SAT-Based Search for Minwise Independent Families

Enrico Iurlano(✉) and Günther R. Raidl

Algorithms and Complexity Group, TU Wien, Austria
{eiurlano,raidl}@ac.tuwien.ac.at

Abstract. Optimal minwise independent families of permutations are combinatorial structures designed to realize a large set of highly symmetric monotonicity patterns while using as few permutations as possible. Such families play a central role in efficient Jaccard similarity estimation and related applications in document analysis and machine learning. While practical variants often tolerate slight relaxations of their attributes, we focus on the computational construction of minwise independent families in their pure form. We introduce a constraint-based framework that (i) provides a direct SAT encoding of the underlying combinatorial conditions and (ii) extends a heuristic by Mathon and van Trung to guide the search within this constrained space. Our results show that this hybrid approach is effective in discovering new, provably optimal families and yields substantial speed-ups compared to previous methods. The framework also offers structural insights into these combinatorial objects and supports them with certified optimality proofs.

Keywords: Minwise independence · SAT solving · Heuristic search · Symmetry breaking · Certified optimality

1 Introduction

A family $\mathscr{F}$ of permutations on a finite universe U is (k-restricted) minwise independent, if it ensures that a random permutation π drawn from $\mathscr{F}$ and applied to any $X \subseteq U$ (of cardinality at most k) exhibits perfect fairness in the sense that each element $x^* \in X$ has the same chance of being mapped to the minimum of the values in $\{\pi(x) : x \in X\}$. These families are a crucial ingredient of the well-known `MinHash` algorithm [4], which was proposed in the 1990s to rapidly estimate the similarity between documents in large World Wide Web page collections. Since then, these structures have inspired thousands of subsequent studies in the broader field of big data and machine learning [35]. More precisely, minwise independence is strongly linked to established techniques such as b-bit minwise hashing [22], one-permutation hashing [23], and locality-sensitive hashing [14], which are used in applications including spam detection [22], dimensionality reduction [35], derandomization [28], and computational geometry [28].

This work was partially supported by the program VGSCO of the Austrian Science Fund (FWF) [10.55776/W1260-N35].

At a technical level, minwise independence is used to accurately approximate the pairwise Jaccard similarity between high-dimensional data based on precomputed *sketches* [5] associated with the data points, i.e., subsets of a large, finite universe U. The Jaccard similarity $r(A, B) := |A \cap B|/|A \cup B|$ of two sets $A, B \subseteq U$ can be estimated by repeatedly subjecting A and B to random permutations of U and computing a simple statistic on the results [5].

The main challenge in this sampling approach is that, for large-scale universes U, it is difficult to draw a permutation of U (even approximately) uniformly at random: The combinatorial explosion of the symmetric permutation group on n elements makes it impossible even for state-of-the-art pseudo random number generators (PRNG) to guarantee full sampling of its elements before reaching the PRNG's period-length [26]; this applies in particular to the PRNG *Mersenne Twister* [26] being the standard[1] of the `C++` programming language. Therefore, Broder et al. [5] introduced minwise independent families of permutations that can be generated with far fewer members than the full symmetric group on U, yet any random permutation drawn from such a smaller family serves as an equivalent replacement for a random permutation. The impact in the field of big data is largely influenced by relaxed and approximate forms of such families, which accept a certain degree of inaccuracy for practical purposes. Often, certain families of practical hash functions are employed as a compromise; see Broder et al. [5] and subsequent work.

Optimality-achieving algorithms or constructions for these families are known only for specific parameter choices [17]. Therefore, providing a general framework for finding and studying provably optimal representatives (at computationally tractable scales) adds valuable insight into their nature. Indeed, constructive approaches for generating minwise independent families [32] and related structures often benefit from knowledge of optimal behavior on smaller scales as they are designed in a recursive [19,32], product-type [7], replicative [34], or extensible [13] manner. Such a tool may also help to disprove or empirically support conjectures on minwise independence that might be inferable by analogy from open conjectures in related fields [21,29].

For related structures [13,24,29] the development of exact computational approaches has been initiated. Notably, in computational (non-)existence proofs, their proposed algorithms exhibit long runtimes even for small parameter values [13,29]. This raises questions about the existence of better performing alternatives. On the other hand, the aforementioned works have the disadvantage that no verification across diverse and independent software platforms for their claims of optimality is available, which would be important to enhance trust.

Our contribution. We propose a framework to address questions on the existence and structural decomposability of minwise independent families of a postulated size, thereby providing insights on (optimal) such representatives with specific characteristics. The framework relies on a reduction to the Boolean satisfiability problem, i.e., a SAT encoding, which is then solved with a modern

[1] https://cplusplus.com/reference/random/mt19937/ (accessed 2026-02-12)

SAT solver. To manage the high-dimensional search space, our encoding employs problem-specific symmetry breaking whereas to accelerate heuristic solution finding, we adapt a group-theoretic decomposition strategy to our setting; originally, it was proposed for a similar structure [24,29] and implemented as a backtracking algorithm. For parameter instances that allow direct comparison with [29], we observe a dramatic speed-up in terms of computation time by a factor of up to 2.09×10^5. Using our SAT framework, we can also detect nonexistence and subsequently certify it with independent software. Furthermore, the framework supports flexible customization and the integration of additional assumptions for further studies. As a byproduct of potentially broader interest, we provide pure-SAT evaluations (without resorting to bitvector arithmetic) of certain permutation attributes arising from symbolic permutation emulation.

The paper is structured as follows. Relevant terminology and related work are reported in Sect. 1.1–1.2. The SAT encoding is derived in Sect. 2 and extended to subgroup-parametrized encodings in Sect. 3. Experimental evaluations are reported in Sect. 4 and concluding remarks in Sect. 5.

1.1 Preliminaries

In the following, we provide the most relevant definitions and properties concerning minwise independence and the methodology we develop later.

Let $[n] := \{1, \ldots, n\}$ and denote by S_n the symmetric group of all $n!$ permutations of $[n]$. Let us introduce sets of *semiordered patterns* $\mathsf{SOP}(n, j) := \{(s_1, \ldots, s_j) \in [n]^j : s_2 < s_3 < \cdots < s_j \text{ and } s_1 \notin \{s_2, \ldots, s_j\}\}$ for $1 \leq j \leq n$. In the following we always assume that appearing families are indexed by $[d]$ for some $d \in \mathbb{N}$. We refer to the number d of (not necessarily distinct) members of a family as its size.

Following [5], we say that a finite non-empty family $\mathscr{F} = (\pi_1, \ldots, \pi_d)$ of permutations $\pi_i \in S_n$, $i = 1, \ldots, d$ is *k-restricted minwise independent* if for each $j \in [k]$, for each $X \subseteq [n]$ with $|X| = j$, and each x^* in X,

$$\Pr\left[\pi(x^*) = \min_{x \in X} \pi(x) \,\middle|\, \pi \text{ is drawn uniformly at random from } \mathscr{F}\right] = \frac{1}{j}. \quad (1)$$

It will be convenient to consider an equivalent replacement for (1) given by

$$|\{i \in [d] : \pi_i(s_1) = \min\{\pi_i(s_1), \pi_i(s_2), \ldots, \pi_i(s_j)\}\}| = \frac{d}{j}, \quad (2)$$

for each $j \in [k]$ and each $(s_1, s_2, \ldots, s_j) \in \mathsf{SOP}(n, j)$. This implies that d must be a multiple of $\operatorname{lcm}([k])$, the least common multiple of the numbers $1, 2, \ldots, k$. We call a k-restricted minwise independent family $\mathscr{F} = (\pi_1, \ldots, \pi_d) \subseteq S_n$ *optimal* or *minimally-sized* if no k-restricted minwise independent family $\mathscr{G} \subseteq S_n$ has fewer than d members.

Example 1. Let $\mathscr{F} := (\pi_1, \ldots, \pi_6) \subseteq S_4$ be the family formed by the following six permutations in Fig. 1. No matter which $d \times j$ submatrix of $j \leq 3$ columns we

$$\begin{array}{lcccc}
\pi_1 = & 2 & 3 & 1 & 4 \\
\pi_2 = & 1 & 4 & 2 & 3 \\
\pi_3 = & 4 & 1 & 2 & 3 \\
\pi_4 = & 2 & 3 & 4 & 1 \\
\pi_5 = & 2 & 1 & 4 & 3 \\
\pi_6 = & 4 & 3 & 2 & 1
\end{array}$$

Fig. 1. A 3-restricted minwise independent and optimal family. Each row corresponds to a permutation: a (throughout the row) unique value is assigned to each column-position.

select together with an attribute of "speciality" for one of its columns, the number of rows of this submatrix in which this special column entry is row-wise the smallest is invariant of the submatrix-choice and subsequent speciality-choice. More precisely, notice that for $(2,1,4) \in \mathsf{SOP}(4,3)$, we have $2 = |\{\pi_3, \pi_5\}| = 6/3$ and $\pi_3(2) = 1 = \min\{1,4,3\} = \min\{\pi_3(2), \pi_3(1), \pi_3(4)\}$ as well as $\pi_5(2) = 1 = \min\{1,2,3\} = \min\{\pi_5(2), \pi_5(1), \pi_5(4)\}$. A respective observation can be made for all 12 elements in $\mathsf{SOP}(4,3)$, all 12 elements in $\mathsf{SOP}(4,2)$, and all four elements in $\mathsf{SOP}(4,1)$. Optimality follows from the fact that the family size must be a multiple of $\operatorname{lcm}(1,2,3) = 6$, and 6 itself already suffices.

Let us denote by $S_{n,k}$ the set of injective functions $\sigma : [k] \to [n]$, henceforth called *k-subpermutations*. The latter are used to strengthen the requirements of k-restricted minwise independent families towards *k-rankwise independent* families [32] satisfying for each subpermutation $\sigma \in S_{n,k}$

$$|\{i \in [d] : \pi_i(\sigma(1)) < \cdots < \pi_i(\sigma(k))\}| = \frac{d}{k!}. \tag{3}$$

In the sequel, we recall some concepts from group theory. Let G be a (not necessarily commutative) group with operation $\circ : G \times G \to G$, neutral element $e \in G$, and inversion $(\cdot)^{-1} : G \to G$. For each subgroup H of G—a subset of G closed under the group operations—a *left-coset* is a set of the form $gH := \{g \circ h : h \in H\}$ for an arbitrary $g \in G$. Similarly, *right-cosets* $Hg := \{h \circ g : h \in H\}$ can be considered for any $g \in G$. We focus on subgroups of the symmetric group S_n, where $\circ$ is function composition. We use the standard function composition convention $f \circ g := f(g(\cdot))$, i.e., f is applied to the output of g.

1.2 Related work

In their foundational work, Broder et al. [5] proposed an exponential-size construction of k-restricted minwise independent families for $k = n$. Subsequently, Itoh et al. [17] gave a provably optimal constant-factor improvement for this case, and Tarui et al. [32] derived polynomial-size asymptotic upper bounds for the cases $k \in \{3,4\}$ via finite-geometric methods. Error-tolerant approximate versions of biased-distribution settings were also examined in [5]. The work [17]

also introduced the concept of rankwise independence, identified it as a proper special case of minwise independence, and provided the subsequent useful observations. In contrast, Matoušek and Stojaković [25] studied slight relaxations.

Remark 1. Tarui et al. [32] observed that (2) for $j \leq 3$ is equivalent to (3) with $k = 3$, and consequently 3-restricted minwise independence coincides with 3-rankwise independence. Itoh et al. [17, p. 141] give a transformation that turns a k-restricted min- (or rankwise) independent family $\mathscr{F} \subseteq S_n$ into one in $S_{\tilde{n}}$ of the same size, for any $\tilde{n} \leq n$. By contrapositive, non-existence of k-restricted minwise independent families $\widetilde{\mathscr{F}} \subseteq S_{\tilde{n}}$ of d members implies non-existence for such families $\mathscr{F} \subseteq S_n$ with larger $n \geq \tilde{n}$, provided $|\mathscr{F}| = d$.

Asymptotic bounds on the size of optimal k-restricted minwise independent families appear in [2,17,18]. Upper bounds for optimal rankwise independent families have also been derived [15,20] as well as lower bounds [2].

It was recently recognized [19] that requirement (3) "up to isomorphy" exactly characterizes perfect sequence covering arrays [34]. For the latter, several construction [12,21,34] and search approaches [13,24,29] have been proposed as alternatives to the ones addressing mainly the asymptotics [2,32] of rankwise independence. The enumerative strategies of Mathon and van Trung [24] and Na et al. [29] are based on the assumption that families are unions of cosets in S_n; both works motivate our SAT-based adaptation of their group-decomposition approach. For minwise independence, the following theorem seems to be the first result proving the existence of a minwise independent family by a particular decomposition into (non-trivial) *right-cosets*. Although originally stated without group-theoretic language, the result is rephrased here to match our framework; see Sect. 3.

Theorem 1 (adapted from [3]). *Let $k \geq 3$ be odd and $\mathscr{F} = (\theta_1, \ldots, \theta_d) \subseteq S_n$ be k-restricted minwise independent. With $G = \{\gamma_1, \gamma_2\} := \{\mathrm{id}, \sigma\}$ denoting the subgroup of S_n containing the identity permutation and the order-reversing permutation $\sigma : [n] \to [n]$, $i \mapsto n + 1 - i$, we have that $(\gamma_\ell \circ \theta_m : \ell \in [2], m \in [d])$ is a $(k+1)$-restricted minwise independent family of size $2d$.*

Another connection of minwise independence to group theory has been theoretically studied within a more restrictive setting in [6].

Remark 2. While deriving the apparently best lower bound on the size of rankwise independent families, Bargachev [2] asked whether there is a "natural bijection" between the class of permutations having k so-called waste indices and the class of k-partial derangements. In passing we highlight that such a bijection follows from a slight adaptation and generalization of [36, Proposition 2.1] which bijects on top of the cycle notation of permutations [9]; more details can be found in the electronic appendix. [2]

[2] https://www.ac.tuwien.ac.at/files/resources/instances/minw-idp/

2 A SAT approach

In this section we provide an initial SAT encoding. In the upcoming Sect. 3 we then refine the encoding into two more sophisticated ones, each constituting a generalization.

The main idea relies on the natural approach to represent permutations via their order-theoretic incidence structure already used, e.g., in [1]. In fact, if $\pi \in S_n$ is a permutation, it is uniquely determined by full knowledge on $X^\pi = (x^\pi_{i,j})_{i,j=1}^n \in \{0,1\}^{n\times n}$ capturing by $x^\pi_{i,j}$ the information if $\pi(i) < \pi(j)$ (1 in the affirmative case, otherwise 0). Note that X^π can be seen as the incidence matrix of a strict total order on $[n]$ representing the permutation. We can, in fact, recover the j-th entry of the permutation via $\pi(j) = 1 + \sum_{i=1}^n x^\pi_{i,j}$. All solutions for $x^\pi_{i,j}$ subject to the following constraints (irreflexivity, asymmetry, totality, and transitivity) correspond to a permutation:

$$\neg x^{\pi_\ell}_{i,i} \quad \text{for } \ell \in [d]\,,\ i \in [n]\,; \tag{4}$$

$$\neg x^{\pi_\ell}_{i,j} \vee \neg x^{\pi_\ell}_{j,i} \quad \text{for } \ell \in [d]\,,\ i \in [n]\,,\ j \in [n]\,; \tag{5}$$

$$x^{\pi_\ell}_{i,j} \vee x^{\pi_\ell}_{j,i} \quad \text{for } \ell \in [d]\,,\ i \in [n]\,,\ j \in [n]\,; \tag{6}$$

$$\neg x^{\pi_\ell}_{i,j} \vee \neg x^{\pi_\ell}_{j,h} \vee x^{\pi_\ell}_{i,h} \quad \text{for } \ell \in [d]\,,\ i \in [n]\,,\ j \in [i+1\,{:}\,n],\ h \in [n] \setminus \{i,j\}. \tag{7}$$

These constraints are already stated in conjunctive normal form (CNF). Here for two integers p and q, we denote by $[p\,{:}\,q] := \{p, p+1, \dots, q\}$. Constraints (5)–(6) in fact are equivalent to $x_{j,i} \leftrightarrow \neg x_{i,j}$ for $i < j$; by substitution, it is hence possible to rely exclusively on the strict upper-diagonal entries of X^π thereby allowing us to omit (4)–(6).

To encode property (2) in the definition of minwise independence, we need to impose specific cardinality constraints which are stated in (8)—note that $j \leq 3$ is intentionally excluded due to a more favorable replacement in (9):

$$\sum_{\ell=1}^{d} \left[\bigwedge_{h=2}^{j} x^{\pi_\ell}_{s_1,s_h} \right] \leq d/j \quad \text{for } j \in [4\,{:}\,k],\ s \in \mathsf{SOP}(n,j); \tag{8}$$

$$\sum_{\ell=1}^{d} \left[\bigwedge_{h=2}^{3} x^{\pi_\ell}_{\sigma(h-1),\sigma(h)} \right] \leq d/3! \quad \text{for } \sigma \in S_{n,3}. \tag{9}$$

In fact, we avoid the higher upper bound $d/3$ (which in (8) would result for $j = 3$) and we can also relinquish the case $j = 2$—this is justified by the aforementioned particularity that 3-restricted minwise independence is precisely 3-rankwise independence (see Remark 1).

Note that upper bounds instead of equalities appear in (8)–(9), which provide a considerable simplification in view of a conversion to CNF. These seemingly weaker conditions are, however, equivalent to the ones with equalities: Any not attained upper bound in (8)–(9) would imply a strict excess of the upper bound for some other $s' \in \mathsf{SOP}(n,j)$ respectively $\sigma' \in S_{n,3}$—infeasibility would be a consequence; in fact, the number of patterns/subpermutation conditions met by

a family $\mathscr{F}$ is fixed. How the cardinality constraints are technically translated into CNF is deferred to Sect. 4.

Remark 3. If the interest is in modeling k-rankwise independence, we can simply replace (8)–(9) with inequalities $\sum_{\ell=1}^{d} \left[\bigwedge_{h=2}^{k} x^{\pi_\ell}_{\sigma(h-1),\sigma(h)}\right] \leq d/k!$ for $\sigma \in S_{n,k}$.

For symmetry breaking, for a permutation π, we consider the binary string resulting from $X^\pi = (x^\pi_{i,j})_{i,j=1}^n$ by concatenating all side-diagonals of X^π lying above the main diagonal to form a string, denoted by $z_{\mathrm{cat}}[X^\pi] := (x^\pi_{i+1,i+j} : j = 2, \ldots n, i = 0, \ldots, n-j)$.

It is then natural, without loss of generality, to enforce an ordering on X^{π_ℓ}, $\ell = 1, \ldots, d$, defined through an ordering of their associated strings $z_{\mathrm{cat}}[X^{\pi_\ell}]$. With $\preceq$ denoting the lexicographical ordering relation on equal-length binary strings, i.e., $s_1 s_2 \cdots, s_M \preceq t_1 t_2 \cdots t_M$ iff $\sum_{h=1}^{M}(t_h - s_h)2^{M-h} \geq 0$, we thus require

$$z_{\mathrm{cat}}[X^{\pi_{\ell+1}}] \preceq z_{\mathrm{cat}}[X^{\pi_\ell}] \quad \text{for } \ell \in [d-1]\,. \tag{10}$$

The identity permutation, whose upper diagonal part of the incidence matrix consists of exclusively 1-entries, is hence the maximum element with respect to this ordering. For $\pi \in S_n$, the string $z_{\mathrm{cat}}[X^\pi]$ is of length $n(n-1)/2$. Therefore, we might prefer the following weakened constraint depending on a user-defined "accuracy" parameter $L \in [0\,{:}\,n(n-1)/2]$ (denoting for a string s the substring consisting of the first L entries of s by $s_{[1..L]}$):

$$z_{\mathrm{cat}}[X^{\pi_{\ell+1}}]_{[1..L]} \preceq z_{\mathrm{cat}}[X^{\pi_\ell}]_{[1..L]} \quad \text{for } \ell \in [d-1]\,. \tag{11}$$

In [30, entry A036604] one can look up $\mathsf{A}(n)$, the minimum number of (dynamically choosable) pairwise comparisons needed to uniquely determine a strict total order on $[n]$; see also the follow-up paper [31]. In the following we regard the choice $L := \mathsf{A}(n)$ as a meaningful guideline but we could also choose a different value for L, if an alternative trade-off between symmetry breaking and lower complexity of the encoding is preferred. Finally, in this setting we can without loss of generality add the following constraint (12) as we preserve minwise independence by composing each permutation in the family with π_1^{-1} and afterwards bringing the transformed family into lexicographic order.

$$\pi_1 = \mathrm{id}\,;\ \text{i.e., unit clauses} \quad x^{\pi_1}_{i,j} \quad \text{for } i \in [n]\,,\ j \in [i+1\,{:}\,n]. \tag{12}$$

3 Incorporating a group-theoretic decomposition

This section emulates in SAT the group-theoretic heuristic of Mathon and van Trung [24] (see also [29]) on top of the encoding in Sect. 2.

As the approaches in the latter two works are—up to isomorphy, see [19]—applied to rankwise independence it is a natural question if they are a useful approach when rankwise independence is now relaxed towards minwise independence: Clearly, this decomposition property is inherited by the relaxation,

but according to asymptotic results a minwise independent family obtained this way could in general be far from optimal [17,18]. Therefore, it is a priori unclear whether this approach can be successful for the search of (near-)optimal minwise independent representatives, too.

Moreover, let us anticipate that it is currently unknown whether solutions always carry the underlying structure which will be postulated in what follows. Hence, this section provides a heuristic; a negative answer here does not entail non-existence in the full search space.

Let us address the required steps for answering this question: In the spirit of [24,29] we will hence assume that our families $(\pi_1, \dots, \pi_d)$ with $\pi_i \in S_n$ such that $\operatorname{lcm}([k])$ is a divisor of d have the following underlying structure: There is a subgroup $G = \{\gamma_1, \dots, \gamma_q\}$ of S_n whose order $q = |G|$ is a divisor of d and for which there are "offsets" $\theta_1, \dots, \theta_{d/|G|} \in S_n$ such that

$$\mathscr{F} = (\pi_1, \dots, \pi_d) = (\theta_\ell \circ \gamma_m)_{(\ell,m)\in[d/|G|]\times[|G|]}. \tag{13}$$

Condition (13) models the coincidence of $(\pi_1, \dots, \pi_d)$ with a union of $d/|G|$ (not necessarily distinct) left-cosets of a subgroup G of S_n. By fixing a non-trivial subgroup G of S_n, the number of decision variables is reduced, as every choice for an offset θ_i automatically fixes a larger set of size $|G|$ of permutations in $\mathscr{F}$. Note that we will always assume that by default $\gamma_1 := \mathrm{id}$ in our enumeration of G.

An encoding in our current setting consists of the following parts: Variables $X^{\theta_\ell} = (x^{\theta_\ell}_{i,r})_{(i,r)\in[n]\times[n]}$, for $\ell = 1, \dots, d/|G|$, with permutation constraints precisely as in (4)–(7), cardinality constraints

$$\sum_{\ell=1}^{d/|G|} \sum_{m=1}^{|G|} \left[\bigwedge_{h=2}^{j} x^{\theta_\ell}_{\gamma_m(s_1),\gamma_m(s_h)} \right] \le d/j \quad \text{for } j \in [4\!:\!k],\ s \in \mathsf{SOP}(n,j); \tag{14}$$

$$\sum_{\ell=1}^{d/|G|} \sum_{m=1}^{|G|} \left[\bigwedge_{h=2}^{3} x^{\theta_\ell}_{\gamma_m(\sigma(h-1)),\gamma_m(\sigma(h))} \right] \le d/3! \quad \text{for } \sigma \in S_{n,3}; \tag{15}$$

and lexicographical symmetry breaking as in (11),

$$z_{\mathrm{cat}}[\theta_{\ell+1}]_{[1..L]} \preceq z_{\mathrm{cat}}[\theta_\ell]_{[1..L]} \quad \text{for } \ell \in [d/|G| - 1]\,. \tag{16}$$

Fixing the trivial subgroup solely consisting of the identity permutation $\{\gamma_1\} = \{\mathrm{id}\}$ of S_n, we recover precisely the original, non-heuristic model (4)–(9), (11). Thus, we derived a generalization parameterized by the group G.

Apart from left-cosets, one can just as well look at *right-cosets* based on the same motivation. This was also analyzed in [29] and the underlying assumption here translates to

$$\mathscr{F} = (\pi_1, \dots, \pi_d) = (\gamma_m \circ \theta_\ell)_{(\ell,m)\in[d/|G|]\times[|G|]}. \tag{17}$$

In the framework of SAT, this needs a more elaborate modeling approach: Although we notice that $\gamma_m \circ \theta_\ell = (\theta_\ell^{-1} \circ \gamma_m^{-1})^{-1}$, leading to the idea of using θ_ℓ^{-1}

directly as a new decision variable, there is then no direct way to recover via an encoding in CNF the incidence matrix of the inverse of a permutation (when the latter is itself specified by its incidence matrix). In fact we now face the difficulty of applying a fixed permutation given as parameter (via G) to a symbolically represented permutation (specified by decision variables) and afterwards accessing its incidence matrix. This considerably differs from the left-coset approach where the need is to access the incidence matrix of a symbolic permutation in entries pre-permuted by fixed permutations passed as parameters.

We circumvent this issue by encoding the offsets θ_m as permutation matrices $T^{\theta_\ell} = (t^{\theta_\ell}_{ij})^n_{i,j=1} \in \{0,1\}^{n\times n}$ whose bi-stochasticity we enforce via

$$\bigvee_{j=1}^{n} t^{\theta_\ell}_{ij} \quad \text{for } \ell \in [d/q]\,,\ i \in [n]\,; \tag{18}$$

$$\neg t^{\theta_\ell}_{i,j} \vee \neg t^{\theta_\ell}_{i,r} \quad \text{for } \ell \in [d/q]\,,\ i \in [n]\,,\ \{j,r\} \in \binom{[n]}{2}; \tag{19}$$

$$\bigvee_{i=1}^{n} t^{\theta_\ell}_{ij} \quad \text{for } \ell \in [d/q]\,,\ j \in [n]\,. \tag{20}$$

Cardinality-2 subsets of $[n]$ are denoted as $\binom{[n]}{2}$. The first two clause families together enforce exactly one 1-entry per row, hence exactly n such entries in total; demanding in addition at least one 1-entry per column then already forces T^{θ_ℓ} to be a permutation matrix keeping the encoding more compact. The subsequent constraint (21) encodes the incidence matrix of $\gamma_m \circ \theta_\ell$ for any fixed γ_m and a symbolic θ_ℓ inside the fresh variables $x^{\gamma_m\circ\theta_\ell}_{i,r}$, $(i,r) \in [n]^2$. The formulation depends on the permutation matrix T^{θ_ℓ} of θ_ℓ:

$$x^{\gamma_m\circ\theta_\ell}_{i,r} \leftrightarrow \left((t^{\theta_\ell}_{\gamma_m(r),j})^n_{j=1} \preceq (t^{\theta_\ell}_{\gamma_m(i),j})^n_{j=1}\right) \quad \text{for } \ell \in [d/q]\,,\ m \in [q]\,,\ (i,r) \in [n]^2\,. \tag{21}$$

Regarding (21), note that firstly, row i is a lexicographical successor of row r of the permutation matrix of an arbitrary permutation π iff $\pi(i) < \pi(r)$; secondly, the permutation matrix of $\gamma_m \circ \theta_\ell$ can be recovered from T^{θ_ℓ} by applying the permutation γ_m to its rows.

The main task to be addressed here is accessing the truth value of $x^{\gamma_m\circ\theta_\ell}_{i,r}$, storing as a Boolean whether the i-th and the r-th row are in ascending lexicographic ordering. A case distinction for the value of $x^{\gamma_m\circ\theta_\ell}_{i,r}$ shows that imposing the subsequent lexicographical ordering on the tuples that suitably link $x^{\gamma_m\circ\theta_\ell}$ to T^{θ_ℓ}-entries precisely ensures such a behavior for $x^{\gamma_m\circ\theta_\ell}_{i,r}$:

$$\text{Relations (23)–(24) are satisfied for } \ell \in [d/q]\,,\ m \in [q]\,,\ (i,r) \in [n]^2\,; \tag{22}$$

$$(x^{\gamma_m\circ\theta_\ell}_{i,r},\ t^{\theta_\ell}_{\gamma_m(r),1},\ \ldots\ ,\ t^{\theta_\ell}_{\gamma_m(r),n}) \preceq (1,\ t^{\theta_\ell}_{\gamma_m(i),1},\ \ldots\ ,\ t^{\theta_\ell}_{\gamma_m(i),n}), \tag{23}$$

$$(0, \neg t^{\theta_\ell}_{\gamma_m(r),1},\ \ldots\ , \neg t^{\theta_\ell}_{\gamma_m(r),n}) \preceq (x^{\gamma_m\circ\theta_\ell}_{i,r}, \neg t^{\theta_\ell}_{\gamma_m(i),1},\ \ldots\ , \neg t^{\theta_\ell}_{\gamma_m(i),n}). \tag{24}$$

The implementation of $\preceq$ in SAT is deferred to Sect. 4. What remains are the analogous cardinality constraints, now stated as

$$\sum_{\ell=1}^{d/|G|} \sum_{m=1}^{|G|} \left[\bigwedge_{h=2}^{j} x_{s_1,s_h}^{\gamma_m \circ \theta_\ell} \right] \leq d/j \quad \text{for } j \in [4\!:\!k],\ s \in \mathsf{SOP}(n,j); \tag{25}$$

$$\sum_{\ell=1}^{d/|G|} \sum_{m=1}^{|G|} \left[\bigwedge_{h=2}^{3} x_{\sigma(h-1),\sigma(h)}^{\gamma_m \circ \theta_\ell} \right] \leq d/3! \quad \text{for } \sigma \in S_{n,3}, \tag{26}$$

without loss of generality, we require

$$z_{\text{cat}}[X^{\gamma_1 \circ \theta_{\ell+1}}]_{[1:L]} \preceq z_{\text{cat}}[X^{\gamma_1 \circ \theta_\ell}]_{[1:L]} \quad \text{for } \ell \in [d/q - 1]\,. \tag{27}$$

The question is now, which subgroups are promising choices, and how many of them should be tested. Potentially, for each q dividing $n!$ and d, there can exist a subgroup of S_n of order q that is fruitful for the coset approach. In [29] it is shown for their problem setting that for the left-coset approach, a single representative per *conjugacy class* $\text{Cl}(G) := \{(\psi G)\psi^{-1} : \psi \in S_n\}$ is sufficient to be considered, as it reflects the behavior of all groups serving as representatives of this class. For the right-coset approach in [29], all subgroups of a given order have to be examined, where, however, without loss of generality one can assume the first offset θ_1 to be the identity permutation. We incorporate these insights from [29] into our experimental setup too, for reducing the number of inspected groups, respectively tightening the search space.

4 Computational experiments

We use `Julia` in version 1.11.1 for generating strict total order constraints. The cardinality constraints are obtained by calling the library `PySAT` in version 0.1.7.-dev15 [16]. The latter not only provides an interface to different SAT solvers, but also contains conversion routines to bring the encountered cardinality constraints (14),(15),(25),(26) into pure CNF: Seven encodings are available and we pick the one recommended by the default strategy of `PySAT`. For accessing the set of all (conjugacy classes of) subgroups of a given symmetric group S_n, we fall back on the `Julia`-package `Oscar` in version 1.2.0-dev [8], having an interface to `GAP` [11]—the access time is negligible and therefore not reported. In the literature, formulations for enforcing that for two binary vectors $a = (a_1, \ldots, a_r)$ and $b = (b_1, \ldots, b_r)$ the lexicographic ordering $a \preceq b$ applies (needed for the symmetry breaking in Sect. 2), have been proposed. However, as these are unsupported by `PySAT`, we opted for a custom implementation of the "AND Encoding using Common Subexpression Elimination" from [10], which depends on fresh variables x_i, $i = 1, \ldots, r-1$, and less than $6(r-1)$ clauses.

We choose for our experimental evaluation the SAT solver `Glucose` in version 4.2.1 which is frequently used for combinatorial problems. The time limit of the solver is set to 3600 seconds per instance if not stated otherwise. Selected

Table 1. Runtimes in seconds for both approaches where $k = 3$. Both approaches were run without postulated decomposition into cosets. A slightly faster CPU (Intel® Xeon® E5-2680 with 2.70GHz) is used by the authors of [29]. The symbol "/" indicates an inconclusive experiment (with unreported time measurement in [29]).

(d, n)	Solver run on (7)–(10),(12)	Algorithm 1 of [29]	Representative exists?
$(12, 6)$	0.00591 s	3.0 s	yes
$(12, 7)$	0.01146 s	2400.0 s	yes
$(12, 8)$	2.03096 s	/	no

non-existence results are independently verified by running the SAT solver in proof-logging modality such that a proof in the DRAT (Deletion Resolution Asymmetric Tautology [33]) file format is generated which is subsequently certified by the software `drat-trim` [33]. The experiments have been run on a cluster with an Intel® Xeon® E5-2640 v4 CPU with 2.40GHz and 160GB RAM on a single thread.

To first motivate competitiveness of our approach via a SAT encoding, we carried out the initial experiment (see Table 1) for $k = 3$ which is the only parameter allowing a comparison with existing literature results—this, given the particular isomorphy mentioned in Sect. 1.2. It can be noted that our proposed encoding can be solved in a fraction of time: The solving time on the smallest instance is smaller by three orders of magnitude, the medium instance by five orders (more precisely, the latter, by a factor of 2.09×10^5). Most remarkably, the largest instance $(12, 8)$ which requires to return non-existence as answer can be solved by the SAT solver in just two seconds while the computation was inconclusive apparently even after hours (although the authors do not explicitly report their maximum runtime; presumably, it would require days or weeks of computation time given the fact that it is a non-existence result). These observations demonstrate the SAT solver's excellent comparative performance, apparently forming a preferable method for such a task. We next transfer to this SAT setting the heuristic decomposition strategy into cosets of S_n.

For the subsequent, we decided not to examine the case $k = 3$ whose properties can be deduced from the considerations in the works [12,29] for $n \leq 8$. The search for optimality when $n = 9$—the smallest open case unanswerable by those works—was inconclusive in our setting despite a week of computation time. Therefore, we begin a systematic series of experiments starting with $k = 4$ relying on almost a month of cumulative computation time: Table 2 presents computational results as follows.

Table 2: Results for $k = 4$; 3600s time limit given for each run of the SAT solver. Asterisks followed by numbers indicate the count of inconclusive experiments, i.e., the solver hit the time limit, for the right-coset approach.

		sg		sg-feas				sg-infeas			
		#exst		#		avg_t[s]		#		avg_t[s]	
(d, n)	$\lvert G\rvert$	L	R	L	R	L	R	L	R	L	R
$(12, 4)$	12	1	1	1	1	$2.1 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$	0	0	/	/
	6	1	4	0	0	/	/	1	4	$3.7 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$
	4	3	7	2	2	$2.2 \cdot 10^{-4}$	$2.9 \cdot 10^{-3}$	1	5	$1.2 \cdot 10^{-3}$	$3.6 \cdot 10^{-3}$
	3	1	4	1	4	$3.3 \cdot 10^{-4}$	$1.3 \cdot 10^{-3}$	0	0	/	/
	2	2	9	1	3	$5.7 \cdot 10^{-4}$	$1.2 \cdot 10^{-3}$	1	6	$1.1 \cdot 10^{-3}$	$6.6 \cdot 10^{-3}$
	1	1	/	1	/	$1.2 \cdot 10^{-3}$	/	0	/	/	/
$(12, 5)$	12	2	15	0	0	/	/	2	15	$8.9 \cdot 10^{-4}$	$1.8 \cdot 10^{-5}$
	6	3	30	1	0	$2.1 \cdot 10^{-3}$	/	2	30	$2.8 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$
	4	3	35	0	0	/	/	3	35	$1.0 \cdot 10^{-2}$	$4.8 \cdot 10^{-3}$
	3	1	10	1	0	$2.7 \cdot 10^{-3}$	/	0	10	/	$2.8 \cdot 10^{0}$
	2	2	25	1	2	$2.8 \cdot 10^{-2}$	$6.1 \cdot 10^{-3}$	1	23	$1.5 \cdot 10^{-3}$	$2.1 \cdot 10^{-1}$
	1	1	/	1	/	$4.1 \cdot 10^{-3}$	/	0	/	/	/
$(24, 6)$	24	6	90	2	8	$2.9 \cdot 10^{-3}$	$2.6 \cdot 10^{-3}$	4	82	$1.2 \cdot 10^{-2}$	$1.5 \cdot 10^{-5}$
*2	12	4	150	2	4	$4.0 \cdot 10^{-3}$	$1.2 \cdot 10^{-2}$	2	144	$3.2 \cdot 10^{-3}$	$2.4 \cdot 10^{1}$
	8	7	255	2	32	$1.4 \cdot 10^{-2}$	$6.7 \cdot 10^{-2}$	5	223	$8.9 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$
*27	6	6	280	3	30	$2.6 \cdot 10^{-2}$	$2.5 \cdot 10^{0}$	3	223	$6.0 \cdot 10^{-2}$	$8.5 \cdot 10^{1}$
*38	4	7	255	5	72	$3.6 \cdot 10^{-2}$	$6.8 \cdot 10^{0}$	2	145	$2.0 \cdot 10^{1}$	$2.5 \cdot 10^{2}$
*15	3	2	40	1	14	$9.1 \cdot 10^{-2}$	$4.6 \cdot 10^{2}$	1	11	$6.2 \cdot 10^{-1}$	$4.4 \cdot 10^{2}$
*38	2	3	75	2	35	$9.8 \cdot 10^{-1}$	$9.0 \cdot 10^{2}$	1	2	$3.3 \cdot 10^{0}$	$2.0 \cdot 10^{3}$
	1	1	/	1	/	$8.5 \cdot 10^{-1}$	/	0	/	/	/
$(24, 7)$	24	14	1435	0	0	/	/	14	1435	$1.9 \cdot 10^{-2}$	$1.5 \cdot 10^{-5}$
	12	13	1715	2	0	$2.6 \cdot 10^{-2}$	/	11	1715	$1.7 \cdot 10^{-1}$	$1.2 \cdot 10^{0}$
*3	8	7	1575	0	0	/	/	7	1572	$1.4 \cdot 10^{-1}$	$3.8 \cdot 10^{-1}$
*30	6	8	1645	3	0	$5.4 \cdot 10^{-2}$	/	5	1615	$8.6 \cdot 10^{-2}$	$2.8 \cdot 10^{1}$
*532	4	7	1295	3	3	$2.7 \cdot 10^{-1}$	$4.0 \cdot 10^{1}$	4	760	$2.4 \cdot 10^{1}$	$4.1 \cdot 10^{2}$
*151	3	2	175	1	0	$6.1 \cdot 10^{0}$	/	1	24	$2.0 \cdot 10^{0}$	$3.5 \cdot 10^{2}$
*230	2	3	231	2	1	$1.4 \cdot 10^{1}$	$4.1 \cdot 10^{2}$	1	0	$5.9 \cdot 10^{1}$	/
	1	1	/	1	/	$3.5 \cdot 10^{1}$	/	0	/	/	/

While the first two columns indicate the choice of parameters d, n, and $|G|$, the remainder contains results for the left- (shaded background) and the right-coset approach, respectively. Column `sg` provides information about the number of *conjugacy classes of subgroups of S_n of order $|G|$* (`#exst L`). It also shows how many *subgroups of S_n of order $|G|$* exist (`#exst R`). The experiment has been conducted for every existing subgroup, i.e., the number of considered subgroups always coincides with `#exst`. The count of subgroups for which the left- respectively right-coset leads to a feasible solution is displayed in `sg-feas #L/R`. This information is accompanied by the average time (in seconds), which the solver needed to find such a feasible solution (`avg_t[s] L/R`). The same structure applies for the group of columns `sg-infeas`, which reports these counts and time measurements for subgroups resulting in infeasibility. The right-coset

approach is never run for $|G| = 1$, symbolized by the entry "/", as this represents the non-heuristic approach where we only employ the left-coset approach.

From the experiments in Table 2 we can derive the following observation.

Observation 1 *For $n \in \{4, 5\}$, respectively $n \in \{6, 7\}$, there are optimal 4-restricted minwise independent families of 12 members, respectively 24 members, which are all coincident with a union of non-trivial cosets of S_n. For $n = 4$, respectively $n = 6$, a family being a coset of cardinality 12, respectively 24, can be found.*

Computer-aided proof. We have $\operatorname{lcm}([4]) = 12$. Therefore the representatives found in Table 2 are minimally-sized for $n \in \{4, 5\}$. In 0.04 seconds the solver showed non-existence of a minwise independent family with 12 members for $n = 6$; an accompanying solver run producing a proof-file of size 334 KB (in binary encoded DRAT format) in 0.21 seconds is certified in 0.13 seconds by `drat-trim`. The next smallest cardinality which is a multiple of 12, namely $d = 24$, is therefore reported. For $d = 24$ and $n \in \{6, 7\}$ such representatives, again being decomposable into cosets, indeed exist; see Table 2. □

It is noteworthy that, consistently across Table 2, for every subgroup order considered, infeasibility on the left-coset side implies infeasibility on the right-coset side. In contrast, for several subgroup orders, for which the left-coset approach was fruitful, the right-coset approach was not successful. Moreover, the more complex SAT encoding for the right-coset approach increases the runtime by one to two orders of magnitude in comparison to the left-coset approach for feasible but also infeasible instances. Due to these two observations we conclude that in particular for higher values of d and n one should rather focus on the left-coset approach: If one faces a fixed and limited margin of computation time, this allows examining more instances (which also seem the more promising ones).

Observation 2 *There is no 4-restricted minwise independent family $\mathscr{F} \subseteq S_8$ consisting of 24 members.*

Computer-aided proof. The associated SAT instance turned out to be unsatisfiable after 64.7 hours of computation time; since unsatisfiability was already suspected, full lexicographical accuracy $L := n(n-1)/2 - 1$ was chosen here. Additionally, we ran the solver in proof-logging modality, which after 184.5 hours returned a 28.3 GB long proof (in binary encoded DRAT format). It took `drat-trim` 59.3 hours to validate the latter. □

Proposition 1. *k-restricted minwise independent families $\mathscr{F} \subseteq S_n$ with $d = |\mathscr{F}|$, which coincide with a union of left-cosets of S_n, exist for constellations*

$$(d, n, k; |G|) \in \{(60, 6, 6; 6), (60, 6, 5; 6), (60, 5, 5; 60), (48, 8, 4; 24)\}.$$

Proof. Certificates for this assertion can be found in the supplementary material of the aforementioned electronic appendix, where some `Julia` routines are also deposited.

We also remark that for $k \in \{5, 6\}$ we have $\operatorname{lcm}([5]) = \operatorname{lcm}([6]) = 60$ and thus the respective instances in Proposition 1 are even optimal. We highlight suboptimality for $(d, n, k) = (48, 8, 4)$, as we can construct an instance with constellation $(d, n, k) = (36, 8, 4)$ by applying Theorem 1 to the 3-restricted minwise independent family deducible from $\mathscr{F} \subseteq S_8$ of 18 members in [29, Proposition 4.6]—due to Observation 2 the end product is even optimal. However, although we can obtain this insight, even with runtimes up to 48 hours, the solver was not able to find any respective family counting 36 members via decomposition into left-cosets. Finally, we point out the particular situation occurring for $(d, k) = (12, 4)$, see Table 2, where for $n = 4$ no decomposition into left-cosets of cardinality 6 is possible, while this is the case for $n = 5$, which intuitively is a stronger constraint setting. This, apparently, is explainable by the structurally different order-6 subgroups present in S_5 compared with those in S_4. At the same time it is notable that for $|G| = 12$ the left-coset approach worked for $n = 4$ but not for $n = 5$.

5 Conclusion

We have developed SAT encodings that are well suited for examining the existence and structural decomposability of (near-)optimal k-restricted minwise independent families $\mathscr{F} \subseteq S_n$ for computationally tractable values of n and k. In addition to discovering several previously unknown minwise independent families, our results reveal the following key insight: All computed (near-)optimal representatives of these families indeed carry the aforementioned group-theoretic structure. A further benefit of the decomposition is that it dramatically reduces running times, making the search for feasible representatives possible where a purely non-heuristic approach would be prohibitively expensive. Collecting further empirical evidence, or even obtaining a generally valid result, affirming the decomposability of optima into unions of non-trivial cosets seems an interesting challenge.

The following aspects offer interesting directions for further work. In Observations 1 and 2, the partial replacement of cardinality constraints with parity constraints (using a solver featuring exclusive-or) or other weakened forms might offer a useful strategy for finding shorter computational non-existence proofs. They might be convertible into considerably shorter Lean proofs [27]. Further experimentation with the cardinality constraints, e.g., by using solvers with built-in cardinality propagators or by employing integer linear programming, seems a worthwhile alternative to study.

Disclosure of Interests. The authors have no competing interests.

References

1. Banbara, M., Tamura, N., Inoue, K.: Generating event-sequence test cases by answer set programming with the incidence matrix. In: Technical Communications of the 28th International Conference on Logic Programming. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2012). https://doi.org/10.4230/lipics.iclp.2012.86

2. Bargachev, V.: An improved lower bound on the size of k-rankwise independent families of permutations. Tech. rep., St. Petersburg Department of Steklov Institute of Mathematics (2004). https://www.pdmi.ras.ru/preprint/2004/04-13.html (last accessed 2026-05-30)
3. Bargachev, V.: On some properties of min-wise independent families and groups of permutations. Journal of Mathematical Sciences **134**(5), 2340–2345 (2006). https://doi.org/10.1007/s10958-006-0110-1
4. Broder, A.Z.: On the resemblance and containment of documents. In: Proceedings of the Compression and Complexity of SEQUENCES. pp. 21–29. IEEE (1997)
5. Broder, A.Z., Charikar, M., Frieze, A.M., Mitzenmacher, M.: Min-wise independent permutations. Journal of Computer and System Sciences **60**(3), 630–659 (2000). https://doi.org/10.1145/276698.276781
6. Cameron, P.J., Spiga, P.: Min-wise independent families with respect to any linear order. Communications in Algebra **35**(10), 3026–3033 (2007). https://doi.org/10.1080/00927870701404812
7. Chee, Y.M., Colbourn, C.J., Horsley, D., Zhou, J.: Sequence covering arrays. SIAM Journal on Discrete Mathematics **27**(4), 1844–1861 (2013). https://doi.org/10.1137/120894099
8. Decker, W., Eder, C., Fieker, C., Horn, M., Joswig, M. (eds.): The Computer Algebra System OSCAR: Algorithms and Examples, Algorithms and Computation in Mathematics, vol. 32. Springer, 1 edn. (2024). https://doi.org/10.1007/978-3-031-62127-7
9. Désarménien, J.: Une autre interprétation du nombre de dérangements. Séminaire Lotharingien de Combinatoire **8**, 1–6 (1984), https://www.emis.de/journals/SLC/opapers/s08desar.html
10. Elgabou, H.: Encoding The Lexicographic Ordering Constraint in Satisfiability Modulo Theories. Ph.D. thesis, University of York (2015), https://etheses.whiterose.ac.uk/10387/
11. The GAP Group: GAP—Groups, Algorithms, and Programming, Version 4.13.1 (2024), https://www.gap-system.org
12. Gentle, A.R.: A polynomial construction of perfect sequence covering arrays. Algebraic Combinatorics **6**(5), 1383–1394 (2023). https://doi.org/10.5802/alco.308
13. Gentle, A.R., Wanless, I.M.: On perfect sequence covering arrays. Annals of Combinatorics **27**(3), 539–564 (2023). https://doi.org/10.1007/s00026-022-00610-6
14. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: Atkinson, M.P., Orlowska, M.E., Valduriez, P., Zdonik, S.B., Brodie, M.L. (eds.) Proceedings of 25th International Conference on Very Large Data Bases. pp. 518–529. Morgan Kaufmann (1999), http://www.vldb.org/conf/1999/P49.pdf
15. Harvey, N., Sahami, A.: Explicit and near-optimal construction of t-rankwise independent permutations. In: Kumar, A., Ron-Zewi, N. (eds.) Approximation, Randomization, and Combinatorial Optimization. LIPIcs, vol. 317, pp. 67:1–67:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2024). https://doi.org/10.4230/lipics.approx/random.2024.67
16. Ignatiev, A., Morgado, A., Marques-Silva, J.: PySAT: A Python toolkit for prototyping with SAT oracles. In: International Conference on Theory and Applications of Satisfiability Testing. pp. 428–437. Springer (2018). https://doi.org/10.1007/978-3-319-94144-8_26
17. Itoh, T., Takei, Y., Tarui, J.: On permutations with limited independence. In: Proceedings of the eleventh annual ACM-SIAM Symposium on Discrete Algorithms. pp. 137–146 (2000). https://doi.org/10.5555/338219.338245

18. Itoh, T., Takei, Y., Tarui, J.: On the sample size of k-restricted min-wise independent permutations and other k-wise distributions. In: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing. pp. 710–719 (2003)
19. Iurlano, E.: Growth of the perfect sequence covering array number. Designs, Codes and Cryptography **91**(4), 1487–1494 (2023). https://doi.org/10.1007/s10623-022-01168-3
20. Kuperberg, G., Lovett, S., Peled, R.: Probabilistic existence of regular combinatorial structures. Geometric and Functional Analysis **27**(4), 919–972 (2017). https://doi.org/10.1007/s00039-017-0416-9
21. Levenshtein, V.I.: On perfect codes in deletion and insertion metric. Discrete Mathematics and Applications **2**(3), 241–258 (1992). https://doi.org/10.1515/dma.1992.2.3.241
22. Li, P., König, A.C.: Theory and applications of b-bit minwise hashing. Communications of the ACM **54**(8), 101–109 (2011). https://doi.org/10.1145/1978542.1978566
23. Li, P., Owen, A., Zhang, C.H.: One permutation hashing. Advances in Neural Information Processing Systems **25** (2012)
24. Mathon, R., van Trung, T.: Directed t-packings and directed t-Steiner systems. Designs, Codes and Cryptography **18**(1), 187–198 (1999). https://doi.org/10.1023/a:1008353723204
25. Matoušek, J., Stojaković, M.: On restricted min-wise independence of permutations. Random Structures & Algorithms **23**(4), 397–408 (2003)
26. Matsumoto, M., Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation **8**(1), 3–30 (1998). https://doi.org/10.1145/272991.272995
27. de Moura, L., Ullrich, S.: The Lean 4 theorem prover and programming language. In: Platzer, A., Sutcliffe, G. (eds.) Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings. pp. 625–635. LNCS, Springer (2021). https://doi.org/10.1007/978-3-030-79876-5_37
28. Mulmuley, K.: Computational geometry: An introduction through randomized algorithms. Prentice-Hall, Englewood Cliffs, N.J. (1994)
29. Na, J., Jedwab, J., Li, S.: A group-based structure for perfect sequence covering arrays. Designs, Codes and Cryptography **91**(3), 951–970 (2023). https://doi.org/10.1007/s10623-022-01132-1
30. OEIS Foundation Inc.: The On-Line Encyclopedia of Integer Sequences (2024), published electronically at https://oeis.org
31. Peczarski, M.: New results in minimum-comparison sorting. Algorithmica **40**, 133–145 (2004). https://doi.org/10.1007/s00453-004-1100-7
32. Tarui, J., Itoh, T., Takei, Y.: A nearly linear size 4-min-wise independent permutation family by finite geometries. In: Arora, S., Jansen, K., Rolim, J.D.P., Sahai, A. (eds.) Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques. LNCS, vol. 2764, pp. 396–408. Springer, Berlin, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45198-3_33
33. Wetzler, N., Heule, M., Hunt Jr., W.A.: DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In: Theory and Applications of Satisfiability Testing—17th International Conference. LNCS, vol. 8561, pp. 422–429. Springer (2014). https://doi.org/10.1007/978-3-319-09284-3_31
34. Yuster, R.: Perfect sequence covering arrays. Designs, Codes and Cryptography **88**(3), 585–593 (2020). https://doi.org/10.1007/s10623-019-00698-7

35. Zamora, J., Mendoza, M., Allende, H.: Hashing-based clustering in high dimensional data. Expert Systems with Applications **62**, 202–211 (2016). https://doi.org/10.1016/j.eswa.2016.06.008
36. Zhang, J., Gray, D., Wang, H., Zhang, X.D.: On the combinatorics of derangements and related permutations. Applied Mathematics and Computation **431**, 127341 (2022). https://doi.org/10.1016/j.amc.2022.127341

A Electronic Appendix

This document contains supplementary material for the following paper:

> Iurlano, E., Raidl, G. R.: SAT-Based Search for Minwise Independent Families. (2026). To appear, PPSN2026

Some citations and references to results or formulas retain the numbering from the corresponding paper.

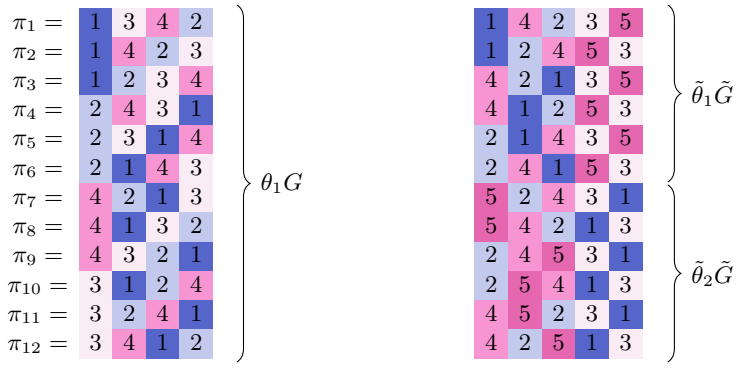

Fig. A1. Subgroups G (respectively $\tilde{G}$) of order 12 (respectively 6) allow generating 12 permutations carrying 4-restricted minwise independence via a search for just one (respectively two) offset permutations. Only one twelfth (respectively one sixth) of the decision variables is required.

Denote by $\mathbb{B}^{m\times n}_{\text{asc}}$ (respectively $\mathbb{B}^{m\times n}_{\text{desc}}$) all binary $m \times n$ matrices in which the i-th row is a lexicographic predecessor (respectively successor) of the $(i+1)$-th one, for $i \in [m-1]$.

Lemma A1. *For an arbitrary Boolean matrix $A = (a_{ij})_{(i,j)\in[2]\times[n]} \in \{0,1\}^{2\times n}$, consider the following system of membership-constraints involving the Boolean $m_{\text{asc}} \in \{0,1\}$:*

$$\begin{pmatrix} m_{\text{asc}} & a_{11}\ a_{12}\ \dots\ a_{1n} \\ 1 & a_{21}\ a_{22}\ \dots\ a_{2n} \end{pmatrix} \in \mathbb{B}^{2\times(1+n)}_{\text{asc}}, \tag{A1}$$

$$\begin{pmatrix} 0 & \neg a_{11}\ \neg a_{12}\ \dots\ \neg a_{1n} \\ m_{\text{asc}} & \neg a_{21}\ \neg a_{22}\ \dots\ \neg a_{2n} \end{pmatrix} \in \mathbb{B}^{2\times(1+n)}_{\text{asc}}. \tag{A2}$$

(i) We have $A \in \mathbb{B}^{2\times n}_{\text{asc}}$ iff $m_{\text{asc}} = 1$ is a solution for (A1)–(A2).
(ii) We have $A \in \mathbb{B}^{2\times n}_{\text{desc}}$ iff $m_{\text{asc}} = 0$ is a solution for (A1)–(A2).

Proof. The leading columns of (A1) and (A2) are $\binom{m_{\text{asc}}}{1}$ and $\binom{0}{m_{\text{asc}}}$, respectively. For each value of $m_{\text{asc}} \in \{0,1\}$, exactly one of these columns equals $\binom{0}{1}$

and so decides the comparison at the leftmost bit (the corresponding condition thus holds unconditionally), while the other equals $\binom{m_{\text{asc}}}{m_{\text{asc}}}$ and ties, reducing the condition to a lexicographic comparison of the remaining n bits.

At $m_{\text{asc}} = 1$ the active reduction is $a_{1\cdot} \preceq a_{2\cdot}$, i.e., $A \in \mathbb{B}^{2\times n}_{\text{asc}}$. At $m_{\text{asc}} = 0$ the active reduction is $\neg a_{1\cdot} \preceq \neg a_{2\cdot}$, equivalently $A \in \mathbb{B}^{2\times n}_{\text{desc}}$. Each reduction is an iff, yielding (i) and (ii). □

Lemma A2. *A binary $n \times n$ matrix is bi-stochastic iff each row has* precisely one 1*-entry and each column has* at most one 1*-entry.*

Proof. Apply the pigeonhole principle to the columns. □

Lemma A3. *Let $\succeq_*$ be any total order on S_n with* $\mathrm{id} \succeq_* \pi$*, for all $\pi \in S_n$.*

(i) Let $\tau \in S_n$ be any permutation. A family $\mathscr{F} = (\pi_1, \ldots, \pi_d)$ is k-restricted minwise independent iff $\widetilde{\mathscr{F}} = (\pi_1 \circ \tau, \ldots, \pi_d \circ \tau)$ has this property.

(ii) There exists a k-restricted minwise independent family $\mathscr{F} = (\pi_1, \ldots, \pi_d)$ iff there exists a k-restricted minwise independent family $\widetilde{\mathscr{F}} = (\widetilde{\pi_1}, \ldots, \widetilde{\pi_d})$ that fulfills $\widetilde{\pi_1} = \mathrm{id}$ and $\widetilde{\pi_i} \succeq_* \widetilde{\pi_{i+1}}$, for $i = 1, \ldots, d-1$.

Proof. Let us first address (i). For an arbitrary $X \subseteq [n]$ and $x^* \in X$, after denoting $y^* := \tau(x^*)$ and $Y := \{\tau(x) : x \in X\}$, we obtain k-restricted minwise independence of $\widetilde{\mathscr{F}}$ from the fact that

$$|\{i \in [d] : \pi_i \circ \tau(x^*) = \min_{x\in X} \pi_i \circ \tau(x)\}| = |\{i \in [d] : \pi_i(y^*) = \min_{y\in Y} \pi_i(y)\}| = \frac{1}{|Y|} = \frac{1}{|X|},$$

due to k-restricted minwise independence of $\mathscr{F}$.

Concerning (ii), setting $\tau := \pi_1^{-1}$, the property holds according to (i) for the family $\widetilde{\mathscr{F}} := (\pi_1 \circ \pi_1^{-1}, \ldots, \pi_d \circ \pi_1^{-1})$ when its members indexed $2, \ldots, d$ are rearranged into a chain with respect to $\succeq_*$. □

The lexicographic "AND" encoding with common subexpression elimination

For two equal-length bit-vectors $a = (a_1, \ldots, a_r)$ and $b = (b_1, \ldots, b_r)$ (interpreted as binary numbers with decreasing significance), we encode the lexicographic comparison $a \preceq b$ following the "AND Encoding using Common Subexpression Elimination" of [10], since the dedicated ordering constraints are unsupported by `PySAT`. The encoding introduces fresh auxiliary variables x_i, $i = 1, \ldots, r-1$, where x_i is constrained to indicate that the two prefixes of length i coincide.

These variables are then suitably linked in [10]; we give a translation to a ready list of clauses:

$$\begin{aligned}
\neg x_1 \vee b_1 \vee \neg a_1,\ \neg x_1 \vee a_1 \vee \neg b_1,\ x_1 \vee \neg a_1 \vee \neg b_1,\ x_1 \vee a_1 \vee b_1,\\
x_i \vee \neg x_{i+1}, \quad i = 1, \ldots, r-2,\\
\neg x_{i+1} \vee b_{i+1} \vee \neg a_{i+1}, \quad i = 1, \ldots, r-2,\\
\neg x_{i+1} \vee a_{i+1} \vee \neg b_{i+1}, \quad i = 1, \ldots, r-2,\\
x_{i+1} \vee \neg b_{i+1} \vee \neg a_{i+1} \vee \neg x_i, \quad i = 1, \ldots, r-2,\\
x_{i+1} \vee b_{i+1} \vee a_{i+1} \vee \neg x_i, \quad i = 1, \ldots, r-2,\\
\neg x_i \vee b_{i+1} \vee \neg a_{i+1}, \quad i = 1, \ldots, r-1.
\end{aligned}$$

Lemma A4 (analogue of [29]). *Let $G, G' \leq S_n$ be conjugate subgroups, say $G' = \psi G \psi^{-1}$ for some $\psi \in S_n$, and let d be a multiple of $|G| = |G'|$. Then a k-restricted minwise independent family of size d that is a union of left-cosets of G exists if and only if such a family that is a union of left-cosets of G' exists. Consequently, for the left-coset approach it suffices to examine a single representative per conjugacy class $\mathrm{Cl}(G) = \{\psi G \psi^{-1} : \psi \in S_n\}$ of subgroups of S_n.*

Proof. As G and G' play symmetric roles (indeed $G = \psi^{-1} G' \psi$), it suffices to prove one implication. Assume there are offsets $\theta'_1, \ldots, \theta'_{d/|G|} \in S_n$ such that $\mathscr{F}' = (\theta'_\ell \circ \gamma')_{\ell \in [d/|G|],\, \gamma' \in G'}$ is k-restricted minwise independent. Every $\gamma' \in G'$ is of the form $\gamma' = \psi \circ \gamma \circ \psi^{-1}$ with $\gamma \in G$, and $\gamma \mapsto \psi \circ \gamma \circ \psi^{-1}$ is a bijection from G onto G'. Right-composing every member of $\mathscr{F}'$ with the fixed permutation ψ preserves k-restricted minwise independence by Lemma A3 (i) (with $\tau := \psi$); the resulting members read

$$(\theta'_\ell \circ \psi \circ \gamma \circ \psi^{-1}) \circ \psi = (\theta'_\ell \circ \psi) \circ \gamma, \qquad \ell \in [d/|G|]\,,\ \gamma \in G.$$

Hence, taking the offsets $\theta_\ell := \theta'_\ell \circ \psi$, the family $(\theta_\ell \circ \gamma)_{\ell \in [d/|G|],\, \gamma \in G}$ is a union of left-cosets of G, has the same size d, and is k-restricted minwise independent. □

Remark A5. A rough estimate of the *size of the encoding* in Sect. 2 can be obtained by the following observations: The d incidence matrices are represented by $dn(n-1)/2 = dO(n^2)$ decision variables. A number of $d\binom{n}{3}3! + d\sum_{4 \leq j \leq k} \binom{n}{j} j = dO(n^k)$ additional variables signals occurrences of a semiordered pattern in each of the d permutations (assuming k is fixed); these have to be linked to the summands in (8)–(9). The number of constraints is dominated by the count of cardinality constraints to be installed for all of the latter summands, i.e., by $O(n^k)$ arithmetic inequalities, which enforce that at most d/j variables out of a set of d variables shall attain the Boolean value 1; here a linear amount of auxiliary fresh variables to realize each counting constraint must typically be introduced; we refer to Sect. 4 for more specific information on how we realize such constraints. Given the latter magnitude of constraints, the count of $dO(n^3)$ transitivity-ensuring clauses, see (7), is negligible in comparison. A number of

$d-1$ lexicographical comparisons have to be modeled; as we see in Sect. 4 their realization asks to introduce a number of auxiliary variables as large as the length of the binary strings representing the upper diagonal matrix, i.e., at most $(d-1)O(n^2)$ where the extremal case refers to the use of accuracy $L = n(n-1)/2 - 1$.

Theorem A6 (Overview of available bounds). *Let $n \geq k \geq 3$, $\mathscr{F} \subseteq S_n$ be k-restricted minwise independent, and $\mathscr{G} \subseteq S_n$ be k-rankwise independent. Then, with the* subfactorial *defined as $z \mapsto !z := (z!) \cdot \sum_{j=0}^{z} (-1)^j/j!$, the following estimates apply:*

(i) $|\mathscr{F}| \geq \max\{n, \operatorname{lcm}([k])\}$*; see [2,17].*
(ii) $|\mathscr{F}| \leq n^{(1+(1/\ln n))k} \operatorname{lcm}([k-1])$*; see [17].*
(iii) $|\mathscr{G}| \geq \sum_{i=0}^{\lfloor k/2 \rfloor} !i \binom{n}{i} =: E(n,k)$ *for even k, otherwise, when k is odd, $|\mathscr{G}| \geq E(n,k) + !\lceil k/2 \rceil \binom{n-1}{\lfloor k/2 \rfloor}$; see [2].*
(iv) There is some constant $C > 0$ (independent of n and k), for which $|\mathscr{G}| \leq (Cn)^{Ck}$; see [20]. There is some constant $D > 0$ (independent of n and k), for which $|\mathscr{G}| \leq (Dn)^{35k}$; see [15].

Bijecting derangements onto non-waste permutations

The computational effort for generating non-existence proofs underlines the value of combinatorial lower bounds. The following explains in detail how to settle the challenge of Bargachev [2, p. 6] to construct a natural bijection between two classes of permutations that he exploited to obtain the tightest-known lower bound in [2, Theorem 2] on the size of k-rankwise independent families. The class of permutations having k *waste indices*, denoted as $\mathsf{Wst}(n,k) \subseteq S_n$ (later defined) is the object of study. Bargachev's approach [2] relies on a technical solution of a recurrence relation for proving the coincidence of the cardinality of $\mathsf{Wst}(n,k)$ with the count of k*-partial derangements* in S_n. The latter are defined as permutations with exactly $k \in [0:n]$ fixed points, whose class we henceforth denote as $\mathsf{Derang}(n,k) \subseteq S_n$. However, Bargachev emphasizes the lack of a known *natural* bijection $\Phi : \mathsf{Derang}(n,k) \to \mathsf{Wst}(n,k)$, even for $k = 0$. By a natural bijection a structure-preserving mapping between two sets clearly explaining why both sets have the same cardinality is meant; the correspondence shall arise directly from the inherent properties of the elements.

We fill this gap by more generally giving such a bijection for *arbitrary* k, with a remarkably succinct description.

Remark A7. The cardinality $|\mathsf{Derang}(n,k)|$ admits a folklore closed form involving selecting the k fixed positions in $\binom{n}{k}$ ways and deranging the remaining $n-k$ elements:

$$|\mathsf{Derang}(n,k)| = !(n-k)\binom{n}{k} = \frac{n!}{k!}\sum_{j=0}^{n-k}\frac{(-1)^j}{j!}.$$

The same subfactorial already occurs in Theorem A6 (iii).

Let $\pi \in S_n$. An index $j \in [n]$ is *waste* [2] if either the special case $j = n$ with $\pi(n) = \min\{\pi(\ell) : \ell = 1, \ldots, n\} = 1$ applies, or alternatively, if $j < n$ with

$$\min\{\pi(\ell) : \ell = 1, \ldots, j\} = \pi(j) \quad \text{and} \quad \pi(j) > \pi(j+1). \tag{A3}$$

Denote by $\mathsf{Wst}(n, k) \subseteq S_n$ the set of all permutations possessing precisely $k \in [0\!:\!n]$ waste indices; *non-waste* permutations refer to $\mathsf{Wst}(n, 0)$.

We now mimic and generalize the approach in [36, Proposition 2.1] relying on the cycle notation of permutations; see also [9].

Theorem A8. *For $n \in \mathbb{N}$ and $k \in [0\!:\!n]$ a natural bijection $\Phi : \mathsf{Derang}(n, k) \to \mathsf{Wst}(n, k)$ exists.*

Proof. Let us define the transformed $\Phi(\pi)$ of $\pi \in \mathsf{Derang}(n, k) \subseteq S_n$ as the result of the following procedure: Write down π in cycle notation where each cycle is formally represented by a tuple; as there is a certain degree of freedom which member of the cycle is encountered at the first entry of the tuple, let us require that the smallest member of the cycle populates the respective first entry. Afterwards, the cycle-representing tuples, as a whole, are rearranged from left to right such that their respective first entries ("leaders") decrease; see Example A9. The final output $\Phi(\pi)$ is then defined as the permutation which maps $j \in [n]$ to the j-th entry in the string corresponding to the ordered concatenation of these tuples.

Now observe well-definedness, as $\Phi(\pi)$ is an element of $\mathsf{Wst}(n, k)$: By the assumption $\pi \in \mathsf{Derang}(n, k)$, there are precisely k length-1 cycles in the cycle notation of π. In the concatenated object, each position of a length-1 cycle is a waste index: Note that the minimization in (A3) but also the second, decreasing, behavior in (A3) hold true due to the decreasing ordering of the cycles' leaders. On the other hand, in cycles of length at least two, by construction, cycle-entries appearing not in the first position cannot be minimizers in (A3); also the first entries of the cycles never meet the second condition of decrease in (A3). No additional waste-indices originate from cycles of length at least two and therefore $\Phi(\pi) \in \mathsf{Wst}(n, k)$.

For bijectivity of Φ we show that its inverse is given by $\Psi : \mathsf{Wst}(n, k) \to \mathsf{Derang}(n, k)$. Given $\tau \in \mathsf{Wst}(n, k)$, an entry τ_j (and, by extension, its position $j \in [n]$) is called a *left-to-right minimum* of τ—briefly, an LR-minimum—when $\tau_j < \tau_m$ for every $m < j$; equivalently, when j satisfies the minimization-condition in (A3). Let $j_1 < j_2 < \cdots < j_\ell$ enumerate the LR-minima of τ. Segment τ at these positions,

$$(\tau_{j_1}, \ldots, \tau_{j_2-1}), (\tau_{j_2}, \ldots, \tau_{j_3-1}), \ldots, (\tau_{j_\ell}, \ldots, \tau_n), \tag{A4}$$

and define $\Psi(\tau) \in S_n$ as the permutation whose cycle decomposition reads these blocks as cycles.

Because τ_{j_i} is an LR-minimum and no LR-minimum lies strictly between j_i and j_{i+1}, τ_{j_i} is strictly smaller than every other entry of its block; moreover, the LR-minima are themselves strictly decreasing, $\tau_{j_1} > \cdots > \tau_{j_\ell}$. These are

precisely the two normalizations under which Φ writes cycles (smallest member first; leaders in decreasing order), so applying Φ to $\Psi(\tau)$ recovers the original concatenation, i.e., $\Phi(\Psi(\tau)) = \tau$. A waste index $j < n$ of τ is itself an LR-minimum whose successor $\tau_{j+1} < \tau_j$ is again an LR-minimum, so the block at j is the singleton (τ_j); for $j = n$ the last block is $(\tau_n) = (1)$ by the special clause of the waste-index definition. Thus the k waste indices of τ produce exactly k length-1 cycles in $\Psi(\tau)$, giving $\Psi(\tau) \in \mathsf{Derang}(n, k)$.

Conversely, for any $\pi \in \mathsf{Derang}(n, k)$ the cycle starts of π are precisely the LR-minima of $\Phi(\pi)$—each cycle begins with its minimum, and the cycles preceding it have strictly larger leaders. Applying the segmentation procedure to $\Phi(\pi)$ therefore recovers the cycle decomposition of π, i.e., $\Psi(\Phi(\pi)) = \pi$. Hence Φ is a bijection. □

Example A9. Let $\rho = (1, 5, 3, 4, 6, 2, 8, 7, 9)$, and thus $\rho \in \mathsf{Derang}(9, 4)$. Then, its cycle notation, respectively $\Phi(\rho)$ constructed in Theorem A8, is given by

$$\rho = (9)(78)(4)(3)(256)(1), \text{ respectively } \Phi(\rho) = (9, 7, 8, 4, 3, 2, 5, 6, 1),$$

where the cycles' leaders obey the decreasing order.

Remark A10. Theorem A8 automatically implies the validity of [2, Lemma 4] affirming that when the concept of waste indices is naturally lifted to subpermutations from $S_{n,m}$ (compare [2, p. 3–4]), then for each fixed subset $X \subseteq [n]$ with $|X| = m$, we have $|\{\sigma \in S_{n,m} : \sigma \text{ has codomain } X \text{ and no waste index}\}| = !m$.

Computationally obtained exemplars

More information on the source code is available online[3].

The following are the certifying instances for Observation 1 and Proposition 1. Readers interested in inspecting these objects can find them below as `Julia` lists together with a standalone feasibility checker in Listing A2—the latter can be used as follows:

```
F = [apply_perm(offset,g) for offset in theta for g in G]
k=5; is_mw_indep(F, k)
```

Listing A1. Certifiers

```
#d,n,k;|G|=60,6,6;6
θ = [[2,4,5,3,6,1],[2,5,1,4,6,3],[4,6,1,3,2,5],[2,6,1,5,3,4],[3,2,1,4,5,6],
     [5,4,1,2,3,6],[4,3,1,5,6,2],[6,5,2,3,4,1],[4,3,2,5,1,6],[6,3,2,1,5,4]] # |θ|=10
G = [[1,2,3,4,5,6],[5,4,2,3,6,1],[6,3,4,2,1,5],[4,5,6,1,2,3],[2,1,5,6,3,4],[3,6,1,5,4,2]] # |G|=6

#d,n,k;|G|=60,6,5;6
θ = [[4,5,6,1,2,3],[3,4,5,2,6,1],[1,3,6,2,5,4],[1,3,6,4,2,5],[2,4,6,3,1,5],
     [1,2,6,5,4,3],[2,5,3,6,1,4],[3,6,4,5,1,2],[2,3,1,6,4,5],[2,5,3,1,6,4]] # |θ|=10
G = [[1,2,3,4,5,6],[5,4,2,3,6,1],[6,3,4,2,1,5],[4,5,6,1,2,3],[2,1,5,6,3,4],[3,6,1,5,4,2]] # |G|=6

#d,n,k;|G|=60,5,5;60
θ = [[1,4,3,2,5]] # |θ|=1
G = [[1,2,3,4,5],[1,4,2,3,5],[1,3,4,2,5],[1,4,3,5,2],[1,5,4,3,2],[1,3,5,4,2],
     [1,5,3,2,4],[1,2,5,3,4],[1,3,2,5,4],[1,5,2,4,3],[1,4,5,2,3],[1,2,4,5,3],
     [2,1,5,4,3],[2,4,1,5,3],[2,5,4,1,3],[2,4,5,3,1],[2,3,4,5,1],[2,5,3,4,1],
```

[3] https://www.ac.tuwien.ac.at/files/resources/instances/minw-idp/

```
      [2,3,5,1,4],[2,1,3,5,4],[2,5,1,3,4],[2,3,1,4,5],[2,4,3,1,5],[2,1,4,3,5],
      [4,1,2,5,3],[4,5,1,2,3],[4,2,5,1,3],[4,5,2,3,1],[4,3,5,2,1],[4,2,3,5,1],
      [4,3,2,1,5],[4,1,3,2,5],[4,2,1,3,5],[4,3,1,5,2],[4,5,3,1,2],[4,1,5,3,2],
      [5,1,4,2,3],[5,2,1,4,3],[5,4,2,1,3],[5,2,4,3,1],[5,3,2,4,1],[5,4,3,2,1],
      [5,3,4,1,2],[5,1,3,4,2],[5,4,1,3,2],[5,3,1,2,4],[5,2,3,1,4],[5,1,2,3,4],
      [3,2,4,1,5],[3,1,2,4,5],[3,4,1,2,5],[3,1,4,5,2],[3,5,1,4,2],[3,4,5,1,2],
      [3,5,4,2,1],[3,2,5,4,1],[3,4,2,5,1],[3,5,2,1,4],[3,1,5,2,4],[3,2,1,5,4]] # |G|=60

#d,n,k;|G|=36,8,4;2 swapped G and θ indicate the only right-coset case considered here
G = [[1,2,3,4,5,6,7,8],[8,7,6,5,4,3,2,1]] # |G|=2
θ = [[1,2,3,4,5,6,7,8],[8,5,6,7,2,3,4,1],[1,6,7,3,2,4,8,5],
     [5,2,4,8,6,7,3,1],[1,4,6,5,8,3,2,7],[7,8,3,2,4,6,5,1],
     [5,1,8,7,3,4,2,6],[6,3,4,2,1,8,7,5],[5,1,8,3,6,4,7,2],
     [2,6,4,7,1,8,3,5],[2,6,1,5,8,4,7,3],[3,8,4,7,6,1,5,2],
     [6,2,1,5,4,3,8,7],[7,4,3,8,2,1,5,6],[5,7,1,3,6,4,2,8],
     [8,6,4,2,7,1,3,5],[4,3,8,1,7,6,2,5],[5,7,6,2,3,8,1,4]] # |θ|=18, inferred from Na et al. 2023, Proposition 4.6(iii)

#d,n,k;|G|=48,8,4;24
θ = [[5,8,7,2,3,6,1,4],[6,5,1,4,2,3,8,7]] # |θ|=2
G = [[1,2,3,4,5,6,7,8],[1,5,7,2,4,3,6,8],[1,4,6,5,2,7,3,8],[8,7,4,3,6,5,2,1],
     [8,6,2,7,3,4,5,1],[8,3,5,6,7,2,4,1],[3,6,8,1,2,7,5,4],[3,2,5,6,1,8,7,4],
     [3,1,7,2,6,5,8,4],[4,5,1,8,7,2,6,3],[4,7,6,5,8,1,2,3],[4,8,2,7,5,6,1,3],
     [6,4,2,7,1,8,3,5],[6,1,3,4,7,2,8,5],[6,7,8,1,4,3,2,5],[5,3,7,2,8,1,4,6],
     [5,8,4,3,2,7,1,6],[5,2,1,8,3,4,7,6],[2,8,5,6,4,3,1,7],[2,4,1,8,6,5,3,7],
     [2,6,3,4,8,1,5,7],[7,1,6,5,3,4,8,2],[7,3,8,1,5,6,4,2],[7,5,4,3,1,8,6,2]] # |G|=24

#d,n,k;|G|=24,7,4;12
θ = [[7,2,5,6,3,4,1],[7,6,4,3,1,5,2]] # |θ|=2
G = [[1,2,3,4,5,6,7],[1,2,3,5,4,7,6],[3,2,1,5,4,6,7],[3,2,1,4,5,7,6],
     [6,2,7,3,1,5,4],[6,2,7,1,3,4,5],[7,2,6,1,3,5,4],[7,2,6,3,1,4,5],
     [5,2,4,7,6,1,3],[5,2,4,6,7,3,1],[4,2,5,6,7,1,3],[4,2,5,7,6,3,1]] # |G|=12

#d,n,k;|G|=24,6,4;24
θ = [[2,3,5,1,6,4]] # |θ|=1
G = [[1,2,3,4,5,6],[2,1,3,4,6,5],[5,6,3,4,1,2],[6,5,3,4,2,1],[1,2,4,3,6,5],[2,1,4,3,5,6],
     [6,5,4,3,1,2],[5,6,4,3,2,1],[3,4,5,6,1,2],[4,3,5,6,2,1],[1,2,5,6,3,4],[2,1,5,6,4,3],
     [3,4,6,5,2,1],[4,3,6,5,1,2],[2,1,6,5,3,4],[1,2,6,5,4,3],[5,6,1,2,3,4],[6,5,1,2,4,3],
     [3,4,1,2,5,6],[4,3,1,2,6,5],[5,6,2,1,4,3],[6,5,2,1,3,4],[4,3,2,1,5,6],[3,4,2,1,6,5]] # |G|=24

#d,n,k;|G|=12,5,4;6
θ = [[1,4,2,3,5],[5,2,4,3,1]] # |θ|=2
G = [[1,2,3,4,5],[1,3,2,5,4],[2,3,1,4,5],[2,1,3,5,4],[3,1,2,4,5],[3,2,1,5,4]] # |G|=6

#d,n,k;|G|=12,4,4;12
θ = [[1,3,4,2]] # |θ|=1
G = [[1,2,3,4],[1,3,4,2],[1,4,2,3],[4,3,2,1],[4,2,1,3],[4,1,3,2],
     [3,4,1,2],[3,1,2,4],[3,2,4,1],[2,1,4,3],[2,4,3,1],[2,3,1,4]] # |G|=12
```

Listing A2. Feasibility checker

```
using Combinatorics

"""
Returns the composition of two permutations ‘outer‘ and ‘inner‘ which both are passed as lists.
"""
function apply_perm(outer, inner)
   return [outer[inner[j]] for j in 1:length(inner)]
end

"""
Returns the union over all k in ‘k_range‘ of all sets of semiordered patterns SOP(n,k).
"""
function gen_SOP_n_k_range(k_range, n)
   SOP = []
   for j in k_range
      tmp_cll = []
      for cmb_j in combinations(1:n, j)
         for idx in 1:length(cmb_j)
            copy_cmb_j = copy(cmb_j)
            el = cmb_j[idx]
            deleteat!(copy_cmb_j, idx)
            push!(tmp_cll, tuple([el; sort(copy_cmb_j)]...))
         end
      end
      sort!(tmp_cll)
      append!(SOP, tmp_cll)
   end
   return SOP
end

"""
Checks if the family ‘F‘ is ‘k‘-restricted minwise independent.
"""
function is_mw_indep(F, k)
   SOP = gen_SOP_n_k_range(2:k, length(F[1]))
   unmet = []
   is_valid = true
   for sop in SOP
      counter = 0
```

```
    for l in 1:length(F)
      if all([F[l][sop[1]] < F[l][sop[j]] for j in 2:length(sop)])
        counter += 1
      end
    end
    if counter != div(length(F), length(sop))
      push!(unmet, (sop, counter))
      is_valid = false
    end
  end
  is_valid == false && println("unmet multiplicity: ", unmet)
  return is_valid
end
```