A group-based structure for perfect sequence covering arrays

Jingzhou Na   Jonathan Jedwab   Shuxing Li

4 February 2022 (revised 28 September 2022)

Abstract

An (n, k)-perfect sequence covering array with multiplicity λ, denoted $\mathrm{PSCA}(n, k, \lambda)$, is a multiset whose elements are permutations of the sequence $(1, 2, \dots, n)$ and which collectively contain each ordered length k subsequence exactly λ times. The primary objective is to determine for each pair (n, k) the smallest value of λ, denoted $g(n, k)$, for which a $\mathrm{PSCA}(n, k, \lambda)$ exists; and more generally, the complete set of values λ for which a $\mathrm{PSCA}(n, k, \lambda)$ exists. Yuster recently determined the first known value of $g(n, k)$ greater than 1, namely $g(5, 3) = 2$, and suggested that finding other such values would be challenging. We show that $g(6, 3) = g(7, 3) = 2$, using a recursive search method inspired by an old algorithm due to Mathon. We then impose a group-based structure on a perfect sequence covering array by restricting it to be a union of distinct cosets of a prescribed nontrivial subgroup of the symmetric group S_n. This allows us to determine the new results that $g(7, 4) = 2$ and $g(7, 5) \in \{2, 3, 4\}$ and $g(8, 3) \in \{2, 3\}$ and $g(9, 3) \in \{2, 3, 4\}$. We also show that, for each $(n, k) \in \{(5, 3), (6, 3), (7, 3), (7, 4)\}$, there exists a $\mathrm{PSCA}(n, k, \lambda)$ if and only if $\lambda \geq 2$; and that there exists a $\mathrm{PSCA}(8, 3, \lambda)$ if and only if $\lambda \geq g(8, 3)$.

1   Introduction

Throughout, let k and n be integers satisfying $2 \leq k \leq n$. Let S_n be the set of permutations of $[n] := \{1, 2, \dots, n\}$, and let $S_{n,k}$ be the set of $\frac{n!}{(n-k)!}$ ordered k-subsets of $[n]$. An (n, k)-perfect sequence covering array with multiplicity λ, denoted $\mathrm{PSCA}(n, k, \lambda)$, is a multiset P with elements in S_n such that each element of $S_{n,k}$ is a k-subsequence of exactly λ elements of P. Equivalently, regarding the elements of P as n-sequences, we say that each element of $S_{n,k}$ is *covered* by exactly λ sequences of P. For example, the subset

$$\{12345, 13254, 14523, 15432, 24315, 25413, 34512, 35214, 42513, 43215, 52314, 53412\}$$

of S_5 is a $\mathrm{PSCA}(5, 3, 2)$. If P is a $\mathrm{PSCA}(n, k, 1)$, then P is a set (not a multiset). The size of the multiset of k-subsequences covered by a $\mathrm{PSCA}(n, k, \lambda)$ P can be counted both as $|S_{n,k}|\lambda$ and as $\binom{n}{k}|P|$, from which we obtain the necessary condition $|P| = k!\lambda$.

Perfect sequence covering arrays are related to several other objects from combinatorial design theory. A PSCA(n, k, λ) P is equivalent to a k-(n, n, λ) directed design $([n], P)$ [2, Sect. VI.20]. If a PSCA(n, k, λ) exists, then it achieves the largest possible size of a k-(n, n, λ) directed packing [16], [5]. Replacing "exactly λ elements" in the definition of a perfect sequence covering array by "at least one element" gives an (n, k) sequence covering array, or equivalently a completely k-scrambling set of $[n]$ [19], [6, Section 5], [10]. For a comprehensive study of constructions, nonexistence results, and search methods for sequence covering arrays, see Chee et al. [3]. Sequence covering arrays are useful in various applications in which faults can arise when certain events occur in a particular order [21, 20, 1, 9, 24, 23, 12]. For example, the faults might be adverse reactions when a sequence of medications is taken in a certain order. In order to determine whether faults arise under all possible ordered subsets of at most k out of n events, we require a set of tests in which each ordering of each subset of k events occurs: this is given by an (n, k)-sequence covering array. A PSCA$(n, k, 1)$, if it exists, is the smallest possible size of an (n, k)-sequence covering array and so represents the most cost-efficient method of carrying out the required set of tests.

We define $g(n, k)$ to be the smallest λ for which a PSCA(n, k, λ) exists. This value is well-defined, because S_n is a trivial PSCA$(n, k, \frac{n!}{k!})$ and so $g(n, k) \leq \frac{n!}{k!}$ for all $k \leq n$. The central objective in the study of perfect sequence covering arrays is to determine, for each pair (n, k), the value of $g(n, k)$ and more generally the complete set of values λ for which there exists a PSCA(n, k, λ). The current state of knowledge for $g(n, k)$ for small values of n and k is shown in Table 1. We are concerned in this paper with exact values rather than asymptotic bounds.

The rest of the paper is organized in the following way. Section 2 describes previous results for the value of $g(n, k)$, including constructions, combinatorial nonexistence results, computer nonexistence results, and asymptotic results. Section 3 describes a recursive algorithm for finding all possible examples of a PSCA(n, k, λ) without repeated elements, for arbitrary $\lambda \geq 1$. Section 4 identifies a group-based structure shared by many examples of perfect sequence covering arrays, and modifies the search algorithm by prescribing this structure. This allows us to determine new values or bounds for $g(n, k)$ for several pairs (n, k). It also allows us to find examples of new parameter sets (n, k, λ) for which a PSCA(n, k, λ) exists, providing evidence for a positive answer to a question posed by Charlie Colbourn (personal communication, Sept. 2021): does the existence of a PSCA(n, k, λ) imply the existence of a PSCA$(n, k, \lambda + 1)$? Section 5 presents open problems arising from our results.

The results of this paper are largely based on the Master's thesis of the first author [17], which contains additional examples and visualizations.

2 Previous results for $g(n, k)$

In this section, we summarize previous results for the value of $g(n, k)$, including constructions, combinatorial nonexistence results, computer nonexistence results, and asymptotic results.

We begin with two trivial constructions.

Lemma 2.1. *Let $n \geq 2$. Then $g(n, 2) = g(n, n) = 1$.*

Proof. The set $\{12 \cdots n, \ n \cdots 21\}$ is trivially a PSCA$(n, 2, 1)$, and so $g(n, 2) = 1$. The set S_n is trivially a PSCA$(n, n, 1)$, and so $g(n, n) = 1$. $\square$

The following composition construction is also trivial.

Lemma 2.2. *Suppose there exists a* $\mathrm{PSCA}(n, k, \lambda)$ *and a* $\mathrm{PSCA}(n, k, \mu)$. *Then their multiset union is a* $\mathrm{PSCA}(n, k, \lambda + \mu)$.

The following two bounds are straightforward to prove.

Lemma 2.3.

(*i*) *Let* $k \leq n - 1$. *Then* $g(n, k) \geq g(n - 1, k)$.

(*ii*) *Let* $k \geq 2$. *Then* $g(n, k) \geq \frac{1}{k} \, g(n, k - 1)$.

Proof. For (*i*), delete the symbol n from each sequence of a $\mathrm{PSCA}(n, k, \lambda)$ to give a $\mathrm{PSCA}(n-1, k, \lambda)$. For (*ii*), regard a $\mathrm{PSCA}(n, k, \lambda)$ as a $\mathrm{PSCA}(n, k - 1, k\lambda)$. $\square$

The following result was proved in terms of perfect codes capable of correcting single deletions.

Theorem 2.4 (Levenshtein [14, Thm. 3.1]). *The set* S_n *can be partitioned into* n *sets of sequences, each of which is a* $\mathrm{PSCA}(n, n - 1, 1)$. *Therefore* $g(n, n - 1) = 1$.

Levenshtein [13, p. 140] conjectured in 1990 that the only values of k for which $g(n, k) = 1$ are those provided by Lemma 2.1 and Theorem 2.4, namely 2, $n - 1$, and n. This was disproved by the following result.

Proposition 2.5 (Mathon 1991, reported in [16, p. 191]). *There exists a* $\mathrm{PSCA}(6, 4, 1)$.

Mathon and van Trung showed by hand that $g(5, 3) > 1$ [16, Thm 3.2] (see [17, p. 13] for a minor correction to the proof), and established the following nonexistence results by computer search.

Proposition 2.6 (Mathon and van Trung [16, Sects. 4 & 6]). *We have* $g(7, 4) > 1$ *and* $g(7, 5) > 1$ *and* $g(8, 6) > 1$.

Mathon and van Trung concluded that 4 might be the only value of k for which Levenshtein's conjecture fails. We express their revised conjecture in the following form, using Lemma 2.3 (*i*).

Conjecture 2.7 (Mathon and van Trung [16, p. 198]). *Let* $k \notin \{2, 4\}$. *Then* $g(k + 2, k) > 1$.

More than twenty years after publication of [16], the smallest open case of Conjecture 2.7 remains $k = 7$.

The search result $g(7, 4) > 1$ stated in Proposition 2.6 was established in 2004 via an elegant combinatorial proof that does not appear to have been widely recognized outside of the published context of perfect deletion-correcting codes. We therefore rephrase it here.

Theorem 2.8 (Klein [11, Thm 3.2]). *We have* $g(7, 4) > 1$.

Proof. Suppose, for a contradiction, that P is a $\mathrm{PSCA}(7, 4, 1)$. We may assume after relabelling that P contains the sequence 1234567. Let T be the set of elements in $P \setminus \{1234567\}$ that contain one of the 3-subsequences in the set

$$U = \{124, 134, 234\},$$

and let T' be the set of elements in $P \setminus \{1234567\}$ that contain one of the 4-subsequences in the set

$$U' = \{3124, 1324, 1243, 2134, 1342, 2314, 2341\}.$$

It is easy to check that $T = T'$. By the PSCA property, the set $P \setminus \{1234567\}$ covers each element of U', so there are at least $|U'| = 7$ elements in $T' = T$.

Now in the sequence $1234567 \in P$, each of the symbols 5, 6, 7 occurs after each of the 3-subsequences in U. Therefore in every element of T, each symbol $5, 6, 7$ occurs before the symbol 4 (otherwise P would cover some 4-subsequence more than once). Since there are at least 7 elements of T, but there are only $3! < 7$ ways to order the symbols $5, 6, 7$ to occur before the symbol 4, we conclude that there are at least two elements of T covering the same 4-subsequence (formed from some permutation of symbols $5, 6, 7$ followed by the symbol 4). This gives the required contradiction. $\square$

The following nonexistence result was proved using matrix rank arguments and by reference to results on covering arrays such as [4].

Theorem 2.9 (Chee et al. [3, Thm 2.3]). *Let $k \geq 3$. Then $g(2k, k) > 1$.*

Although our interest in this paper is in determining the exact value of $g(n, k)$ for small n and k, we summarize in Theorems 2.10 and 2.11 below the best known asymptotic bounds on the growth rate of $g(n, k)$ as n and k grow. These results improve on previous asymptotic results for completely scrambling sets [19, 10, 6, 18].

Theorem 2.10 holds for general k, and was proved by combining combinatorial arguments with a result due to Wilson [22, Thm. 1] on the rank of a set inclusion matrix over a finite field.

Theorem 2.10 (Yuster [25, Thm. 1]). *Let $k \geq 4$ be an integer.*

(i) *If $k/2$ is a prime, then for all $n \geq k$ we have*

$$g(n, k) \geq \frac{\binom{n}{k/2} - \binom{n}{k/2-1}}{k!}.$$

(ii) *Let n and k grow such that $n \gg k$. Then $g(n, k) > n^{k/2 - o_k(1)}$ (where $o_k(1)$ represents a function that approaches 0 as $k \to \infty$).*

Theorem 2.11 holds for the case $k = 3$. The proof of the upper bound arises from a recursive construction that builds a $\mathrm{PSCA}(n^2, 3, 2(n+1)\lambda)$ from a $\mathrm{PSCA}(n, 3, \lambda)$, using a finite affine plane of order n where n is a power of 3.

Theorem 2.11 (Yuster [25, Thm. 2]). *Let $n \geq 3$. Then $n/6 \leq g(n, 3) \leq Cn(\log_2 n)^{\log_2 7}$ for some absolute constant C.*

In 2020, Yuster [25] determined that $g(5, 3) = 2$ by exhibiting a $\mathrm{PSCA}(5, 3, 2)$. This was the first exact value of $g(n, k)$ greater than 1 to be determined. To describe how this result was found, we introduce two definitions.

Definition 2.12. *The (n, k)-incidence matrix is the $n! \times \frac{n!}{(n-k)!}$ array whose rows are indexed by the elements of S_n, whose columns are indexed by the elements of $S_{n,k}$, and whose (x, y) entry is 1 if $x \in S_n$ covers $y \in S_{n,k}$ and is 0 otherwise.*

Each row sum of the (n, k)-incidence matrix is $\binom{n}{k}$ and each column sum is $\frac{n!}{k!}$.

Definition 2.13. *Let X be a multiset with elements in S_n. The* repetition vector of X *with respect to k, written $\mathrm{rv}_k(X)$, is the sum of the rows of the (n,k)-incidence matrix that are indexed by the elements of X.*

The vector $\mathrm{rv}_k(X)$ has length $\frac{n!}{(n-k)!}$, and its jth entry is the number of sequences in X that cover k-subsequence j. The multiset X is therefore a $\mathrm{PSCA}(n,k,\lambda)$ if and only if each entry of $\mathrm{rv}_k(X)$ is λ.

We can now describe the $\mathrm{PSCA}(5,3,2)$ found by Yuster. We shall reinterpret this example in Section 4.1.

Example 2.14 (Yuster [25, Prop. 3.4])**.** *Let*

$$X = \{12345, 43215, 35214, 14523, 25413, 53412\}.$$

Then the length 60 vector $\mathrm{rv}_3(X)$ has four entries 0 (in the positions indexed by the 3-subsequences 132, 231, 154, 451), four entries 2 (in the positions indexed by the 3-subsequences 123, 321, 145, 541), and the remaining 52 entries 1.

Let $\sigma = 13254 \in S_5$ and write $X\sigma = \{x\sigma : x \in X\}$, where we follow the convention that the composition of permutations π, σ given by $\pi\sigma$ represents the action of π followed by σ. Then the repetition vector of

$$X\sigma = \{13254, 52314, 24315, 15432, 34512, 42513\}$$

has the same property as $\mathrm{rv}_3(X)$, but the positions in $\mathrm{rv}_3(X\sigma)$ of the 0 and 2 entries are interchanged with those in $\mathrm{rv}_3(X)$. This ensures that every entry of $\mathrm{rv}_3(X) + \mathrm{rv}_3(X\sigma)$ is 2, and therefore that $X \cup X\sigma$ is a $\mathrm{PSCA}(5,3,2)$. Since $g(5,3) > 1$ [16, Thm. 3.2], we conclude that $g(5,3) = 2$.

The original motivation for this paper was the challenge provided by Yuster's concluding statement [25, p. 592] that

> "Proving additional exact values of $g(n,k)$ which are not of unit multiplicity in addition to $g(5,3)$ also seems challenging".

Table 1 summarizes all previously known exact values of $g(n,k)$ for small n and k.

3 Recursive search algorithm for $\mathrm{PSCA}(n,k,\lambda)$

In this section, we describe a recursive algorithm for finding all possible examples of a $\mathrm{PSCA}(n,k,\lambda)$ without repeated elements, for arbitrary $\lambda \geq 1$. The algorithm is a tree search that attempts to build a $\mathrm{PSCA}(n,k,\lambda)$ one n-sequence at a time without covering any k-subsequence more than λ times, backtracking when this is not possible. Although we have followed Yuster [25, p. 586] in allowing a perfect sequence covering array to be a multiset, we consider that in many respects it is more natural to restrict the definition of a $\mathrm{PSCA}(n,k,\lambda)$ to a set and have therefore formulated the algorithm to exclude repeated elements. For $\lambda = 1$, this restriction is redundant: if the algorithm terminates without output for parameters $(n,k,1)$, then we can conclude that no $\mathrm{PSCA}(n,k,1)$ exists and so $g(n,k) > 1$.

Table 1: Previously known exact values of $g(n,k)$ for small n and k, and their sources. (s1): Lemma 2.1. (s2): Theorem 2.4. (s3): Example 2.14. (s4): Proposition 2.5. (s5): Proposition 2.6. (s6): Theorem 2.8. (s7): Lemma 2.3 (i).

n \ k	2	3	4	5	6	7
2	1 (s1)					
3	1 (s1)	1 (s1)				
4	1 (s1)	1 (s2)	1 (s1)			
5	1 (s1)	2 (s3)	1 (s2)	1 (s1)		
6	1 (s1)	≥ 2 (s7)	1 (s4)	1 (s2)	1 (s1)	
7	1 (s1)	≥ 2 (s7)	≥ 2 (s5,s6)	≥ 2 (s5)	1 (s2)	1 (s1)
8	1 (s1)	≥ 2 (s7)	≥ 2 (s7)	≥ 2 (s7)	≥ 2 (s5)	1 (s2)
9	1 (s1)	≥ 2 (s7)	≥ 2 (s7)	≥ 2 (s7)	≥ 2 (s7)	?

3.1 Idea of algorithm

Let $A = (a_{x,y})$ be the (n,k)-incidence matrix (where $x \in S_n$ and $y \in S_{n,k}$). We wish to construct a PSCA(n,k,λ) by finding a $(k!\lambda)$-subset X of S_n for which each entry of $\mathrm{rv}_k(X)$ is λ, which is equivalent by Definition 2.12 to

$$\sum_{x \in X} a_{x,y} = \lambda \quad \text{for each } y.$$

We initialize X to be empty. We then add one element of S_n to X at a time, subject to the condition that

$$\sum_{x \in X} a_{x,y} \leq \lambda \quad \text{for each } y. \tag{1}$$

The algorithm succeeds in finding a PSCA(n,k,λ) if $|X|$ reaches $k!\lambda$. If it is not possible to add an element of S_n to X subject to (1), we backtrack.

We keep track of two sets, Y and L. The set Y contains the k-subsequences not yet covered λ times by the n-sequences in X, namely the values y for which $\sum_{x \in X} a_{x,y} < \lambda$. The set L contains the candidates for enlarging X, namely the n-sequences that do not cover a k-subsequence already covered λ times. We also keep track of the repetition vectors $R = \mathrm{rv}_k(X) = (\sum_{x \in X} a_{x,y}) = (r_y)$ and $M = \mathrm{rv}_k(L) = (\sum_{\ell \in L} a_{l,y}) = (m_y)$, although only at the positions y indexed by Y. We seek to enlarge X so that the vector R attains the value λ in every position. Each position of the vector M contains the largest amount by which R can be increased in that position (if every candidate in L were added to X).

Suppose that X currently contains fewer than $k!\lambda$ elements. If there is a k-subsequence that cannot be covered λ times, even by adding every candidate in L to X (that is, $r_y + m_y < \lambda$ for some $y \in Y$), then we terminate the branch early and backtrack. Otherwise, we find the set Y' of $y \in Y$ for which r_y attains its maximum value (that is, the k-subsequences that are not yet covered λ times but are closest to being so). At the next iteration of the algorithm, we choose one $y \in Y'$ and recurse by attempting to add to X each of the elements of $\{\ell \in L : a_{l,y} = 1\}$ in turn (each such addition causing the k-subsequence y to be covered one more time). In order to reduce the number of tree branches that must be searched, we choose a value of $y \in Y'$ for which $|\{\ell \in L : a_{l,y} = 1\}| = m_y$

is minimized: call this value y^*. Although the choice of y^* is restricted to the subset Y' of Y, and might not be unique within Y', the algorithm is exhaustive over the possibilities for incrementing the value of r_{y^*}. Since we require each entry of r_y for $y \in Y$ to eventually reach the value λ, the algorithm finds every possible example of a PSCA(n, k, λ) regardless of the sequence of values y^* chosen.

For each $x \in L$ in turn for which $a_{x,y^*} = 1$, we update the sets X, Y, L and the repetition vectors R, M and then recurse. We update X by adding the n-sequence x. We update Y by removing all k-subsequences newly covered λ times. We update R by adding the row of A indexed by x to it. We update L by removing x (so that the same n-sequence x cannot be added to X a second time) and by removing each n-sequence covering a k-subsequence that is newly covered λ times (because the later inclusion of such an n-sequence in X would violate (1)). We update M by subtracting the rows of A that have just been removed from L.

Pseudocode implementing this search procedure is given in Algorithm 1. By calling the procedure SEARCH with $X = \emptyset$ and $Y = S_{n,k}$ and $R = (0, \ldots, 0)$ and $L = S_n$ and $M = (\frac{n!}{k!}, \ldots, \frac{n!}{k!})$, we obtain every possible PSCA(n, k, λ) without repeated elements. Furthermore, we may assume after relabelling symbols that the perfect sequence covering array contains the sequence $12 \cdots n$. We therefore replace, for the first iteration of Algorithm 1, the set $L(y^*)$ at Line 18 by the single-element set $\{12 \cdots n\}$.

3.2 Discussion

The nonexistence results for a PSCA$(n, k, 1)$ given in Proposition 2.6 were obtained by Mathon and van Trung using a search algorithm for a directed t-packing [16, p. 163], that in turn relies on a algorithm due to Mathon [15] for finding spreads in an incidence structure. A key feature of this algorithm is that each successive iteration removes rows and columns of the incidence matrix describing the incidence structure, leading to reduced time and space complexity. Mathon noted [15, p.164] that "An actual implementation of this algorithm requires good data structures to facilitate fast and efficient computations in, and updating of the various point and line sets", without explicitly describing these data structures.

Algorithm 1 for finding a PSCA(n, k, λ) without repeated elements is inspired by Mathon's algorithm, and reduces to a broadly equivalent form in the special case $\lambda = 1$. In particular, the idea of improving efficiency by restricting the search to the subset Y' of Y in Algorithm 1 is taken directly from Mathon's paper [15]. The cases $\lambda > 1$ do not have a corresponding form in the context of a directed t-packing and so were not considered in [15]. Algorithm 1 also contains a feature not described in [15] or [16] that leads to a significant speed advantage for all cases $\lambda \geq 1$: keeping track of the vector M and passing it as a recursion parameter, which avoids having to calculate $|Y|$ column sums over $|L|$ rows when carrying out the early termination test at Line 13. The calculations in Lines 13, 16, 17 can be accomplished in linear time with a single pass through the positions indexed by Y.

The input space requirement of Algorithm 1 is determined by the representation of the (n, k)-incidence matrix. There is no need to store this matrix explicitly as an $n! \times \frac{n!}{(n-k)!}$ matrix over $\{0, 1\}$: it is sufficient to store the positions of the 1 entries in each row, and the positions of the 1 entries in each column.

Algorithm 1 Tree search for $\mathrm{PSCA}(n, k, \lambda)$ by backtracking

---

Input: PSCA parameters (n, k, λ) and $A = \big(a_{x,y} : x \in S_n,\ y \in S_{n,k}\big) = (n, k)$-incidence matrix

1: **procedure** $\mathrm{SEARCH}(X, Y, R, L, M)$
2:     % X = sequences of partial $\mathrm{PSCA}(n, k, \lambda)$ P
3:     % write $(r_y) = \mathrm{rv}_k(X) = (\sum_{x \in X} a_{x,y}) = $ sum of rows of A indexed by X
4:     % $Y = \{y \in S_{n,k} : r_y < \lambda\} = k$-subsequences not yet covered λ times by P
5:     % $R = (r_y : y \in Y) = $ entries of $\mathrm{rv}_k(X)$ indexed by Y
6:     % $L = \{\ell \notin X : a_{\ell,y} = 0 \text{ for all } y \notin Y\} = $ candidates for enlarging X
7:     % write $(m_y) = \mathrm{rv}_k(L) = (\sum_{\ell \in L} a_{\ell,y}) = $ sum of rows of A indexed by L
8:     % $M = (m_y : y \in Y) = $ entries of $\mathrm{rv}_k(L)$ indexed by Y
9:     **if** $|X| = k!\lambda$ **then**
10:        record X as a $\mathrm{PSCA}(n, k, \lambda)$.
11:        **return**
12:     **end if**
13:     **if** $r_y + m_y < \lambda$ for some $y \in Y$ **then**
14:        **return**   % terminate branch early
15:     **end if**
16:     let Y' be the set of $y \in Y$ for which r_y attains its maximum value.
17:     choose an arbitrary $y = y^* \in Y'$ for which m_y attains its minimum value.
18:     $L(y^*) := \{x \in L : a_{x,y^*} = 1\}$
19:     **for each** $x \in L(y^*)$ **do**
20:        $X_{\mathrm{new}} := X \cup \{x\}$
21:        $Y_{\mathrm{new}} := Y \setminus \{y : r_y + a_{x,y} = \lambda\}$
22:        $R_{\mathrm{new}} := (r_y + a_{x,y} : y \in Y_{\mathrm{new}})$
23:        $B := \{x\} \cup \{\ell \in L : a_{\ell,y} = 1 \text{ for at least one } y \in Y \setminus Y_{\mathrm{new}}\}$
24:        $L_{\mathrm{new}} := L \setminus B$
25:        $M_{\mathrm{new}} := (m_y - \sum_{\ell \in B} a_{\ell,y} : y \in Y_{\mathrm{new}})$
26:        $\mathrm{SEARCH}(X_{\mathrm{new}}, Y_{\mathrm{new}}, R_{\mathrm{new}}, L_{\mathrm{new}}, M_{\mathrm{new}})$
27:     **end for**
28: **end procedure**

3.3 New values for $g(n,k)$

Algorithm 1 finds the following examples of a PSCA$(6,3,2)$ and a PSCA$(7,3,2)$. By reference to Table 1, this gives the new results $g(6,3) = g(7,3) = 2$.

Proposition 3.1.

(i) *The following* 12 *sequences form a* PSCA$(6,3,2)$

$$123456 \quad 154326 \quad 216543 \quad 245613 \quad 354162 \quad 361452$$
$$423165 \quad 461325 \quad 516234 \quad 532614 \quad 632541 \quad 645231$$

and therefore $g(6,3) = 2$.

(ii) *The following* 12 *sequences form a* PSCA$(7,3,2)$

$$1234567 \quad 1573426 \quad 3275641 \quad 3617524 \quad 4261735 \quad 4756123$$
$$5164327 \quad 5243176 \quad 6257314 \quad 6345721 \quad 7216453 \quad 7431625$$

and therefore $g(7,3) = 2$.

4 PSCA(n,k,λ) as union of cosets of a prescribed subgroup

In this section, we identify an algebraic structure shared by many examples of perfect sequence covering arrays. We then modify Algorithm 1 to search for perfect sequence covering arrays having this prescribed structure, thereby determining new values or bounds for $g(n,k)$ for several pairs (n,k). The algorithm also finds examples of new parameter sets (n,k,λ) for which a PSCA(n,k,λ) exists. As in Section 3, we restrict the search to perfect sequence covering arrays without repeated elements.

4.1 Motivation

The PSCA$(5,3,2)$ given in Example 2.14 was constructed by Yuster as $X \cup X\sigma$ for a 6-subset X of S_5 and a permutation $\sigma \in S_5$. We can equivalently interpret this perfect sequence covering array as the union of six left cosets of the order 2 subgroup $\langle\sigma\rangle$ of S_5, by reading the following table not by rows (as $X \cup X\sigma$) but by columns (as $\bigcup_{x\in X} x\langle\sigma\rangle$, where X is a set of left coset representatives for $\langle\sigma\rangle$).

PSCA$(5,3,2)$ with $\sigma = 13254$

	$\langle\sigma\rangle$	$43215\,\langle\sigma\rangle$	$35214\,\langle\sigma\rangle$	$14523\,\langle\sigma\rangle$	$25413\,\langle\sigma\rangle$	$53412\,\langle\sigma\rangle$
X	12345	43215	35214	14523	25413	53412
$X\sigma$	13254	52314	24315	15432	34512	42513

We can likewise reinterpret the PSCA$(6,3,2)$ and PSCA$(7,3,2)$ of Proposition 3.1, found using Algorithm 1, as the union of six left cosets of a suitable order 2 subgroup $\langle\sigma\rangle$ of S_6 and S_7, respectively.

PSCA(6, 3, 2) with $\sigma = 154326$						
	$\langle\sigma\rangle$	$216543\,\langle\sigma\rangle$	$354162\,\langle\sigma\rangle$	$461325\,\langle\sigma\rangle$	$532614\,\langle\sigma\rangle$	$645231\,\langle\sigma\rangle$
X	123456	216543	354162	461325	532614	645231
$X\sigma$	154326	516234	423165	361452	245613	632541

PSCA(7, 3, 2) with $\sigma = 4261735$						
	$\langle\sigma\rangle$	$1573426\,\langle\sigma\rangle$	$3275641\,\langle\sigma\rangle$	$3617524\,\langle\sigma\rangle$	$5164327\,\langle\sigma\rangle$	$5243176\,\langle\sigma\rangle$
X	1234567	1573426	3275641	3617524	5164327	5243176
$X\sigma$	4261735	4756123	6257314	6345721	7431625	7216453

This motivates us to seek a PSCA(n, k, λ) as the union of $\frac{k!\lambda}{|G|}$ distinct cosets of a nontrivial subgroup G of S_n, where the parameter λ need not take the value 2; the subgroup G need not be cyclic, nor have order 2; and we can choose either right cosets of a subgroup or left cosets. However, we shall see in Section 4 that the use of left cosets leads to a simplification in the search (using conjugacy classes), and a richer existence pattern (involving a larger subgroup G).

We note that a single PSCA(n, k, λ) can admit more than one representation as a union of distinct cosets of a subgroup of S_n. For example, the PSCA$(6, 3, 2)$ represented above as six left cosets of the order 2 subgroup $\langle 154326\rangle$ can also be represented as the entire order 12 subgroup $\langle 154326, 216543\rangle \cong D_{12}$. Likewise, the PSCA$(7, 3, 2)$ represented above as six left cosets of the order 2 subgroup $\langle 4261735\rangle$ can also by represented as six right cosets of the order 2 subgroup $\langle 3617524\rangle$, and as two left cosets of the order 6 subgroup $\langle 3275641, 4261735\rangle \cong S_3$.

We remark that several aspects of our approach can be recognized in Mathon and van Trung's work [16]. We note in particular the following refinement of Proposition 2.5, found by computer search. We write D_{2n} for the dihedral group of order $2n$.

Proposition 4.1 (Mathon and van Trung [16, Thm 4.1])**.** *Up to isomorphism, there are exactly two examples P_1, P_2 of a* PSCA$(6, 4, 1)$:

(i) *the 24 sequences of P_1 comprise a subgroup $G_1 \cong S_4$ of S_6; the automorphism group of P_1 is isomorphic to S_4.*

(ii) *the 24 sequences of P_2 comprise a union of three right cosets of a subgroup $G_2 \cong D_8$ of S_6; the automorphism group of P_2 is isomorphic to D_8. (See [17, p. 29] for a correction to the listing of P_2 in [16, p. 192].)*

4.2 Left or right cosets

We next show how, for a perfect sequence covering array comprising a union of distinct cosets of a nontrivial subgroup, there is a fundamental distinction between left and right cosets. We firstly note that the right action of a permutation on a subset X of S_n permutes the entries of the repetition vector $\mathrm{rv}_k(X)$.

Lemma 4.2. *Let X be a subset of S_n and let $\sigma \in S_n$. Then the vector $\mathrm{rv}_k(X\sigma)$ is obtained by permuting the entries of the vector $\mathrm{rv}_k(X)$.*

Proof. Under the convention that the permutation composition $\pi\sigma$ represents the action of π followed by σ, the right action of σ on the set X permutes the symbols in $[n]$ and so permutes the elements of $S_{n,k}$. The result follows by Definition 2.13. $\square$

It follows that the right action of a permutation $\sigma \in S_n$ maps one $\text{PSCA}(n, k, \lambda)$ to another.

Corollary 4.3. *Suppose P is a $\text{PSCA}(n, k, \lambda)$, and let $\sigma \in S_n$. Then $P\sigma$ is also a $\text{PSCA}(n, k, \lambda)$.*

Subgroups G and H of S_n are *conjugate* in S_n if $H = \sigma G \sigma^{-1}$ for some $\sigma \in S_n$. Conjugacy is an equivalence relation on the set of subgroups of S_n, and the equivalence class of G under conjugation is the *conjugacy class* $\text{Cl}(G) := \{\sigma G \sigma^{-1} : \sigma \in S_n\}$. Consider searching over all nontrivial subgroups G of S_n and all sets $\mathcal{R}$ (of cardinality $\frac{k!\lambda}{|G|}$) of left coset representatives for G to find a perfect sequence covering array of the form $\bigcup_{\pi \in \mathcal{R}} \pi G$. We now use Corollary 4.3 to show that it is sufficient to restrict attention to a single representative G from each conjugacy class of subgroups of S_n. This drastically reduces the required computation for an exhaustive search. For example, S_7 contains 11299 nontrivial subgroups, but only 95 nontrivial conjugacy classes of subgroups.

Theorem 4.4. *Let G be a subgroup of S_n and let $H \in \text{Cl}(G)$. Suppose there is a $\text{PSCA}(n, k, \lambda)$ that is a union of distinct left cosets of H. Then there is a $\text{PSCA}(n, k, \lambda)$ which is a union of distinct left cosets of G.*

Proof. Let P be a $\text{PSCA}(n, k, \lambda)$ that can be written as

$$P = \bigcup_{\pi \in \mathcal{R}} \pi H \tag{2}$$

for a set $\mathcal{R}$ of left coset representatives for H. Since $H \in \text{Cl}(G)$, for some $\sigma \in S_n$ we can write

$$P = \bigcup_{\pi \in \mathcal{R}} \pi (\sigma G \sigma^{-1})$$

$$= \left(\bigcup_{\mu \in \mathcal{R}\sigma} \mu G \right) \sigma^{-1}.$$

Then $P\sigma = \bigcup_{\mu \in \mathcal{R}\sigma} \mu G$ is a union of distinct left cosets of G, and is a $\text{PSCA}(n, k, \lambda)$ by Corollary 4.3. $\qquad \square$

Consider instead searching for a perfect sequence covering array as a union $\bigcup_{\sigma \in \mathcal{R}} G\sigma$ of distinct right cosets of G. Theorem 4.4 no longer holds when we replace left cosets by right cosets, because it relies on Corollary 4.3 which fails when the right action of σ on P is replaced by the left action. Therefore an exhaustive search must consider all nontrivial subgroups of S_n. However, there is now a worthwhile simplification in that we may assume the subgroup G itself is one of the right cosets contained in the perfect sequence covering array $\bigcup_{\sigma \in \mathcal{R}} G\sigma$: let $\mu \in \mathcal{R}$, and note from Corollary 4.3 that $\left(\bigcup_{\sigma \in \mathcal{R}} G\sigma \right) \mu^{-1} = \bigcup_{\sigma \in \mathcal{R}} G(\sigma \mu^{-1})$ is a perfect sequence covering array, and that it contains the right coset $G(\mu \mu^{-1}) = G$.

4.3 Algorithm description

We can now describe an algorithm for finding all possible examples of a $\text{PSCA}(n, k, \lambda)$ that is a union of $\frac{k!\lambda}{|G|}$ distinct cosets of a prescribed nontrivial subgroup G of S_n. The algorithm attempts to build a $\text{PSCA}(n, k, \lambda)$ one coset of G at a time without covering any k-subsequence more than

λ times. If it terminates without output then there is no PSCA(n, k, λ) that is a union of distinct cosets of G.

The algorithm follows the same principles as Algorithm 1, but operates on the following compressed version of the (n, k)-incidence matrix in which the $|G|$ repetition vectors indexed by the sequences of a coset are replaced by their sum, because the entire coset is either contained or not contained in the perfect sequence covering array. The advantage of this approach is that prescribing the subgroup G reduces the maximum search depth by a factor of $|G|$. This gives a dramatic speed improvement, even for $|G| = 2$.

Definition 4.5. *Let G be a subgroup of S_n, and let $\mathcal{R}$ be a complete set of left (or right) coset representatives for G in S_n. The* left (or right) (n, k)-incidence matrix for G *is the $\frac{n!}{|G|} \times \frac{n!}{(n-k)!}$ array over $\{0, 1, \ldots, |G|\}$ whose rows are indexed by $\mathcal{R}$, whose columns are indexed by the elements of $S_{n,k}$, and whose (x, y) entry is the number of times the coset xG (or Gx) covers $y \in S_{n,k}$.*

Pseudocode implementing the search procedure is given in Algorithm 2 (which reduces to Algorithm 1 if we take G to be the trivial subgroup). Algorithm 1 keeps track of a shrinking set L of rows and a shrinking set Y of columns of the (n, k)-incidence matrix; Algorithm 2 does the same, but in relation to the (n, k)-incidence matrix for G. This requires the following modifications to Algorithm 1, because the entries $a_{x,y}$ of the latter matrix are no longer restricted to $\{0, 1\}$:

Line 9. The target size of $|X|$ is the number $\frac{k!\lambda}{|G|}$ of cosets, rather than the number $k!\lambda$ of n-sequences.

Line 18. The entries of the (n, k)-incidence matrix for G lie in $\{0, 1, \ldots, |G|\}$ rather than $\{0, 1\}$, so the test $a_{x,y^*} = 1$ is replaced by $a_{x,y^*} \neq 0$.

Line 6. The set L of candidates for enlarging X is more constrained, because inclusion of a row containing an entry $a_{\ell,y} > 1$ could cause the sum $a_{\ell,y} + r_y$ to exceed λ. We therefore impose that $a_{\ell,y} + r_y \leq \lambda$ over all values of y (not just those lying outside Y). We can rewrite L as specified in Line 6 as

$$\{\ell \notin X : a_{\ell,y} = 0 \text{ for all } y \notin Y\} \cap \{\ell \notin X : a_{\ell,y} + r_y \leq \lambda \text{ for all } y \in Y\},$$

to see that the set L in Algorithm 2 is a subset of the set L in Algorithm 1.

Line 23. The set B used to update L at Line 24 follows from the definition of L at Line 6 and the updated value of R at Line 22.

The procedure SEARCH of Algorithm 2 searches for a PSCA(n, k, λ) comprising a union of $\frac{k!\lambda}{|G|}$ distinct cosets of a single prescribed nontrivial subgroup G of S_n, where $|G|$ divides $k!\lambda$. In the case of left cosets, it is sufficient to search over a single representative G of each conjugacy class of nontrivial subgroups of S_n (see Section 4.2). For each such subgroup G, we may exclude from the initial candidate set L each coset representative that indexes a row of the left (n, k)-incidence matrix for G containing some entry greater than λ, and initialize M accordingly. To determine whether a PSCA(n, k, λ) exists using this procedure, it is most efficient to examine the set of suitable subgroups G for Algorithm 2 in decreasing order of $|G|$, because a larger value of $|G|$ gives a more dramatic speed improvement over Algorithm 1. We do not take G to the trivial group, because then Algorithm 2 reduces to Algorithm 1 and, even if a perfect sequence covering array is found, no additional structure is identified.

Algorithm 2 Tree search for union-of-cosets $\mathrm{PSCA}(n,k,\lambda)$ by backtracking

---

Input: PSCA parameters (n,k,λ) and $A = \big(a_{x,y} : x \in \mathcal{R},\ y \in S_{n,k}\big)$ = left or right (n,k)-incidence matrix for a subgroup G of S_n and complete set of coset representatives $\mathcal{R}$, where $|G|$ divides $k!\lambda$.

1: **procedure** SEARCH(X, Y, R, L, M)
2:          % X = coset representatives of partial $\mathrm{PSCA}(n,k,\lambda)$ P
3:          % write $(r_y) = \mathrm{rv}_k(X) = (\sum_{x \in X} a_{x,y})$ = sum of rows of A indexed by X
4:          % $Y = \{y \in S_{n,k} : r_y < \lambda\}$ = k-subsequences not yet covered λ times by P
5:          % $R = (r_y : y \in Y)$ = entries of $\mathrm{rv}_k(X)$ indexed by Y
6:          % $L = \{\ell \notin X : a_{\ell,y} + r_y \le \lambda \text{ for all } y\}$ = candidates for enlarging X
7:          % write $(m_y) = \mathrm{rv}_k(L) = (\sum_{\ell \in L} a_{\ell,y})$ = sum of rows of A indexed by L
8:          % $M = (m_y : y \in Y)$ = entries of $\mathrm{rv}_k(L)$ indexed by Y
9:      **if** $|X| = \frac{k!\lambda}{|G|}$ **then**
10:          record X as a set of coset representatives for a $\mathrm{PSCA}(n,k,\lambda)$.
11:          **return**
12:      **end if**
13:      **if** $r_y + m_y < \lambda$ for some $y \in Y$ **then**
14:          **return**   % terminate branch early
15:      **end if**
16:      let Y' be the set of $y \in Y$ for which r_y attains its maximum value.
17:      choose an arbitrary $y = y^* \in Y'$ for which m_y attains its minimum value.
18:      $L(y^*) := \{x \in L : a_{x,y^*} \ne 0\}$
19:      **for each** $x \in L(y^*)$ **do**
20:          $X_{\mathrm{new}} := X \cup \{x\}$
21:          $Y_{\mathrm{new}} := Y \setminus \{y : r_y + a_{x,y} = \lambda\}$
22:          $R_{\mathrm{new}} := (r_y + a_{x,y} : y \in Y_{\mathrm{new}})$
23:          $B := \{x\} \cup \{\ell \in L : a_{\ell,y} + r_y + a_{x,y} > \lambda \text{ for some } y \in Y\}$
24:          $L_{\mathrm{new}} := L \setminus B$
25:          $M_{\mathrm{new}} := (m_y - \sum_{\ell \in B} a_{\ell,y} : y \in Y_{\mathrm{new}})$
26:          SEARCH$(X_{\mathrm{new}}, Y_{\mathrm{new}}, R_{\mathrm{new}}, L_{\mathrm{new}}, M_{\mathrm{new}})$
27:      **end for**
28: **end procedure**

In the case of right cosets, we must instead search over all subgroups G of S_n, although we may take the initial set X to be $\{1_G\}$ and initialize Y, R, L, M accordingly (see Section 4.2). We need examine the subgroup G only if the row indexed by 1_G of the right (n,k)-incidence matrix for G has all entries at most λ; if so, then the same holds for all other rows by Lemma 4.2.

Table 2 illustrates the differences in CPU time required to search for all possible examples of a $\mathrm{PSCA}(n,k,\lambda)$ without repeated elements using Algorithm 1, and using distinct left or right cosets with Algorithm 2 over all subgroups G of a specified order. These search times refer to a C implementation on a single core of an Intel Xeon E5-2680, excluding the precalculation time for incidence matrices in GAP [7] for Algorithm 2 (which carries negligible overhead for larger searches). Comparison of timings for $(n,k,\lambda) = (6,3,2)$ shows that Algorithm 2, when it succeeds, is significantly faster than Algorithm 1 even when $|G| = 2$. Comparison of timings for $(n,k,\lambda) = (7,3,2)$ shows that Algorithm 2, when it succeeds for a larger $|G|$, is dramatically faster than with a smaller $|G|$. Comparison of timings for $(n,k,\lambda) = (7,4,2)$ shows that Algorithm 2 using left cosets can succeed when Algorithm 2 using right cosets fails for the same $|G|$, and in that case a successful search using left cosets is significantly faster than using right cosets. The comparison times for the parameter sets $(7,5,1)$ and $(8,6,1)$ taken from [16] refer to exhaustive searches carried out in 1999 on an Ultra SPARCstation 5 to establish Proposition 2.6.

Table 2: CPU time to search for all possible examples of a $\mathrm{PSCA}(n,k,\lambda)$

method	cosets	(n,k,λ)	$|G|$	examples?	CPU time	1999 search [16]
Algorithm 1		$(7,5,1)$		no	0.1 seconds	5 minutes
Algorithm 1		$(8,6,1)$		no	100 minutes	100 hours
Algorithm 1		$(6,3,2)$		yes	3 seconds	
Algorithm 1		$(7,3,2)$		yes	40 minutes	
Algorithm 2	left	$(6,3,2)$	2	yes	0.4 seconds	
Algorithm 2	left	$(7,3,2)$	2	yes	50 seconds	
Algorithm 2	left	$(7,3,2)$	6	yes	0.05 seconds	
Algorithm 2	left	$(7,4,2)$	6	yes	3 seconds	
Algorithm 2	right	$(7,4,2)$	6	no	10 seconds	
Algorithm 2	right	$(7,4,2)$	2	yes	210 minutes	

4.4 New values and bounds for $g(n,k)$

Algorithm 2 finds the following examples of a $\mathrm{PSCA}(7,4,2)$, a $\mathrm{PSCA}(7,5,4)$, a $\mathrm{PSCA}(8,3,3)$, and a $\mathrm{PSCA}(9,3,4)$ as a union of left cosets. By reference to Table 1, this gives the new results $g(7,4) = 2$ and $g(7,5) \in \{2,3,4\}$ and $g(8,3) \in \{2,3\}$ and $g(9,3) \in \{2,3,4\}$.

Proposition 4.6.

(i) *The following* 48 *sequences form a* $\mathrm{PSCA}(7,4,2)$ *as a union of* 8 *left cosets of the subgroup*

$G = \langle 4735621 \rangle \cong C_6$ *of* S_7*:*

left coset		sequences				
$1254736\,G$	1254736	2517634	4765132	5126437	6472531	7641235
$1347256\,G$	1347256	2376514	4351762	5364127	6325471	7312645
$1352746\,G$	1352746	2315674	4367152	5321467	6374521	7346215
$1362745\,G$	1362745	2345671	4327156	5371462	6314527	7356214
$1365472\,G$	1365472	2341765	4326517	5372641	6317254	7354126
$1462537\,G$	1462537	2745136	4527631	5671234	6214735	7156432
$1745236\,G$	1745236	2671534	4156732	5462137	6527431	7214635
$1765234\,G$	1765234	2641537	4126735	5472136	6517432	7254631

Therefore $g(7,4) = 2$.

(ii) *The following* 480 *sequences (listed only by reference to cosets) form a* PSCA$(7,5,4)$ *as* 20 *left cosets of the subgroup* $G = \langle 7261354,\ 4216537 \rangle \cong S_4$ *of* S_7*:*

$$
\begin{array}{ccccc}
1234567\,G & 1236574\,G & 1324576\,G & 1324675\,G & 1325476\,G \\
1325674\,G & 1347265\,G & 1354267\,G & 1357642\,G & 1364275\,G \\
1364572\,G & 1367542\,G & 1374256\,G & 1375624\,G & 1623574\,G \\
1635247\,G & 1637425\,G & 2137465\,G & 2137654\,G & 2163457\,G
\end{array}
$$

Therefore $g(7,5) \in \{2,3,4\}$.

(iii) *The following* 18 *sequences form a* PSCA$(8,3,3)$ *as a union of* 9 *left cosets of the subgroup* $G = \langle 85672341 \rangle \cong C_2$ *of* S_8*:*

left coset		sequences	
$12345678\,G$	12345678	85672341
$15468237\,G$	15468237	82731564
$17624385\,G$	17624385	84357612
$27561843\,G$	27561843	54238176
$28461573\,G$	28461573	51738246
$31864275\,G$	31864275	68137542
$32654187\,G$	32654187	65327814
$37461528\,G$	37461528	64738251
$47218653\,G$	47218653	74581326

Therefore $g(8,3) \in \{2,3\}$.

(iv) *The following* 24 *sequences form a* PSCA$(9,3,4)$ *as a union of* 4 *left cosets of the subgroup* $G = \langle 768241593 \rangle \cong C_6$ *of* S_9*:*

left coset		sequences				
$123456897\,G$	123456897	258714936	473165892	519627384	649572381	768241935
$154372968\,G$	154372968	217865349	461327958	526941873	675914823	742856319
$198426537\,G$	198426537	239754186	498175632	583617294	683542791	739261485
$318697425\,G$	318697425	348592176	829436751	879135264	953784612	963281547

Therefore $g(9,3) \in \{2,3,4\}$.

Table 3: Updated table of $g(n,k)$ for small n and k, with new values or bounds in bold. (s1): Lemma 2.1. (s2): Theorem 2.4. (s3): Example 2.14. (s4): Proposition 2.5. (s5): Proposition 2.6. (s7): Lemma 2.3 (i). (s8): Proposition 3.1. (s9): Proposition 4.6.

n \ k	2	3	4	5	6	7
2	1 (s1)					
3	1 (s1)	1 (s1)				
4	1 (s1)	1 (s2)	1 (s1)			
5	1 (s1)	2 (s3)	1 (s2)	1 (s1)		
6	1 (s1)	**2** (s8)	1 (s4)	1 (s2)	1 (s1)	
7	1 (s1)	**2** (s8)	**2** (s9)	**2** or **3** or **4** (s9)	1 (s2)	1 (s1)
8	1 (s1)	**2** or **3** (s9)	≥ 2 (s7)	≥ 2 (s7)	≥ 2 (s5)	1 (s2)
9	1 (s1)	**2** or **3** or **4** (s9)	≥ 2 (s7)	≥ 2 (s7)	≥ 2 (s7)	?

Table 3 is an updated table of values of $g(n,k)$ for small n and k, taking account the new values and bounds from Proposition 3.1 and Proposition 4.6.

A search for a PSCA$(8,3,2)$ using left cosets in Algorithm 2 completed without output: if a PSCA$(8,3,2)$ exists, then it does not occur as a union of distinct left cosets of a nontrivial subgroup of S_8. A search for a PSCA$(8,4,2)$ and a PSCA$(9,3,2)$ using left cosets in Algorithm 2 completed without output for a representative of each conjugacy class of subgroups of order greater than 2: if a PSCA$(8,4,2)$ or PSCA$(9,3,2)$ occurs as a union of distinct left cosets of a nontrivial subgroup of S_8 or S_9, respectively, then the subgroup has order 2.

Although we obtained some positive results using Algorithm 2 for right cosets, the structure uncovered was less rich than for left cosets. In particular, in each case tested we found that if a PSCA(n,k,λ) occurs as a union of distinct right cosets of a nontrivial subgroup G of S_n, then a PSCA(n,k,λ) also occurs as a union of distinct left cosets of a subgroup G' of S_n, where G' is isomorphic to G. There were also several instances when the largest $|G|$ was smaller than the largest $|G'|$, in which case the search using right cosets was signficantly slower (see Section 4.3). We therefore did not attempt to carry out some of the larger searches for right cosets.

4.5 New parameter sets (n,k,λ) for a PSCA(n,k,λ)

Charlie Colbourn (personal communication, Sept. 2021) posed the following question.

Question 4.7. *Does the existence of a* PSCA(n,k,λ) *imply the existence of a* PSCA$(n,k,\lambda+1)$?

This prompts the following observation as a direct consequence of Lemma 2.2.

Lemma 4.8. *Let $k \leq n$ and let $g = g(n,k)$. Suppose there exists a* PSCA(n,k,λ) *for each*

$$\lambda \in \{g, g+1, \ldots, 2g-1\}.$$

Then there exists a PSCA(n,k,λ) *if and only if $\lambda \geq g$, and the answer to Question 4.7 is yes for the parameter pair (n,k).*

Algorithm 2 finds the following examples of a PSCA$(5, 3, 3)$, a PSCA$(6, 3, 3)$, a PSCA$(7, 3, 3)$, a PSCA$(7, 4, 3)$, a PSCA$(8, 3, 4)$, and a PSCA$(8, 3, 5)$ as a union of left cosets. Each of these parameter sets is new. Combination of these examples with the results in Table 3 shows that the answer to Question 4.7 is yes for each of the parameter pairs

$$(n, k) \in \{(5, 3), (6, 3), (7, 3), (7, 4), (8, 3)\},$$

regardless of whether $g(8, 3) = 2$ or $g(8, 3) = 3$, and application of Lemma 4.8 gives Corollary 4.10. We do not currently know of a parameter set (n, k, λ) for which the answer to Question 4.7 is no.

Proposition 4.9.

(*i*) *The following* 18 *sequences form a* PSCA$(5, 3, 3)$ *as a union of* 9 *left cosets of the subgroup* $G = \langle 43215 \rangle \cong C_2$ *of* S_5:

left coset	sequences	
$12345\,G$	12345	43215
$12435\,G$	12435	43125
$13452\,G$	13452	42153
$15324\,G$	15324	45231
$21543\,G$	21543	34512
$23514\,G$	23514	32541
$24513\,G$	24513	31542
$51423\,G$	51423	54132
$52314\,G$	52314	53241

(*ii*) *The following* 18 *sequences form a* PSCA$(6, 3, 3)$ *as a union of* 3 *left cosets of the subgroup* $G = \langle 634215, 456123 \rangle \cong S_3$ *of* S_6:

left coset	sequences					
$123456\,G$	123456	215634	361542	456123	542361	634215
$134265\,G$	134265	256143	315624	461532	523416	642351
$162435\,G$	162435	241653	326514	435162	514326	653241

(*iii*) *The following* 18 *sequences form a* PSCA$(7, 3, 3)$ *as a union of* 9 *left cosets of the subgroup* $G = \langle 3412765 \rangle \cong C_2$ *of* S_7:

left coset	sequences	
$1234567\,G$	1234567	3412765
$1253764\,G$	1253764	3471562
$1643572\,G$	1643572	3621754
$5147362\,G$	5147362	7325164
$5241673\,G$	5241673	7423651
$5276341\,G$	5276341	7456123
$5432617\,G$	5432617	7214635
$6175324\,G$	6175324	6357142
$6245371\,G$	6245371	6427153

(*iv*) *The following* 72 *sequences form a* PSCA(7, 4, 3) *as a union of* 18 *left cosets of the subgroup* $G = \langle 1576342 \rangle \cong C_4$ *of* S_7:

left coset	sequences			
1234567 G	1234567	1324765	1576342	1756243
1256437 G	1256437	1376425	1534672	1724653
2136754 G	2136754	3126574	5174236	7154326
2164573 G	2164573	3164752	5146327	7146235
2315476 G	2315476	3217456	5713624	7512634
2436751 G	2436751	3426571	5674231	7654321
2537614 G	2537614	3725614	5372416	7253416
2574613 G	2574613	3754612	5326417	7236415
2716354 G	2716354	3516274	5214736	7314526
2741536 G	2741536	3541726	5261374	7361254
2764351 G	2764351	3564271	5246731	7346521
4127356 G	4127356	4135276	6152734	6173524
4162753 G	4162753	4163572	6145237	6147325
4251673 G	4251673	4371652	6531427	6721435
4263175 G	4263175	4362157	6547123	6745132
4523716 G	4523716	4732516	6275314	6357214
4536172 G	4536172	4726153	6254137	6374125
4576312 G	4576312	4756213	6234517	6324715

(*v*) *The following* 24 *sequences form a* PSCA(8, 3, 4) *as the single left coset* 12354678 G *of the subgroup* $G = \langle 67142358, 46572381 \rangle \cong \mathrm{SL}(2, 3)$ *of* S_8:

12354678	16543872	18435276	24851637	26518734	27185436	31746825	35674128
38467521	41752683	43275186	46527381	51738264	52387461	54873162	63241857
67124358	68412753	73268514	74826315	75682413	82316745	85631247	87163542

(*vi*) *The following* 30 *sequences form a* PSCA(8, 3, 5) *as a union of* 5 *left cosets of the subgroup* $G = \langle 65872143, 45712836 \rangle \cong S_3$ *of* S_8:

left coset	sequences					
12345678 G	12345678	35182764	45712836	65872143	72465381	82635417
12485736 G	12485736	35842617	45162378	65732481	72615843	82375164
17384625 G	17384625	36148752	43761852	64837152	78416325	81673425
17564823 G	17564823	36278451	43281657	64217358	78536124	81543726
21685347 G	21685347	27315468	28475631	53742186	54862713	56132874

Corollary 4.10.

(*i*) *For each* $(n, k) \in \{(5, 3), (6, 3), (7, 3), (7, 4)\}$, *there exists a* PSCA$(n, k, \lambda)$ *if and only if* $\lambda \geq 2$.

(*ii*) *There exists a* PSCA$(8, 3, \lambda)$ *if and only if* $\lambda \geq g(8, 3)$, *and* $g(8, 3) \in \{2, 3\}$.

4.6 Examples of a group-based $\mathrm{PSCA}(n, n-1, 1)$

Algorithm 2 finds the following examples of a $\mathrm{PSCA}(n, n-1, 1)$ as a union of $\frac{(n-1)!}{|G|}$ left cosets of a nontrivial subgroup G of S_n, for each $n \in \{4, 5, 6, 7\}$. This suggests the possibility of a group-based proof of Theorem 2.4, as an alternative to Levenshtein's coding-theoretic proof.

Proposition 4.11. *The following sets of $(n-1)!$ sequences (listed only by reference to cosets) form a $\mathrm{PSCA}(n, n-1, 1)$ as a union of $\frac{(n-1)!}{|G|}$ left cosets of the subgroup G of S_n.*

(i) $n = 4$ *and the subgroup* $G = \langle 3412 \rangle \cong C_2$ *of* S_4:

$$1234\,G \quad 1432\,G \quad 2413\,G$$

(ii) $n = 5$ *and the subgroup* $G = \langle 34125, 43215 \rangle \cong C_2 \times C_2$ *of* S_5:

$$12345\,G \quad 13254\,G \quad 14253\,G \quad 15243\,G \quad 15342\,G \quad 51432\,G$$

(iii) $n = 6$ *and the subgroup* $G = \langle 125634, 346521, 345612 \rangle \cong S_4$ *of* S_6:

$$123456\,G \quad 132546\,G \quad 132645\,G \quad 135642\,G \quad 136524\,G$$

(iv) $n = 7$ *and the subgroup* $G = \langle 7235461, 1756432 \rangle \cong S_4$ *of* S_7:

$$
\begin{array}{llllll}
1234657\,G & 1235476\,G & 1237456\,G & 1273564\,G & 1324576\,G & 1325467\,G \\
1326475\,G & 1326574\,G & 1342756\,G & 1345267\,G & 1352764\,G & 1356427\,G \\
1357246\,G & 3124756\,G & 3125746\,G & 3126745\,G & 3127654\,G & 3145627\,G \\
3154267\,G & 3154762\,G & 3156742\,G & 3412657\,G & 3415276\,G & 3421567\,G \\
3425176\,G & 3426715\,G & 3427651\,G & 3456172\,G & 3457126\,G & 3457621\,G
\end{array}
$$

5 Open problems

We have established new values and bounds for $g(n, k)$, as shown in Table 3. We have established the following new parameter sets for a $\mathrm{PSCA}(n, k, \lambda)$: $(5, 3, 3)$, $(6, 3, 3)$, $(7, 3, 3)$, $(7, 4, 3)$, $(8, 3, 4)$, and $(8, 3, 5)$ (see Section 4.5). We have given an example of a $\mathrm{PSCA}(n, n-1, 1)$ as a union of left cosets of a nontrivial subgroup of S_n for $n \in \{4, 5, 6, 7\}$ (see Section 4.6). Examples of a $\mathrm{PSCA}(n, k, \lambda)$ formed as a union of distinct left cosets of a nontrivial subgroup appear to be widespread, and prescribing this structure brings into reach various searches that would otherwise be intractable.

  We propose several open problems arising from our results.

(i) Determine further exact values and bounds for $g(n, k)$.

(ii) Find a $\mathrm{PSCA}(n, k, \lambda)$ for new parameter sets (n, k, λ).

(iii) Find a group-based construction for a $\mathrm{PSCA}(n, n-1, 1)$ for each $n \geq 2$.

(iv) Is there a parameter set (n, k, λ) for which a $\mathrm{PSCA}(n, k, \lambda)$ exists but there is no example that is a union of left cosets of a nontrivial subgroup of S_n?

(v) Resolve Question 4.7 by determining whether the existence of a $\mathrm{PSCA}(n, k, \lambda)$ implies the existence of a $\mathrm{PSCA}(n, k, \lambda + 1)$.

(*vi*) The recursive search methods presented here appear to encounter memory constraints when attempting to settle the smallest open case of Conjecture 2.7, namely whether $g(9,7) > 1$. Are there theoretical techniques or improved search methods for handling this case?

(*vii*) Find more combinatorial nonexistence results for perfect sequence covering arrays.

Comments

The authors thank Aidan Gentle and Ian Wanless for kindly sharing a preprint of the paper [8], which describes how they used different methods from ours to computationally determine existence and nonexistence results for perfect sequence covering arrays that complement ours. They independently showed that $g(6,3) = g(7,3) = 2$. They also recovered some of the results originally reported in [17], in particular that $g(7,4) = 2$. They further established two results that our methods were not able to find: $g(8,3) > 2$ and $g(8,4) > 2$, the first of which combines with the PSCA$(8,3,3)$ of Proposition 4.6 to show that $g(8,3) = 3$. Conversely, our methods found results that were not obtained in [8], including that $g(7,5) \leq 4$ and $g(9,3) \leq 4$. The paper [8] also gives various bounds on the value of $g(n,k)$ that arise from examples of perfect sequence covering arrays comprising a complete subgroup of S_n.

The authors gratefully acknowledge helpful discussions with Karen Meagher and Charlie Colbourn.

References

[1] B. Apilli. Fault-based combinatorial testing of web services. In *Proc. 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, pages 731–732, 2009.

[2] F. E. Bennett and A. Mahmoodi. Directed designs. In C. J. Colbourn and J. H. Dinitz, editors, *Handbook of Combinatorial Designs*, pages 441–444. Chapman and Hall/CRC, Boca Raton, 2nd edition, 2007.

[3] Y. M. Chee, C. J. Colbourn, D. Horsley, and J. Zhou. Sequence covering arrays. *SIAM J. Discrete Math.*, 27(4):1844–1861, 2013.

[4] C. J. Colbourn. Covering arrays and hash families. In D. Crnković and V. Tonchev, editors, *Information Security, Coding Theory and Related Combinatorics*, pages 99–135, Amsterdam, 2011. IOS Press.

[5] J. E. Dawson, J. Seberry, and D. B. Skillicorn. The directed packing numbers DD(t, v, v), $t \geq 4$. *Combinatorica*, 4(2–3):121–130, 1984.

[6] Z. Füredi. Scrambling permutations and entropy of hypergraphs. *Random Struct. Algor.*, 8(2):97–104, 1996.

[7] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.11.0*, 2020.

[8] A.R. Gentle and I.M. Wanless. On perfect sequence covering arrays. *Ann. Comb.*, 2022. To appear. arXiv:2202.01960.

[9] S. Huang, M. B. Cohen, and A. M. Memon. Repairing GUI test suites using a genetic algorithm. In *2010 Third International Conference on Software Testing, Verification and Validation*, pages 245–254, 2010.

[10] Y. Ishigami. An extremal problem of d permutations containing every permutation of every t elements. *Discrete Math.*, 159(1–3):279–283, 1996.

[11] A. Klein. On perfect deletion-correcting codes. *J. Comb. Des.*, 12(1):72–77, 2004.

[12] D. R. Kuhn, J. M. Higdon, J. F. Lawrence, R. N. Kacker, and Y. Lei. Combinatorial methods for event sequence testing. In *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*, pages 601–609, 2012.

[13] V. I. Levenshtein. Perfect deletion-correcting codes as combinatorial designs. *II International Workshop Algebraic and Combinatorial Coding Theory*, pages 137–140, 1990.

[14] V. I. Levenshtein. Perfect codes in the metric of deletions and insertions. *Diskr. Mat.*, 3(1):3–20, 1991. English translation in: *Discrete Math. Appl.*, 2(3):241–258, 1992.

[15] R. Mathon. Searching for spreads and packings. In J. W. P. Hirschfeld, S. S. Magliveras, and M. J. de Resmini, editors, *Geometry, Combinatorial Designs and Related Structures*, Proc. First Pythagorean Conference, pages 161–176. Cambridge University Press, 1997.

[16] R. Mathon and T. van Trung. Directed t-packings and directed t-Steiner systems. *Des. Codes Cryptog.*, 18:187–198, 1999.

[17] J. Na. Perfect sequence covering arrays. Master's thesis, Simon Fraser University, 2021.

[18] J. Radhakrishnan. A note on scrambling permutations. *Random Struct. Algor.*, 22(4):435–439, 2003.

[19] J. Spencer. Minimal scrambling sets of simple orders. *Acta Math. Hungarica.*, 22(3–4):349–353, 1971.

[20] W. Wang, Y. Lei, S. Sampath, R. Kacker, R. Kuhn, and J. Lawrence. A combinatorial approach to building navigation graphs for dynamic web applications. In *2009 IEEE International Conference on Software Maintenance*, pages 211–220, 2009.

[21] W. Wang, S. Sampath, Y. Lei, and R. Kacker. An interaction-based test sequence generation approach for testing web applications. In *2008 11th IEEE High Assurance Systems Engineering Symposium*, pages 209–218, 2008.

[22] R. M. Wilson. A diagonal form for the incidence matrices of t-subsets vs. k-subsets. *Europ. J. Combinatorics*, 11:609–615, 1990.

[23] X. Yuan, M. B. Cohen, and A. M. Memon. GUI interaction testing: Incorporating event context. *IEEE Trans. Softw. Eng.*, 37(4):559–574, 2011.

[24] X. Yuan and A. M. Memon. Generating event sequence-based test cases using GUI runtime state feedback. *IEEE Trans. Softw. Eng.*, 36(1):81–95, 2010.

[25] R. Yuster. Perfect sequence covering arrays. *Des. Codes Cryptog.*, 88:585–593, 2020.