeness of a computational problem related to Blitchfeld's theorem and PPP-completeness (resp. PWPP-completeness) of a problem motivated by the Short Integer Solution problem. Hubáček and Václavek [HV21] showed that some general formalizations of the discrete logarithm problem are complete for PWPP and PPP and, motivated by classical constructions of collision-resistant hashing, they characterized PWPP via the problem of breaking claw-free (pseudo-)permutations.

1.5 Open Problems

Our work suggests various interesting directions for future research:

- We exploit the power of strong statements in extremal combinatorics for establishing PPP-completeness. The notorious lack of tight bounds for the Erdős-Rado sunflower lemma and Ramsey's theorem implies that we have no strong version of these theorems, which may explain why showing the inclusion of the corresponding problems in, e.g., PPP has eluded researchers.

- We introduced total search problems corresponding to Mantel's theorem, Turán's theorem, and Ward-Szabo's theorem. In this work, we only prove hardness results for these problems but no inclusion results. Hence, it is still open whether they are complete for the classes PPP and PWPP, or whether they could define a new subclass of TFNP.

- The $\textsc{Turán}_r$ problem is defined in a similar fashion to $\textsc{Mantel}$, yet, unlike for $\textsc{Mantel}$, we currently do not have a proof of PPP-hardness for it. Thus, the question of PPP-hardness of $\textsc{Turán}_r$ is immediate. Alternatively, it would be interesting to define a different PPP-hard problem in a natural way from Turán's theorem.

- Another exciting question is whether the efficient Baranyai assumption (Assumption 4.18) holds, as well as whether it is possible to prove the inclusion results of the problems associated to the general version of Erdős-Ko-Rado's theorem *without* that assumption. Showing reductions between $\textsc{general-Erdős-Ko-Rado}_k$ and $\textsc{general-Erdős-Ko-Rado}_l$ for $k \neq l$ without the efficient Baranyai assumption would also be intriguing.

- Finally, we believe the problems $\textsc{General-Pigeon}_k^m$ deserve a more thorough investigation to further our understanding of the classes they define and their interrelation.

2 Preliminaries

We denote by $\log x$ the binary logarithm of x. We denote by $[n]$ the set $\{1, 2, 3, \ldots, n-1, n\}$. We interpret elements of $\{0,1\}^*$ as strings and write them as $x = x_1 x_2 \cdots x_n$ for $x_i \in \{0,1\}$. Each element x_i is also called a bit. We say n is the length of $x \in \{0,1\}^n$, and say x is an n-bit string. We denote by 0^n (resp. 1^n) the n-bit string consisting of all 0 (resp. 1). If $x, y \in \{0,1\}^*$ are two strings of lengths n, m, respectively, we denote by $x \parallel y = x_1 x_2 \cdots x_n y_1 y_2 \cdots y_m$ the *concatenation* of x and y. We denote by $\leq$ the lexicographical order on strings. Note that $\leq$ is a partial order as it is only well-defined for strings of the same length. We use $x < y$ to denote $x \leq y$ and $x \neq y$. We may occasionally abuse notation and write $x < k$ where $k \in \mathbb{N}$, in which case we mean the binary encoding of k on the same number of bits as x. If $\lceil \log k \rceil$ exceeds the length of x, we define $x < k$ such that the order is total.

If Ω is a set of size n, we associate the set 2^Ω with the characteristic vectors from $\{0,1\}^n$ for some arbitrary (but fixed) order on Ω. We denote by $\subseteq$ the partial order on $\{0,1\}^n$ where $x \subseteq y$ iff $x_i \leq y_i$ for every $i = 1 \ldots n$. If $x \in \{0,1\}^n$ is a string, we denote by $\overline{x} := \overline{x}_1 \overline{x}_2 \cdots \overline{x}_n$ the *complement of* x, defined by $\overline{x}_i = 1 - x_i$. We also use other set-theoretic operators $\cap, \cup, \setminus$ that are defined in a natural way. We also denote by $|x| = \sum_{i=1}^n x_i$ the number of 1s in x when the length is implicit from the context.

2.1 Total Search Problems

A search problem is defined by a binary relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ – a string $s \in \{0,1\}^*$ is a *solution* for an *instance* $x \in \{0,1\}^*$ if $(x,s) \in R$. A search problem defined by relation R is *total* if for every x, there exists an s such that $(x,s) \in R$. We define TFNP as the class of all total search problems that can be efficiently verified, i.e., there is a deterministic polynomial-time Turing machine that, given (x,s), outputs 1 if and only if $(x,s) \in R$ and, for every instance x, there exists a solution s of polynomial length in the size of x.

To avoid unnecessarily cumbersome phrasing throughout the paper, we define TFNP relations implicitly by presenting the set of valid *instances* $X \subseteq \{0,1\}^*$ recognizable in polynomial time (in the length of an instance) and, for each instance $i \in X$, the set of admissible *solutions* $Y_i \subseteq \{0,1\}^*$ for the instance i. It is then implicitly assumed that, for any invalid instance $i \in \{0,1\}^* \setminus X$, we define the corresponding solution set as $Y_i = \{0,1\}^*$.

Next, we recall the definitions of the complexity classes PWPP and PPP via their canonical complete problems WEAK-PIGEON and PIGEON.

Definition 2.1 (WEAK-PIGEON and PWPP [Jeř16])**.** *The problem* WEAK-PIGEON *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^n \to \{0,1\}^{n-1}$.

Solution: $x_1 \neq x_2$ *s.t.* $C(x_1) = C(x_2)$.

The class of all TFNP problems reducible to WEAK-PIGEON *is called PWPP.*

Definition 2.2 (PIGEON and PPP [Pap94])**.** *The problem* PIGEON *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^n \to \{0,1\}^n$.

Solution: *One of the following:*

i) x *s.t.* $C(x) = 0^n$,

ii) $x \neq y$ *s.t.* $C(x) = C(y)$.

The class of all TFNP problems reducible to PIGEON *is called PPP.*

3 Property-Preserving Encodings

A key ingredient to our proofs of inclusion in PWPP and PPP is the use of efficient encodings. We rely on two different types of encodings. The first one simply consists of bijections between two different representations of the same set of objects, the first one being more natural and more convenient to work with, and the second one being more concise. The second type of encodings, which we call property-preserving encodings, consists of shrinking functions, in the sense that the range of the encoding is smaller than the domain, whose collisions exactly correspond to elements sharing some property. The following definition gives a precise description of the features we require from these encodings.

Definition 3.1 (Property-preserving encoding)**.** *Let* $\mathcal{X} \subseteq \{0,1\}^k, \mathcal{Y}$ *be sets, and let* $\sim$ *be an equivalence relation on* $\mathcal{X}$*. Let* $E : \{0,1\}^k \to \mathcal{Y}$ *be a surjection. We say that* E *constitutes a* property-preserving encoding for $\sim$ on $\mathcal{X}$ *if it satisfies.*

- (Efficiency)*.* *E can be computed in polynomial time.*
- (Compression)*.* $|\mathcal{Y}| \leq |\mathcal{X}|$.
- ($\sim$-correctness)*.* *E is constant on every coset of $\mathcal{X}$ for $\sim$.*

We first describe some bijective encodings before studying some property-preserving encodings.

3.1 Cover Encodings

Our reductions in Section 4 make use of *Cover encodings* [Cov73] that efficiently encode subsets of a specified size in optimal space: namely, we may encode every subset $S \subseteq \{0,1\}^m$ such that $|S| = k$ by considering the lexicographic order of all $\binom{m}{k}$ such sets (in fact we consider the lexicographic order over their characteristic vectors $\in \{0,1\}^m$), and mapping this into binary strings: this requires $\lceil \log \binom{m}{k} \rceil$ bits, which is optimal. We denote the encoding and decoding functions as follows, with $\alpha(k,m) = \lceil \log \binom{m}{k} \rceil$.

$$E^{k,m}_{\mathsf{Cover}} : \{0,1\}^m \to \{0,1\}^{\alpha(k,m)}$$
$$D^{k,m}_{\mathsf{Cover}} : \{0,1\}^{\alpha(k,m)} \to \{0,1\}^m$$

We set $E_{\mathsf{Cover}} = E^{n,2n}_{\mathsf{Cover}}$ and $D_{\mathsf{Cover}} = D^{n,2n}_{\mathsf{Cover}}$, and $\alpha = \alpha(n,2n)$. As described in [Cov73], these functions can be made efficient.

Lemma 3.2. *For every $k \leq m$, $D^{k,m}_{\mathsf{Cover}} \circ E^{k,m}_{\mathsf{Cover}}$ is the identity over all k-subsets of $\{0,1\}^m$. Similarly, $E^{k,m}_{\mathsf{Cover}} \circ D^{k,m}_{\mathsf{Cover}}$ is the identity over the first $\binom{k}{m}$ elements in the lexicographic order of $\{0,1\}^{\alpha(k,m)}$.*

Note that the behavior of $D^{k,m}_{\mathsf{Cover}}$ is undefined for the last $2^{\alpha(k,m)} - \binom{m}{k}$ inputs. Furthermore, by design, $E^{k,m}_{\mathsf{Cover}}$ is well-defined on any subset of $[m]$ (even if this subset does not have size k), but the encoding only makes sense for subsets of size k. We also note the following identity which will be useful later when dealing with n-subsets of $[2n]$.

$$D_{\mathsf{Cover}}(0^\alpha) = 0^n 1^n = \overline{[n]} \tag{1}$$

Remark 3.3. *When we encode n-subsets of $[2n]$, since we encode sets according to their rank of their characteristic vector in the lexicographic order, any set that does not contain element 1 is one of the $\binom{2n-1}{n-1} = \frac{1}{2}\binom{2n}{n} \leq 2^{\alpha-1}$ first ones in the lexicographic order, hence its encoding starts with a 0. Conversely, if we decode an element whose first two bits are 0's, this means that the corresponding n-subset of $[2n]$ is one of the first $2^{\alpha-2} \leq \binom{2n-1}{n-1}$ in the lexicographic order, hence that it does not contain the element 1.* ◇

3.2 Encoding 2-subsets of $[2^n]$

In Section 7, we need to encode the subsets of $[2^n]$ with 2 distinct elements in an injective way. Unfortunately, since the base set is large, we cannot use Cover encodings to do so. However, we can use the idea behind Cover encodings, that is to encode the subsets by their rank in the lexicographic order. Consider $(x,y) \in [2^n] \times [2^n]$, with $x < y$. What is its rank in the lexicographic order?

All subsets whose smallest element is smaller than x have a lower rank. The number of such subsets is

$$\begin{aligned}(2^n-1)+(2^n-2)+\ldots+(2^n-x+1) &= \sum_{j=2^n-x+1}^{2^n-1} j \\ &= \sum_{j=1}^{2^n-1} j - \sum_{j=1}^{2^n-x} j \\ &= \frac{2^n(2^n-1)}{2} - \frac{(2^n-x)(2^n-x+1)}{2}\end{aligned}$$

All subsets whose smallest element is x and whose second smallest element is smaller than y also have a lower rank. There are exactly $y-x-1$ such subsets.

Hence, the rank of the subset (x,y) in the lexicographic order is

$$\frac{2^n(2^n-1)}{2} - \frac{(2^n-x)(2^n-x+1)}{2} + y - x - 1$$

Note that since there are $\binom{2^n}{2} < 2^{2n-1}$ subsets of $[2^n]$ with 2 distinct elements, the rank of any subset (x,y)

with $x < y$ can be written in binary using $2n-1$ bits. Now, denote as $E_{lex} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n-1}$ the following circuit. On input (x,y), it proceeds as follows.

1. If $x = y$, it returns 0^{2n-1}.

2. If $x < y$, it computes and returns the binary encoding on $2n-1$ bits of $\frac{2^n(2^n-1)}{2} - \frac{(2^n-x)(2^n-x+1)}{2} + y - x - 1$.

3. If $x > y$, it computes and returns the binary encoding on $2n-1$ bits of $\frac{2^n(2^n-1)}{2} - \frac{(2^n-y)(2^n-y+1)}{2} + x - y - 1$.

Note that E_{lex} has polynomial size, and is injective on the set of subsets of $[2^n]$ with 2 distinct elements by construction.

Remark 3.4. *In fact, this encoding is a bijection from the set of 2-subsets of* $[2^n]$ *to the set* $[\binom{2^n}{2}]$*. The reciprocal of that bijection can also be computed by a circuit* D_{lex} *of polynomial size.*

3.3 Prüfer Codes

In Section 6, we make use of Prüfer codes [Pru18] that give an efficiently computable bijection between the set of labelled spanning trees on n vertices and the set of sequences of $n-2$ elements of $[n]$. They were originally used by Heinz Prüfer [Pru18] to prove Classical Theorem 5

We denote by $E_{\mathsf{Prüfer}}$ a circuit that efficiently computes the Prüfer encoding of a spanning tree described by an element of $\{0,1\}^{\binom{n}{2}}$. Similarly, let $D_{\mathsf{Prüfer}}$ be a circuit that efficiently computes the spanning tree associated with a Prüfer code. By looking at the algorithm to compute Prüfer encodings, it is clear that we can assume these circuits to have polynomial size. We also assume that $E_{\mathsf{Prüfer}}$ outputs elements of the right form even on inputs which do not correspond to spanning trees. Consider the lexicographic order on $[n]^{n-2}$. Let R be a circuit that efficiently computes the rank of an element of $[n]^{n-2}$, and let $\tilde{E}_{\mathsf{Prüfer}} = R \circ E_{\mathsf{Prüfer}}$. Given a spanning tree, $\tilde{E}_{\mathsf{Prüfer}}$ returns the rank of its Prüfer code in the lexicographic order.

Let R' be a circuit which on input x computes the sequence of $[n]^{n-2}$ whose rank in the lexicographic order is x. Let $\tilde{D}_{\mathsf{Prüfer}} = D_{\mathsf{Prüfer}} \circ R'$. Given a rank, $\tilde{D}_{\mathsf{Prüfer}}$ returns the spanning tree whose Prüfer code has the corresponding rank in the lexicographic order. Note that $\tilde{D}_{\mathsf{Prüfer}}$ and $\tilde{E}_{\mathsf{Prüfer}}$ both have polynomial size. Now, if $\beta = \lceil (n-2)\log(n) \rceil$, then $\tilde{E}_{\mathsf{Prüfer}} : \{0,1\}^{\binom{n}{2}} \to \{0,1\}^{\beta}$, $\tilde{D}_{\mathsf{Prüfer}} : \{0,1\}^{\beta} \to \{0,1\}^{\binom{n}{2}}$. By construction, we have the following.

Lemma 3.5. *The following statements are true.*

1. $\tilde{D}_{\mathsf{Prüfer}} \circ \tilde{E}_{\mathsf{Prüfer}}$ *is the identity over the set of labelled spanning trees on* n *vertices.*

2. $\tilde{E}_{\mathsf{Prüfer}} \circ \tilde{D}_{\mathsf{Prüfer}}$ *is the identity over the first* n^{n-2} *elements of* $\{0,1\}^{\beta}$*.*

Remark 3.6. *The behavior of* $\tilde{D}_{\mathsf{Prüfer}}$ *on its last* $2^{\beta} - n^{n-2}$ *inputs is undefined.*

Remark 3.7. *Let* T_1 *be the tree composed of the edges* $(1,2), (1,3), \dots, (1,n)$*. Then,* $\tilde{E}_{\mathsf{Prüfer}}(T_1) = 0^{\beta}$ *and* $\tilde{D}_{\mathsf{Prüfer}}(0^{\beta}) = T_1$*.* ◇

3.4 Catalan Factorization

Catalan factorization [EK99] is an encoding of subsets of $[2n]$ that allows us to decompose the partially ordered set $(2^{[2n]}, \subseteq)$ into $\binom{2n}{n}$ chains and to move efficiently within each chain to find a canonical representative, namely the only n-subset of the chain.

Let $x \in \{0,1\}^{2n}$ be a bitmap representing an element of $[2n]$. We introduce a new symbol z, and construct the Catalan factorization as follows. We temporarily record for each symbol whether or not it is underlined.

1. Underline the leftmost substring that starts with a non-underlined 1, followed by a (possibly empty) sequence of underlined symbols, and ends in a non-underlined 0. If no such substring exists, go to step 3.

2. Go to step 1.

3. Record the number k of non-underlined 1's.

4. Replace all non-underlined symbols in x with z, and let $x' \in \{0,1,z\}^{2n}$ be the resulting string (with underlinings removed).

5. Output (x', k).

We denote the output of the Catalan factorization as $E_{\mathsf{Catalan}}(x) \in \{0,1,z\}^{2n} \times [2n]$. We say $x' = \tilde{E}_{\mathsf{Catalan}}(x)$ is the *Catalan string* of x. If $x' \in \{0,1,z\}^{2n}$ and m is the number of z's in x', then for any $l \leq m$, we define $D_{\mathsf{Catalan}}(x', l)$ as the string obtained from x' by replacing the l last z's by 1 and the rest by 0.

Example 3.8. *Let $n = 4$ and let $x = 01101100$ be the string corresponding to the set $\{2,3,5,6\}$. Then, we construct the Catalan factorization by repeating step 1 to get the underlined version.*

$$01101100 \to 01\underline{10}1100 \to 01\underline{10}1\underline{10}0 \to 01\underline{10}\underline{1\underline{10}0}$$

We terminate as there are no non-underlined 0's with a 1 on its left. We record that there is $k = 1$ non-underlined 1. We then replace all non-underlined symbols with z to obtain the Catalan factorization.

$$(x', k) = (zz101100, 1)$$

Note that we have $D_{\mathsf{Catalan}}(x', k) = 01101100 = x$ so the encoding and decoding operations behave as expected. Note also that $D_{\mathsf{Catalan}}(x', 0) = 00101100$ corresponds to the set $\{3,5,6\}$ and $D_{\mathsf{Catalan}}(x', 2) = 11101100$ corresponds to the set $\{1,2,3,5,6\}$. For this reason, we say that the Catalan string x' identifies the following chain.

$$\{3,5,6\} \subset \{2,3,5,6\} \subset \{1,2,3,5,6\}$$

In that chain, k identifies that x is the 1^{st} element, counting from 0. ◇

Lemma 3.9. *$D_{\mathsf{Catalan}} \circ E_{\mathsf{Catalan}}$ acts as identity over $\{0,1\}^{2n}$.*

Proof. Let $x \in \{0,1\}^{2n}$, and $(x', k) = E_{\mathsf{Catalan}}(x)$ be its Catalan factorization. Let m be the number of z's in x. We claim that at the end of the underlining phase of the Catalan factorization of x, the entries that are not underlined are first $m - k$ 0's and then k 1's. Indeed, by definition, k of them are 1, so $m - k$ of them are 0. Furthermore, if we had a non-underlined 1 before a non-underlined 0, then we could consider the rightmost non-underlined 1 that is before a non-underlined 0. This 1 is followed by a sequence of underlined symbols and then by a non-underlined 0 so this 1 and the corresponding 0 should have been underlined. Thus, we indeed have that the entries that are not underlined are first $m - k$ 0's and then k 1's. These are the entries that are turned into z's when we go from x to x'.

Now, when we compute $D_{\mathsf{Catalan}}(x', k)$, we replace the last k z's in x' by 1's and the $m - k$ other ones by 0's, which is exactly what we had in x. Hence, $D_{\mathsf{Catalan}} \circ E_{\mathsf{Catalan}}(x) = D_{\mathsf{Catalan}}(x', k) = x$. □

We also denote by $D^{(l)}_{\mathsf{Catalan}} : \{0,1,z\}^{2n} \to \{0,1\}^{2n}$ the map $x' \mapsto D_{\mathsf{Catalan}}(x', l)$. If on input x', l is larger than the number of z symbols in x', all z symbols are be replaced with 1; this ensures the map is defined for all $l \geq 0$.

Lemma 3.10. *For every $l \geq 0$, $\tilde{E}_{\mathsf{Catalan}} \circ D^{(l)}_{\mathsf{Catalan}}$ acts as identity on the set of Catalan strings. That is, if x' is a Catalan string, then for every l, the Catalan string of $D^{(l)}_{\mathsf{Catalan}}(x')$ is x'.*

Proof. Let $x \in \{0,1\}^{2n}$ and let $x' = \tilde{E}_{\mathsf{Catalan}}(x)$ be the Catalan string of x. Now let $l \geq 0$, $y = D_{\mathsf{Catalan}}(x', l)$ and $y' = \tilde{E}_{\mathsf{Catalan}}(y)$ be the Catalan string of y. We want to show that $y' = x'$.

We proceed using induction on the steps of the algorithm. At first, no entries are underlined in either string. Next, suppose that after some number of steps, the underlined bits are exactly the same in x and in y. Now, consider two bits that get underlined in x at the next step. Then, all the bits between them are underlined in x at this point, so this is also the case in y by induction hypothesis. Furthermore, since these two bits get underlined in x, they are not turned into z's at the end of the algorithm, which means that they are still the same bits in x' and therefore in y. Hence, in y we have these 2 bits, first a 1 and then a 0, such that every entry between them is underlined, so they get underlined at this step.

Conversely, consider two bits that get underlined in y at the next step. Then, all the entries between them in y are underlined at this point, so it is the case in x too by induction hypothesis. By contradiction, suppose that the corresponding bits in x do not get underlined at this step. By the previous observation, it means that this pair of bits in x is not $(1, 0)$. There are three cases to consider:

1. In x, these two bits are 0's. Then, the first gets turned into a 1 in y, which means that it never gets underlined in x (otherwise it would remain the same). Then, since all the bits in x between these two are already underlined, and since the first never gets underlined, this means that the second never gets underlined (there will never be a non-underlined 1 before it such that all entries between them are underlined). Hence, these two bits never get underlined in the algorithm, and are finally turned into z's. Then, to go from x' to y, we replace the l last z's by 1's and the others by 0's, thus making it impossible for the first of these two bits to be turned into a 1 while the second is turned into a 0.

2. In x, these two bits are respectively 0 and 1. Then, both these bits are changed between x and y, which means that they never get underlined in x, hence they are z's in x'. Thus, like previously, it is impossible that the first one is turned into a 0 while the second is turned into a 1.

3. In x, these two bits are 1's. Then, the second bit gets turned into a 0 in y, which means that it never gets underlined in x. Like in the first case, we get that the first bit never gets underlined neither, once more making it impossible for these two bits to be turned respectively in 1 and 0.

In all three cases, we get a contradiction. Thus, the corresponding bits in x are also underlined at this step. Then, by induction, we get that at each step, the same bits are underlined in x and y. Finally, we turn all the bits that are not underlined into z's to get x' and y', hence $x' = y'$. □

Remark 3.11. *We can define an equivalence relation $\sim$ over the subsets of $[2n]$ by saying that two subsets are equivalent if and only if they have the same Catalan string.*
By combining Catalan factorization and Cover encodings, we can obtain a property-preserving encoding for $\sim$ on $\{0,1\}^{2n}$. We use this in Section 5.

4 Erdős-Ko-Rado Theorem on Intersecting Families

In this section, we define total search problems motivated by the well-known Erdős-Ko-Rado theorem on intersecting families and study their computational complexity. First, we present a PWPP-complete variant of the problem. Next, we modify the problem using a strong statement of the Erdős-Ko-Rado theorem to get a PPP-complete variant.

Recall the definition of an intersecting family and the statement of the Erdős-Ko-Rado theorem.

Definition 4.1 (Intersecting family)**.** *Let Ω be any set. A family of sets $\mathcal{F} \subseteq 2^{\Omega}$ is an* intersecting family *if no two sets are disjoint, i.e., if for any $A, B \in \mathcal{F}$, it holds that $A \cap B \neq \emptyset$.*

Classical Theorem 1 (Erdős-Ko-Rado [EKR61])**.** *Any intersecting family where each set has k elements on a universe of size m contains at most $\binom{m-1}{k-1}$ sets, and this bound is tight.*

We start by defining a total search problem motivated by a special case of the Erdős-Ko-Rado theorem for families of n-sets in a universe of size $2n$ presented in the following corollary.

Corollary 4.2. *Any intersecting family where each set has n elements on a universe of size $2n$ contains at most $\binom{2n-1}{n-1}$ sets, and this bound is tight. Furthermore, if $\mathcal{F}$ is an intersecting family of maximum size, then for every n-subset S, exactly one of S and $\overline{S}$ is in $\mathcal{F}$.*

Suppose that we have a collection, containing more than $\binom{2n-1}{n-1}$ sets of size n on $2n$ elements. Then, by Classical Theorem 1, there must be two sets that do not intersect. This induces a total search problem of finding two such disjoint sets. We consider an implicit representation of such a collection by a circuit C whose inputs serve as indices in the collection. The output of the circuit is a representation of the corresponding set as a characteristic vector of the $2n$ elements. Of course, this representation does not guarantee that C satisfies the conditions required for Classical Theorem 1 to apply, which would make the problem not total; in this case, we allow evidence of this fact to be a solution to the problem. Namely, if for a given input x, we do not have $|C(x)| = n$, or two distinct indices x, y represent the same set, i.e., $C(x) = C(y)$, we allow such inputs as solutions.

Definition 4.3 (weak-Erdős-Ko-Rado)**.** *The problem* weak-Erdős-Ko-Rado *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{\lceil \log(\binom{2n-1}{n-1}) \rceil + 1} \to \{0,1\}^{2n}$.

Solution: *One of the following:*

i) x *s.t.* $|C(x)| \neq n$,

ii) $x \neq y$ *s.t.* $C(x) = C(y)$,

iii) x, y *s.t.* $C(x) \cap C(y) = \emptyset$.

As we discussed in the introduction, the totality of this problem is proved using a "weak" statement in extremal combinatorics, namely the first part of Corollary 4.2, hence the name Weak. However, the analogy with weak-Pigeon goes further. Indeed, our first main theorem is the following.

Theorem 4.4. weak-Erdős-Ko-Rado *is PWPP-complete.*

Throughout this section, we maintain $\alpha = \lceil \log \binom{2n}{n} \rceil = \left\lceil \log \binom{2n-1}{n-1} \right\rceil + 1$.

Lemma 4.5. weak-Erdős-Ko-Rado $\in$ *PWPP.*

Proof. At a high level, we want to encode the sets using a shrinking circuit, in such a way that collisions correspond to disjoint sets. Observe that for n-sets in a universe of size $2n$, the only disjoint sets are complements, hence we get an equivalent instance of weak-Erdős-Ko-Rado if we map each set to either itself or its complement, arbitrarily. In our construction, we map each set S to the representative not containing 1. That is, if $1 \notin S$, the set is left unchanged and, otherwise, it is mapped to its complement $\overline{S}$. Note that by the pigeonhole principle, two sets that do not contain 1 must have a non-empty intersection since we work with n-subsets of $[2n]$. To obtain a shrinking circuit, we make use of Cover encodings (Section 3.1) that give an optimal encoding of all n-sets by considering their lexicographic order. Notice that if the input S is not an n-set, we may map it arbitrarily to any n-set, as a collision, in this case, yields a solution to the weak-Erdős-Ko-Rado instance.

Formally, recall that we have $E_{\mathsf{Cover}} : \{0,1\}^{2n} \to \{0,1\}^{\alpha}$ and $D_{\mathsf{Cover}} : \{0,1\}^{\alpha} \to \{0,1\}^{2n}$. Now let $C : \{0,1\}^{\alpha} \to \{0,1\}^{2n}$ be an instance of Erdős-Ko-Rado. We proceed to construct an instance $C' : \{0,1\}^{\alpha} \to \{0,1\}^{\alpha-1}$ of weak-Pigeon as follows:

$$C'(x) = \begin{cases} E_{\mathsf{Cover}}(C(x)) & \text{if } C(x)_1 = 0 \\ E_{\mathsf{Cover}}(\overline{C(x)}) & \text{if } C(x)_1 = 1 \end{cases}$$

Note that since we only encode sets whose first bit is a 0, by Remark 3.3, we get that the first bit of the encoding always is a 0, so we can consider only the $\lceil \log(\binom{2n}{n}) \rceil - 1 = \alpha - 1$ last bits of $C'(x)$ for every x, which is why we say that C' only outputs $\alpha - 1$ bits. Note also that if for some x, $C(x)$ does not have size n, then $E_{\mathsf{Cover}}(C(x))$ and $E_{\mathsf{Cover}}(\overline{C(x)})$ are still well-defined, even if they are meaningless.

Now, suppose that we have a solution to C', that is $x \neq y$ such that $C'(x) = C'(y)$. There are four cases to consider, depending on the first bits of $C(x), C(y)$. If $C(x)_1 = C(y)_1 = 0$, then $E_{\mathsf{Cover}}(C(x)) = C'(x) = C'(y) = E_{\mathsf{Cover}}(C(y))$. If both $C(x)$ and $C(y)$ have size n, then by injectivity of E_{Cover} on inputs of size n (see Lemma 3.2), we get $C(x) = C(y)$, which is a solution to WEAK-ERDŐS-KO-RADO. If one of them does not have size n, we also get a solution to WEAK-ERDŐS-KO-RADO. The other cases are similar. □

Remark 4.6. *Consider the circuit $E : \{0,1\}^{2n} \to \{0,1\}^{\alpha-1}$, defined as follows.*

$$E(x) = \begin{cases} 0^{\alpha-1} & \text{if } |x| \neq n \\ E_{\mathsf{Cover}}(x) & \text{if } x_1 = 0 \text{ and } |x| = n \\ E_{\mathsf{Cover}}(\overline{x}) & \text{if } x_1 = 1 \text{ and } |x| = n \end{cases}$$

Let $\mathcal{X} \subseteq \{0,1\}^{2n}$ be the subset of $\{0,1\}^{2n}$ corresponding to the n-subsets of $[2n]$. We define an equivalence relation $\sim$ on $\mathcal{X}$ by saying that two strings are equivalent if the corresponding subsets are either equal or disjoint. Note that this relation is transitive only because we work with n-subsets of $[2n]$.
Then, we have that E is a property-preserving encoding for $\sim$ on $\mathcal{X}$.
Furthermore, the property that is preserved by E is such that if two of its inputs collide, they form a solution to the problem we're interested in.
Then, to prove the inclusion of WEAK-ERDŐS-KO-RADO *into PWPP, it suffices to compose our instance of* WEAK-ERDŐS-KO-RADO *with E.*

Lemma 4.7. WEAK-ERDŐS-KO-RADO *is PWPP-hard.*

Proof. Our goal is for the ERDŐS-KO-RADO solver to find collisions in an instance C' of WEAK-PIGEON. We use a variation of the graph hash product [Kra05, KNY19]. The idea is to interpret the output of C' as an index into the collection of all n-sets that do not contain 1. We then use the Cover decoding function to obtain a representation of the corresponding set, and by correctness of the encoding, any such set must have exactly n elements – and all the sets intersect since they do not contain 1. Hence, the only solutions to the WEAK-ERDŐS-KO-RADO instance are collisions, that yield solutions to the original circuit C'.

Formally, let $C' : \{0,1\}^m \to \{0,1\}^{m-1}$ be an instance of WEAK-PIGEON. Let n be the minimal integer such that $2^{m+1} \leq \binom{2n}{n}$. Then, $m + 1 \leq \alpha$. We proceed to build a circuit $A : \{0,1\}^\alpha \to \{0,1\}^{\alpha-2}$ whose size is polynomial in m and such that from any collision in A we can efficiently find a collision in C'. Recall that we have $E_{\mathsf{Cover}} : \{0,1\}^{2n} \to \{0,1\}^\alpha$ and $D_{\mathsf{Cover}} : \{0,1\}^\alpha \to \{0,1\}^{2n}$. We define $C : \{0,1\}^\alpha \to \{0,1\}^{2n}$ by

$$C(x) = D_{\mathsf{Cover}}(00 \parallel A(x))$$

By Remark 3.3, since for every x, $(00 \parallel A(x))$ is one of the $\binom{2n-1}{n-1}$ first possible inputs, we have that the set $D_{\mathsf{Cover}}(00 \parallel A(x))$ is an n-subset of $[2n]$ which does not contain the element 1. We observe that C defines an instance of WEAK-ERDŐS-KO-RADO. Now suppose that we have a solution to this instance. By correctness of the decoding, we can only have solutions of type iii), that is $x \neq y$ such that $C(x) = C(y)$. By injectivity of D_{Cover} on its first $\binom{2n}{n}$ inputs (see Lemma 3.2), we get that $(00 \parallel A(x)) = (00 \parallel A(y))$ hence $A(x) = A(y)$ and from there we can retrieve a collision for C'. □

PPP-completeness using the tight bound We remark that Corollary 4.2 gives a *tight* upper bound on the size of the collection. Furthermore, we know some structure of any collection whose size is exactly one $\binom{2n-1}{n-1}$: it must either not be an intersecting family, or it must contain either $[n]$ or $\overline{[n]}$. This is an example of a "strong" theorem in extremal combinatorics. As discussed in the introduction, this observation allows us to modify the problem to be create a variant of WEAK-ERDŐS-KO-RADO that is to WEAK-ERDŐS-KO-RADO what PIGEON is to WEAK-PIGEON. The idea is to let C encode a collection whose size exactly matches the threshold. We then let C represent a collection of exactly $\binom{2n-1}{n-1}$ sets, and also allow preimages of $[n]$ and $\overline{[n]}$ as solutions. We show that modifying the problem in this manner makes it PPP-complete, thus strengthening the analogy with Pigeon. This technique is quite general, and we utilise it again in later sections.

Definition 4.8 (Erdős-Ko-Rado)**.** *The problem* Erdős-Ko-Rado *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{\lceil \log(\binom{2n-1}{n-1}) \rceil} \to \{0,1\}^{2n}$.

Solution: *One of the following:*

i) x *s.t.* $|C(x)| \neq n$ *and* $x < \binom{2n-1}{n-1}$,

ii) $x \neq y$ *s.t.* $C(x) = C(y)$ *and* $x, y < \binom{2n-1}{n-1}$,

iii) x, y *s.t.* $C(x) \cap C(y) = \emptyset$ *and* $x, y < \binom{2n-1}{n-1}$,

iv) x *s.t.* $C(x) = [n]$ *or* $\overline{[n]}$ *and* $x < \binom{2n-1}{n-1}$.

Theorem 4.9. Erdős-Ko-Rado *is PPP-complete.*

Lemma 4.10. Erdős-Ko-Rado *is PPP-hard.*

Proof. This proof is similar in spirit to that of Lemma 4.7, except for some minor changes. The first one is that the instance of Pigeon might be a permutation, and thus not have collisions. We then need to be able to find the preimage of 0. This is done by solutions of type *iv*). The second one is that we only look at the first $\binom{2n-1}{n-1}$ inputs of the Pigeon instance, so we have to modify it to make sure that all the possible solutions come from here. This is why we build the circuit A.

Formally, let $C' : \{0,1\}^m \to \{0,1\}^m$ be an instance of Pigeon, and let n be the minimal integer such that $2^m < \binom{2n-1}{n-1}$. Since $\alpha = \left\lceil \log \binom{2n-1}{n-1} \right\rceil + 1$, we have $m < \alpha - 1$. Define $A : \{0,1\}^{\alpha-1} \to \{0,1\}^{\alpha-1}$ by,

$$A(x) = \begin{cases} C'(x) & \text{if } x < 2^m \\ x & \text{o.w.} \end{cases}$$

It might be the case that the output of A has less than $\alpha - 1$ bits, in which case we pad it with 0 on the left to make it an $(\alpha-1)$-bit string. Recall that we have $E_{\mathsf{Cover}} : \{0,1\}^{2n} \to \{0,1\}^{\alpha}$ and $D_{\mathsf{Cover}} : \{0,1\}^{\alpha} \to \{0,1\}^{2n}$.

We proceed to build an instance $C : \{0,1\}^{\alpha-1} \to \{0,1\}^{2n}$ of Erdős-Ko-Rado by setting $C(x) = D_{\mathsf{Cover}}(0 \parallel A(x))$. Note that for any $x < \binom{2n-1}{n-1}$, we have $A(x) < \binom{2n-1}{n-1}$, thus $C(x) \subseteq [2n]$ is an n-subset and does not contain the element 1 by Remark 3.3.

Now, suppose that we have a solution to C. Since the index of a solution is $< \binom{2n-1}{n-1}$, the corresponding subset(s) must have size n and can't contain 1. If the solution is of the form x, y such that $C(x) \cap C(y) = \emptyset$ then we have $|C(x) \ \cup \ C(y)| = |C(x)| + |C(y)| = 2n$ so we must have either $1 \in C(x)$ or $1 \in C(y)$, which is not possible.

Thus, any solution must be $x \neq y$ such that $C(x) = C(y)$ or x such that $C(x) = [n]$ or $\overline{[n]}$. There are two cases to consider:

- Case $D_{\mathsf{Cover}}(0 \parallel A(x)) = D_{\mathsf{Cover}}(0 \parallel A(y))$. Then $A(x) = A(y)$ since D_{Cover} is injective on its first $\binom{2n}{n}$ inputs. But C' has range $\subseteq [2^m - 1]$ so any collision in A must result from a collision in C'. Hence, we get that $x, y < 2^m$ give us a solution to C'.
- Case $D_{\mathsf{Cover}}(A(x)) = [n]$ or $\overline{[n]}$. Since $A(x) < \binom{2n-1}{n-1}$ then $D_{\mathsf{Cover}}(0 \parallel A(x))$ does not contain element 1, so $C(x) = \overline{[n]} = D_{\mathsf{Cover}}(0^{\alpha})$, thus $A(x) = 0^{\alpha-1}$. This means that we have $x < 2^m$ and x corresponds to a preimage of 0^m for C'.

In each case, we get a solution to our original problem. □

Remark 4.11. *We often use that technique of creating a circuit A from a circuit C, such that any collision (resp. preimage of 0) in A must come from a collision (resp. preimage of 0) in C, and happen in the first inputs of A (in the range where we want it to happen).*

Lemma 4.12. Erdős-Ko-Rado $\in$ *PPP*.

Proof. This proof is quite the same as the proof of Lemma 4.7, with two minor differences. The first one is that in the instance of Pigeon we create, there might be preimages of 0. These solutions to Pigeon correspond to solutions of type *iv)* for Erdős-Ko-Rado. The second difference is that we only perform the reduction on the first $\binom{2n-1}{n-1}$ inputs, and then map the others in such a way that they neither create a collision nor result in a preimage of 0.

Formally, suppose that we have an instance of Erdős-Ko-Rado, i.e., a circuit $C : \{0,1\}^{\alpha-1} \to \{0,1\}^{2n}$. We proceed to construct an instance $C' : \{0,1\}^{\alpha-1} \to \{0,1\}^{\alpha-1}$ of Pigeon as follows:

$$C'(x) = \begin{cases} E_{\mathsf{Cover}}(C(x)) & \text{if } C(x)_1 = 0 \text{ and } x < \binom{2n-1}{n-1} \\ E_{\mathsf{Cover}}(\overline{C(x)}) & \text{if } C(x)_1 = 1 \text{ and } x < \binom{2n-1}{n-1} \\ x & \text{if } x \geq \binom{2n-1}{n-1} \end{cases}$$

In the case $x < \binom{2n-1}{n-1}$, since we only encode sets whose first bit is a 0, by Remark 3.3, we get that the first bit of the encoding always is a 0, so we can consider only the $\lceil \log(\binom{2n}{n}) \rceil - 1 = \alpha - 1$ last bits of $C'(x)$ for every such x. Furthermore, if we consider the output of E_{Cover} as an integer, we get that this integer is $< \binom{2n-1}{n-1}$ (because the set we encode is one of the first $\binom{2n-1}{n-1}$ in the lexicographic order). Note also that if for some x such that $x < \binom{2n-1}{n-1}$, $C(x)$ does not have size n, then $C'(x)$ is still well-defined and less than $\binom{2n-1}{n-1}$, even if it is meaningless.

Now, suppose that we have a solution to C' of the form $x \neq y$ such that $C'(x) = C'(y)$. Again there are four cases to consider, depending on the first bits of $C(x), C(y)$. If $C(x)_1 = C(y)_1 = 0$ then $E_{\mathsf{Cover}}(C(x)) = C'(x) = C'(y) = E_{\mathsf{Cover}}(C(y))$. If both $C(x)$ and $C(y)$ have size n, then by injectivity of E_{Cover} on inputs of size n (see Lemma 3.2), we get $C(x) = C(y)$, which is a solution to Erdős-Ko-Rado. If one of them does not have size n, we also get a solution to Erdős-Ko-Rado. The other cases are similar.

Now, suppose that we have a solution to C' of the form x such that $C'(x) = 0^{\alpha-1}$. Like previously, we get that $x < \binom{2n-1}{n-1}$. If $C(x)$ does not have size n then x is a solution. Now, suppose that $C(x)$ has size n. There are two cases to consider, depending on the first bit of $C'(x)$. If the first bit of $C(x)$ is 0, then, $E_{\mathsf{Cover}}(C(x)) = 0^\alpha$ so $C(x) = 0^n \parallel 1^n$ by Eq. (1) and Lemma 3.2. Thus, $C(x) = \overline{[n]}$. Instead, if the first bit of $C(x)$ is 1, then $E_{\mathsf{Cover}}(\overline{C(x)}) = 0^\alpha$ so $\overline{C(x)} = \overline{[n]}$ and thus $C(x) = [n]$. In either case, we get a solution to our original problem. □

Remark 4.13. *Like previously, the idea behind that proof is to compose our instance of* Erdős-Ko-Rado *with the property-preserving encoding we defined in Remark 4.6. However, this time it is not only the collisions that are of interest to us, but also the preimages of the 0 string.*

4.1 A Generalized Erdős-Ko-Rado Problem

For the previous problems, we were only considering a very restricted version of the Erdős-Ko-Rado theorem, namely for an intersecting family of n-subsets of $[2n]$. We now consider a more general version where we consider an intersecting family of n-subsets of $[kn]$ for some $k > 2$.

We now fix some $k > 2$ for the rest of this section. The Erdős-Ko-Rado theorem states that if $\mathcal{F}$ is an intersecting family where each set has n elements on a universe of size kn, then $\mathcal{F}$ contains at most $\binom{kn-1}{n-1}$ sets. Then, we can define the following TFNP problem, very similar to weak-Erdős-Ko-Rado.

Definition 4.14 (weak-general-Erdős-Ko-Rado$_k$)**.** *The problem* weak-general-Erdős-Ko-Rado$_k$ *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{\lceil \log(\binom{kn-1}{n-1}) \rceil + 1} \to \{0,1\}^{kn}$.

Solution: *One of the following:*

i) x *s.t.* $|C(x)| \neq n$,

ii) $x \neq y$ *s.t.* $C(x) = C(y)$,

iii) x, y *s.t.* $C(x) \cap C(y) = \emptyset$.

Proposition 4.15. weak-general-Erdős-Ko-Rado$_k$ *is PWPP-hard.*

Proof. This proof is very similar to the proof of Lemma 4.7, except that instead of working with n-subsets of $[2n]$, we work with n-subsets of $[kn]$. There is also a technical change, which is that this time we work with n-subsets of $[kn]$ that *do* contain the element 1. This is necessary to make sure that we have an intersecting family, but it adds some more technicality. For the same reason, we need A to shrink more than in the previous proof. However, the idea behind the proof is exactly the same, with the same use of the graph-hash product on a large intersecting family.

Formally, let $C' : \{0,1\}^m \to \{0,1\}^{m-1}$ be an instance of weak-Pigeon. Let n be the minimal integer such that $2^{m+1} \leq \binom{kn}{n}$. Now, let $\alpha = \left\lceil \log \binom{kn}{n} \right\rceil$. Then, $m + 1 \leq \alpha$. We also define $a = \lceil \log(k) \rceil$. By definition of α, we have $\binom{kn}{n} \geq 2^{\alpha-1}$. We also have $\frac{1}{k} \geq \frac{1}{2^a}$, so $\binom{kn-1}{n-1} = \frac{1}{k}\binom{kn}{n} \geq \frac{2^{\alpha-1}}{k} \geq 2^{\alpha-1-a}$. Like in the proof of Lemma 6.5, we can build a circuit $A' : \{0,1\}^\alpha \to \{0,1\}^{\alpha-1-a}$ whose size is polynomial in m and such that from any collision in A' we can efficiently find a collision in C'. Let $s \in \{0,1\}^\alpha$ be the binary encoding on α bits of $\binom{kn}{n} - \binom{kn-1}{n-1}$. We use the Cover encoding functions for n-subsets of $[kn]$: $E^{n,kn}_{\mathsf{Cover}} : \{0,1\}^{kn} \to \{0,1\}^\alpha$ and $D^{n,kn}_{\mathsf{Cover}} : \{0,1\}^\alpha \to \{0,1\}^{kn}$.

We define $C : \{0,1\}^\alpha \to \{0,1\}^{kn}$ by $C(x) = D^{k,kn}_{\mathsf{Cover}}(s \oplus 0^{a+1} \parallel A'(x))$. For every x, we have that $(0^{a+1} \| A'(x))$ is one of the first $2^{\alpha-1-a}$ elements of $\{0,1\}^\alpha$ in the lexicographic order, hence it is one of the first $\binom{kn-1}{n-1}$ first. Thus, the rank of $s \oplus 0^{a+1} \| A'(x)$ in the lexicographic order is between $\binom{kn}{n} - \binom{kn-1}{n-1}$ and $\binom{kn}{n} - 1$ counting from 0. The last $\binom{kn-1}{n-1}$ n-subsets of $[kn]$ in the lexicographic order correspond to subsets that contain the element 1. Hence, for every x, we have that the set $D^{n,kn}_{\mathsf{Cover}}(s \oplus 0^{1+a} \parallel A'(x))$ is an n-subset of $[kn]$ which contains the element 1. We observe that C defines an instance of weak-general-Erdős-Ko-Rado$_k$.

Now, suppose that we have a solution to this instance. We consider each solution type separately.

i) It cannot be x such that $|C(x)| \neq n$ because $C(x) = D^{n,kn}_{\mathsf{Cover}}(s \oplus 0^{1+a} \parallel A'(x))$ is an n-subset of $[kn]$.

ii) By the previous, $1 \in C(x)$ and $1 \in C(y)$ so $1 \in C(x) \cup C(y)$, which is a contradiction.

iii) By injectivity of $D^{n,kn}_{\mathsf{Cover}}$ on its first $\binom{kn}{n}$ inputs (see Lemma 3.2), we get that $(s \oplus 0^{1+a} \parallel A'(x)) = (s \oplus 0^{1+a} \parallel A'(y))$ hence $A'(x) = A'(y)$ and from there we can retrieve a collision for C'. □

To prove that weak-general-Erdős-Ko-Rado$_k$ $\in$ PWPP, we present some useful definitions and results related to the Erdős-Ko-Rado theorem.

Definition 4.16. *If k divides m, a (k,m)*-parallel class *is a set of m/k k-subsets of $[m]$ which partition $[m]$.*

Classical Theorem 2 (Baranyai, [Bar73])**.** *If k divides m, we can define $\binom{m-1}{k-1}$ (k,m)-parallel classes $\mathcal{A}_1, \ldots, \mathcal{A}_{\binom{m-1}{k-1}}$ such that each k-subset of $[m]$ appears in exactly one $\mathcal{A}_i$.*

Remark 4.17. *Note that this result proves the Erdős-Ko-Rado theorem in the case where the size of the subsets divides the size of the universe.*
Note also that up to renaming the elements, we can assume that $\mathcal{A}_1$ consists exactly of the sets $\{1, 2, \ldots, n\}, \{n+1, n+2, \ldots, 2n\}, \ldots,$ and $\{(k-1)n+1, (k-1)n+2, \ldots, kn\}$.

However, all known proofs of this theorem are inefficient, in the sense that there is no known way to define $\mathcal{A}_1, \ldots, \mathcal{A}_{\binom{m-1}{k-1}}$ such that given a k-subset of $[m]$, we can find in polynomial time the only i such that this subset appears in $\mathcal{A}_i$. We make this assumption explicit.

Assumption 4.18 (efficient Baranyai assumption)**.** *There is an efficient procedure to define $\mathcal{A}_1, \ldots, \mathcal{A}_{\binom{m-1}{k-1}}$ and a circuit $Bar : \{0,1\}^m \to [\binom{m-1}{k-1}]$ which takes as input a k-subset of $[m]$ and returns the only index i such that this subset appears in $\mathcal{A}_i$. Furthermore, we assume that $\mathcal{A}_1$ consists exactly of the sets $\{1, 2, \ldots, n\}, \{n+1, n+2, \ldots, 2n\}, \ldots,$ and $\{(k-1)n+1, (k-1)n+2, \ldots, kn\}$.*

Proposition 4.19. *Under Assumption 4.18,* WEAK-GENERAL-ERDŐS-KO-RADO$_k$ $\in$ *PWPP.*

Proof. At a high level, the proof goes as follows. We are given strictly more than $\binom{kn-1}{n-1}$ subsets of $[kn]$. We map them to elements of $[\binom{kn-1}{n-1}]$ in the following way. If one set does not have size n, we map it anywhere. If it has size n, we map it to the only i such that the set is in $\mathcal{A}_i$. This defines an instance of WEAK-PIGEON. In any collision for this instance, we must have either a set that does not have size n, or two sets in the same parallel class, which means that either they are equal, or they do not intersect.

Formally, by assumption, we have a circuit $Bar : \{0,1\}^{kn} \to [\binom{kn-1}{n-1}]$ which takes as input an n-subset of $[kn]$ and returns the only index i such that this subset appears in $\mathcal{A}_i$. We define a circuit $Bar' : \{0,1\}^{kn} \to \{0,1\}^{\lceil \log \binom{kn-1}{n-1} \rceil}$ which takes as input an n-subset of $[kn]$ and returns the binary encoding on $\left\lceil \log \binom{kn-1}{n-1} \right\rceil$ bits of the only index i such that this subset appears in $\mathcal{A}_i$. Now, suppose that we have an instance $C : \{0,1\}^{\lceil \log(\binom{kn-1}{n-1}) \rceil + 1} \to \{0,1\}^{kn}$ of WEAK-GENERAL-ERDŐS-KO-RADO$_k$. We set $C' = Bar' \circ C$. Then, we have $C' : \{0,1\}^{\lceil \log(\binom{kn-1}{n-1}) \rceil + 1} \to \{0,1\}^{\lceil \log \binom{kn-1}{n-1} \rceil}$ so C' is an instance of WEAK-PIGEON.

Now, suppose that we have a solution to this instance of WEAK-PIGEON, that is $x \neq y \in \{0,1\}^{\lceil \log(\binom{kn-1}{n-1}) \rceil + 1}$ such that $C'(x) = C'(y)$. Then, $Bar'(C(x)) = Bar'(C(y))$. If one of $C(x), C(y)$ does not have size n, we have a solution to our instance of WEAK-GENERAL-ERDŐS-KO-RADO$_k$, and similarly if $C(x) = C(y)$. Otherwise, it means that $C(x), C(y)$ are distinct n-subsets of $[kn]$ that appear in the same (n, kn)-parallel class. By definition of a parallel class, it means that these 2 sets are part of a partition of $[kn]$, hence they don't intersect and they form a solution to our original instance of WEAK-GENERAL-ERDŐS-KO-RADO$_k$. □

Remark 4.20. *Let $\mathcal{X}$ be the set of n-subsets of $[kn]$. We define an equivalence relation $\sim$ on $\mathcal{X}$ by saying that two n-subsets X and Y of $[kn]$ are equivalent if and only $Bar(X) = Bar(Y)$, meaning that they are in the same (n, kn)-parallel class in the partition induced by Bar.*
Then, we have that Bar is a property-preserving encoding for $\sim$ on $\mathcal{X}$.
Note that two equivalent subsets are either equal or disjoint. Hence, the property that is preserved by Bar is such that if two of its inputs collide, they form a solution to our problem.
Then, to prove the inclusion of WEAK-GENERAL-ERDŐS-KO-RADO$_k$ *into PPP, it suffices to compose our instance of* WEAK-GENERAL-ERDŐS-KO-RADO$_k$ *with Bar.*

The previous two propositions establish the following result.

Theorem 4.21. *Under Assumption 4.18,* WEAK-GENERAL-ERDŐS-KO-RADO$_k$ *is PWPP-complete.*

PPP-completeness using the tight bound Like for the case of n-subsets of $[2n]$, we can define a "tight" version of the previous problem, which is very similar to ERDŐS-KO-RADO.

Definition 4.22 (GENERAL-ERDŐS-KO-RADO$_k$). *The problem* GENERAL-ERDŐS-KO-RADO$_k$ *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{\lceil \log(\binom{kn-1}{n-1}) \rceil} \to \{0,1\}^{kn}$.

Solution: *One of the following:*

i) x *s.t.* $|C(x)| \neq n$ *and* $x < \binom{kn-1}{n-1}$,

ii) $x \neq y$ *s.t.* $C(x) = C(y)$ *and* $x, y < \binom{kn-1}{n-1}$,

iii) x, y *s.t.* $C(x) \cap C(y) = \emptyset$ *and* $x, y < \binom{kn-1}{n-1}$,

iv) x *s.t.* $C(x) = \{1, 2, \ldots, n\}$ *or* $\{n+1, n+2, \ldots, 2n\}$, *or...,* *or* $\{(k-1)n+1, (k-1)n+2, \ldots, kn\}$ *and* $x < \binom{kn-1}{n-1}$.

First, let's see why this problem is total. Suppose that we have a list of $\binom{kn-1}{n-1}$ subsets of $[kn]$. If one of the sets does not have n elements, if two of the sets are equal, or if two of the sets don't intersect, we have a solution. Now, suppose that we have an intersecting family of $\binom{kn-1}{n-1}$ distinct n-subsets of $[kn]$.

Now, consider a collection of (n, kn)-parallel classes $\mathcal{A}_1, \ldots, \mathcal{A}_{\binom{kn-1}{n-1}}$ such that each n-subset of $[kn]$ appears in exactly one $\mathcal{A}_i$ (which exists by Classical Theorem 2). Up to renaming the elements, we can assume that $\mathcal{A}_1$ is composed of the k n-subsets $\{1, 2, \ldots, n\}$, $\{n+1, n+2, \ldots, 2n\}$, ... and $\{(k-1)n+1, (k-1)n+2, \ldots, kn\}$. Since we have an intersecting family of distinct subsets, no two subsets can be in the same $\mathcal{A}_i$, and we have as many subsets as $\mathcal{A}_i$'s, which means that one of the subsets is in $\mathcal{A}_1$, hence that it is one of the particular subsets we are looking for. This proves that GENERAL-ERDŐS-KO-RADO$_k$ $\in$ TFNP. We then have the following result.

Proposition 4.23. GENERAL-ERDŐS-KO-RADO$_k$ *is PPP-hard.*

Proof. Informally, this proof is very much like the proof of Proposition 4.15, with the same technicalities as in the proof of Lemma 4.10. The idea is again to interpret the outputs of an instance of PIGEON as indices into the collection of all the n-subsets of $[kn]$ which contain the element 1. Solutions of type *iv)* correspond to preimages of 0. Like for Lemma 4.10, we need to define A to make sure that all solutions to our instance of GENERAL-ERDŐS-KO-RADO$_k$ indeed come from the instance of PIGEON.

Formally, let $C' : \{0,1\}^m \to \{0,1\}^m$ be an instance of PIGEON, and let n be the minimal integer such that $2^m \leq \binom{kn-1}{n-1}$. We set $\alpha = \left\lceil \log \binom{kn}{n} \right\rceil$ and $\beta = \left\lceil \log \binom{kn-1}{n-1} \right\rceil + 1$. Then, $\beta - 1 \geq m$. Define $A : \{0,1\}^{\beta-1} \to \{0,1\}^{\beta-1}$ by,

$$A(x) = \begin{cases} C'(x) & \text{if } x < 2^m \\ x & \text{if } x \geq 2^m \end{cases}$$

It might be the case that the output of A has less than $\beta - 1$ bits, in which case we pad it with 0 on the left to make it an $(\beta - 1)$-bit string. Let $s \in \{0,1\}^\alpha$ be the binary encoding on α bits of $\binom{kn}{n} - 1$. Recall that we have $E^{n,kn}_{\mathsf{Cover}} : \{0,1\}^{kn} \to \{0,1\}^\alpha$ and $D^{n,kn}_{\mathsf{Cover}} : \{0,1\}^\alpha \to \{0,1\}^{kn}$.

We proceed to build an instance $C : \{0,1\}^{\beta-1} \to \{0,1\}^{kn}$ of GENERAL-ERDŐS-KO-RADO$_k$ by setting $C(x) = D^{n,kn}_{\mathsf{Cover}}(s - 0^{\alpha+1-\beta} \| A(x))$ where - represents the subtraction in binary (mod 2^α). For every $x < \binom{kn-1}{n-1}$, we have that $(0^{\alpha+1-\beta} \parallel A(x))$ is one of the first $\binom{kn-1}{n-1}$ elements of $\{0,1\}^\alpha$ in the lexicographic order. Thus, the rank of $s - 0^{\alpha+1-\beta} \parallel A(x)$ in the lexicographic order is between $\binom{kn}{n} - \binom{kn-1}{n-1}$ and $\binom{kn}{n} - 1$ counting from 0. The last $\binom{kn-1}{n-1}$ n-subsets of $[kn]$ in the lexicographic order correspond to subsets that contain the element 1. Hence, for every $x < \binom{kn-1}{n-1}$, we have that the set $D^{n,kn}_{\mathsf{Cover}}(s - 0^{\alpha+1-\beta} \parallel A(x))$ is an n-subset of $[kn]$ which contains the element 1. We observe that C defines an instance of GENERAL-ERDŐS-KO-RADO$_k$.

Now, suppose that we have a solution to this instance. We consider each solution type separately.

i) It cannot be x such that $|C(x)| \neq n$ because $C(x) = D^{n,kn}_{\mathsf{Cover}}(s - 0^{\alpha+1-\beta} \parallel A(x))$ is an n-subset of $[kn]$.

ii) By the previous, $1 \in C(x)$ and $1 \in C(y)$ so $1 \in C(x) \cup C(y)$, which is a contradiction.

iii) By injectivity of $D^{n,kn}_{\mathsf{Cover}}$ on its first $\binom{kn}{n}$ inputs (see Lemma 3.2), we get that $(s - 0^{\alpha+1-\beta} \parallel A(x)) = (s - 0^{\alpha+1-\beta} \parallel A(y))$ hence $A(x) = A(y)$ and from there we can retrieve a collision for C' by design of A.

iv) If it is x such that $C(x)$ is one of the k particular subsets we're looking for, since we know that $1 \in C(x)$, it means that $C(x) = [n]$. When we consider n-subsets of $[kn]$, the characteristic vector of $[n]$ is the last one in the lexicographic order, which means that $[n] = D^{n,kn}_{\mathsf{Cover}}(s)$. Furthermore, $[n] = C(x) = D^{n,kn}_{\mathsf{Cover}}(s - 0^{\alpha+1-\beta} \parallel A(x))$, the rank of $s - 0^{\alpha+1-\beta} \parallel A(x)$ in the lexicographic order is between $\binom{kn}{n} - \binom{kn-1}{n-1} + 1$ and $\binom{kn}{n}$ and $D^{n,kn}_{\mathsf{Cover}}$ is injective on its first $\binom{kn}{n}$ inputs. Thus, $s - 0^{\alpha+1-\beta} \parallel A(x) = s$, which implies that $A(x) = 0$. By definition of A, this can only mean that $C'(x) = 0^m$.

In either case, we get a solution to our original problem. □

Proposition 4.24. *Under Assumption 4.18,* GENERAL-ERDŐS-KO-RADO$_k$ $\in$ *PPP.*

Proof. The proof of this result resembles a lot the proof of Proposition 4.19. The idea is the same: we are given $\binom{kn-1}{n-1}$ subsets of $[kn]$. We map each of them to an element of $[\binom{kn-1}{n-1}$ as follows. If a set does not have n elements, we map it anywhere, and if it has n elements, we map it to the only i such that this set is in $\mathcal{A}_i$. This defines an instance of Pigeon. If we have a collision, it results in a solution like before. If we have a preimage of 0, it is a set in $\mathcal{A}_1$, which means it is one of the sets we are looking for. The definition of C' has some technicality since we need to take care of the last inputs to make sure that they are not involved in a collision or result in a preimage of 0.

More formally, we have by assumption a circuit $Bar : \{0,1\}^{kn} \to [\binom{kn-1}{n-1}]$ which takes as input an n-subset of $[kn]$ and returns the only index i such that this subset appears in $\mathcal{A}_i$. We define a circuit $Bar' : \{0,1\}^{kn} \to \{0,1\}^{\lceil \log \binom{kn-1}{n-1} \rceil}$ which takes as input an n-subset of $[kn]$ and returns the binary encoding on $\left\lceil \log \binom{kn-1}{n-1} \right\rceil$ bits of $i-1$ where i is the only index such that this subset appears in $\mathcal{A}_i$.

Now, suppose that we have an instance $C : \{0,1\}^{\lceil \log(\binom{kn-1}{n-1}) \rceil} \to \{0,1\}^{kn}$ of weak-general-Erdős-Ko-Rado$_k$. We set

$$C'(x) = \begin{cases} Bar' \circ C(x) & \text{if } x < \binom{kn-1}{n-1} \\ x & \text{if } x \geq \binom{kn-1}{n-1} \end{cases}$$

Then, we have $C' : \{0,1\}^{\lceil \log(\binom{kn-1}{n-1}) \rceil} \to \{0,1\}^{\lceil \log \binom{kn-1}{n-1} \rceil}$ so C' is an instance of Pigeon.

Now, suppose that we have a solution to this instance of Pigeon. There are two cases to consider.

1. It is $x \neq y \in \{0,1\}^{\lceil \log(\binom{kn-1}{n-1}) \rceil}$ such that $C'(x) = C'(y)$. By construction of C' (and by definition of Bar'), this means that $x, y < \binom{kn-1}{n-1}$. We have $Bar'(C(x)) = Bar'(C(y))$. If one of $C(x), C(y)$ does not have size n, we have a solution to our instance of general-Erdős-Ko-Rado$_k$, and similarly if $C(x) = C(y)$. Otherwise, it means that $C(x), C(y)$ are distinct n-subsets of $[kn]$ that appear in the same (n, kn)-parallel class. By definition of a parallel class, it means that these 2 sets are part of a partition of $[kn]$, hence they don't intersect and they form a solution to our original instance of general-Erdős-Ko-Rado$_k$.

2. It is x such that $C'(x) = 0^{\lceil \log(\binom{kn-1}{n-1}) \rceil}$. By construction of C', it means that $x < \binom{kn-1}{n-1}$. We have $Bar'(C(x)) = 0^{\lceil \log(\binom{kn-1}{n-1}) \rceil}$. If $C(x)$ does not have size n, it is a solution to our original instance. If it has size n, it means that it is an n-subset of $[kn]$ which is in $\mathcal{A}_1$. By assumption, the only such subsets are the particular ones we're looking for. Hence, x is a solution to our original instance of general-Erdős-Ko-Rado$_k$. □

Remark 4.25. *As before, the idea behind that proof is to compose our instance of* general-Erdős-Ko-Rado$_k$ *with the property-preserving encoding Bar. However, this time it is not only the collisions that are of interest to us, but also the preimages of the 0 string.*

The previous two propositions establish the following result.

Theorem 4.26. *Under Assumption 4.18,* general-Erdős-Ko-Rado$_k$ *is PPP-complete.*

5 Sperner's Theorem on Largest Antichains

We now turn our attention to a different existence theorem from extremal combinatorics, concerning antichains. We say a family of sets $\mathcal{F} \subseteq 2^{\Omega}$ is an *antichain* if for every $A \neq B \in \mathcal{F}$, it holds that $A \nsubseteq B$. A well-known theorem by Sperner gives a characterization of the largest antichain. As before, for an appropriate input size, this induces a total search problem of finding two distinct sets A, B for which $A \subseteq B$. As in the previous section, we consider both a weak and a strong version, and prove the weak version to be PWPP-complete, and the strong one PPP-complete.

Classical Theorem 3 (Sperner [Spe28])**.** *The largest antichain on any universe of* $2n$ *elements is unique and consists of all subsets of size* n*.*

Like before, we consider an implicit representation of the collection of subsets via a circuit C whose input corresponds to an index into the collection, and whose output is the characteristic vector of the corresponding set.

Definition 5.1 (WEAK-SPERNER-ANTICHAIN)**.** *The problem* WEAK-SPERNER-ANTICHAIN *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{\lceil \log(\binom{2n}{n})\rceil+1} \to \{0,1\}^{2n}$.

Solution: $x \neq y$ *s.t.* $C(x) \subseteq C(y)$.

Theorem 5.2. WEAK-SPERNER-ANTICHAIN *is* PWPP*-complete*

For the rest of this section, we set $\alpha = \lceil \log \binom{2n}{n} \rceil = \left\lceil \log \binom{2n-1}{n-1} \right\rceil + 1$.

Lemma 5.3. WEAK-SPERNER-ANTICHAIN *is* PWPP*-hard.*

Proof. We explain the reduction at a high level. We reduce from WEAK-ERDŐS-KO-RADO and create an instance of WEAK-SPERNER-ANTICHAIN by including each set from the WEAK-ERDŐS-KO-RADO instance, as well as its complement. If we find a solution to WEAK-SPERNER-ANTICHAIN, one of the sets must be contained within another. If one of the two sets does not have size n, we obtain a solution to WEAK-ERDŐS-KO-RADO of type i). Otherwise, the duplicated sets must be equal, and hence the original sets are either equal, or one of the sets is the complement of the other.

Formally, suppose that we have an instance $C : \{0,1\}^{\alpha} \to \{0,1\}^{2n}$ of WEAK-ERDŐS-KO-RADO. Write $x = yb$ where b is a bit. We build an instance $C' : \{0,1\}^{\alpha+1} \to \{0,1\}^{2n}$ of SPERNER-ANTICHAIN as follows.

$$C'(x) = \begin{cases} C(y) & \text{if } b = 0 \\ \overline{C(y)} & \text{if } b = 1 \end{cases}$$

Now, suppose that we have a solution to this instance of SPERNER-ANTICHAIN, that is $x \neq x'$ such that $C'(x) = C'(x')$. Write $x = yb$ and $x' = y'b'$. There are four cases to consider. If $b = b' = 0$. Then $y \neq y'$ and $C(y) = C'(x) \subseteq C'(x') = C(y')$. If $C(y)$ and $C(y')$ both have size n, then $C(y) = C(y')$, and if this is not the case we get a solution for C. In both cases, we get a solution for WEAK-ERDŐS-KO-RADO. The other cases are similar; in all four cases, we get a solution to our original problem, so WEAK-SPERNER-ANTICHAIN is PWPP-hard. □

Classical Theorem 4 (Dilworth's Theorem, [Dil50])**.** *The size of the largest antichain in* $(2^{[2n]}, \subseteq)$ *is equal to the size of the smallest chain partition, namely* $\binom{2n}{n}$.

Lemma 5.4. WEAK-SPERNER-ANTICHAIN $\in$ *PWPP.*

Proof. We give a high-level overview of the reduction from WEAK-SPERNER-ANTICHAIN to WEAK-PIGEON.

Fix an arbitrary partition into chains of $(2^{[2n]}, \subseteq)$ of size $\binom{2n}{n}$ (which exists by Classical Theorems 3 and 4). Since we have more than $\binom{2n}{n}$ inputs in an instance of WEAK-SPERNER-ANTICHAIN, by the pigeonhole principle, two distinct inputs must end up in the same chain. We want to give an identifier to each of these chains, using α bits, such that for any subset we are be able to quickly find the identifier of the chain to which it belongs. To do so, in each chain, we choose as representative the n-subset of the chain, that is guaranteed to exist by Classical Theorem 4. Then, the identifier of the chain is the Cover encoding on this subset. To map a subset to the representative of its chain, we make use Catalan factorizations (Section 3.4). Once we have this, from each subset we can efficiently get the n-subset in its chain and therefore the identifier of the chain. Finally, a collision in the identifiers is equivalent to two elements in the same chain, which means a solution for WEAK-SPERNER-ANTICHAIN.

Formally, let $C : \{0,1\}^{\alpha+1} \to \{0,1\}^{2n}$ be an instance of WEAK-SPERNER-ANTICHAIN. We proceed to construct an instance of WEAK-PIGEON as follows: if $x \in \{0,1\}^{\alpha+1}$, we have $X := C(x) \in \{0,1\}^{2n}$ which represents a subset of $[2n]$. Let $(X', k) = E_{\mathsf{Catalan}}(X)$ be the Catalan factorization of X, l be the number of z's in X' and m the number of bits underlined during the construction of X'. Note that every time we underline bits we underline simultaneously a 0 and a 1, thus m is even. Then, $l = 2n - m$ is an even number. Now, let $S(x) = D^{(l/2)}_{\mathsf{Catalan}}(X')$. Then, since X' has the same number of 1's and 0's and since we replaced half of the z's by 1's and the other half by 0's, we have that $S(x)$ represents an n-subset of $[2n]$. Informally, it is the n-subset of the chain that contains X, and replacing z's by 1's enables us to move inside that chain. Finally, we set $C'(x) = E_{\mathsf{Cover}}(S(x)) \in \{0,1\}^{\alpha}$. We observe that C' is an instance of WEAK-PIGEON.

Now suppose that we have a solution to this instance of WEAK-PIGEON, that is $x \neq y$ such that $C'(x) = C'(y)$. Then, by injectivity of E_{Cover} on the n-subsets of $[2n]$ (see Lemma 3.2), we get that $S(x) = S(y)$. Informally, this means that $C(x)$ and $C(y)$ belong to the same chain and thus that one is contained is the other. Let's now prove it formally. Let $(X', k) = E_{\mathsf{Catalan}}(X) = E_{\mathsf{Catalan}}(C(x))$ be the Catalan factorization of X and l be the number of z's in X', and let $(Y', k') = E_{\mathsf{Catalan}}(Y) = E_{\mathsf{Catalan}}(C(y))$. We have $S(x) = D_{\mathsf{Catalan}}(X', l/2)$ so by Lemma 3.10, the Catalan string that corresponds to $S(x)$ is X'. Similarly, the Catalan string that corresponds to $S(y)$ is Y'. Since $S(x) = S(y)$, we get $X' = Y'$. We have that $X = D_{\mathsf{Catalan}}(E_{\mathsf{Catalan}}(X))$ and that $Y = D_{\mathsf{Catalan}}(E_{\mathsf{Catalan}}(Y))$ by Lemma 3.9, so $X = D_{\mathsf{Catalan}}(X', k)$ and $Y = D_{\mathsf{Catalan}}(Y', k') = D_{\mathsf{Catalan}}(X', k')$. By symmetry of x and y we can assume that $k \leq k'$. Then, to go from X' to X we added k elements (the ones corresponding to the last k z's in X') while to go from X' to Y we added these same k elements plus $k' - k$ others. Hence, $C(x) = X \subseteq Y = C(y)$. □

Remark 5.5. *Consider the circuit $E : \{0,1\}^{2n} \to \{0,1\}^{\alpha}$, defined as follows. On input $X \in \{0,1\}^{2n}$, it computes (X', k) the Catalan factorization of X, l the number of z in X'. Then, it computes $S(X) = D^{(l/2)}_{\mathsf{Catalan}}(X')$ and finally returns $E_{\mathsf{Cover}}(S(X))$.*
Let $\mathcal{X} = 2^{[2n]}$. We define an equivalence relation on $\mathcal{X}$ by saying that two subsets are equivalent if and only if they have the same Catalan string.
Then, we showed in the previous proof that E is a property-preserving encoding for $\sim$ on $\mathcal{X}$. Note that we also showed that if we have two equivalent subsets, one is included in the other. Hence, the property that is preserved by E is such that if two of its inputs collide, they form a solution to our problem.
Then, to prove the inclusion of WEAK-SPERNER-ANTICHAIN *into PWPP, it suffices to compose our instance of* WEAK-SPERNER-ANTICHAIN *with E.*

PPP-completeness using the tight bound As with Erdős-Ko-Rado, we observe that the bound in theorem is tight, and we know the unique antichain of size $\binom{2n}{n}$, so we have some structural information about any collection of size $\binom{2n}{n}$. From that strong theorem, employing the same technique as before, we modify the problem to let the circuit represent a collection of that exact size. By Classical Theorem 3, we observe that if $\mathcal{F}$ is an antichain with $|\mathcal{F}| = \binom{2n}{n}$, then $\mathcal{F}$ must contain $\overline{[n]}$. This leads us to define the following problem.

Definition 5.6 (SPERNER-ANTICHAIN)**.** *The problem* SPERNER-ANTICHAIN *is defined by the relation*

Instance: *A Boolean circuit $C \colon \{0,1\}^{\lceil \log(\binom{2n}{n}) \rceil} \to \{0,1\}^{2n}$.*

Solution: *One of the following:*

i) $x \neq y$ s.t. $C(x) \subseteq C(y)$ and $x, y < \binom{2n}{n}$,

ii) x s.t. $C(x) = \overline{[n]}$ and $x < \binom{2n}{n}$.

Theorem 5.7. SPERNER-ANTICHAIN *is PPP-complete.*

Lemma 5.8. SPERNER-ANTICHAIN *is PPP-hard.*

Proof. Same proof as for Lemma 5.3, by reduction from ERDŐS-KO-RADO. Observe that if we have a solution of type ii) for SPERNER-ANTICHAIN, the corresponding set in the ERDŐS-KO-RADO instance is either $[n]$ or $\overline{[n]}$, which is one of the desired solutions to ERDŐS-KO-RADO. □

Lemma 5.9. SPERNER-ANTICHAIN $\in$ *PPP.*

Proof. Informally, this proof is the same as the proof of Lemma 5.3, with some additional technical details. First, we need to take care of preimages of 0. The indices corresponding to preimages of 0 correspond to solutions of type *ii*). Second, since we only care about the first $\binom{2n}{n}$ inputs, we have to make sure that the last ones are not part of a collision, or result in a preimage of 0.

Formally, let $C : \{0,1\}^\alpha \to \{0,1\}^{2n}$ be an instance of Sperner-Antichain. We proceed to construct an instance of Pigeon as follows: if $x \in \{0,1\}^\alpha$, we have $X := C(x) \in \{0,1\}^{2n}$ which is a subset of $[2n]$. Let $(X', k) = E_{\mathsf{Catalan}}(X)$ be the Catalan factorization of X, l be the number of z's in X' and m the number of bits underlined during the construction of X'. Note that every time we underline bits we underline simultaneously a 0 and a 1, thus m is even. Then, $l = 2n - m$ is an even number. Now, let

$$S(x) = D^{(l/2)}_{\mathsf{Catalan}}(X')$$

Then, since X' has the same number of 1's and 0's and since we replaced half of the z's by 1's and the other half by 0's, we have that $S(x)$ represents an n-subset of $[2n]$. Informally, it is the n-subset of the chain that contains X, and replacing z's by 1's enables us to move inside that chain. Finally, we set,

$$C'(x) = \begin{cases} E_{\mathsf{Cover}}(S(x)) & \text{if } x < \binom{2n}{n} \\ x & \text{if } x \geq \binom{2n}{n} \end{cases}$$

Then $C' : \{0,1\}^\alpha \to \{0,1\}^\alpha$ is an instance of Pigeon and has polynomial size. Suppose that we have a solution to this instance of Pigeon of the form x such that $C'(x) = 0^\alpha$. Then, $x < \binom{2n}{n}$ and $E_{\mathsf{Cover}}(S(x)) = 0^\alpha$ so $S(x) = \overline{[n]}$. Let $(X', k) = E_{\mathsf{Catalan}}(X) = E_{\mathsf{Catalan}}(C(x))$ be the Catalan factorization of X. Like previously, we get that the Catalan string that corresponds to $S(x)$ is X'. However, $S(x) = \overline{[n]}$ and the Catalan string that corresponds to $\overline{[n]}$ is $0^n \parallel 1^n$. Thus, $X' = 0^n \parallel 1^n$. Now, $C(x) = D_{\mathsf{Catalan}} \circ E_{\mathsf{Catalan}}(C(x)) = D_{\mathsf{Catalan}}(0^n \parallel 1^n, k) = 0^n \parallel 1^n$, so $C(x) = \overline{[n]}$.

Suppose instead that we have a solution to this instance of Pigeon, of the form $x \neq y$ such that $C'(x) = C'(y)$. Like before, we have $x, y < \binom{2n}{n}$. Then, by injectivity of E_{Cover} on the n-subsets of $[2n]$ (see Lemma 3.2), we get that $S(x) = S(y)$. Informally, this means that $C(x)$ and $C(y)$ belong to the same chain and thus that one is contained is the other. Let $(X', k) = E_{\mathsf{Catalan}}(X) = E_{\mathsf{Catalan}}(C(x))$ be the Catalan factorization of X and l be the number of z's in X', and let $(Y', k') = E_{\mathsf{Catalan}}(Y) = E_{\mathsf{Catalan}}(C(y))$. We have $S(x) = D_{\mathsf{Catalan}}(X', l/2)$ so by Lemma 3.10, the Catalan string that corresponds to $S(x)$ is X'. Similarly, the Catalan string that corresponds to $S(y)$ is Y'. Since $S(x) = S(y)$, we get $X' = Y'$. We have that $X = D_{\mathsf{Catalan}}(E_{\mathsf{Catalan}}(X))$ and that $Y = D_{\mathsf{Catalan}}(E_{\mathsf{Catalan}}(Y))$ by Lemma 3.9, so $X = D_{\mathsf{Catalan}}(X', k)$ and $Y = D_{\mathsf{Catalan}}(Y', k') = D_{\mathsf{Catalan}}(X', k')$. By symmetry of x and y we can assume that $k \leq k'$. Then, to go from X' to X we added k elements (the ones corresponding to the last k z's in X') while to go from X' to Y we added these same k elements plus $k' - k$ others. Hence, $C(x) = X \subseteq Y = C(y)$. □

Remark 5.10. *Like previously, the idea behind that proof is to compose our instance of* Sperner-Antichain *with the property-preserving encoding we defined in Remark 5.5. However, this time it is not only the collisions that are of interest to us, but also the preimages of the 0 string.*

6 Cayley's Tree Formula

We consider yet another classic theorem from combinatorics, related to spanning trees. A classic result by Cayley establishes the number of spanning trees of the complete graph on n vertices. We observe then that if we have a collection of sufficiently many such graphs, either one of the graphs is not a spanning tree, or two spanning trees collide. Note that two isomorphic trees on distinct vertices are not considered a collision. This allows us to define a total search problem of either finding a collision or finding an index not corresponding to a spanning tree. We represent trees using a bitmap on all possible edges, ordered arbitrarily. We show that this problem is equivalent to weak-Pigeon, in a more direct way than for the previous results. As before, the problem can be modified using the same technique as previously to become equivalent to Pigeon, and thus PPP-complete.

Classical Theorem 5 (Cayley [Cay89])**.** *There are exactly n^{n-2} spanning trees of the complete graph on n vertices.*

Definition 6.1 (weak-Cayley)**.** *The problem* weak-Cayley *is defined by the relation*

Instance: *A Boolean circuit $C\colon \{0,1\}^{\lceil (n-2)\log(n)\rceil+1} \to \{0,1\}^{\binom{n}{2}}$.*

Solution: *One of the following:*

i) x s.t. $C(x)$ is not a spanning tree (i.e., is not spanning, not connected or contains a cycle),

ii) $x \neq y$ s.t. $C(x) = C(y)$.

Theorem 6.2. weak-Cayley *is PWPP-complete.*

For the rest of this section, we set $\beta = \lceil (n-2)\log(n)\rceil$.

Lemma 6.3. weak-Cayley $\in$ *PWPP.*

Proof. We reduce to weak-Pigeon. Unlike the previous problems, here, we are interested in a very simple algebraic structure, namely equality. Thus, we want collisions in our encoding to correspond to equality. This means that we want an efficiently computable injective encoding of spanning trees. For this, we use Prüfer codes (Section 3.3). We map any input x to the Prüfer encoding of $C(x)$ and, therefore, a collision either yield a collision in the trees or a graph that is not a spanning tree.

Formally, suppose that we have $C\colon \{0,1\}^{\lceil (n-2)\log(n)\rceil+1} \to \{0,1\}^{\binom{n}{2}}$ an instance of Cayley. We may define an instance of weak-Pigeon by setting $C'(x) = \tilde{E}_{\mathsf{Prüfer}}(C(x))$. We observe that $C' : \{0,1\}^{\beta+1} \to \{0,1\}^{\beta}$ is indeed an instance of weak-Pigeon. By definition, $C'(x)$ is the rank in the lexicographic order of the Prüfer code of $C(x)$. Now, suppose that we have a solution to this instance, that is $x \neq y \in \{0,1\}^{\beta+1}$ such that $C'(x) = C'(y)$. Then, $\tilde{E}_{\mathsf{Prüfer}}(C(x)) = \tilde{E}_{\mathsf{Prüfer}}(C(y))$. If $C(x)$ or $C(y)$ is not a spanning tree, then we have a solution to our original instance of Cayley. Otherwise, $C(x)$ and $C(y)$ are spanning trees, so by injectivity of $\tilde{E}_{\mathsf{Prüfer}}$ on the set of labelled spanning trees on n vertices (see Lemma 3.5), we have $C(x) = C(y)$ which is a solution to our original instance of weak-Cayley. □

Remark 6.4. *Here, we can interpret $\tilde{E}_{\mathsf{Prüfer}}$ as a property-preserving encoding on the set of labelled spanning trees on n vertices, where the equivalence relation is equality. Hence, this is another proof of inclusion using property-preserving encodings, where we compose the instance of our problem with an appropriate property-preserving encoding. The equivalence relation has to be equality since the only spanning trees that are solutions of* weak-Cayley *are spanning trees that are equal.*

Lemma 6.5. weak-Cayley *is PWPP-hard.*

Proof. We interpret the output of the weak-Pigeon instance as an index into the collection of all labelled spanning trees on n vertices. By correctness of the encoding, the output necessarily is a spanning tree and, hence, the only solutions are collisions. We also detail some technical work to get a circuit with the right input size and output size, for which finding collisions allows solving the original instance of weak-Pigeon.

Formally, let $C' : \{0,1\}^{m+1} \to \{0,1\}^{m}$ be an instance of weak-Pigeon. We define a circuit $A : \{0,1\}^{m+2} \to \{0,1\}^{m}$ as follows. For any $x \in \{0,1\}^{m+2}$, write $x = y \parallel z$ with $y \in \{0,1\}^{m+1}$ and $z \in \{0,1\}$. Then, we set $A(x) = C'(C'(y) \parallel z)$. Note that A still has polynomial size and that any collision in A allows us to retrieve a collision for C' (like in the Merkle-Damgård construction, see [Mer79]).

Let n be the smallest integer such that $m + 1 \leq (n-2)\log(n)$. Note that n is polynomial in m. Let $\beta = \lceil (n-2)\log(n)\rceil$. Then, $m+1 \leq \beta$, hence $m+2 \leq \beta+1$. Now, we define a circuit $A' : \{0,1\}^{\beta+1} \to \{0,1\}^{\beta-1}$ as follows. For any $x \in \{0,1\}^{\beta+1}$, write $x = y \parallel z$ with $y \in \{0,1\}^{m+2}$ and $z \in \{0,1\}^{\beta+1-m-2}$. Then, we set $A'(x) = A(y) \parallel z$. Note that A' also has polynomial size and that any collision in A' allows us to retrieve a collision for A hence for C'.

Recall that we have $\tilde{E}_{\mathsf{Prüfer}} : \{0,1\}^{\binom{n}{2}} \to \{0,1\}^{\beta}$ and $\tilde{D}_{\mathsf{Prüfer}} : \{0,1\}^{\beta} \to \{0,1\}^{\binom{n}{2}}$. We now define an instance C of Cayley by setting $C(x) = \tilde{D}_{\mathsf{Prüfer}}(0 \parallel A'(x))$. Now, suppose that we have a solution to this instance of Cayley. For every x, $0 \parallel A'(x)$ is one of the first n^{n-2} elements of $\{0,1\}^\beta$ in the lexicographic order, so $\tilde{D}_{\mathsf{Prüfer}}$ is well-defined and correct (i.e., it indeed returns a spanning tree) on input $0 \parallel A'(x)$. Then, this solution must be $x \neq y$ such that $C(x) = C(y)$. By injectivity of $\tilde{D}_{\mathsf{Prüfer}}$ on its first n^{n-2} inputs (Lemma 3.5), we get that $A'(x) = A'(y)$ and from this we can retrieve a solution to our original instance of weak-Pigeon. □

PPP-completeness using the tight bound Again, we observe that Classical Theorem 5 gives an exact bound, namely that there are exactly n^{n-2} labelled spanning trees on n vertices. As before, this leads us to defining the following problem.

Definition 6.6 (Cayley)**.** *The problem* Cayley *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{\lceil (n-2)\log(n)\rceil} \to \{0,1\}^{\binom{n}{2}}$.

Solution: *One of the following:*

i) x *s.t.* $C(x)$ *is not a spanning tree and* $x < n^{n-2}$*,*

ii) $x \neq y$ *s.t.* $C(x) = C(y)$ *and* $x < n^{n-2}$*,*

iii) x *s.t.* $C(x) = T_1$ *and* $x < n^{n-2}$*, with* T_1 *defined as in Remark 3.7.*

Theorem 6.7. Cayley *is PPP-complete.*

Lemma 6.8. Cayley *is PPP-hard.*

Proof. This proof is in spirit similar to the proof of Lemma 6.5. We interpret the outputs of the instance of Pigeon as indices in the list of all spanning trees of the complete graph on n vertices. Like in previous proofs, we have to define a circuit A with sufficiently many inputs such that from any collision (resp. preimage of 0) in A we can find a collision (resp. preimage of 0) in the instance of Pigeon. In the instance of Cayley we create, preimages of T_1 correspond to preimages of 0.

Let $C' : \{0,1\}^m \to \{0,1\}^m$ be an instance of Pigeon, and let n be the smallest integer such that $m \leq (n-2)\log(n)$. Note that n is polynomial in m. Let $\beta = \lceil (n-2)\log(n)\rceil$. We define $A : \{0,1\}^\beta \to \{0,1\}^\beta$ as follows.

$$A(x) = \begin{cases} C'(x) & \text{if } x < 2^m \\ x & \text{if } x \geq 2^m \end{cases}$$

If necessary, we pad the outputs of A on the left by 0's so that they have length β (this might be necessary for $x < 2^m$). Note that $A([2^m - 1]) \subseteq [2^m - 1]$ and A acts as the identity over $[2^\beta - 1] \setminus [2^m - 1]$, hence any solution to A as an instance of Pigeon immediately gives a solution to C'. Recall that we have $\tilde{E}_{\mathsf{Prüfer}} : \{0,1\}^{\binom{n}{2}} \to \{0,1\}^\beta$ and $\tilde{D}_{\mathsf{Prüfer}} : \{0,1\}^\beta \to \{0,1\}^{\binom{n}{2}}$. Then, we define an instance C of Cayley by setting $C(x) = \tilde{D}_{\mathsf{Prüfer}}(A(x))$.

Now, suppose that we have a solution to this instance of Cayley. Every solution must consist of inputs $< n^{n-2}$ but $A([n^{n-2} - 1]) \subseteq [n^{n-2} - 1]$ by construction of A, and $\tilde{D}_{\mathsf{Prüfer}}$ is well-defined, correct and injective on this set by Lemma 3.5. This implies that this solution can not be x such that $C(x)$ is not a spanning tree. Then, suppose that this solution is $x \neq y$ such that $C(x) = C(y)$. By injectivity of $\tilde{D}_{\mathsf{Prüfer}}$ on $[n^{n-2} - 1]$, we get that $A(x) = A(y)$ and from this we can retrieve a solution to our original instance of Pigeon. Now, if this solution is x such that $C(x) = T_1$ then this means that $A(x) = 0^\beta$ by Remark 3.7 and injectivity of $D_{\mathsf{Prüfer}}$ over $[n^{n-2} - 1]$ so $C'(x) = 0^m$. □

Lemma 6.9. Cayley $\in$ *PPP.*

Proof. The idea behind the proof is similar to that of Lemma 6.3, using $\tilde{E}_{\mathsf{Prüfer}}$ to create an instance of Pigeon except that we restrict the circuit to only apply the first n^{n-2} elements of the collection, and set it to the identity on the rest of the inputs. Any preimage of 0 correspond to a preimage of T_1, and collisions arise from graphs that are not spanning trees, as well as collisions in the Cayley instance. □

7 Ward-Szabo Theorem on Swell Colorings

We now focus on a different theorem from extremal combinatorics, and more precisely from extremal graph theory. Let $G = (V, E)$ be the complete graph on N vertices. An edge-coloring $c : E \to [r]$ for some r is called a *swell coloring* of G if it uses at least 2 colors and if every triangle is either monochromatic or trichromatic. It is rather straightforward to see that in any 2-coloring of G, there must exist a bichromatic triangle. On the contrary, if we color each edge with a different color, we trivially get a swell coloring. The natural question that appears is then to determine the minimal number of colors required to swell-color the complete graph on N vertices. This was solved in some cases by Ward and Szabo in 1995.

Classical Theorem 6 (Ward-Szabo [WS95])**.** *The complete graph on N vertices cannot be swell-colored with fewer than $\sqrt{N}+1$ colors, and this bound is tight.*

From that theorem, we can define a TFNP problem as follows: the input is a coloring C of the edges of the complete graph on 2^{2n} vertices with 2^n colors, as well as three vertices a, b, c such that $C(a, b) \neq C(a, c)$ to guarantee that at least 2 colors are used in the coloring. A solution is then the vertices of a bichromatic triangle (which is guaranteed to exist by Classical Theorem 6). We also allow extra solutions, one to specify that the edges (a, b) and (a, c) have the same color, and one if the coloring of the graph is not consistent.

Definition 7.1 (Ward-Szabo)**.** *The problem* Ward-Szabo *is defined by the relation*

Instance: *The following:*

1. *A Boolean circuit $C\colon \{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}^n$; and,*
2. *Distinct $a, b, c \in \{0,1\}^{2n}$.*

Solution: *One of the following:*

i) 0 *if $C(a, b) = C(a, c)$,*

ii) x, y s.t. $C(x, y) \neq C(y, x)$,

iii) Distinct x, y, z s.t. $C(x, y) = C(y, z) \neq C(x, z)$.

We also define two variants of this problem, whose totality is a consequence of the totality of Ward-Szabo. In the first one, we allow an extra type of solution, namely the vertices of two distinct triangles with the same "color profile".

Definition 7.2 (Ward-Szabo-Collisions)**.** *The problem* Ward-Szabo-Collisions *is defined by the relation*

Instance: *The following:*

1. *A Boolean circuit $C\colon \{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}^n$; and,*
2. *Distinct $a, b, c \in \{0,1\}^{2n}$.*

Solution: *One of the following:*

i) 0 *if $C(a, b) = C(a, c)$,*

ii) x, y s.t. $C(x, y) \neq C(y, x)$,

iii) Distinct x, y, z s.t. $C(x, y) = C(y, z) \neq C(x, z)$,

iv) Two triples, $(x, y, z), (x', y', z')$, each with 3 distinct elements, s.t. $\{x, y, z\} \neq \{x', y', z'\}$ and $C(x, y) = C(x', y')$, $C(x, z) = C(x', z')$, $C(y, z) = C(y', z')$.

In the second variant, we allow the same extra type of solution, namely the vertices of two distinct triangles with the same "color profile", with the additional constraint that these triangles should be trichromatic.

Definition 7.3 (Ward-Szabo-Colorful-Collisions)**.** *The problem* Ward-Szabo-Colorful-Collisions *is defined by the relation*

Instance: *The following:*

1. *A Boolean circuit* $C\colon \{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}^n$*; and,*
2. *Distinct* $a, b, c \in \{0,1\}^{2n}$.

Solution: *One of the following:*

i) 0 *if* $C(a,b) = C(a,c)$*,*

ii) x, y *s.t.* $C(x,y) \neq C(y,x)$*,*

iii) Distinct x, y, z *s.t.* $C(x,y) = C(y,z) \neq C(x,z)$*,*

iv) Two triples $(x,y,z), (x',y',z')$*, each with 3 distinct elements, s.t.* $\{x,y,z\} \neq \{x',y',z'\}$*,* $C(x,y) = C(x',y')$*,* $C(x,z) = C(x',z')$*,* $C(y,z) = C(y',z')$ *and the triangle* (x,y,z) *is trichromatic.*

Theorem 7.4. weak-Pigeon $\leq$ Ward-Szabo-Collisions $\leq$ Ward-Szabo-Colorful-Collisions $\leq$ Ward-Szabo*.*

Proof. At a high level, we use the weak-Pigeon circuit as the coloring of the graph. If we find a bichromatic triangle, we have found a collision. If we find two triangles with the same "color-profile", we have also found a collision.

Formally, let us prove that weak-Pigeon reduces to Ward-Szabo-Collisions. Let $C : \{0,1\}^{n+1} \to \{0,1\}^n$ be an instance of weak-Pigeon. By the Merkle-Damgård construction, we can build a circuit $A : \{0,1\}^{4n} \to \{0,1\}^n$ of polynomial size such that finding a collision for A allows finding a collision for C. We set $a = 0^{2n}, b = 1^{2n}$ and $c = 0^{2n-1} \parallel 1$. If $A(a,b) = A(a,c)$ then we have a collision for A. Otherwise, we have $A(a,b) \neq A(a,c)$. We define a circuit $A' : \{0,1\}^{4n} \to \{0,1\}^n$ as follows.

$$A'(x,y) = \begin{cases} A(x,y) & \text{if } x \leq y \\ A(y,x) & \text{if } x > y \end{cases}$$

Then, we define an instance of Ward-Szabo-Collisions by saying that the coloring is A' and that $A'(a,b) \neq A'(a,c)$.

Now, suppose that we have a solution to this instance of Ward-Szabo-Collisions. Note that solution cannot be x, y such that $A'(x,y) \neq A'(y,x)$ by definition of A'. If this solution is distinct x, y, z such that $A'(x,y) = A'(x,z) \neq A'(y,z)$ then $A'(x \parallel y) = A'(x \parallel z)$. which implies a collision for A in any case. If this solution is two triples $(x,y,z) \neq (x',y',z')$ such that $A'(x,y) = A'(x',y')$, $A'(x,z) = A'(x',z')$, $A'(y,z) = A'(y',z')$, then by symmetry of x, y and z, and of x', y' and z', we can assume $x \neq x'$. If $x = y'$ and $y = x'$, then $A'(x,z) = A'(x',z') = A'(y,z')$ and $x \neq y$ so this gives us a collision for A. Otherwise, from $A'(x \parallel y) = A'(x' \parallel y')$, from which we can find a collision for A.
In all cases, we get a collision for A from which we can get a collision for C. □

Theorem 7.5. Ward-Szabo-Collisions $\in$ *PWPP.*

Proof. We describe informally the proof. There are only 2^{3n} different "color profiles" possible, which is less than the number of triangles containing the vertex 0^{2n}. Hence, if we map sufficiently many distinct triangles containing that vertex to their color profile, it defines an instance of weak-Pigeon, and any solution to this instance gives us a solution of type *iv*).

Formally, let $C : \{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}^n$, $a, b, c \in \{0,1\}^{2n}$be an instance of Ward-Szabo-Collisions. We consider the "color profile" of some triangles containing the vertex indexed by 0^{2n}. Let $C' : \{0,1\}^{3n+1} \to \{0,1\}^{3n}$ be the circuit defined as follows. For every $x \in \{0,1\}^{3n+1}$, write $x = (y \parallel z)$ with $y \in \{0,1\}^{n+3}$ and $z \in \{0,1\}^{2n-2}$. Then, let $y' = (1^{n-2} \parallel y)$ and $z' = (10 \parallel z) \in \{0,1\}^{2n}$. Then, we set $C'(x) = (C(0^{2n}, y'), C(0^{2n}, z'), C(y', z'))$. C' defines an instance of weak-Pigeon. Suppose now that we

have a solution to this instance of WEAK-PIGEON, that is $x_1 \neq x_2$ such that $C'(x_1) = C'(x_2)$.
Then, define y'_1, z'_1, y'_2 and z'_2 as above. Since $x_1 \neq x_2$, by construction we have that $\{0^{2n}, y'_1, z'_1\} \neq \{0^{2n}, y'_2, z'_2\}$ and that each of these two sets has three distinct elements. Furthermore, $C'(x_1) = C'(x_2)$ implies that $C(0^{2n}, y'_1) = C(0^{2n}, y'_2), C(0^{2n}, z'_1) = C(0^{2n}, z'_2)$ and $C(y'_1, z'_1) = C(y'_2, z'_2)$. Hence, we have a solution of type *iv*) to WARD-SZABO-COLLISIONS. □

Remark 7.6. *The last two theorems prove that* WARD-SZABO-COLLISIONS *is PWPP-complete. However, notice that the proof of inclusion into PWPP does not use solutions of the first three types. Hence, if we call* WARD-SZABO-COLLISIONS*' the problem similar to* WARD-SZABO-COLLISIONS *but without the first three types of solutions, this new problem is also PWPP-complete. Indeed, the proof of inclusion into PWPP would be similar, and the proof of hardness too, only with less cases to consider. Thus, it seems (at least that is how we prove it) that what makes* WARD-SZABO-COLLISIONS *PWPP-complete is only its last type of solutions. Now, one could wonder how hard this problem becomes if we slightly modify this last type of solutions to make them harder to find. This is exactly what* WARD-SZABO-COLORFUL-COLLISIONS *does.*

Theorem 7.7. WARD-SZABO-COLORFUL-COLLISIONS $\in$ *PPP*.

Proof. We first give an overview of the proof. It is quite similar in spirit to the previous one, but we need to work to avoid getting collisions that would give us 2 monochromatic triangles. This costs an extra bit, hence the inclusion in PPP and not in PWPP. We are given three vertices $a, b, c \in \{0,1\}^{2n}$ such that the colors $C(a, b), C(a, c)$ and $C(a, c)$ are distinct (otherwise we have an easy solution to the instance). We create an instance of PIGEON by mapping any vertex x to the pair of colors $(C(x, b), C(x, c))$ if we don't have $C(x, b) = C(x, c) = C(b, c)$ which would be a monochromatic triangle, and to the color $C(x, a)$ otherwise. We need $2n$ bits to make sure that these two types of outputs don't collide. We make sure that 0 has no preimage. Then, any solution to the instance of PIGEON must be a collision. If it is a collision from the first case, we found 2 distinct non-monochromatic triangles with the same profile, hence a solution of type *iii*) or *iv*). If it is a collision from the second case, we found 2 non-monochromatic triangles with the same profile.

Formally, let $C : \{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}^n$ and $a, b, c \in \{0,1\}^{2n}$ be an instance of WARD-SZABO-COLORFUL-COLLISIONS. If $C(a, b) = C(a, c)$ then we have a solution to this instance of WARD-SZABO-COLORFUL-COLLISIONS. Now, suppose $C(a, b) \neq C(a, c)$. If $C(b, c) = C(a, b)$ or $C(b, c) = C(a, c)$, then we have a solution of type *iii*) to this instance of WARD-SZABO-COLORFUL-COLLISIONS. Hence, we can suppose that the colors $C(a, b), C(a, c)$ and $C(b, c)$ are all distinct. Furthermore, if $C(c, b) \neq C(b, c)$, we have a solution of type *ii*), so we also assume that $C(c, b) = C(b, c)$. We use the circuit $E_{lex} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n-1}$ defined in Section 3.2, to encode 2-subsets of $\{0,1\}^n$ using $2n - 1$ bits.

We define an instance $C' : \{0,1\}^{2n} \to \{0,1\}^{2n}$ of PIGEON as follows.

$$C'(x) = \begin{cases} 01110^{2n-4} & \text{if } x = a \\ 010^{2n-2} & \text{if } x = b \\ 0110^{2n-3} & \text{if } x = c \\ 01^{n-1} \parallel C(x, a) & \text{if } C(x, b) = C(x, c) = C(b, c) \\ 1 \parallel E_{lex}(C(x, b), C(x, c)) & \text{otherwise} \end{cases}$$

Now, suppose that we have a solution to this instance of PIGEON. By construction of C', it cannot be $x \in \{0,1\}^{2n}$ such that $C'(x) = 0^{2n}$. Then, it must be $x \neq y \in \{0,1\}^{2n}$ such that $C'(x) = C'(y)$. Furthermore, by design of C', we have $x, y \notin \{a, b, c\}$. We consider two cases, depending on the first bit of $C'(x)$.

1. Suppose the first bit of $C'(y) = C'(x)$ is a 1. Then, $E_{lex}(C(x, b), C(x, c)) = E_{lex}(C(y, b), C(y, c))$. If $C(x, b) = C(x, c)$, then we have that $C(x, b) = C(x, c) \neq C(b, c)$ otherwise the first bit of $C'(x)$ would be a 0. Then, the triangle (x, b, c) is bichromatic so it's a solution to our instance of WARD-SZABO-COLORFUL-COLLISIONS. Similarly, if $C(y, b) = C(y, c)$, then the triangle (y, b, c) is bichromatic. Now, if $C(x, b) \neq C(x, c)$ and $C(y, b) \neq C(y, c)$, then $\{C(x, b), C(x, c)\} = \{C(y, b), C(y, c)\}$ by injectivity of E_{lex} on subsets of 2 distinct elements of $\{0,1\}^n$. Then, $\{x, b, c\} \neq \{y, b, c\}$, each has three distinct elements, and either $C(x, b) = C(y, b)$, $C(x, c) = C(y, c)$ and $C(b, c) = C(b, c)$, or $C(x, b) = C(y, c)$,

$C(x,c) = C(y,b)$ and $C(b,c) = C(c,b)$. The triangle (x,b,c) is not monochromatic so this gives us a solution to our instance of WARD-SZABO-COLORFUL-COLLISIONS, either of type *iv)* if it is trichromatic, or of type *iii)* if it is bichromatic.

2. Otherwise, suppose that the first bit of $C'(y) = C'(x)$ is a 0. By construction of C', this means that $C(x,b) = C(x,c) = C(b,c) = C(y,c) = C(y,b)$. Furthermore, since $C'(x) = C'(y)$, we get that $C(x,a) = C(y,a)$. Then, $\{x,a,b\} \neq \{y,a,b\}$, each has three distinct elements, and $C(x,a) = C(y,a)$, $C(x,b) = C(y,b)$ and $C(a,b) = C(a,b)$. The triangle (x,a,b) is not monochromatic since $C(x,b) = C(b,c) \neq C(a,b)$ so this gives us a solution to our instance of WARD-SZABO-COLORFUL-COLLISIONS, either of type *iv)* if it is trichromatic, or of type *iii)* if it is bichromatic. □

7.1 A Hierarchy of Total Search Problems between weak-Pigeon and Pigeon?

In the last proof, we define a reduction to PIGEON where the circuit C' only has a range of $2^{2n-1} + 2^{n-1}$ elements. Indeed, we need exactly $\binom{2^n}{2} = 2^{2n-1} - 2^{n-1}$ elements to encode the pairs of colors. We also need exactly 2^n elements for the fourth case. However, we can map the x anywhere in that case if $C(x,a) \in \{C(a,b), C(a,c), C(b,c)\}$ because such an x would give us a bichromatic triangle. Hence, we need $2^n - 3$ colors for this case. We also need 3 extra elements for a, b and c. Hence, overall, we only need a range of $2^{2n-1} + 2^{n-1}$ elements. Thus, we get a reduction from WARD-SZABO-COLORFUL-COLLISIONS to a problem that is weaker than PIGEON (but stronger than WEAK-PIGEON), which is the following : given a circuit from $2n$ bits to $2n$ bits, either find a collision, or a preimage of one of the first $2^{2n} - (2^{2n-1} + 2^{n-1})$ elements.

More generally, we can define the problem GENERAL-PIGEON$_k^m$ as follows.

Definition 7.8 (GENERAL-PIGEON$_k^m$). *The problem* GENERAL-PIGEON$_k^m$ *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^m \to \{0,1\}^m$.

Solution: *One of the following:*

i) $x \neq y \in \{0,1\}^m$ *s.t.* $C(x) = C(y)$,

ii) $x \in \{0,1\}^m$ *s.t.* $C(x)$ *is one of the first* k *elements of* $\{0,1\}^m$.

Note that this problem gets harder as k decreases. It is trivial for $k = 2^m$, equivalent to WEAK-PIGEON for $k = 2^{m-1}$ and to PIGEON for $k = 1$.
This problem induces an entire family of intermediary problems between WEAK-PIGEON and PIGEON. It is not clear how many non-equivalent problems appear in that hierarchy. It is also unclear whether each PWPP-hard problem that is in PPP is in fact equivalent to one of these.

8 Mantel's Theorem on Triangle-Free Graphs

Next, we move on to another classical theorem in extremal graph theory. It answers the following question: What is the maximum number of edges in a triangle-free graph on N vertices?

Classical Theorem 7 (Mantel [Man07]). *If $G = (V, E)$ is a triangle-free graph on N vertices then $|E| \leq N^2/4$, and this bound is tight.*

This gives rise to the following search problem. Suppose that we are given a collection of strictly more than $N^2/4$ distinct edges for a graph on N vertices. Then, by Mantel's theorem, there must be three of these edges forming a triangle in the graph. The search problem is then to find them. We can turn this problem into a TFNP problem if we also allow evidence that two edges in the collection are in fact the same, or that an edge is in fact a loop. For practical reasons, we demand that the endpoints of every edge are given in the lexicographic order. When the edges are represented implicitly by a poly-sized circuit, we get the following problem.

Definition 8.1 (WEAK-MANTEL)**.** *The problem* WEAK-MANTEL *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{2n-1} \to \{0,1\}^n \times \{0,1\}^n$.

Solution: *One of the following:*

i) Distinct i, j, k *s.t.* $C(i), C(j), C(k)$ *form a triangle,*

ii) i *s.t.* $C(i) = (u, v)$ *with* $u \geq v$ *in the lexicographic order,*

iii) $i \neq j$ *s.t.* $C(i) = C(j)$.

Remark 8.2. *Like in the other problems, the size of the collection we receive (in this case, edges) is twice the threshold size (here,* 2^{n-2}*). However, here, we observe that the number of edges we receive as input is greater than the number of possible edges since* $2^{n-1} > \binom{2^n}{2}$*. Thus, in any instance of* WEAK-MANTEL*, there* must *be solutions of type ii) or iii).*

Theorem 8.3. WEAK-MANTEL *is PWPP-hard.*

Proof. To prove this result, we apply the graph-hash product to the complete balanced bipartite graph on 2^n vertices.

Formally, let $C : \{0,1\}^n \to \{0,1\}^{n-1}$ be an instance of WEAK-PIGEON. We define $C' : \{0,1\}^{2n-1} \to \{0,1\}^{2n-2}$ as follows. For every $x \in \{0,1\}^{2n-1}$, write $x = y \parallel z$ with $y \in \{0,1\}^n$ and $z \in \{0,1\}^{n-1}$. We then set $C'(x) = C(y) \parallel z$. Note that from any collision for C' we can retrieve a collision for C (by looking at the first n bits). Now, we define $C'' : \{0,1\}^{2n-1} \to \{0,1\}^n \times \{0,1\}^n$ as follows. For every $x \in \{0,1\}^{2n-1}$, write $C'(x) = (y \parallel z)$ with $y, z \in \{0,1\}^{n-1}$. We then set $C''(x) = (0 \parallel y, 1 \parallel z)$. We observe that C'' defines an instance of MANTEL. Note that the edges given by C'' correspond to edges of the complete balanced bipartite graph on 2^n vertices where one side of the bipartition consists of the 2^{n-1} first elements in the lexicographic order. In particular, the graph described by C'' is triangle-free, so there is no solution of type *i*). Similarly, by construction of C'', there can be no solution of type *ii*). Thus, any solution to this instance of WEAK-MANTEL is $i \neq j$ such that $C''(i) = C''(j)$. By construction of C'', this means that $C'(i) = C'(j)$ and from there we can find a collision for C. □

Theorem 8.4. WEAK-MANTEL $\in$ *PPP.*

Proof. We give a high-level overview of the proof. Since we have more edges than there are possible distinct edges, we encode the edges injectively, mapping only ill-defined edges to 0. This defines an instance of PIGEON, where a solution can only be a collision, meaning two different indices corresponding to the same edge.

With the circuit $E_{lex} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n-1}$ defined in Section 3.2, we can encode 2-subsets of $\{0,1\}^n$ using optimally many bits, that is $\left\lceil \log \binom{2^n}{2} \right\rceil = 2n - 1$.

Now, consider the following circuit $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n-1}$,

$$E(u, v) = \begin{cases} 0^{2n-1} & \text{if } u \geq v \\ E_{lex}(u, v) + 0^{2n-2}1 & \text{if } u < v \end{cases}$$

where + represents the addition in binary. Note that since the range of E_{lex} is exactly the first $\binom{2^n}{2}$ elements of $\{0,1\}^{2n-1}$ in the lexicographic order, if $E(u, v) = 0^{2n-1}$, it must be that $u \geq v$.

Let $C : \{0,1\}^{2n-1} \to \{0,1\}^n \times \{0,1\}^n$ be an instance of WEAK-MANTEL. For every $x \in \{0,1\}^{2n-1}$, we set $C'(x) = E(C(x))$. Then, $C' : \{0,1\}^{2n-1} \to \{0,1\}^{2n-1}$ is an instance of PIGEON.

Now, suppose that we have a solution to this instance of PIGEON. If it is x such that $C'(x) = 0^{2n-1}$, then $E(C(x)) = 0^{2n-1}$ which means that $C(x) = (u, v)$ with $u \geq v$ so x is a solution to our instance of WEAK-MANTEL. If it is $x \neq y$ such that $C'(x) = C'(y)$. If $C'(x) = 0^{2n-1}$, by the first case we have that x is a solution to the instance of WEAK-MANTEL. Now, if $C'(x) \neq 0^{2n-1}$, then it means that $E(C(x)) + 0^{2n-2}1 = E(C(y)) + 0^{2n-2}1$ so $E(C(x)) = E(C(y))$. By injectivity of E on well-defined inputs (that is inputs of the form (u, v) with $u < v$), this means that $C(x) = C(y)$ which is a solution to our original instance of WEAK-MANTEL. □

Remark 8.5. *Similarly to the proof that* WARD-SZABO-COLLISIONS $\in$ *PPP, we only use the last two types of solutions, which suggests that what makes this problem easier than* PIGEON *is only the fact that we are given more edges than there are different possible edges in a graph on* 2^n *vertices.*

Remark 8.6. *In fact, this last proof shows that* WEAK-MANTEL *reduces to* GENERAL-PIGEON$^{2n-1}_{2^{n-1}}$*.*

Mantel's theorem states that there is a unique triangle-free graph on $2N$ vertices that has N^2 edges, it is the complete bipartite graph $K_{N,N}$. Now, consider any labelling of the vertices of $K_{N,N}$. If for every label x, the vertices labelled x and $x+1 \mod 2N$ were on the same side of the bipartition, then all the vertices would be on the same side of the bipartition, which is impossible. Hence, there must be 2 vertices labelled x and $x+1 \mod 2N$ on different sides of the bipartition, and therefore there must be an edge between them. Thus, the following problem is total.

Definition 8.7 (MANTEL)**.** *The problem* MANTEL *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{2n-2} \to \{0,1\}^n \times \{0,1\}^n$*.*

Solution: *One of the following:*

i) *Distinct* i, j, k *s.t.* $C(i), C(j), C(k)$ *form a triangle,*

ii) i *s.t.* $C(i) = (u, v)$ *with* $u \geq v$ *in the lexicographic order,*

iii) $i \neq j$ *s.t.* $C(i) = C(j)$*,*

iv) i *s.t.* $C(i) = (u, v)$ *with* $v = u + 1 \mod 2^n$ *when we consider* u *and* v *as integers.*

Theorem 8.8. MANTEL *is PPP-hard.*

Proof. To prove this result, we do the graph-hash product on the complete balanced bipartite graph on 2^n vertices, where one side of the bipartition consists of the first 2^{n-1} vertices in the lexicographic order. We make sure to map 0 into the edge $(01^{n-1}, 10^{n-1})$, which is the only edge satisfying *iv*) in that graph.

Formally, let $C : \{0,1\}^{2n-2} \to \{0,1\}^{2n-2}$ be an instance of PIGEON.
We define a circuit $C' : \{0,1\}^{2n-2} \to \{0,1\}^n \times \{0,1\}^n$ as follows. Let $x \in \{0,1\}^{2n-2}$. If $C(x) = 0^{2n-2}$, we set $C'(x) = (0 \parallel 1^{n-1}, 1 \parallel 0^{n-1})$. If $C(x) = 1^{n-1} \parallel 0^{n-1}$, we set $C'(x) = (0^n, 1 \parallel 0^{n-1})$. Otherwise, if $C(x) = (u, v)$, we set $C'(x) = (0 \parallel u, 1 \parallel v)$. C' has polynomial size and defines an instance of MANTEL.

Now, suppose that we have a solution to this instance of MANTEL. Like in the proof of Theorem 8.8, this solution cannot be of type *i*) because the graph described by C' is bipartite hence triangle-free, and it cannot be of type *ii*) neither, by construction. If this solution is of the form $i \neq j$ such that $C'(i) = C'(j)$, by construction of C' it means that $C(i) = C(j)$ which is a collision for C. If this solution is of the form i such that $C'(i) = (u, v)$ with $v = u+1 \mod 2^n$, then by definition of C', it can only be that $C'(i) = (0 \| 1^n, 1 \| 0^n)$. By construction of C', this means that $C(i) = 0^{2n-2}$ hence x is a solution to the original instance of PIGEON. □

8.1 Generalization with Turán's Theorem

Mantel's theorem investigates the maximum number of edges in a triangle-free graph on N vertices. Similarly, one could wonder about the maximum number of edges in a graph on N vertices that does not contain a clique on r vertices, where $r \geq 3$ is an arbitrary constant. This problem was solved by Turán in 1941.

Classical Theorem 8 (Turán [Tur41])**.** *If* $G = (V, E)$ *is a graph on* $N = |V|$ *vertices that does not contain any* $r+1$*-clique, then* $|E| \leq (1 - \frac{1}{r})\frac{N^2}{2}$ *and this bound is tight when* r *divides* N*.*

Now, suppose that we are given a list of strictly more than $(1 - \frac{1}{r})\frac{N^2}{2}$ edges for a graph on N vertices. Then, by Turán's theorem, if all these edges are distinct, the graph must contain an $r+1$-clique. This induces a total search, namely that of finding the vertices of such a clique. If the edges are given implicitly via a Boolean circuit which on input i returns the endpoints of the i-th edge, we get the following TFNP problem.

Definition 8.9 (weak-Turán$_r$)**.** *The problem* weak-Turán$_r$ *is defined by the relation*

Instance: *A Boolean circuit* $C\colon \{0,1\}^{2n-1} \to \{0,1\}^n \times \{0,1\}^n$.

Solution: *One of the following:*

i) *Distinct* $i_1, i_2, \dots i_{(r+1)(r+2)/2}$ *such that* $C(i_1), C(i_2), \dots C(i_{(r+1)(r+2)/2})$ *are the edges of an* $r+1$*-clique,*

ii) i *s.t.* $C(i) = (u,v)$ *with* $u \geq v$ *in the lexicographic order,*

iii) $i \neq j$ *s.t.* $C(i) = C(j)$.

Remark 8.10. *Note that* r *can be any polynomial in* n *in the previous definition and it would still define a* TFNP *problem.*

Theorem 8.11. *For every* $r_1 < r_2$*, there is a reduction from* weak-Turán$_{r_1}$ *to* weak-Turán$_{r_2}$.

Proof. Let $C : \{0,1\}^{2n-1} \to \{0,1\}^n \times \{0,1\}^n$ be an instance of weak-Turán$_{r_1}$. Now, we interpret it as an instance of weak-Turán$_{r_2}$. Suppose that we have a solution to this instance of weak-Turán$_{r_2}$.
If we have $(r_2+1)(r_2+2)/2$ edges that form an r_2+1-clique, it suffices to remove some of them to get the edges of an r_1+1-clique. Otherwise, any solution of type *ii*) or *iii*) for weak-Turán$_{r_2}$ immediately translates into a solution of the same type for weak-Turán$_{r_1}$. □

Theorem 8.12. *For every* $r \geq 2$*,* weak-Turán$_r$ *is* PWPP*-hard.*

Proof. It is enough to notice that WeakTurán$_2$ is exactly weak-Mantel, which is PWPP-hard by Theorem 8.3. Then, apply Theorem 8.11. □

Theorem 8.13. *For every* $r > 2$*,* weak-Turán$_r \in$ PPP*.*

The proof is exactly similar to the proof of *Theorem* 8.4. In this case too, it appears that what makes the problem easier than Pigeon is that we are given too many edges.

Turán's theorem states that there if r divides N, there is a unique graph on N vertices that does not contain any $r+1$-clique and that has the maximum number of edges. This graph is the complete r-partite graph, where each part has size N/r. Like previously, there must be 2 vertices labelled x and $x+1$ mod $2N$ with an edge between them. We denote by N the largest multiple of r that is at most 2^n, and set $M = (1-\frac{1}{r})\frac{N^2}{2}$. Thus, the following problem is in TFNP.

Definition 8.14 (Turán$_r$)**.** *The problem* Turán$_r$ *is defined by the relation*

Instance: *The following:*

1. *A Boolean circuit* $C\colon \{0,1\}^{2n-1} \to \{0,1\}^n \times \{0,1\}^n$*; and,*
2. *Two integers* N *and* M.

Solution: *One of the following:*

i) 0 *if* r *does not divide* N*, or if* $N > 2^n$*, or if* $N + r \leq 2^n$*, or if* $M \neq (1-\frac{1}{r})\frac{N^2}{2}$,

ii) i *s.t.* $C(i) = (u,v)$ *with* $u \geq N$ *or* $v \geq N$*, and* $i < M$

iii) *Distinct* $i_1, i_2, \dots i_{(r+1)(r+2)/2}$ *such that* $C(i_1), C(i_2), \dots C(i_{(r+1)(r+2)/2})$ *are the edges of an* $r+1$*-clique, and* $i_j < M$ *for every* j,

iv) i *s.t.* $C(i) = (u,v)$ *with* $u \geq v$ *in the lexicographic order, and* $i < M$,

v) $i \neq j$ *s.t.* $C(i) = C(j)$*, and* $i, j < M$,

vi) i *s.t.* $C(i) = (u,v)$ *with* $v = u + 1 \mod 2^n$ *when we consider* u *and* v *as integers, and* $i < M$.

This last problem is in TFNP. However, we cannot adapt the proof of PPP-hardness of Mantel to it in a straightforward way and, in fact, it is open whether this problem is PPP-hard.

References

[Bar73] Zsolt Baranyai. Infinite and finite sets, vol. 1. proceedings of a colloquium held at Keszthely, June 25 – July 1, 1973. Dedicated to Paul Erdős on his 60th Birthday. *J. Symb. Log.*, 1:91–108, 1973.

[BJP$^+$19] Frank Ban, Kamal Jain, Christos H. Papadimitriou, Christos-Alexandros Psomas, and Aviad Rubinstein. Reductions in PPP. *Inf. Process. Lett.*, 145:48–52, 2019.

[Cay89] Arthur Cayley. A theorem on trees. *Quarterly Journal of Mathematics*, 23:376–378, 1889.

[Cov73] Thomas M. Cover. Enumerative source encoding. *IEEE Transactions on Information Theory*, 19(1):73–77, 1973.

[DGP09] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM J. Comput.*, 39(1):195–259, 2009.

[Dil50] Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics 51*, pages 161–166, 1950.

[EK99] Ömer Egecioglu and Alastair King. Random walks and Catalan factorization. 1999.

[EKR61] Paul Erdős, Chao Ko, and Richard Rado. Intersection theorems for systems of finite sets. *The Quarterly Journal of Mathematics*, 12(1):313–320, 01 1961.

[ER60] Paul Erdös and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, s1-35(1):85–90, 1960.

[Erd47] Paul Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53(4):292–294, 1947.

[FG18] Aris Filos-Ratsikas and Paul W. Goldberg. Consensus halving is PPA-complete. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 51–64. ACM, 2018.

[HV21] Pavel Hubáček and Jan Václavek. On search complexity of discrete logarithm. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia*, volume 202 of *LIPIcs*, pages 60:1–60:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[Jeř16] Emil Jeřábek. Integer factoring and modular square roots. *J. Comput. Syst. Sci.*, 82(2):380–394, 2016.

[JPY88] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988.

[KNY19] Ilan Komargodski, Moni Naor, and Eylon Yogev. White-box vs. black-box complexity of search problems: Ramsey and graph property testing. *J. ACM*, 66(5), jul 2019.

[Kra05] Jan Krajíček. Structured pigeonhole principle, search problems and hard tautologies. *J. Symb. Log.*, 70(2):619–630, 2005.

[Man07] Willem Mantel. Problem 28 (Solution by H. Gouwentak, W. Mantel, J. Teixeira de Mattes, F. Schuh and W. A. Wythoff). *Wiskundige Opgaven*, 18:60–61, 1907.

[Meh18] Ruta Mehta. Constant rank two-player games are PPAD-hard. *SIAM J. Comput.*, 47(5):1858–1887, 2018.

[Mer79] Ralph Charles Merkle. *Secrecy, Authentication, and Public Key Systems.* PhD thesis, Stanford, CA, USA, 1979. AAI8001972.

[MP91] Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2):317–324, 1991.

[Pap94] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994.

[Pru18] Heinz Prufer. Neuer Beweis eines Satzes über Permutationen. *Archiv der Mathematischen Physik*, 27:742–744, 1918.

[Ram30] Frank P. Ramsey. On a Problem of Formal Logic. *Proceedings of the London Mathematical Society*, s2-30(1):264–286, 01 1930.

[Spe28] Emanuel Sperner. Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift*, 27(1):544–548, 1928.

[SZZ18] Katerina Sotiraki, Manolis Zampetakis, and Giorgos Zirdelis. Ppp-completeness with connections to cryptography. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 148–158. IEEE Computer Society, 2018.

[Tur41] Paul Turán. On an extremal problem in graph theory (in hungarian). *Matematikai és Fizikai Lapok*, 48:436–452, 1941.

[WS95] Coburn Ward and Sandor Szabo. On swell-colored complete graphs. 06 1995.

A Efficient algorithm for the explicit Ramsey problem

The following proof of Ramsey's theorem is folklore. Recall the statement of the theorem

Ramsey [Ram30] *Any edge-coloring of the complete graph on n vertices with two colors contains a monochromatic clique of size at least $\frac{1}{2}\log n$.*

Proof. Let $G = (V, E)$ be the complete graph on n vertices, and $c : E \to \{0, 1\}$ be a two-coloring of its edges.
Pick an arbitrary vertex $v_1 \in V$.
v_1 has $n - 1$ adjacent edges so at least $n/2$ of them have the same color by the pigeonhole principle.
Let c_1 be that color and $V_1 = \{v \in V \setminus \{v_1\}, c(v, v_1) = c_1\}$.
Then, V_1 has at least $n/2$ elements.

Next, pick an arbitrary vertex $v_2 \in V_1$.
There are at least $n/2 - 1$ edges between v_2 and another vertex in V_1. Like before, at least $n/4$ of them have the same color by the pigeonhole principle.
Let c_2 be that color and $V_2 = \{v \in V_1 \setminus \{v_2\}, c(v, v_2) = c_2\}$.

That way, we proceed to build by induction a finite family of vertices (v_i), a finite family of colors (c_i) and a finite family of sets of vertices (V_i) with the following properties :
• For every i, $V_i \subset V_{i-1}$.
• For every i, V_i has size at least $n/2^i$.
• For every i, $v_{i+1} \in V_i$.
• For every i and for every $u \in V_i$, we have $c(v_i, u) = c_i$.

In particular, note that the second point implies that we have at least $\log(n) - 1$ V_i's, thus we can construct at least $\log(n)$ v_i's (since we need that V_i is not empty to build v_{i+1}).
This means that we define at least $\log(n) - 1$ colors c_i. By the pigeonhole principle, at least $\log(n)/2$ of them are the same, say color $c \in \{0, 1\}$.
Let $k = \log(n)/2$.
Pick $i_1, i_2, \ldots, i_k$ such that $c_{i_1} = c_{i_2} = \ldots = c_{i_k} = c$.
We claim that the subgraph whose vertices are $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ is monochromatic.
Indeed, let $j < l \in [k]$.
Then, $v_{i_l} \in V_{i_l - 1} \subset V_{i_l - 2} \subset \ldots \subset V_{i_j}$, so by the fourth point, we get that $c(v_{i_j}, v_{i_l}) = c_{i_j} = c$. □

Now, note that this proof is constructive and yields an algorithm to find a monochromatic subgraph of size $k = \log(n)/2$ of the complete graph on n vertices.
In this algorithm, we have $\log(n)$ iterations, and each of them can be done in time $O(n)$, so overall we get an algorithm running in $O(n \log(n))$ time.