A group-based structure for perfect sequence covering arrays

Jingzhou Na · Jonathan Jedwab · Shuxing Li

Abstract

An \((n, k)\)-perfect sequence covering array with multiplicity \(\lambda\), denoted \(PSCA(n, k, \lambda)\), is a multiset whose elements are permutations of the sequence \((1, 2, \ldots, n)\) and which collectively contain each ordered length \(k\) subsequence exactly \(\lambda\) times. The primary objective is to determine for each pair \((n, k)\) the smallest value of \(\lambda\), denoted \(g(n, k)\), for which a \(PSCA(n, k, \lambda)\) exists; and more generally, the complete set of values \(\lambda\) for which a \(PSCA(n, k, \lambda)\) 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 \(PSCA(n, k, \lambda)\) if and only if \(\lambda \geq 2\); and that there exists a \(PSCA(8, 3, \lambda)\) if and only if \(\lambda \geq g(8, 3)\).

Keywords Combinatorial design theory · Perfect sequence covering array · Group theory · Search algorithm

Mathematics Subject Classification 05B40 · 20B25

Communicated by C. J. Colbourn.

J. Na is supported by a PhD Scholarship from the China Scholarship Council. J. Jedwab is supported by an NSERC Discovery Grant. S. Li is supported by a PIMS Postdoctoral Fellowship.

Jonathan Jedwab
jed@sfu.ca

Jingzhou Na
jingzhou_na@sfu.ca

Shuxing Li
shuxing_li@sfu.ca

1 Department of Mathematics, Simon Fraser University, 8888 University Drive, Burnaby, BC V5A 1S6, Canada
1 Introduction

Throughout, let $k$ and $n$ be integers satisfying $2 \leq k \leq n$. Let $S_n$ be the set of permutations of $[n] := \{1, 2, \ldots, n\}$, 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 $\lambda$, denoted 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 $\lambda$ 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 $\lambda$ sequences of $P$. For example, the subset

\[ \{12345, 13254, 14523, 24315, 25413, 34512, 35214, 42513, 43215, 52314, 53412\} \]

of $S_5$ is a PSCA$(5, 3, 2)$. If $P$ is a PSCA$(n, k, 1)$, then $P$ is a set (not a multiset). The size of the multiset of $k$-subsequences covered by a PSCA$(n, k, \lambda)$ 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$(n, k, \lambda)$ is equivalent to a $k$-$(n, n, \lambda)$ directed design $([n], P)$ [2, Sect. VI.20]. If a PSCA$(n, k, \lambda)$ exists, then it achieves the largest possible size of a $k$-$(n, n, \lambda)$ directed packing [5, 15]. Replacing “exactly $\lambda$ 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 $[n]$ [6, Sect. 5], [9, 18]. 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 [1, 8, 11, 20, 21, 23, 24]. 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 $\lambda$ for which a PSCA$(n, k, \lambda)$ 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 $\lambda$ for which there exists a PSCA$(n, k, \lambda)$. 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$(n, k, \lambda)$ 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 $(n, k, \lambda)$ for which a PSCA$(n, k, \lambda)$ exists, providing evidence for a positive answer to a question posed by Charlie Colbourn (personal communication, Sept. 2021): does the existence of a PSCA$(n, k, \lambda)$ 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 [16], 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 $\{12 \ldots n, n \ldots 21\}$ 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 PSCA$(n, k, \lambda)$ and a PSCA$(n, k, \mu)$. Then their multiset union is a 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 PSCA$(n, k, \lambda)$ to give a PSCA$(n - 1, k, \lambda)$. For (ii), regard a PSCA$(n, k, \lambda)$ as a 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 [13, Thm. 3.1]) The set $S_n$ can be partitioned into $n$ sets of sequences, each of which is a PSCA$(n, n - 1, 1)$. Therefore $g(n, n - 1) = 1$.

Levenshtein [12, 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 [15, p. 191]) There exists a PSCA$(6, 4, 1)$.

Mathon and van Trung showed by hand that $g(5, 3) > 1$ [15, Thm 3.2] (see [16, 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 [15, Sects. 4 and 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 [15, p. 198]) Let $k \notin \{2, 4\}$. Then $g(k + 2, k) > 1$.

More than twenty years after publication of [15], 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 [10, Thm 3.2]) We have \( g(7, 4) > 1 \).

**Proof** Suppose, for a contradiction, that \( P \) is a 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 Theorem 2.10, 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 [6, 9, 17, 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 PSCA\((n^2, 3, 2(n + 1)\lambda)\) from a 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 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 \( \text{rv}_k(X) \), is the sum of the rows of the \((n, k)\)-incidence matrix that are indexed by the elements of \( X \). The vector \( \text{rv}_k(X) \) has length \( \frac{n!}{(n-k)!} \), and its \( j \)th entry is the number of sequences in \( X \) that cover \( k \)-subsequence \( j \). The multiset \( X \) is therefore a PSCA \((n, k, \lambda)\) if and only if each entry of \( \text{rv}_k(X) \) is \( \lambda \).

We can now describe the PSCA \((5, 3, 2)\) found by Yuster. We shall reinterpret this example in Sect. 4.1.

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

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

Then the length 60 vector \( \text{rv}_3(X) \) 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 \( \pi, \sigma \) given by \( \pi\sigma \) represents the action of \( \pi \) followed by \( \sigma \). Then the repetition vector of

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

has the same property as \( \text{rv}_3(X) \), but the positions in \( \text{rv}_3(X\sigma) \) of the 0 and 2 entries are interchanged with those in \( \text{rv}_3(X) \). This ensures that every entry of \( \text{rv}_3(X) + \text{rv}_3(X\sigma) \) is 2, and therefore that \( X \cup X\sigma \) is a PSCA \((5, 3, 2)\). Since \( g(5, 3) \geq 1 \) [15, 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 PSCA \((n, k, \lambda)\)

In this section, we describe a recursive algorithm for finding all possible examples of a PSCA \((n, k, \lambda)\) without repeated elements, for arbitrary \( \lambda \geq 1 \). The algorithm is a tree search that attempts to build a PSCA \((n, k, \lambda)\) one \( n \)-sequence at a time without covering any \( k \)-subsequence more than \( \lambda \) times, backtracking when this is not possible. Although we have
Table 1: Previously known exact values of $g(n,k)$ for small $n$ and $k$, and their sources

| $n$ | 2     | 3     | 4     | 5     | 6     | 7     |
|-----|-------|-------|-------|-------|-------|-------|
|     | 1 (s1)| 1 (s1)| 1 (s1)| 1 (s1)| 1 (s1)| 1 (s1)|
| 3   | 1 (s1)| 1 (s2)| 1 (s1)|       |       |       |
| 4   | 1 (s1)| 2 (s3)| 1 (s2)| 1 (s1)|       |       |
| 5   | 1 (s1)| 2 (s7)| 1 (s4)| 1 (s2)| 1 (s1)|       |
| 6   | 1 (s1)| 2 (s7)| 1 (s4)| 1 (s2)| 1 (s1)|       |
| 7   | 1 (s1)| 2 (s7)| 1 (s4)| 1 (s2)| 1 (s1)|       |
| 8   | 1 (s1)| 2 (s7)| 1 (s4)| 1 (s2)| 1 (s1)|       |
| 9   | 1 (s1)| 2 (s7)| 1 (s4)| 1 (s2)| 1 (s1)|       |

(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

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 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 PSCA $(n,k,1)$ exists and so $g(n,k) > 1$.

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 $(n,k,\lambda)$ by finding a $(k!\lambda)$-subset $X$ of $S_n$ for which each entry of $rv_k(X)$ is $\lambda$, 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

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

The algorithm succeeds in finding a PSCA $(n,k,\lambda)$ 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 $\lambda$ 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 $\lambda$ times. We also keep track of the repetition vectors $R = rv_k(X) = (\sum_{x \in X} a_{x,y}) = (r_y)$ and $M = rv_k(L) = (\sum_{\ell \in L} a_{\ell,y}) = (m_y)$, although only at the positions $y$ indexed by $Y$. We seek to enlarge $X$ so that the vector $R$ attains the value $\lambda$ 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 $\lambda$ 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 $\lambda$ 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 $\{\ell \in L : a_{i, y} = 1\}$ 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 $|\{\ell \in L : a_{i, y} = 1\}| = m_y$ 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 $\lambda$, the algorithm finds every possible example of a PSCA($n, k, \lambda$) 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 $\lambda$ 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 $\lambda$ 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 $L = \emptyset$ and $M = (\frac{n!}{\lambda k!}, \ldots, \frac{n!}{\lambda k!})$, we obtain every possible PSCA($n, k, \lambda$) 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 $\{12 \cdots n\}$.

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

| Input: | PSCA parameters ($n, k, \lambda$) and $A = (a_{x, y}: x \in S_n, y \in S_{n,k}) = (n, k)$-incidence matrix |
|---|---|
| 1: | procedure SEARCH($X, Y, R, L, M$) |
| 2: | % $X$ = sequences of partial PSCA($n, k, \lambda$) $P$ |
| 3: | % write ($r_y$) = $rv_k(X) = (\sum_{x \in X} a_{x, y})$ = sum of rows of $A$ indexed by $X$ |
| 4: | % $Y = \{y \in S_{n,k}: r_y < \lambda\}$ = $k$-subsequences not yet covered $\lambda$ times by $P$ |
| 5: | % $R = (r_y : y \in Y) = \text{entries of } rv_k(X) \text{ indexed by } Y$ |
| 6: | % $L = \{\ell \notin X: a_{x, y} = 0 \text{ for all } y \notin Y\}$ = candidates for enlarging $X$ |
| 7: | % write ($m_y$) = $rv_k(L) = (\sum_{\ell \in L} a_{\ell, y})$ = sum of rows of $A$ indexed by $L$ |
| 8: | % $M = (m_y : y \in Y) = \text{entries of } rv_k(L) \text{ indexed by } Y$ |
| 9: | if $|X| = k! \lambda$ then |
| 10: | record $X$ as a 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: | $X_{\text{new}} := X \cup \{x\}$ |
| 21: | $Y_{\text{new}} := Y \\{y : r_y + a_{x, y} = \lambda\}$ |
| 22: | $R_{\text{new}} := (r_y + a_{x, y} : y \in Y_{\text{new}})$ |
| 23: | $B := \{x\} \cup \{\ell \in L : a_{\ell, x} = 1 \text{ for at least one } y \in Y \\{y \in Y_{\text{new}}\}$ |
| 24: | $L_{\text{new}} := L \setminus B$ |
| 25: | $M_{\text{new}} := (m_y - \sum_{\ell \in B} a_{\ell, y} : y \in Y_{\text{new}})$ |
| 26: | SEARCH($X_{\text{new}}, Y_{\text{new}}, R_{\text{new}}, L_{\text{new}}, M_{\text{new}}$) |
| 27: | end for |
| 28: | end procedure |
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 [15, p. 163], that in turn relies on an algorithm due to Mathon [14] 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 [14, 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 \((n, k, \lambda)\) 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 [14]. The cases \(\lambda > 1\) do not have a corresponding form in the context of a directed \(t\)-packing and so were not considered in [14]. Algorithm 1 also contains a feature not described in [14] or [15] 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.

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\((6, 3, 2)\)

\[
\begin{align*}
123456 &\quad 154326 &\quad 216543 &\quad 245613 &\quad 354162 &\quad 361452 \\
423165 &\quad 461325 &\quad 516234 &\quad 532614 &\quad 632541 &\quad 645231 &
\end{align*}
\]

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

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

\[
\begin{align*}
1234567 &\quad 1573426 &\quad 3275641 &\quad 3617524 &\quad 4261735 &\quad 4756123 \\
5164327 &\quad 5243176 &\quad 6257314 &\quad 6345721 &\quad 7216453 &\quad 7431625 &
\end{align*}
\]

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

4 PSCA\((n, k, \lambda)\) 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

\[ \text{Springer} \]
A group-based structure for perfect sequence covering arrays

$(n, k, \lambda)$ for which a PSCA($n, k, \lambda$) exists. As in Sect. 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 $\langle \sigma \rangle$).

$$
\begin{array}{c|cccccc}
\text{PSCA}(5, 3, 2) \text{ with } \sigma = 13254 & \langle \sigma \rangle & 43215 \langle \sigma \rangle & 35214 \langle \sigma \rangle & 14523 \langle \sigma \rangle & 25413 \langle \sigma \rangle & 53412 \langle \sigma \rangle \\
X & 12345 & 43215 & 35214 & 14523 & 25413 & 53412 \\
X\sigma & 13254 & 52314 & 24315 & 15432 & 34512 & 42513 \\
\end{array}
$$

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.

$$
\begin{array}{c|cccccc}
\text{PSCA}(6, 3, 2) \text{ with } \sigma = 154326 & \langle \sigma \rangle & 216543 \langle \sigma \rangle & 354162 \langle \sigma \rangle & 461325 \langle \sigma \rangle & 532614 \langle \sigma \rangle & 645231 \langle \sigma \rangle \\
X & 123456 & 216543 & 354162 & 461325 & 532614 & 645231 \\
X\sigma & 154326 & 516234 & 423165 & 361452 & 245613 & 632541 \\
\end{array}
$$

$$
\begin{array}{c|cccccc}
\text{PSCA}(7, 3, 2) \text{ with } \sigma = 4261735 & \langle \sigma \rangle & 1573426 \langle \sigma \rangle & 3275641 \langle \sigma \rangle & 3617524 \langle \sigma \rangle & 5164327 \langle \sigma \rangle & 5243176 \langle \sigma \rangle \\
X & 1234567 & 1573426 & 3275641 & 3617524 & 5164327 & 5243176 \\
X\sigma & 4261735 & 4756123 & 6257314 & 6345721 & 7431625 & 7216453 \\
\end{array}
$$

This motivates us to seek a PSCA($n, k, \lambda$) as the union of $k^\lambda$ distinct cosets of a nontrivial subgroup $G$ of $S_n$, where the parameter $\lambda$ 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 Sect. 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($n, k, \lambda$) 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 \((3617524)\), and as two left cosets of the order 6 subgroup \((3275641, 4261735) \cong S_3\).

We remark that several aspects of our approach can be recognized in Mathon and van Trung’s work [15]. 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 [15, 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 [16, p. 29] for a correction to the listing of \(P_2\) in [15, 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 \(r_{vk}(X)\).

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

**Proof** Under the convention that the permutation composition \(\pi \sigma\) represents the action of \(\pi\) followed by \(\sigma\), the right action of \(\sigma\) 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 PSCA\((n, k, \lambda)\) to another.

**Corollary 4.3** Suppose \(P\) is a PSCA\((n, k, \lambda)\), and let \(\sigma \in S_n\). Then \(P\sigma\) is also a 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 \(|\mathcal{R}| = \frac{k^k}{k!}\)) 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 PSCA\((n, k, \lambda)\) that is a union of distinct left cosets of \(H\). Then there is a PSCA\((n, k, \lambda)\) which is a union of distinct left cosets of \(G\).

**Proof** Let \(P\) be a PSCA\((n, k, \lambda)\) that can be written as

\[
P = \bigcup_{\pi \in \mathcal{R}} \pi H
\]  

(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

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

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

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

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 \( \sigma \) 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 PSCA\((n, k, \lambda)\) that is a union of \( k! \lambda \) distinct cosets of a prescribed nontrivial subgroup \( G \) of \( S_n \). The algorithm attempts to build a PSCA\((n, k, \lambda)\) one coset of \( G \) at a time without covering any \( k \)-subsequence more than \( \lambda \) times. If it terminates without output then there is no PSCA\((n, k, \lambda)\) 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 \(\lambda\). 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.

**Algorithm 2** Tree search for union-of-cosets PSCA\((n, k, \lambda)\) by backtracking

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

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

The procedure SEARCH of Algorithm 2 searches for a PSCA\((n, k, \lambda)\) comprising a union of \(\frac{k!}{|G|}\) distinct cosets of a single prescribed nontrivial subgroup \(G\) of \(S_n\), where \(|G|\) divides \(k!\). 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 Sect. 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 \(\lambda\), and initialize \(M\) accordingly. To determine whether a PSCA\((n, k, \lambda)\) 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.

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 Sect. 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 $\lambda$; 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 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 [19] 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 [15] refer to exhaustive searches carried out in 1999 on an Ultra SPARCstation 5 to establish Proposition 2.6.

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

Algorithm 2 finds the following examples of a PSCA$(7, 4, 2)$, a PSCA$(7, 5, 4)$, a PSCA$(8, 3, 3)$, and a 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 PSCA$(7, 4, 2)$ as a union of 8 left cosets of the subgroup $G = \langle 4735621 \rangle \cong C_6$ of $S_7$: Therefore $g(7, 4) = 2$. 

![Springer](https://example.com/springer-logo)
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 \( S_7 \):

\[
\begin{align*}
1234567 & G \quad 12345678 \quad 85672341 \quad 4216537
\end{align*}
\]

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

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 \( S_8 \): Therefore \( g(8, 3) \in \{2, 3\} \).

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 \( S_9 \): Therefore \( g(9, 3) \in \{2, 3, 4\} \).
A group-based structure for perfect sequence covering arrays

Table 3 Updated table of \( g(n, k) \) for small \( n \) and \( k \), with new values or bounds in bold

| \( n \) | 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) | \( \geq 2 \) (s7) | \( \geq 2 \) (s7) | \( \geq 2 \) (s5) | 1 (s2) |
| 9 | 1 (s1) | 2 or 3 or 4 (s9) | \( \geq 2 \) (s7) | \( \geq 2 \) (s7) | \( \geq 2 \) (s7) | ? |

(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

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 \((n, k, \lambda)\) occurs as a union of distinct right cosets of a nontrivial subgroup \( G \) of \( S_n \), then a PSCA \((n, k, \lambda)\) 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 significantly slower (see Sect. 4.3). We therefore did not attempt to carry out some of the larger searches for right cosets.

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

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

**Question 4.7** *Does the existence of a PSCA \((n, k, \lambda)\) 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 \((n, k, \lambda)\) for each \( \lambda \in \{g, g + 1, \ldots, 2g - 1\} \). Then there exists a PSCA \((n, k, \lambda)\) 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 $(n, k, \lambda)$ 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  |
|------------|------------|
| 12345G     | 12345      | 43215      |
| 12435G     | 12435      | 43125      |
| 13452G     | 13452      | 42153      |
| 15324G     | 15324      | 45231      |
| 21543G     | 21543      | 34512      |
| 23514G     | 23514      | 32541      |
| 24513G     | 24513      | 31542      |
| 51423G     | 51423      | 54132      |
| 52314G     | 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  |
|------------|------------|
| 123456G    | 123456     | 215634     |
| 134265G    | 134265     | 256143     |
| 162435G    | 162435     | 241653     |
| 1361452     | 361542     | 456123     |
| 1354621     | 315624     | 461532     |
| 1643521     | 326514     | 435162     |
| 614325     | 514326     | 53241      |

(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  |
|------------|------------|
| 1234567G   | 1234567    | 3412765    |
| 1253764G   | 1253764    | 3471562    |
| 1653572G   | 1643572    | 3621754    |
| 5147362G   | 5147362    | 7325164    |
| 5241673G   | 5241673    | 7423651    |
| 5276341G   | 5276341    | 7456123    |
| 5432617G   | 5432617    | 7214635    |
| 6175324G   | 6175324    | 6357142    |
| 6245371G   | 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 $S_7$:
A group-based structure for perfect sequence covering arrays 967

Left coset  Sequences
1234567G  1234567  1324765  1576342  1756243  1756243
1256437G  1256437  1376425  1534672  1724653  1724653
2136754G  2136754  3126574  5174236  7154326  7154326
2164573G  2164573  3164752  5146327  7146235  7146235
2315476G  2315476  3217456  5713624  7512634  7512634
2346751G  2346751  3426571  5674231  7654321  7654321
2537614G  2537614  3754612  5372416  7253416  7253416
2574613G  2574613  3754612  5372416  7253416  7253416
2716354G  2716354  3516274  5214736  7314526  7314526
2741536G  2741536  3541726  5261374  7361254  7361254
2764351G  2764351  3564271  5246731  7346521  7346521
4127356G  4127356  4135276  6152734  6173524  6173524
4162753G  4162753  4163572  6145237  6147325  6147325
4523716G  4523716  4371652  6531427  6721435  6721435
4536172G  4536172  4362157  6547123  6745132  6745132
4576312G  4576312  4756213  6245137  6374125  6374125

(v) The following 24 sequences form a PSCA \((8, 3, 4)\) as the single left coset 12354678 \(G\) of the subgroup \(G = \langle 67142358, 46572381 \rangle \cong SL(2, 3)\) 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
12345678G  1234567  35182764  45712836  65872143  72465381  82635417
12485736G  1248573  35842617  45162378  6531427  6145237  6147325
17384625G  17384625  36148752  43761852  6433512  7846325  81673425
17564823G  17564823  36278451  43281657  64271358  78536124  81543726
21685347G  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 PSCA \((n, n - 1, 1)\)

Algorithm 2 finds the following examples of a 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 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 \(S_4\):
\[
1234 \quad G \quad 1432 \quad G \quad 2413 \quad G
\]

\[(ii)\) \(n=5\) and the subgroup \(G = \langle 34125, 43215 \rangle \cong C_2 \times C_2\) of \(S_5\):
\[
12345 \quad G \quad 13254 \quad G \quad 14253 \quad G \quad 15243 \quad G \quad 15342 \quad G
\]

\[(iii)\) \(n=6\) and the subgroup \(G = \langle 125634, 346521, 345612 \rangle \cong S_4\) of \(S_6\):
\[
123456 \quad G \quad 132546 \quad G \quad 132645 \quad G \quad 135642 \quad G \quad 136524 \quad G
\]

\[(iv)\) \(n=7\) and the subgroup \(G = \langle 7235461, 1756432 \rangle \cong S_4\) of \(S_7\):
\[
1234657 \quad G \quad 1235476 \quad G \quad 1237456 \quad G \quad 1273564 \quad G \quad 1324576 \quad G \quad 1325467 \quad G
\]

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 PSCA\((n, k, \lambda)\): \((5, 3, 3), (6, 3, 3), (7, 3, 3), (7, 4, 3), (8, 3, 4), \) and \((8, 3, 5)\) (see Sect. 4.5). We have given an example of a 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 Sect. 4.6). Examples of a 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 PSCA\((n, k, \lambda)\) for new parameter sets \((n, k, \lambda)\).
\[(iii)\) Find a group-based construction for a PSCA\((n, n-1, 1)\) for each \(n \geq 2\).
\[(iv)\) Is there a parameter set \((n, k, \lambda)\) for which a 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 PSCA\((n, k, \lambda)\) implies the existence of a 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 [7], 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 [16], 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 [7], including that $g(7, 5) \leq 4$ and $g(9, 3) \leq 4$. The paper [7] 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 $S_n$.

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

**Data availability** All data on which the conclusions of this paper depend are included here.

**References**

1. Apilli B.: Fault-based combinatorial testing of web services. In: Proceedings of 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications, pp. 731–732 (2009).
2. Bennett F.E., Mahmoodi A.: Directed designs. In: Colbourn C.J., Dinitz J.H. (eds.) Handbook of Combinatorial Designs, 2nd edn, pp. 441–444. Chapman and Hall, Boca Raton (2007).
3. Chee Y.M., Colbourn C.J., Horsley D., Zhou J.: Sequence covering arrays. SIAM J. Discret. Math. 27(4), 1844–1861 (2013).
4. Colbourn C.J.: Covering arrays and hash families. In: Crnković D., Tonchev V. (eds.) Information Security, pp. 99–135. Coding Theory and Related Combinatorics. IOS Press, Amsterdam (2011).
5. Dawson J.E., Seberry J., Skillicorn D.B.: The directed packing numbers $DD(t, v, v)$, $t \geq 4$. Combinatorica 4(2–3), 121–130 (1984).
6. Füredi Z.: Scrambling permutations and entropy of hypergraphs. Random Struct. Algor. 8(2), 97–104 (1996).
7. Gentle A.R., Wanless I.M.: On perfect sequence covering arrays. Ann. Comb. (2022), arXiv:2202.01960.
8. Huang S., Cohen M.B., Memon A.M.: Repairing GUI test suites using a genetic algorithm. In: 2010 Third International Conference on Software Testing, Verification and Validation, pp. 245–254 (2010).
9. Ishigami Y.: An extremal problem of $d$ permutations containing every permutation of every $t$ elements. Discret. Math. 159(1–3), 279–283 (1996).
10. Klein A.: On perfect deletion-correcting codes. J. Comb. Des. 12(1), 72–77 (2004).
11. Kuhn D.R., Higdon J.M., Lawrence J.F., Kacker R.N., Lei Y.: Combinatorial methods for event sequence testing. In: 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation, pp. 601–609 (2012).
12. Levenshtein V.I.: Perfect deletion-correcting codes as combinatorial designs. In: II International Workshop Algebraic and Combinatorial Coding Theory, pp. 137–140 (1990).
13. Levenshtein V.I.: Perfect codes in the metric of deletions and insertions. Diskr. Mat. 3(1), 3–20 (1991). English translation in: Discret. Math. Appl. 2(3), 241–258 (1992).
14. Mathon R.: Searching for spreads and packings. In: Hirschfeld J.W.P., Magliveras S.S., de Resmini M.J. (eds.) Geometry, Combinatorial Designs and Related Structures. Proceedings of First Pythagorean Conference, pp. 161–176. Cambridge University Press, Cambridge (1997).
15. Mathon R., van Trung T.: Directed $t$-packings and directed $t$-Steiner systems. Des. Codes Cryptogr. 18, 187–198 (1999).
16. Na J.: Perfect sequence covering arrays. Master’s thesis, Simon Fraser University (2021).
17. Radhakrishnan J.: A note on scrambling permutations. Random Struct. Algor. 22(4), 435–439 (2003).
18. Spencer J.: Minimal scrambling sets of simple orders. Acta Math. Hung. 22(3–4), 349–353 (1971).
19. The GAP Group. GAP—Groups, Algorithms, and Programming, Version 4.11.0 (2020).
20. Wang W., Sampath S., Lei Y., Kacker R.: An interaction-based test sequence generation approach for testing web applications. In: 2008 11th IEEE High Assurance Systems Engineering Symposium, pp. 209–218 (2008).
21. Wang W., Lei Y., Sampath S., Kacker R., Kuhn R., Lawrence J.: A combinatorial approach to building navigation graphs for dynamic web applications. In: 2009 IEEE International Conference on Software Maintenance, pp. 211–220 (2009).

22. Wilson R.M.: A diagonal form for the incidence matrices of $t$-subsets vs. $k$-subsets. Eur. J. Comb. 11, 609–615 (1990).

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

24. Yuan X., Cohen M.B., Memon A.M.: GUI interaction testing: incorporating event context. IEEE Trans. Softw. Eng. 37(4), 559–574 (2011).

25. Yuster R.: Perfect sequence covering arrays. Des. Codes Cryptogr. 88, 585–593 (2020).

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.